

UCLA

UCLA Electronic Theses and Dissertations

Title

Closed-Loop Multimodal Generative Artificial Intelligence under Real-World Constraints

Permalink

<https://escholarship.org/uc/item/35q9142r>

ISBN

9798186340089

Author

Zhang, Yasi

Publication Date

2026-08-11

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Closed-Loop Multimodal Generative Artificial Intelligence
under Real-World Constraints

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Statistics

by

Yasi Zhang

2026

© Copyright by

Yasi Zhang

2026

ABSTRACT OF THE DISSERTATION

Closed-Loop Multimodal Generative Artificial Intelligence under Real-World Constraints

by

Yasi Zhang

Doctor of Philosophy in Statistics

University of California, Los Angeles, 2026

Professor Yingnian Wu, Co-Chair

Professor Oscar Leong, Co-Chair

Multimodal generative models are increasingly used in interactive and scientific settings where success requires more than perceptual sample quality. A deployable system must evaluate structured behavior, align model optimization with graded objectives, and generate reliable outputs when training data or measurements are imperfect. This dissertation develops these three capabilities as a closed loop of evaluation, alignment, and generation.

The first part studies evaluation for instruction-based image editing. EdiVal-Agent decomposes scenes into objects and evaluates instruction following, content consistency, and visual quality across multiple editing turns. MT-EditFlow then uses these multi-axis signals to optimize sequential editing trajectories and reduce exposure bias and error propagation.

The second part studies alignment. REAL formulates regression-aware reinforcement learning for language models used as numeric judges, preserving the ordinal information discarded by binary rewards. EBAMA instead aligns text-to-image diffusion through object-conditioned energy-based attention-map objectives that improve attribute binding and re-

duce object neglect.

The third part studies generation under corrupted information. Restoration Score Distillation learns one-step generators using only degraded observations, while Flow Priors uses flow-matching likelihood structure to solve linear inverse problems through iterative corrupted trajectory matching. Together, these methods show how evaluation signals, alignment objectives, and generative priors can be designed around real-world constraints rather than idealized clean supervision.

The dissertation of Yasi Zhang is approved.

Mark Handcock

Hongquan Xu

Oscar Leong, Committee Co-Chair

Yingnian Wu, Committee Co-Chair

University of California, Los Angeles

2026

To Ying Nian Wu, Oscar Leong, and to myself.

TABLE OF CONTENTS

1	Introduction	1
1.1	Evaluation	1
1.2	Alignment	2
1.3	Generation	2
1.4	Thesis Organization	2
 I	 Evaluation	 3
 2	 EdiVal-Agent: An Object-Centric Framework for Automated, Fine-Grained Evaluation of Multi-Turn Editing	 4
2.1	Introduction	4
2.2	EdiVal-Agent	8
2.2.1	Overview	8
2.2.2	Step 1: Decomposition	9
2.2.3	Step 2: Instruction Generation	9
2.2.4	Step 3: Evaluation	11
2.2.5	Measuring Human Agreement	13
2.3	Benchmarking Multi-turn Editing	14
2.3.1	Instruction Following	16
2.3.2	Content Consistency	17
2.3.3	Visual Quality	17
2.3.4	Multi-turn Editing vs. Complex Editing	18

2.3.5	Pareto Front	19
2.3.6	Tool Swap Analysis and Prompt Compression Strategies	19
2.4	Conclusion	19
2.5	Appendix	20
2.5.1	Artificial Analysis Leaderboard	20
2.5.2	Failure Case	26
2.5.3	Discussion on single-shot complex editing	27
2.5.4	Discussion on Visual Quality	27
2.5.5	VLMS failing to judge visual quality	29
2.5.6	Additional Evaluation Results	38
2.5.7	Related Work	45
2.5.8	Limitation and Discussion	47
2.5.9	Pareto Plot	47
2.5.10	Ablation Study on Complex Editing Compression	48
2.5.11	Tool Swap Analysis	49
3	MT-EditFlow: Reinforcement Learning for Multi-Turn Image Editing with Flow Matching	59
3.1	Introduction	59
3.2	Related Work	62
3.2.1	Instruction-Based Image Editing	62
3.2.2	Image Editing Evaluation and Benchmarks	63
3.2.3	Reinforcement Learning with Diffusion and Flow Matching	64
3.3	Background	64

3.3.1	Flow Matching	64
3.3.2	Reinforcement Learning for Image Editing	65
3.4	MT-EditFlow: RL for Multi-Turn Image Editing	67
3.4.1	MT-EditFlow Reward Objective	67
3.4.2	Scoring Strategy: Granularity and Sparsity	69
3.4.3	Evaluator Mode: Thinking vs. Non-thinking	70
3.4.4	Fusion Level: Reward vs. Advantage	71
3.4.5	Trajectory-Level Advantage Broadcasting	71
3.5	Experiments	72
3.5.1	Training Details	73
3.5.2	Main Results	73
3.5.3	Analysis on Reward Signal Design	75
3.5.4	Ablation Studies	77
3.6	Limitations	78
3.7	Conclusion	78

II Alignment 79

4	REAL: Regression-Aware Reinforcement Learning for LLM-as-a-Judge	80
4.1	Introduction	80
4.2	Background	82
4.2.1	LLMs and Reinforcement Learning	82
4.2.2	Regression-Aware Inference and Finetuning in LLMs	83
4.3	Optimal Decision Rule for LLM-as-a-Judge Evaluation Metrics	85

4.4	REAL: Regression-Aware Reinforcement Learning	86
4.4.1	REAL Objective	87
4.4.2	Generalized Policy Gradient with Policy-Dependent Reward Functions	87
4.4.3	Stabilization and Implementation	88
4.4.4	Relations with Prior Methods	89
4.5	Experiments	91
4.5.1	Main Results	93
4.5.2	On Regression-Aware Policy-Independent Rewards	93
4.5.3	Ablation Studies	94
4.5.4	Generalizability to Other Settings	96
4.6	Limitations	96
4.7	Conclusion	97
4.8	Appendix	97
4.8.1	Extension to pairwise difference modeling	97
4.8.2	Proofs of the theoretical results	98
4.8.3	Algorithm and Experiments	105
5	Object-Conditioned Energy-Based Attention Map Alignment in Text-to-Image Diffusion Models	115
5.1	Introduction	115
5.2	Related Work	117
5.3	Background	118
5.3.1	Stable Diffusion Models	118
5.3.2	Cross-Attention Mechanism	119

5.4	Method	119
5.4.1	Extraction of the Object-Oriented Structure	120
5.4.2	Object-Conditioned Energy-Based Model	120
5.4.3	Object-Conditioned Energy-Based Attention Map Alignment	121
5.4.4	Workflow	122
5.5	Experiments	124
5.5.1	Quantitative Comparison	124
5.5.2	Qualitative Comparison	125
5.5.3	Human Evaluation	126
5.5.4	Ablation Study	127
5.5.5	Augmented Attribute Editing	128
5.6	Conclusion	128

III Generation 137

6	Score Distillation Beyond Acceleration: Generative Modeling from Corrupted Data	138
6.1	Introduction	138
6.2	Background	141
6.2.1	Diffusion Models	141
6.2.2	Diffusion Models for Corruptions	142
6.2.3	Score Distillation for Generative Modeling	143
6.3	Restoration Score Distillation	144
6.3.1	Phase I: Flexible Corruption-Aware Pretraining	145

6.3.2	Phase II: One-Step Generator Distillation	145
6.4	Experiments	148
6.4.1	A 2D Toy Example	149
6.4.2	Denoising as a Special Case of Restoration Score Distillation	149
6.4.3	General corruptions mixed with noise	150
6.4.4	Beyond natural images: Multi-coil Magnetic Resonance Imaging (MRI)	152
6.4.5	Ablations: Data Scale, Distillation Loss Choice, and More	153
6.4.6	Sampling and Training Efficiency for Distillation	154
6.4.7	After Distillation: Solving Conditional Inverse Problems	156
6.4.8	Model Selection Criterion under Corruption: Proximal FID	156
6.4.9	Why Score Distillation Beyond Acceleration?	157
6.5	Conclusion	158
6.6	Appendix	159
6.6.1	Proximal FID	159
6.6.2	Training and Inference Efficiency	161
6.6.3	Additional Experiments	162
6.6.4	Discussions and Limitations	167
6.6.5	Theoretical Results and Proofs	168
6.6.6	Phase I: Variants of Pretraining Details	196
6.6.7	Phase II: Distillation	201
6.6.8	Implementation Details	202

7	Flow Priors for Linear Inverse Problems via Iterative Corrupted Trajectory Matching	204
----------	--	------------

7.1	Introduction	204
7.2	Background	207
7.2.1	Flow-Based Models	207
7.2.2	Probability Computation for Flow Priors	208
7.3	Method	209
7.3.1	Flow-Based MAP	209
7.3.2	Flow-Based MAP Approximation	210
7.3.3	Toy Example Validation	213
7.4	Experiments	214
7.4.1	Natural Images	216
7.4.2	Medical application	217
7.4.3	Ablation studies	219
7.5	Conclusion	220
7.6	Limitations and Future Work	221
7.7	Appendix	221
7.7.1	Proof	223
7.7.2	Compliance of Trajectory	228
7.7.3	Computational Efficiency	230
7.7.4	Implementation Details	230
8	Conclusion and Future Directions	237
8.1	Future Directions	237
	References	239

LIST OF FIGURES

2.1	Overview of our workflow and representative model’s performance. For visualization, we adopt two thresholds: a consistency score of at least 90 and a visual quality score of at least 6. Details of the automated evaluation pipeline are provided in Figure 2.2 and Section 2.2. In multi-turn editing, models exhibit distinct weaknesses: <i>GPT-Image-1</i> struggles with content consistency, <i>Qwen-Image-Edit</i> underperforms in both visual quality and content consistency, and <i>FLUX.1-Kontext-dev</i> lags in instruction following, whereas <i>Nano Banana</i> shows no single dominant weakness.	5
2.9	Turn-3 instruction following: Multi-turn vs. single-shot complex prompts. . . .	18
2.2	Framework of EdiVal-Agent . It first decomposes images into semantically meaningful objects, such as <i>metal yellow sign</i> and <i>metal brown pole</i> , and identifies their contextual relationships, e.g., they are both in <i>foreground</i> . It then generates diverse and proper editing scenarios at scale which are based on the initial analysis, e.g., <i>Change the color of metal brown pole to gray</i> . Finally, it systematically evaluates editing model outputs from multiple axes with our proposed metrics: EdiVal-IF, EdiVal-CC, and EdiVal-VQ. Our agentic pipeline is agnostic to the expert tools used and can be readily enhanced with more advanced tools in the future.	21
2.3	Beautification vs. preservation under the prompt: “ <i>Change the background to a library.</i> ” <i>GPT-Image-1</i> tends to increase HPSv3 via beautification, while <i>FLUX.1-Kontext-max</i> emphasizes fidelity to the input.	22
2.4	Results of human agreement. Dashed lines represent the average accuracy of each method. EdiVal-IF achieves 81.3% human agreement accuracy, significantly outperforming the VLM (<i>Qwen2-VL</i>) at 75.2% and thresholded CLIP_dir at 65.4%. Note that the CLIP_dir threshold is tuned separately for each task. . .	22

2.5	Marginal Task Success rate across turns.	23
2.6	Marginal task success rate grouped by task types for Nano Banana.	24
2.7	Illustration of background consistency. Instruction: “Remove beige brick house.” The grounding box is the union of all object regions from the raw and edited images.	24
2.8	Illustration of object consistency. Instruction: “Remove brick beige house.” The grounding box, extracted from the raw input image, highlights the localized region used to compute unchanged-object consistency. The corresponding consistency score is shown in brackets.	24
2.10	Pareto front plot for Turn 3 editing across EdiVal-IF, EdiVal-CC, and EdiVal-VQ.	25
2.11	Artificial Analysis leaderboard (Sep 12, 2025). Screenshot of the public leaderboard used for comparison in Table 2.5.	25
2.12	Failure due to detector false positives. Although the edit visually replaces the railing with a wooden fence, Grounding-DINO fires on both “stone gray railing” and “wooden fence” in overlapping regions, causing an incorrect failure in our instruction-following metric.	26
2.13	Marginal task-success rate of the <i>last</i> instruction as a function of complex prompt length (levels $C = 1, 2, 3$).	27
2.14	Per-image 99.9% luminance quantile across turns. Higher values indicate more extreme bright pixels and greater risk of over-exposure.	28
2.15	Representative Qwen-Image-Edit examples illustrating over-exposure and bright artifact formation across turns. Although editing instructions are often satisfied, the images show elevated luminance and high-frequency bright streaks that accompany the edits (visible especially in T2/T3). Editing instructions: [Remove polyester white skirt, Change the count of tennis ball to 4, Change the color of tank top to blue]. Note that VLM gives a <i>positive</i> score to all the images.	29

2.16	Pareto plot for Turn 1.	48
2.17	Pareto plot for Turn 2.	48
2.18	Failure due to detector false positives. Although the edit visually replaces the railing with a wooden fence, Grounding DINO fires on both “stone gray railing” and “wooden fence” in overlapping regions, causing an incorrect failure in our instruction-following metric.	55
2.19	OWL-ViT fails to detect the queried objects regardless of whether they are actually present in the image, and therefore does not resolve this failure case. . . .	55
2.20	GLIP produces substantially more false positives than Grounding DINO, hallucinating objects such as a “stone gray railing” and misclassifying the tower region as a “wooden fence” or “stone gray railing”.	56
2.21	Per-turn EdiVal-CC scores under DINOv3 (x-axis) vs. DINOv2 (y-axis) for the four representative models. Each point is a (model, turn) pair; Pearson correlation is 0.9987 and Spearman correlation is 1.0000, confirming that the feature-extractor swap only shifts the absolute scale while preserving the ranking. . . .	58
3.1	Qualitative results of MT-EditFlow (Ours). In multi-turn image editing, a single failed turn compromises the entire sequence, even if other turns succeed (e.g., a sheep failing to be removed). Furthermore, <i>error accumulation</i> is a common pitfall: if an intermediate edit is incorrect, subsequent turns are often derailed. For instance, using the FLUX.2 base model, if a white horse is mistakenly transformed into a deer during the second turn, the third edit intended for the horse is effectively doomed. By contrast, MT-EditFlow enables the model to execute robust, successful edits across multiple consecutive turns, effectively preventing error propagation.	60

3.2	Illustration of the reward bias and variance trade-off between thinking and non-thinking modes. Thinking mode consistently leads to reduced reward bias and increased variance across different temperatures. For example, the mean evaluation score under thinking mode for poor edits is closer to 1, the lowest score.	70
3.3	Multi-turn editing performance increases with the number of training turns K used.	73
3.4	Marginal task success across turns. With MT-EditFlow, performance remains stable, reducing exposure bias.	73
3.5	Ablation on Reward Signal Design. The top row shows training reward, and the bottom row shows held-out evaluation reward. Fine-grained 1–5 scoring provides a more effective learning signal than binary variants, thinking mode produces better performance with more accurate but higher-variance rewards, and advantage-level fusion consistently outperforms reward-level fusion.	75
3.6	Ablation on λ_{CC}. Higher λ_{CC} improves CC at the cost of IF; $\lambda_{CC} = 0.3$ achieves the best overall balance.	76
3.7	Ablation on group size G and discretization steps T. We found that a smaller group size $G = 16$ and a larger discretization step $T = 12$ lead to better performance.	77
4.1	Overview of the REAL framework. REAL addresses the limitations of standard RL in LLM-as-a-Judge tasks by optimizing a policy-dependent regression reward. The framework employs a generalized policy gradient that leads to a gradient update that decomposes into two terms: (1) Exploration Over Reasoning Trajectory ; and (2) Regression-Aware Prediction Refinement . This enables principled optimization of ordinal structures that standard RL with binary rewards typically ignores. The full algorithm is in Alg. 11.	107

4.2	Evaluation performance during RL training. Both standard RL with binary reward (i.e., $r_{\text{acc}} = \mathbf{1}(y = y^*)$) and REAL with regression-aware reward (i.e., Eq. 4.5) were initialized from the SOTA SFT checkpoint, i.e., TRACT [CLL25]. Standard RL results in suboptimal performance in correlation metrics compared to our proposed approach: REAL.	108
4.3	Performance gains on Qwen3-32B. Our method achieves an average increase of +8.40 Pearson and +7.20 Spearman correlation over the SFT baseline, and +18.30/+11.20 over the base model.	109
4.4	Response length and entropy during REAL training. Response length increases, and the per-token entropy of the policy model decreases steadily.	109
5.1	Key observations of the generation process of diffusion models. The given prompt is “a <u>purple crown</u> and a <u>blue suitcase</u> ”. In panel (c), we hypothesize that if the intensity level of any object in the prompt does not remain high during the first half of the denoising process, e.g. the crown in SD and SG, the model would fail to generate the object in the final image. The panel (d) suggests that if the attention map distributions of any attribute-object pair are not aligned, the model would struggle to correctly bind attributes to their respective objects, e.g. ‘purple’ and ‘crown’ in SD and AnE. The generated images are displayed in the panel (e). All methods share the same random seed.	130
5.2	An overview of our workflow for optimizing diffusion models. It includes aggregation of attention maps, computation of object-centric attention loss, and updates to z_t	131
5.3	Full Sim. results on DVMP and ABC-6K datasets. We randomly sample 200 prompts from each dataset and generate 4 images for each prompt.	131
5.4	Qualitative comparison on the AnE dataset. Each column shares the same random seed.	132

5.5	Qualitative comparison on the DVMP dataset. Each column shares the same random seed.	133
5.6	Qualitative comparison on the ABC-6K dataset. Each column shares the same random seed.	134
5.7	Preference ratio percentage on text-image alignment by human evaluation. Ours(avg) represents the average preference ratio of our method compared with the other four methods.	135
5.8	Ablation demonstration for intensity weight λ. (a) a sliced <u>apple</u> and a <u>purple</u> <u>camera</u> and a <u>teal</u> <u>lion</u> ; (b) a <u>brown</u> <u>bear</u> with <u>red</u> <u>hat</u> and <u>scarf</u> and a small stuffed <u>bear</u> ; (c) a gray <u>crown</u> and a <u>purple</u> <u>apple</u> . We have selected one prompt from each dataset to showcase the stability of our method. Each column shares the same random seed.	135
5.9	Augmented attribute editing with our method. The left two columns demonstrates the attribute editing results from PtP, while the right two columns demonstrates the results from PtP w. ours.	136
6.1	Overview of <i>Restoration Score Distillation</i> (RSD) and some qualitative results across diverse operators: Gaussian deblurring, random inpainting, and super-resolution.	139
6.2	A toy example of learning from a noisy dataset with $\sigma = 0.05$.	146
6.3	Multi-coil MRI: qualitative reconstructions from RSD across acceleration levels R . See Tab. 6.6 for corresponding FID trends.	153
6.4	Efficiency of RSD (CIFAR-10, $\sigma = 0.2$).	155
6.5	CIFAR-10 (denoising, $\sigma = 0.2$): Proximal FID closely tracks true FID and selects a near-optimal checkpoint.	157

6.6	Across datasets: True FID vs. <i>Proximal FID</i> during distillation on FFHQ, CelebA-HQ, and AFHQ-v2 (denoising, $\sigma = 0.2$). In all cases, Proximal FID reliably identifies checkpoints with near-optimal true FID.	160
6.7	Across corruption operators: True FID vs. <i>Proximal FID</i> during training for Gaussian deblurring, random inpainting, and super-resolution. Proximal FID consistently tracks the true FID and supports reliable model selection across operators.	160
6.8	Mismatched Corruption Levels during Distillation. Our method remains robust even when the assumed corruption level is mismatched during distillation.	163
7.1	Illustration of the idea of ICTM. The corrupted trajectory $u_t := \mathcal{A}(x_t)$ follows the corrupted flow ODE $du_t = \mathcal{A}(v_\theta(x_t, t))dt$	212
7.2	Results of a toy example modeling 1,000 FFHQ faces as a Gaussian distribution. Subfigure (a) shows the qualitative results of our method; Subfigure (b) presents the histogram of the differences between ours and the true MAP; Subfigure (c) displays the MSE values as the NFEs varies.	214
7.3	Qualitative comparison results on the CelebA-HQ dataset. The reconstructions generated by our method align more faithfully with the ground truth and exhibit a higher degree of refinement.	216
7.4	Qualitative comparison results on compressed sensing. Our method produces more faithful reconstructions with fewer artifacts, ensuring higher accuracy and clarity in the details.	218
7.5	Ablation results of step size η and guidance weight λ . The choice of hyperparameters for our algorithm is fairly consistent across all tasks. We choose $\eta = 10^{-2}$ for all experiments on CelebA-HQ. For λ , we choose $\lambda = 10^3$ for Gaussian deblurring and $\lambda = 10^4$ for the other tasks.	219

7.6	Ablation results of iteration number K on different tasks. For super-resolution and the other three tasks, $K = 1$ is sufficient to achieve the best performance with the optimal step size η and guidance weight λ . However, for compressed sensing, it is necessary to increase K to obtain the best performance. We hypothesize that this is due to the increased complexity of the compressed sensing operator, which requires more iteration steps to ensure the correct optimization direction.	220
7.7	Generated samples from the flow trained on 10,000 Gaussian samples.	232
7.8	Generated samples from the flow trained on 10,000 HCP T2w images.	233
7.9	Hyperparameter t' selection results for OT-ODE on the CelebA-HQ dataset. We select $t' = 0.2, 0.1, 0.2, 0.2$ for super-resolution, inpainting(random), Gaussian deblurring, and inpainting(box), respectively.	234
7.10	Hyperparameter t' selection results for OT-ODE on the HCP T2w dataset. We select $t' = 0.1$ for all the experiments.	235
7.11	Hyperparameter η selection results for DPS-ODE. We select $\eta = 1000, 750, 200, 75$ for super-resolution, inpainting(random), Gaussian deblurring, and inpainting(box), respectively.	236
7.12	Hyperparameter η selection results for DPS-ODE on the HCP T2w dataset. We select $\eta = 200$ for all the experiments.	236

LIST OF TABLES

2.1	Key attributes of open-source edit benchmarks. Note that ImgEdit [YHL25] does not include multi-turn editing experiments in the paper.	7
2.2	Instruction types in EdiVal-Bench created by EdiVal-Agent , grouped by semantic category. Counts are shown per turn (T1–T3).	10
2.3	Results of multi-turn editing. EdiVal-IF, EdiVal-CC, and EdiVal-O across three sequential editing turns. Best per column in dark red ; second-best in lighter red . * indicates a closed-source model, and † indicates an open-source model.	15
2.4	EdiVal-VQ and EdiVal-VQ Δ results across turns. dark red denotes the <i>best</i> value in the column; lighter red denotes the <i>second-best</i> . For HPS, higher values are stronger aesthetics. For Δ , smaller values are stronger fidelity preservation. . . .	18
2.5	Model rankings on the overlapping set. Relative ranks from Artificial Analysis (human votes) vs. EdiVal (ours) as of Sep 12, 2025.	20
2.6	Turn-3 instruction following: Multi-turn vs. single-shot complex prompts, grouped by technique. Bold indicates which setting is higher for each model. . .	27
2.7	Image success rates and overall task success rates across turns. (Multi-turn model)	39
2.8	Task success rates (%) across five instruction types and three turns (multi-turn mode).	39
2.9	Task success rates (%) across four instruction types and three turns (multi-turn mode).	40
2.10	Image rates, overall task rates, and marginal means across three turns (complex mode).	40
2.11	Success rates (%) for five instruction types across three turns (complex mode). .	41

2.12	Success rates (%) for four instruction types across three turns (complex mode).	41
2.13	Consistency scores (%) across DINOv3-based and L1-based object/background metrics. (multi-turn mode)	42
2.14	Consistency and $1 - L_1$ metrics across three turns (multi-turn mode for square image).	42
2.15	Consistency and L_1 metrics across three turns (multi-turn model for unsquared image).	43
2.16	Consistency scores (%) across object/background DINOv3 and L_1 metrics (complex mode).	43
2.17	Human preference scores, p999, and p99 across three turns (multi-turn mode).	44
2.18	Updated human preference scores, p999 scores, and p99 scores across three turns (complex mode).	44
2.19	Complex Editing Performance for compression variants.	49
2.20	Tool swap analysis (correlations).	49
2.21	Correlations between per-turn instruction-following success rates under the default VLM (Qwen2.5-7B-VL) and three strong vision-language baselines. All correlations are computed over 12 (model, turn) points.	51
2.22	Per-turn image success, task success, and overall scores under different evaluation settings (ranked).	52
2.23	Performance of popular detectors including OWL-ViT, GLIP, and Grounding DINO (Swin-L). Numbers are copied from Table 1 of Grounding DINO 1.5 [RJL24]. In the open-set setting ODinW, which is most similar to our scenario, Grounding DINO outperforms GLIP. It also achieves the best AP on COCO. LVIS is a large-scale benchmark with over 1k categories spanning rare, common, and frequent objects.	54

2.24	DINO-v2 feature backbone results with EdiVal-CC Rank.	57
2.25	DINO-v3 feature backbone results with EdiVal-CC Rank.	57
3.1	Results of multi-turn editing on EdiVal-Bench. EdiVal-IF (Instruction Following), EdiVal-CC (Content Consistency), and EdiVal-O (Overall) across three sequential editing turns.	72
3.2	Single-turn editing results of MT-EditFlow on ImgEdit-Bench. GPT-4.1 is used for evaluation. Our multi-turn RL algorithm naturally leads to improved single-turn performance.	74
3.3	Comparison of MT-EditFlow variants on FLUX.1-Kontext-dev.	74
4.1	Conceptual comparison of selected SFT and RL training methods on non-ordinal and ordinal targets. $r(\theta, x, c) = -(\hat{y}_\theta(x, c) - y^*)^2 + \lambda \log \pi_\theta(y^* x, c)$ represents the regression-aware reward. REAL is the first framework to combine a regression-aware reward with reinforcement learning for reasoning exploration. Note that JEPO further approximates the gradient with the multi-sample Jensen’s Bound.	90
4.2	Evaluation results ($\times 100$) across benchmarks using Pearson (r), Spearman (ρ), and Kendall (τ). All baseline reward model scores are taken from TRACT [CLL25] and Prometheus [KSC23]. Results for the Mistral and Qwen models are obtained from our own runs. <u>Underline</u> indicates the initialization checkpoint used by REAL. We find that TRACT saturates on the Qwen3 series models, so we choose RAFT as the initialization checkpoint. Mistral2-7B’s base model failed to follow instructions and respond in the correct format, so we applied a 100-step warmup. Due to computational constraints we run only RAFT and REAL for Qwen3-32B; we include all baselines for other models.	110

4.3	Average-of-N results at inference. Averaging the results of N generations slightly improves performance, although REAL already enjoys efficiency and strong performance at $N = 1$	111
4.4	Comparison between standard RL baselines and REAL on Mistral2-7B. Standard RL(reg) with regression-aware policy-independent rewards also fail to improve upon the SFT checkpoint.	111
4.5	Ablation on group size K. Larger values of K generally result in better correlation performance.	111
4.6	Ablation on log-likelihood weight λ. Using regression-aware reward alone ($\lambda = 0$) already yields excellent results, while extra log-likelihood reward further improves.	112
4.7	Ablation on dynamic sampling when group size $K = 8$. Partially correct prompts provide the best learning signal.	112
4.8	Initialization from different checkpoints. Initializing from a high-quality SFT checkpoint can lead to better SOTA performance. Our algorithm works with different checkpoint types.	113
4.9	Ablation on prediction weight β. We choose $\beta = 0.01$ (See Eq. 4.8) for initialization from both base and SFT checkpoints.	113
4.10	Ablation on temperature T. $T = 1$ is chosen by default. The temperature parameter is applied in both the training and inference stages, for both CoT generation and score prediction.	114
4.11	Results on score-difference prediction using the Preference_Collection dataset. REAL achieves better out-of-domain generalization while maintaining competitive in-domain performance.	114
4.12	Comparison with JEPO with Mistral2-7B.	114

5.1	Comparison of Full Sim., Min. Sim., and T-C Sim. across different methods on the AnE dataset. Note that the performance of SG on Animal-Animal is degraded to SD, as the prompts do not contain any attribute-object pairs. The best and second-best performances are marked in bold numbers and underlines, respectively; tables henceforth follows this format.	123
5.2	Ablation results on repulsive term. Both Ours and Ours($\lambda = 0$) benefit from the repulsive term as defined in Eqn. (5.5).	132
5.3	Ablations on object-conditioned perspective and energy function choice.	133
6.1	Summary of corruption-aware diffusion objectives used for pretraining. Our framework can be seamlessly integrated with existing advanced corruption-aware diffusion objectives.	142
6.2	Denoising results on CIFAR-10 and CelebA-HQ at $\sigma = 0.2$. Rows with $\sigma=0.0$ are clean-data upper bounds. Few-shot methods use 50 clean images. Baseline numbers are from the original papers or [BWC24, LWY25]. The distilled student (RSD, one-step) improves over teacher models (Full/Truncated/Consistency) and surpasses all few-shot baselines.	150
6.3	RSD vs. teachers on CIFAR-10/FFHQ/AFHQ-v2. RSD (distilled) consistently surpasses teacher diffusion (<i>Full</i> , <i>Truncated</i>) across datasets and noise levels. (FID).	151
6.4	CelebA-HQ restoration under noiseless/noisy settings. Baselines are taken from original papers when available, otherwise reproduced. EM-Diffusion uses 50 clean images for initialization, while RSD uses none. Best results are highlighted.	151
6.5	CelebA-HQ random inpainting vs. missing rate p. Teacher diffusion is trained as in [DSD23].	152

6.6	FID across acceleration levels R in multi-coil MRI. Teacher diffusion is trained as in [ADL24].	152
6.7	Comparison of RSD with teacher diffusion and baselines on CIFAR-10. Metrics for DDPM and Rectified Flow are from [LGL22b].	152
6.8	Data-size ablation on CelebA-HQ.	153
6.9	Conditional inverse problem results of denoising on CIFAR10 at $\sigma = 0.2$. Results for the baselines are taken from [BWC24]. We follow [BWC24] and sample 250 test images and compute the average PSNR and LPIPS.	156
6.10	Model selection by Proximal FID. Best true FID obtained by selecting checkpoints via Proximal FID vs. selecting via true FID (oracle). The red (+) indicates the gap to the best achievable true FID during training.	157
6.11	Comparison between the best true FID and the FID of the model selected by proximal FID for the general corruption task.	161
6.12	Training and inference efficiency of our method. During training, the additional distillation phase introduces only a minor overhead, as FID decreases rapidly and surpasses the teacher diffusion model, Teacher-Truncated, within just 4 hours. For inference, our one-step generator enables the generation of 50k images in only 20 seconds, achieving a $30\times$ speedup.	162
6.13	Comparison of RSD with teacher diffusion and baselines on CIFAR-10. *Metrics for DDPM and Rectified Flow (ODE) are copied from the Rectified Flow [LGL22b] paper’s Table 1.	162
6.14	Unknown noise: true σ vs. $\hat{\sigma}$ on CIFAR-10.	164
6.15	Misspecification study on CIFAR-10. RSD is robust to slight overestimation when the true $\sigma = 0.2$	164

6.16	$\times 2$ super-resolution on CelebA-HQ (100 images). We compare a pseudo-inverse baseline ($\mathcal{A}^\dagger y$), EM-Diffusion (few-shot) with DPS, Teacher Diffusion with DPS, and our RSD prior with latent optimization (Eq. 6.6). Higher PSNR/SSIM and lower LPIPS/FID are better.	165
6.17	Comparison of FID scores. Lower is better.	166
6.18	Ablation of distillation losses on CIFAR-10 denoising ($\sigma \in \{0.1, 0.2, 0.4\}$; FID)	167
7.1	Quantitative comparison results in terms of PSNR and SSIM on the CelebA-HQ dataset. Our algorithm surpasses all other baselines across all tasks. The best values are highlighted in blue and the second-best are underlined.	215
7.2	Results of compressed sensing with varying compression rate ν on the HCP T2w dataset. Note that compressed sensing is more challenging due to the complexity of the forward operator, as evidenced by the poor performance of OT-ODE, which assumes a Gaussian distribution of measurement y given x_t . The best values are highlighted in blue.	217
7.3	Computational time comparison. We compare the time required to recover 100 images for the super-resolution task on a single GPU.	231

ACKNOWLEDGMENTS

My Ph.D. journey began with a desire for freedom: the freedom to discover myself in a foreign land, to pursue truth, and to follow my curiosity to the frontiers of artificial intelligence. At the same time, it was also driven by fear and a sense of lack: a desire to escape from my origins and a constant urge to find a place with more resources and opportunities. Even now, my journey at UCLA feels like both a mystery and an extraordinary gift.

In case I have never said this clearly enough: since the day Ying Nian admitted me to the program, he has completely changed the course of my life. The past four years have been the best years of my life. I have been deeply moved by Ying Nian's ambition and by his unceasing pursuit of ideas worthy of it. To me, he is a true scholar and an extraordinary educator. His incisive guidance and far-reaching predictions have taken root in me like seeds. They gave me an unfamiliar sense of possibility: a desire to seek something enduring. It is rare to encounter such a feeling while one is young, and I count myself extraordinarily fortunate to have received his guidance and encouragement. He has made my life feel more expansive, unpredictable, and exhilarating than I had imagined. I now look forward quietly to the arrival of great things, while also finding a calm happiness in the ordinary days of life.

I met Oscar in the summer of 2023. I was immediately drawn to his patient, lucid explanations of inverse problems and to the warmth and openness that made every conversation with him a pleasure. That encounter led to our first collaboration on inverse problems. To me, Oscar represents an ideal of how to live a full and meaningful life: with care for one's career and dreams, but also for family, love, and the deeper meaning of life. He has shown me, through his example, how generosity and care can flow toward the people around us, and I am deeply grateful to have received both from him. I have come to believe that learning to love others and oneself is among life's deepest meanings. In every aspect of my research, Oscar has supported me with patience, encouragement, and a remarkable sense of calm.

I thank Professor Hongquan Xu and Professor Mark Handcock for their time and effort

in serving on my dissertation committee.

I am profoundly grateful to all my collaborators, especially Tianyu Chen, a truly gifted genius whose curiosity and knowledge seem boundless. Our discussions of technical details and promising research directions, as well as the late nights we spent working toward deadlines, will always remain among my fondest memories. How fortunate I am to have met someone with such a gentle nature, an indomitable spirit, remarkable talent, and an extraordinary work ethic. I also extend my special thanks to Professor Mingyuan Zhou, Dr. Junfeng He, and Dr. Michal Lukasik for their generous support and collaboration.

I am also grateful to Jose Salas, Chie Ryu, Kyle Chang, Mike Kang, and Laurie Leyden for their exceptional administrative support at our department. Their care and efficiency made the practical side of this project remarkably smooth from beginning to end.

I would also like to thank every collaborator who has shared in my research: Jiahui Huang, Zhi Zhang, Peiyu Yu, Shu Wang, Zhendong Wang, Kevin Lin, Xiaofei Wang, Zhengyuan Yang, Linjie Li, Cheng-Han Chiang, Jianwen Xie, Lijuan Wang, Nanzhu Wang, Wenxiao Zhao, Renxi Wang, Yi Yi, Chengyuan Ma, Deqian Kong, Pan Lu, Benjamin Z. Yao, Dinghuai Zhang, Hengzhi He, Xiaojian Ma, Ruiyao Miao, Yifan Lu, Ruiqi Gao, Guang Cheng, Chris Chow, Yanchao Sun, Haochen Zhang, Eric Hanchen Jiang, Han Liu, Furong Huang, Yuchen Cui, Oscar Hernan Madrid Padilla, Yaxuan Zhu, Yingshan Chang, Feng Gao, Jacob Zhiyuan Fang, Yonatan Bisk, Hengfang Wang, Xiaojun Mao, Zhonglei Wang, Johannes Hertrich, Hok Shing Wong, Alexander Denker, Stanislas Ducotterd, Zhenghan Fang, Markus Haltmeier, Zeljko Kereta, Erich Kobler, Mohammad Sadegh Salehi, Carola-Bibiane Schönlieb, Johannes Schwab, Zakhar Shumaylov, Jeremias Sulam, German Shama Wache, Martin Zach, Matthias J. Ehrhardt, and Sebastian Neumayer. Each collaboration has shaped how I think, work, and understand research, and I am grateful for all that I have learned from you.

Beyond research, I want to thank Binwei Yao and Ruiyi Wang for your enduring friendship and for the experiences we have shared. The lives we have lived alongside one another have given us a deep mutual understanding and allowed us to support each other in ways that

words cannot fully capture. I cannot name every friend who has appeared in my life, and I am sorry that I cannot thank each of you individually here. Please know, however, that I have always appreciated your presence and every moment of joy we have shared.

Finally, I thank my parents. My deepest wish for you is that you enjoy your lives fully and embrace with open hearts all that has happened and all that is yet to come.

VITA

2022–present Ph.D. student in Statistics, University of California, Los Angeles.

2018–2022 B.S. in Data Science and Big Data Technology, Fudan University.

Spring & Student Researcher, ByteDance Seed.

Summer 2026

Spring & Student Researcher, Google Research.

Summer 2025

Summer 2025 Applied Scientist Intern, Amazon AWS AI Labs.

PUBLICATIONS

*Equal contribution. †Equal advising.

Jiahui Huang*, **Yasi Zhang***, Tianyu Chen, Shu Wang, Jianwen Xie, Oscar Leong, Mingyuan Zhou, Nanzhu Wang, and Ying Nian Wu. *MT-EditFlow: Reinforcement Learning for Multi-Turn Image Editing with Flow Matching*. **Under Review**.

Wenxiao Zhao, Shu Wang, Renxi Wang, Yi Yi, **Yasi Zhang**, Chengyuan Ma, Deqian Kong, Pan Lu, Ying Nian Wu, and Benjamin Z. Yao. *BoardGameArena: A Multi-Dimensional Benchmark for Strategic Reasoning of LLMs in Board Games*. **Under Review**.

Yasi Zhang*, Tianyu Chen*, Mingyuan Zhou, Oscar Leong, Ying Nian Wu, and Michal Lukasik. *REAL: Regression-Aware Reinforcement Learning for LLM-as-a-Judge*. **ICML 2026**.

Tianyu Chen*, **Yasi Zhang***, Zhi Zhang, Peiyu Yu, Shu Wang, Zhendong Wang, Kevin Lin, Xiaofei Wang, Zhengyuan Yang, Linjie Li, Cheng-Han Chiang, Jianwen Xie, Oscar Leong†, Lijuan Wang†, Ying Nian Wu†, and Mingyuan Zhou†. *EdiVal-Agent: An Object-Centric Framework for Automated, Fine-Grained Evaluation of Multi-Turn Editing*. **ICLR 2026**.

Yasi Zhang*, Tianyu Chen*, Zhendong Wang, Ying Nian Wu, Mingyuan Zhou†, and Oscar Leong†. *Score Distillation Beyond Acceleration: Generative Modeling from Corrupted Data*. **ICLR 2026**.

Peiyu Yu, Dinghuai Zhang, Hengzhi He, Xiaojian Ma, Ruiyao Miao, Yifan Lu, **Yasi Zhang**, Deqian Kong, Ruiqi Gao, Jianwen Xie, Guang Cheng, and Ying Nian Wu. *Latent Energy-Based Odyssey: Black-Box Optimization via Expanded Exploration in the Energy-Based Latent Space*. **ICLR 2026**.

Zhi Zhang, Chris Chow, **Yasi Zhang**, Yanchao Sun, Haochen Zhang, Eric Hanchen Jiang, Han Liu, Furong Huang, Yuchen Cui, and Oscar Hernan Madrid Padilla. *Statistical Guarantees for Lifelong Reinforcement Learning using PAC-Bayesian Theory*. **AISTATS 2025**.

Yasi Zhang, Peiyu Yu, Yaxuan Zhu, Yingshan Chang, Feng Gao, Ying Nian Wu, and Oscar Leong. *Flow Priors for Linear Inverse Problems via Iterative Corrupted Trajectory Matching*. **NeurIPS 2024**.

Yasi Zhang, Peiyu Yu, and Ying Nian Wu. *Object-Conditioned Energy-Based Attention Map Alignment in Text-to-Image Diffusion Models*. **ECCV 2024**.

Yingshan Chang, **Yasi Zhang**, Jacob Zhiyuan Fang, Ying Nian Wu, Yonatan Bisk, and Feng Gao. *Skews in the Phenomenon Space Hinder Generalization in Text-to-Image Generation*. **ECCV 2024**.

Hengfang Wang, **Yasi Zhang**, Xiaojun Mao, and Zhonglei Wang. *Transductive Matrix Completion with Calibration for Multi-Task Learning*. **ICASSP 2023**.

Johannes Hertrich, Hok Shing Wong, Alexander Denker, Stanislas Ducotterd, Zhenghan Fang, Markus Haltmeier, Zeljko Kereta, Erich Kobler, Oscar Leong, Mohammad Sadegh Salehi, Carola-Bibiane Schönlieb, Johannes Schwab, Zakhar Shumaylov, Jeremias Sulam, German Shama Wache, Martin Zach, **Yasi Zhang**, Matthias J. Ehrhardt, and Sebastian Neumayer. *Learning Regularization Functionals: A Comparative Study*. **Handbook of Numerical Analysis 2026**.

Yasi Zhang and Oscar Leong. *Learning Difference-of-Convex Regularizers for Inverse Problems: A Flexible Framework with Theoretical Guarantees*. **Under Submission**.

CHAPTER 1

Introduction

Generative artificial intelligence is often organized around model architecture: diffusion models, flow matching, autoregressive models, or reinforcement learning. In deployment, however, the more consequential distinction is between an open-loop generator and a closed-loop system. An open-loop model produces an output from a prompt or measurement. A closed-loop system must also determine whether the output satisfies a structured request, translate that judgment into a useful learning signal, and remain reliable when either the supervision or the observations are incomplete.

This dissertation studies that loop through three connected problems: *evaluation*, *alignment*, and *generation*. Evaluation defines what successful behavior means and makes failure measurable. Alignment turns the resulting signal into an optimization objective that respects the structure of the task. Generation supplies models and priors that can operate when clean data are unavailable or inverse problems are ill posed. The six chapters that follow develop two complementary contributions in each part.

1.1 Evaluation

Instruction-based image editing exposes the limitations of aggregate similarity scores. A successful edit must change the requested object while preserving unrelated content, maintaining perceptual quality, and supporting iterative refinement. Chapter 2 introduces an object-centric evaluation framework that separates instruction following, content consistency,

and visual quality over multiple turns. Chapter 3 closes the first feedback loop by using these structured signals to train flow-matching editors over complete editing trajectories.

1.2 Alignment

The same principle extends beyond visual editing: the reward must preserve the structure of the desired judgment. Chapter 4 studies numeric LLM-as-a-Judge tasks, where binary correctness discards the distance between a prediction and its target. Regression-aware reinforcement learning instead optimizes graded outcomes and correlation. Chapter 5 addresses alignment inside text-to-image diffusion by matching object-conditioned attention maps, targeting attribute binding and catastrophic object neglect.

1.3 Generation

The final part considers settings in which clean training examples or direct observations are not available. Chapter 6 distills a corruption-aware diffusion teacher into a one-step generator trained only from degraded data. Chapter 7 treats a pretrained flow-matching model as an image prior and develops an efficient approximation to maximum-a-posteriori inference for linear inverse problems.

1.4 Thesis Organization

The progression is intentional. Evaluation reveals failure modes and produces fine-grained signals; alignment determines how those signals should guide learning; generation incorporates the resulting principles into models that must function under realistic data constraints. The concluding chapter returns to this closed-loop view and identifies shared limitations and future directions.

Part I

Evaluation

CHAPTER 2

EdiVal-Agent: An Object-Centric Framework for Automated, Fine-Grained Evaluation of Multi-Turn Editing

2.1 Introduction

What truly defines the success of an image editor? At its core, editing requires making targeted, instruction-driven changes while preserving contextual consistency and perceptual realism—often across multiple refinement turns. Yet current evaluation practice struggles to capture this multifaceted objective.

When ground-truth edited images are available, a common strategy is to compare model outputs against these references (e.g., MagicBrush [ZMC23], UltraEdit [ZMC24], AnyEdit [YCY25], EmuEdit [SPS24]). Typical metrics include pixel-level distances (e.g., L1/L2) and semantic similarities (e.g., DINO [CTM21] and CLIP [RKH21a]). While informative, such metrics suffer from two structural issues: (i) the space of acceptable edits is inherently large, whereas a single reference provides only one realization; and (ii) references are frequently synthesized by existing editing models (e.g., Prompt-to-Prompt [HMT23a], SDXL [PEL24], DALLÉ-2 [RDN22a]), thereby importing their biases and limitations into the evaluation itself. Consequently, high reference similarity does not necessarily imply faithful instruction following, preservation of irrelevant content, or aesthetically plausible outcomes.

A complementary line of work employs zero-shot VLMs as interpretable evaluators (e.g.,

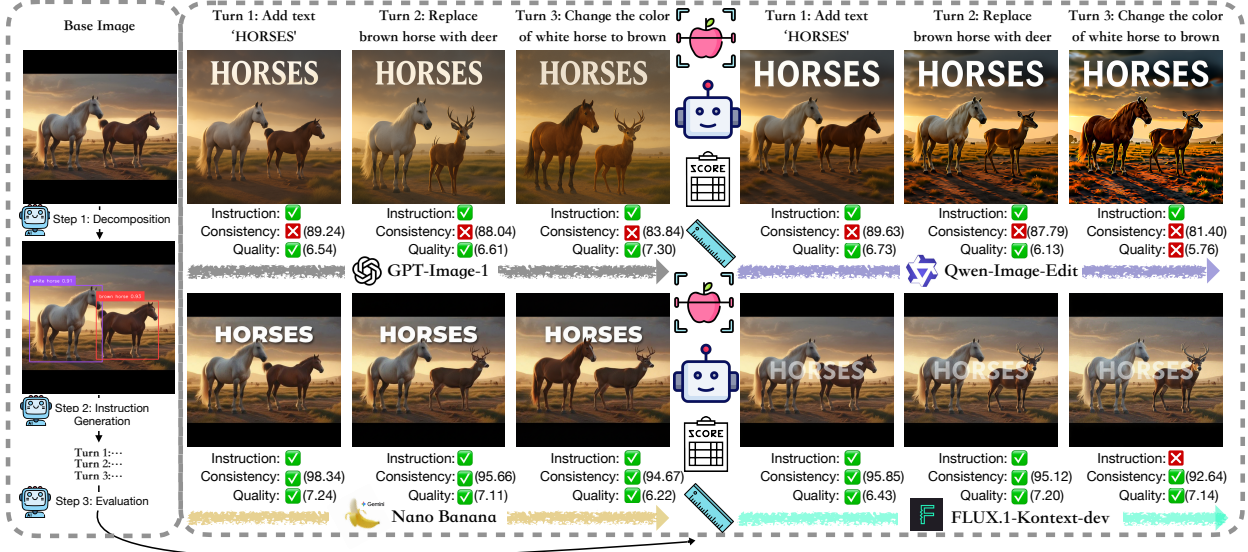


Figure 2.1: Overview of our workflow and representative model’s performance. For visualization, we adopt two thresholds: a consistency score of at least 90 and a visual quality score of at least 6. Details of the automated evaluation pipeline are provided in Figure 2.2 and Section 2.2. In multi-turn editing, models exhibit distinct weaknesses: *GPT-Image-1* struggles with content consistency, *Qwen-Image-Edit* underperforms in both visual quality and content consistency, and *FLUX.1-Kontext-dev* lags in instruction following, whereas *Nano Banana* shows no single dominant weakness.

VIEScore [KJW23], GEdit-Bench [LHX25], I²EBench [MJY24], HQ-Edit [HYZ24], Complex-Edit [YHZ25], and ImgEdit [YHL25]) and queries VLMs about specific aspects of an edit. While VLMs offer holistic, language-mediated judgments, they remain insufficient for precise editing assessment for several reasons. First, for instruction-following evaluation, they are notoriously poor at spatial reasoning [ZHL25, CXK24, CZF25] and are prone to object hallucinations in existence, category, attributes, and relations [BWX24]. These issues together undermine their ability to assess common object-related edit instructions. Second, they have limited sensitivity to pixel-level changes and frequently miss subtle, localized modifications [VNT25] (e.g., fine structures, small attribute shifts, etc.), which are crucial for evaluating content consistency. Third, since they are predominantly pretrained on natural images rather than synthetic generations, their priors are miscalibrated for artifacts and aesthetics, leading to failures in detecting common generative defects (e.g., extra fingers) and in modeling perceptual “naturalness” [LHL24, XLW23, MWS25], which humans are sensitive to. Consequently, VLM-only scoring lacks the precision and reliability required for fine-grained evaluation across instruction following, content consistency, and visual quality. However, we find recently open-source state-of-the-art editing models (e.g., Qwen-Image-Edit [WLZ25], Step1X-Edit [LHX25], Bagel[DZL25]) *solely* rely on VLMs for evaluation.

To address these challenges, we introduce **EdiVal-Agent**: an automated and fine-grained evaluation agent for multi-turn instruction-based image editing from an object-centric perspective, designed to assess not only standard single-turn but also multi-turn instruction-based editing with precision. As shown in Fig. 2.2, **EdiVal-Agent** first decomposes each image into semantically meaningful objects, then synthesizes diverse, context-aware editing instructions while dynamically updating object pools across turns. These two stages enable two novel object-centric metrics tailored for multi-turn evaluation and one global metric of visual quality: 1) EdiVal-IF, which measures instruction following by combining open-vocabulary object detectors for symbolic checks with VLMs for semantic verification on detector-guided crops; 2) EdiVal-CC, which evaluates content consistency by calculating

Table 2.1: Key attributes of open-source edit benchmarks. Note that ImgEdit [YHL25] does not include multi-turn editing experiments in the paper.

Benchmark	# Size	Object-centric	Automated	Multi-Turn	Free from Ref. Images	Tools used
EmuEdit[SPS24]	3,055	✗	✓	✗	✗	L1, CLIP, DINO
MagicBrush[ZMC23]	1,053	✗	✗	✗	✗	L1, L2, DINO, CLIP
AnyEdit[YCY25]	1,250	✗	✗	✗	✗	L1, CLIP, DINO
I2EBench[MJY24]	2,240	✗	✗	✗	✓	VLM
GEdit-Bench[LHX25]	606	✗	✗	✗	✓	VLM
HQ-Edit[HYZ24]	1,651	✗	✓	✗	✓	VLM
ImgEdit-Bench[YHL25]	811	✗	✓	✗	✓	VLM
Kontext-Bench[LBB25]	1,026	✗	✗	✗	✓	Human Annotation
EdiVal-Bench (ours)	1,716	✓	✓	✓	✓	Detector, VLM, L1, DINO, HPS

semantic similarity of unchanged objects and background using the evolving object pools; and 3) EdiVal-VQ, which quantifies changes in overall visual quality with human preference models. We show that EdiVal-IF yields stronger agreement with human judgments in instruction-following evaluation than thresholded CLIP directional (CLIP_dir) scores [GPM22] or VLMs alone, as evidenced in Sec. 2.2.5.

Instantiating the agentic pipeline, we curate a new multi-turn image editing benchmark, **EdiVal-Bench**, featuring 9 instruction types and 16 existing editing models—spanning in-context, flow-matching, and diffusion paradigms, across both closed- and open-source systems—conduct fine-grained analyses, and draw actionable insights. Empirically, as demonstrated in Fig. 2.1 and Tab. 2.3, *GPT-Image-1* excels at instruction following yet ranks near the bottom in content consistency, whereas *Seedream 4.0* and *Nano Banana* perform strongly on both axes. In addition, open-source models such as *Qwen-Image-Edit* degrade significantly in instruction following and visual quality as the number of editing turns increases, while *FLUX.1-Kontext-max* and *FLUX.1-Kontext-dev* lag in instruction following. We further contrast multi-turn editing with single-shot complex prompts [YHZ25], highlighting complementary strengths and failure modes. We hope that our agent pipeline, benchmark, and analyses accelerate the transition of multi-turn editing toward practical applications.

Key contributions. 1) *Agent*: **EdiVal-Agent** is a fully automated evaluator that performs object-centric decomposition, generates diverse multi-turn editing instructions, and measures overall editing quality using two object-centric metrics (EdiVal-IF and EdiVal-CC) plus EdiVal-VQ for visual quality. 2) *Benchmark*: using **EdiVal-Agent**, we construct **EdiVal-Bench** with 1,716 instructions across nine types and three turns on 572 real-world images, with comparisons to prior benchmarks in Tab. 2.1. 3) *Human agreement*: EdiVal-IF attains 81.3% agreement with human ratings for instruction following, outperforming zero-shot VLMs and CLIP-based baselines. 4) *Evaluation*: we assess 13 editors (diffusion, flow-matching, and close source) along instruction following, content consistency, and visual quality. 5) *Insights*: overall ranking—Seedream 4.0 > Nano Banana > FLUX.1-Kontext-max > GPT-Image-1; the strongest open-source editor, Qwen-Image-Edit, exhibits exposure bias under multi-turn editing. 6) *Artifacts & settings*: we reveal luminance drift across turns, and contrast multi-turn against complex single-shot editing to delineate strengths and weaknesses across model families.

2.2 EdiVal-Agent

2.2.1 Overview

As illustrated in Fig. 2.2. The pipeline comprises three stages: (1) *Decomposition* uses a VLM (e.g., GPT-4o; other VLMs are viable alternatives) to extract structured, object-level descriptions—objects, attributes, and relations—enabling symbolic reasoning; (2) *Instruction Generation* produces multi-turn, diverse, compositional prompts by maintaining an explicit object pool and sampling from nine instruction types spanning subject-, attribute-, relational-, text-, count-, and global-level edits; (3) *Evaluation* scores edited images with EdiVal-IF, EdiVal-CC, and EdiVal-VQ.

2.2.2 Step 1: Decomposition

Given an image, a VLM-based agent parses clearly visible foreground objects and returns per-object JSON with fields `object`, `color`, `material`, `text`, `count`, and a boolean `foreground`. Names follow "{material} {color} {object}"; unknown fields are omitted; person identity is never recorded (only wearables/accessories). Example: {"metal yellow sign": {"object": "sign", "color": "yellow", "material": "metal", "text": "SCHOOL", "count": 1, "foreground": true}}. An aggregated `all_objects` string concisely lists objects (e.g., "metal yellow sign . metal brown pole"). We apply this stage to GEdit-Bench [LHX25] (606 images), exclude 34 images with sensitive personal content, and retain 572 images. After extraction, Grounding-DINO validates objects and detects bounding boxes; only reliable detections are kept to seed instruction generation and evaluation. The filtered objects are stored in the `All_Objects_Pool` and later used to initialize three distinct object pools that dynamically track the evolving state of instruction generation.

2.2.3 Step 2: Instruction Generation

From the decomposed scene, the agent generates multi-turn edits that are grounded in the current object state. The instruction taxonomy (nine types; six categories) appears in Table 2.2. We maintain three evolving pools at turn t : $\mathcal{P}_t^{\text{all}}$ (all objects ever present), $\mathcal{P}_t^{\text{unch}}$ (original objects not edited up to t), and $\mathcal{P}_t^{\text{avail}}$ (objects currently editable). With a turn budget `MAX_TURNS`, at each turn the agent (i) selects a type—defaulting to `subject_add` if $\mathcal{P}_t^{\text{avail}} = \emptyset$, otherwise sampling a type not yet used in the chain; (ii) selects object(s) from $\mathcal{P}_t^{\text{avail}}$; (iii) emits a natural-language instruction via GPT-4o referencing those objects and the scene state; and (iv) updates $\mathcal{P}_{t+1}^{\text{all}}$, $\mathcal{P}_{t+1}^{\text{avail}}$, and $\mathcal{P}_{t+1}^{\text{unch}}$ according to the intended edit. When a `background_change` edit applies at turn t , background-consistency scoring is disabled since this turn, and we append "make {objects_in_foreground} unchanged" to the instruction to preserve object-level comparability, where `objects_in_foreground` =

$\{o \in P_t^{\text{avail}} : o.\text{foreground} = \text{true}\}$. The loop is adaptive by expanding/contracting $\mathcal{P}_t^{\text{avail}}$ and naturally compositional. Our default sets **MAX_TURNS**= 3 (In our implementation, each turn is assigned a distinct instruction type.), though longer chains are easily obtained by allowing repetition or adding types.

Table 2.2: **Instruction types in EdiVal-Bench** created by **EdiVal-Agent**, grouped by semantic category. Counts are shown per turn (T1–T3).

Category	Instruction Type	Example Instruction	T1	T2	T3	Total
Subject-centric	subject_add	Add bench on the left of metal red fire hydrant.	67	77	93	237
	subject_remove	Remove wooden brown door.	75	69	61	205
	subject_replace	Replace stone gray railing with wooden fence.	54	57	55	166
Attribute-centric	color_alter	Change the color of metal white air-plane to blue.	56	73	57	186
	material_alter	Change the material of plastic black pen to metal.	66	50	72	188
Text-related	text_change	Replace the text 'BEARS CONTROL' on cotton black cap with 'WILD PATH'.	64	70	54	188
Relational	position_change	Change the position of ceramic white cup to right of plastic white laptop.	52	63	48	163
Counting	count_change	Change the count of fur brown bear to 3.	73	58	60	191
Global	background_change	Change the background to forest, remain the brown fur bear unchanged.	65	55	72	192

2.2.4 Step 3: Evaluation

The first two stages enable two novel object-centric metrics for multi-turn editing evaluation for instruction following and content consistency, respectively, and one global metric for visual quality:

EdiVal-IF To evaluate instruction following, we introduce EdiVal-IF, which assesses multi-turn edits by comparing the image from the previous turn, I^{t-1} , to the current image, I^t . For a given instruction P^t at turn t , the score is determined differently for symbolically and semantically verifiable tasks. Symbolically verifiable types (T_{sym})—such as `subject_add`, `subject_remove`, `subject_replace`, `position_change`, and `count_change`—are evaluated using an open-vocabulary object detector $\mathcal{M}_{\text{detect}}$ [LZR24a]. The detector’s outputs, including bounding boxes and confidence, are assessed against geometric and logical criteria $\mathcal{F}_{\text{sym}}$ derived from the instruction. For example, for a `position_change` instruction “Move [A] to the left of [B]”, $\mathcal{F}_{\text{sym}}$ verifies that the x-coordinate of A’s bounding box $\mathcal{B}$ center is less than that of B in I^t , i.e., $\text{center}_x(\mathcal{B}_A^t) < \text{center}_x(\mathcal{B}_B^t)$. In this case,

$$\text{EdiVal-IF}(I^t, I^{t-1}, P^t \in T_{\text{sym}}) = \mathcal{F}_{\text{sym}}(\mathcal{M}_{\text{detect}}(I^{t-1}, I^t | P^t)). \quad (2.1)$$

Semantically verifiable types (T_{sem})—`color_alter`, `material_alter`, `text_change`, and `background_change`—are evaluated with a VLM $\mathcal{M}_{\text{VLM}}$ [YYZ24]. To focus the evaluation, the VLM is applied to detector-guided object crops (I_o) using instruction-specific templates.

$$\text{EdiVal-IF}(I^t, I^{t-1}, P^t \in T_{\text{sem}}) = \mathcal{M}_{\text{VLM}}(I_o^{t-1}, I_o^t | P^t) = \mathcal{M}_{\text{VLM}}(\mathcal{M}_{\text{detect}}(I^{t-1}, I^t | P^t)). \quad (2.2)$$

We show that EdiVal-IF achieves superior human agreement (Sec. 2.2.5). The multi-turn editing success rate is defined as the logical AND of the EdiVal-IF scores across all edits along the chain, whereas the marginal task rate at turn t is defined according to the formulas 2.1 and 2.2 provided above.

EdiVal-CC To assess content consistency, EdiVal-CC measures the preservation of non-target content between the base image I^0 and the current image I^t . Given editing instructions $P^{1:t}$ from turn 1 to turn t , the object pools $\mathcal{P}_t^{\text{unch}}$ and $\mathcal{P}_t^{\text{all}}$ are dynamically updated. Let Ω denote the entire image area. Using object bounding boxes from the base image ($\mathcal{B}_o^0$) and the current image ($\mathcal{B}_o^t$), extracted by the detector $\mathcal{M}_{\text{detect}}$, the background region is defined as $\Omega_{\text{bg}}^t = \Omega - \bigcup_{o \in \mathcal{P}_t^{\text{all}}} (\mathcal{B}_o^0 \cup \mathcal{B}_o^t)$, i.e., the region obtained by excluding all objects that have appeared. Background consistency is then computed as $s_{\text{bg}}^t = \phi(I_{\text{bg}}^0, I_{\text{bg}}^t)$, where $I_{\text{bg}}^t = \Omega_{\text{bg}}^t \circ I^t$ denotes the background of the image, and ϕ is a similarity function such as L_1 distance or DINO-based similarity. For unchanged objects, we compute the per-object consistency $s_o^t = \phi(I_o^0, I_o^t)$ for each $o \in \mathcal{P}_t^{\text{unch}}$, and then average them. Formally, the final EdiVal-CC score emphasizes semantic preservation by averaging the feature-level similarities of the background and unchanged objects (see Appendix. 2.5.5.1 for details):

$$\text{EdiVal-CC}(I^t, I^0, P^{1:t}) = \frac{1}{2} \left(s_{\text{bg}}^t + \frac{1}{|\mathcal{P}_t^{\text{unch}}|} \sum_{o \in \mathcal{P}_t^{\text{unch}}} s_o^t \right). \quad (2.3)$$

EdiVal-CC aligns with the *intuitive* notion of consistency, providing a precise measurement.

EdiVal-VQ. Zero-shot VLMs are not trained for reliable assessment of image quality—particularly artifacts and aesthetics—and we find they are imprecise as scoring functions (see Appx. 2.5.5). Consequently, we adopt Human Preference Score v3 (HPSv3) [MWS25] as our visual-quality (VQ) evaluator. In practice, applying preference models to unedited, real photographs often yields relatively low aesthetic scores. We also observe divergent behaviors across editors (See Fig. 2.3): some (e.g., GPT-Image-1) tend to *beautify* images and increase HPSv3, whereas others (e.g., FLUX.1-Kontext-max) *preserve* the original appearance with minimal aesthetic change. Because aesthetic preference is inherently user- and task-dependent, and beautification can trade off with content consistency (already incorporated into our overall score), we report EdiVal-VQ separately and do not fold it into the aggregate metric.

EdiVal-O. We aggregate *Instruction Following* (EdiVal-IF) and *Content Consistency* (EdiVal-CC) into a single overall score. Since both metrics are unit-free and normalized to $[0, 1]$ but capture complementary aspects, we follow prior work and use the geometric mean to balance them and penalize imbalance [LHX25, KJW23]. Formally, $\text{EdiVal-O} = \sqrt{\text{EdiVal-IF} \cdot \text{EdiVal-CC}}$.

Design Scope and Limitations. We omit `style_change` from our taxonomy because style categories are inherently ill-defined, which makes instruction-following (EdiVal-IF) difficult to evaluate reliably. Extending **EdiVal-Agent** with style-aware recognition is promising future work. After language-based extraction, we validate objects using Grounding-DINO [LZR24b] and prune low-confidence or ambiguous detections. This stabilizes the object pool and reduces error propagation during instruction generation and IF evaluation. By default, we employ Grounding-DINO as the open-vocabulary detector, Qwen2-VL as the VLM, and DINOv3 [SVS25] as the image feature extractor due to their state-of-the-art performance and open-source availability, which facilitates community use. The agentic pipeline is tool-agnostic and can be readily strengthened by substituting more advanced modules in the future.

2.2.5 Measuring Human Agreement

Setup. We conduct human study on edits made by four exemplary models, Step1X-Edit, AnyEdit, Gemini 2.0 Flash and Flux.1-Kontext-dev, on **EdiVal-Bench**, generated by **EdiVal-Agent** as described in Sec. 2.2.3. In total, we collect 4,576 human annotations of edits. During evaluation, raters were shown the original image, the edited image, and the corresponding instruction, and asked a binary question: “*Evaluate whether the edited image successfully follows the given instruction.*”

Results. Figure 2.4 summarizes the findings. EdiVal-IF achieves a human agreement accuracy of **81.3%**, significantly higher than VLM-only (*Qwen-2-VL*, 75.2%), CLIP_dir (65.4%), and other zero-shot VLMs. These results verify that integrating VLMs reasoning with object detection leads to better alignment with human judgment compared to existing methods. The inter-annotator’s agreement rate (85.5%) indicates the best performance any evaluation tool can reach.

We attribute the improvement in instruction-following evaluation to two factors. First, for symbolically verifiable instruction types—`subject_add`, `subject_remove`, `subject_replace`, `position_change`, and `count_change`—EdiVal-IF relies solely on Grounding-DINO. It determines the success of an edit by checking object presence/absence, the positions of object centers, and the number of bounding boxes. Results for `position_change` and `subject_remove` show that these fixed rules, combined with Grounding-DINO, can significantly outperform Qwen2-VL in edit evaluation. We hypothesize that errors in `position_change` stem from poor spatial reasoning, while failures in `subject_remove` are due to hallucinations regarding object existence. Second, semantically verifiable types—`color_alter`, `material_alter`, `text_change`, and `background_change`—are evaluated using Qwen2-VL combined with Grounding-DINO. The decomposition stage in **EdiVal-Agent** can supports evaluation by localizing text regions, enabling the LLM to reason more precisely about text edits. These findings indicate that EdiVal-IF not only enhances interpretability but also improves the practical applicability of evaluation pipelines in real-world settings that demand human-like understanding. Nonetheless, EdiVal-IF has failure modes, which we document and analyze in Appendix. 2.5.2.

2.3 Benchmarking Multi-turn Editing

Summary of Results. Table 2.3 shows that the benchmark has a very tight top tier. *Seedream 4.0* remains the best overall model (Rank 1), combining the strongest

Table 2.3: **Results of multi-turn editing.** EdiVal-IF, EdiVal-CC, and EdiVal-O across three sequential editing turns. Best per column in **dark red**; second-best in **lighter red**. * indicates a closed-source model, and † indicates an open-source model.

Technique	Model	In-Context	Date	Latency (s/img)	EdiVal-IF			EdiVal-CC			EdiVal-O			Rank
					T1	T2	T3	T1	T2	T3	T1	T2	T3	
Unknown	* Seedream 4.0 🐸	✗	25.09.10	14.55	75.93	55.58	41.59	92.51	88.03	85.86	83.81	69.95	59.76	1
	* GPT-Image-1.5 🐸	✓	25.12.16	35.55	75.19	55.92	40.08	94.49	91.20	88.49	84.29	71.41	59.55	2
	* Nano Banana 2 🐸	✓	26.03.01	23.79	73.89	54.17	38.61	93.54	90.52	88.61	83.14	70.02	58.49	3
	* Nano Banana	✓	25.08.26	9.20	70.70	50.66	35.35	93.91	90.48	89.48	81.48	67.70	56.24	5
	* GPT-Image-1	✓	25.07.16	26.47	73.12	54.89	38.35	81.00	77.78	75.50	76.96	65.34	53.81	6
	* Gemini 2.0 Flash	✓	25.02.05	8.34	68.07	45.96	28.42	90.58	85.10	80.88	78.52	62.54	47.94	8
Flow Matching	* FLUX.2-max	✗	25.12.16	36.87	75.55	55.27	39.36	92.91	88.30	85.78	83.78	69.86	58.10	4
	* FLUX.1-Kontext-max	✗	25.06.03	10.13	69.49	46.89	31.83	93.93	90.90	88.40	80.79	65.29	53.04	7
	† Qwen-Image-Edit	✗	25.08.04	115.08	72.90	44.06	22.55	84.22	80.52	77.98	78.36	59.56	41.93	9
	† StepIX-Edit	✗	25.04.25	20.42	61.89	34.97	17.83	92.76	88.52	85.21	75.77	55.64	38.98	10
	† FLUX.1-Kontext-dev	✗	25.06.25	29.21	59.97	32.69	16.61	95.32	92.24	90.22	75.61	54.91	38.71	11
	† OmniGen	✗	24.09.11	19.70	54.72	24.48	10.66	93.00	88.42	83.92	71.34	46.52	29.91	12
Diffusion	† AnyEdit	✗	24.11.24	3.93	41.07	16.32	7.22	86.42	78.91	70.10	59.58	35.89	22.50	14
	† UltraEdit	✗	24.07.07	3.15	51.37	17.70	6.36	86.80	84.50	82.40	66.78	38.67	22.89	13
	† MagicBrush	✗	23.06.16	4.08	42.31	15.73	4.90	86.96	81.26	76.86	60.66	35.75	19.41	15
	† IP2P	✗	23.12.15	4.09	37.41	10.66	2.80	76.85	68.36	60.30	53.62	26.99	12.99	16

instruction-following performance at T1/T3 (75.93/41.59), strong cross-turn consistency (92.51/88.03/85.86), and relatively low latency (14.55s/img). *GPT-Image-1.5* is a close second and achieves the highest overall *EdiVal-O* at T1/T2 (84.29/71.41), substantially outperforming *GPT-Image-1*; compared with its predecessor, it improves *EdiVal-CC* by 13.49/13.42/12.99 points and *EdiVal-O* by 7.33/6.07/5.74 points across the three turns, although at higher latency (35.55 vs. 26.47s/img). A similar trend appears in the Nano series: *Nano Banana 2* consistently improves over *Nano Banana* in instruction following and overall quality, raising *EdiVal-O* from 81.48/67.70/56.24 to 83.14/70.02/58.49, but these gains come with a large speed penalty (23.79 vs. 9.20s/img) and only marginal changes in consistency. Among the flow-matching baselines, *FLUX.2-max* is the strongest competitor, while *Qwen-Image-Edit* still degrades sharply as the number of turns increases (78.36→59.56→41.93 in *EdiVal-O*), indicating pronounced error accumulation. Overall, newer model versions improve multi-turn robustness and instruction retention, but often by trading off latency. Among open-source systems, *Qwen-Image-Edit* performs well initially (*EdiVal-O* 78.36 at T1) but degrades rapidly with additional turns, likely due to exposure bias as discussed below. We can see that there is a clear gap between the performance of closed-source and open-source systems. With the exception of *Qwen-Image-Edit*, our model rankings exactly

match those reported on the Artificial Analysis leaderboard (rank by human vote) as of September 12, 2025; see Appendix. 2.5.1.

2.3.1 Instruction Following

Marginal Task Success Rate. For a given turn, the *marginal task success rate* (Eqns. 2.1 and 2.2) is the proportion of prompts for which the edit requested at that turn is successfully executed. By contrast, the *instruction-following* score in Table 2.3 reports the *multi-turn task success rate* at turn i : the logical AND of the EdiVal-IF scores across all edits along the chain. Figure 2.5 summarizes per-turn performance. High-ranking models—such as Seedream 4.0, Nano Banana, and FLUX.1-Kontext-max—maintain relatively stable EdiVal-IF across turns, even though Seedream 4.0 and FLUX.1-Kontext-max are *not* in-context editors (they do not condition on prior prompts or intermediate images). In contrast, several other models exhibit a clear decline in marginal success as the number of turns increases.

A particularly salient case is **Qwen-Image-Edit**. Although it is the strongest open-source system at turn 1 (EdiVal-O 78.36 vs. 81.48 for Nano Banana), its performance degrades more rapidly over subsequent turns. We hypothesize that this stems from *exposure bias* [NLS23, Sch19]: many single-turn editors are trained to operate on real images and ground-truth inputs rather than on their own previous outputs. When asked to edit their own generations, small distributional mismatches compound across turns, reducing stability; this effect is further aggravated when the model can only attend to a limited history.

Marginal Task Success Rate Across Instruction Types. We analyze marginal sub-task success rates across turns for different instruction types. The results for Nano Banana are shown in Fig. 2.6. Other editing models exhibit similar behavior. Nano Banana performs relatively well on attribute-centric tasks such as `color_alter` and `material_alter`, but poorly on `position_change` and `count_change`, indicating weaknesses in spatial and numerical reasoning, respectively.

2.3.2 Content Consistency

We evaluate two aspects: (i) **unchanged-object consistency** (Fig. 2.8), which measures whether objects that are not edited up to turn i remain unchanged, and (ii) **background consistency** (Fig. 2.7), which assesses whether the background remains stable when it is not explicitly modified. When calculating consistency, the grounding box is extracted from the raw input image and applied to all edited images. We therefore choose to report DINOv3 over L_1 distance for consistency computation because even small shifts in object location can lead to large variations in pixel-wise L_1 loss, even if unchanged objects are well preserved. By relying on DINO features, we ensure that consistency is measured semantically, capturing attributes such as object identity, attributes, and texture, etc. Nevertheless, the consistency scores from DINOv3 remain highly correlated with those computed using pixel-wise L_1 loss (See results in the Appendix. 2.5.6). Based on the results, the most consistent editing model is **FLUX.1-Kontext-dev**, followed by **Nano Banana** and **FLUX.1-Kontext-max**. In contrast, **GPT-Image-1** ranks near the bottom, showing notably poor consistency across turns. Representative qualitative examples are shown in Figure 2.8 and Figure 2.7.

2.3.3 Visual Quality

Besides EdiVal-VQ, we also report the absolute change in visual quality (VQ) relative to the base image: $\text{EdiVal-VQ}\Delta_i = |\text{EdiVal-VQ}_{\text{turn } i} - \text{EdiVal-VQ}_{\text{base}}|$. A smaller Δ indicates stronger style fidelity to the base image, whereas a larger Δ suggests greater beautification or stylistic drift. As summarized in Table 2.4, **GPT-Image-1** achieves the highest aesthetic scores across turns and the largest Δ , indicating a substantial stylistic shift (Fig. 2.3). In contrast, **GPT-Image-1.5** places greater emphasis on consistency, reducing Δ from 2.18 to 0.15. For preserving the base image’s appearance (small Δ), **Gemini 2.0 Flash** exhibits the least drift, with **Nano Banana** also performing strongly. We provide a low-level exposure statistics analysis in Appendix 2.5.4.

Table 2.4: EdiVal-VQ and EdiVal-VQ Δ results across turns. **dark red** denotes the *best* value in the column; **lighter red** denotes the *second-best*. For HPS, higher values are stronger aesthetics. For Δ , smaller values are stronger fidelity preservation.

Technique	Model	EdiVal-VQ			EdiVal-VQ Δ		
		T1	T2	T3	T1	T2	T3
Unknown	Seedream 4.0	5.14	5.15	5.15	0.76	0.77	0.77
	GPT-Image-1.5	4.16	4.08	4.23	0.22	0.30	0.15
	GPT-Image-1	6.65	6.59	6.56	2.27	2.21	2.18
	Nano Banana 2	6.59	6.51	5.95	2.21	2.13	1.57
	Nano Banana	4.94	5.12	5.26	0.56	0.73	0.88
	Gemini 2.0 Flash	4.44	4.23	4.07	0.05	0.15	0.32
Flow Matching	FLUX.2-max	5.03	5.21	5.36	0.65	0.83	0.98
	FLUX.1-Kontext-max	5.12	5.07	5.04	0.74	0.69	0.66
	Qwen-Image-Edit	5.86	5.72	5.15	1.47	1.34	0.77
	Step1X-Edit	4.06	3.34	2.76	0.33	1.04	1.63
	FLUX.1-Kontext-dev	5.12	5.07	5.04	0.73	0.69	0.65
	OmniGen	4.61	4.07	3.50	0.23	0.31	0.89
Diffusion	AnyEdit	3.66	2.80	1.95	0.72	1.58	2.44
	UltraEdit	4.79	4.68	4.36	0.41	0.30	0.02
	MagicBrush	3.85	3.08	2.36	0.54	1.30	2.02
	IP2P	3.20	2.38	1.44	1.18	2.01	2.94

2.3.4 Multi-turn Editing vs. Complex Editing

We compare two strategies for composing multiple edits. In *multi-turn* editing, instructions are executed sequentially—apply instruction 1, then apply instruction 2 to the result, and so on. In *complex* editing, we concatenate C instructions into a single prompt and perform one edit (“complex level” C , with $C \in \{1, 2, 3\}$). Empirically (Fig. 2.9), when a model does *not* suffer from exposure bias, multi-turn editing tends to yield higher success rates, consistent with a step-by-step “chain of edits” (analogous to chain-of-thought in reasoning). For instance, Nano Banana benefits from the multi-turn formulation. Conversely, when exposure bias is pronounced, compressing instructions into a single, complex prompt can perform better; see Qwen-Image-Edit in Fig. 2.9.

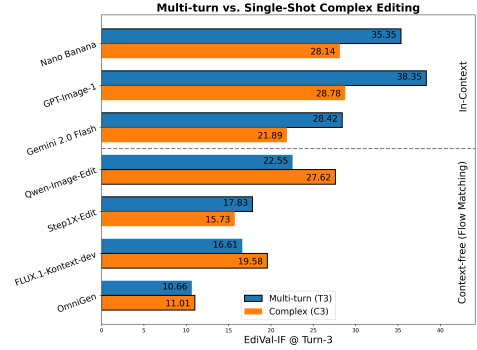


Figure 2.9: Turn-3 instruction following: Multi-turn vs. single-shot complex prompts.

2.3.5 Pareto Front

After constructing the leaderboard using EdiVal-O, we further analyze the trade-offs between different evaluation dimensions. To ensure that no model “games” the benchmark by excelling in only one dimension, we plot the Pareto boundary at Turn 3 for all pairwise combinations of our three evaluated dimensions: EdiVal-IF, EdiVal-CC, and EdiVal-VQ (see Figure 2.10). Additional Pareto plots for Turn 1 and Turn 2 are provided in Figures 2.16 and 2.17.

2.3.6 Tool Swap Analysis and Prompt Compression Strategies

We evaluate the robustness of our evaluation stack by systematically swapping key components, including the VLM (Appendix 2.5.11.1), the detector and its threshold (Appendices 2.5.11.3 and 2.5.11.2), and the image feature extractor (Appendix 2.5.11.4). Our analysis indicates that these modifications have only a negligible impact on the final evaluation outcomes. Additionally, we conduct an ablation study on prompt compression by testing three variants: random shuffling, sequential connectors, and explicit “keep-unchanged” constraints. The results demonstrate that these structural variations yield only a mild effect on the overall complex editing success rate, suggesting the model is relatively insensitive to the specific formatting of compressed instructions. More details are in Appendix 2.5.10.

2.4 Conclusion

We introduced **EdiVal-Agent**, an automated, and interpretable framework for evaluating instruction-based image editing. By leveraging symbolic object decomposition, structured instruction generation, and a hybrid evaluation pipeline integrating both specialist tools and vision-language reasoning models, **EdiVal-Agent** enables fine-grained, object-centric assessment of modern multi-turn editing systems. Limitations and discussions are deferred to Appendix 2.5.8.

2.5 Appendix

2.5.1 Artificial Analysis Leaderboard

We report the leaderboard from the Artificial Analysis website as of **September 12, 2025** (Fig. 2.11). To ensure a fair comparison, we align on the intersection of models evaluated by both platforms and *exclude* Qwen-Image-Edit. Among the overlapping systems—Seedream 4.0, Nano Banana (Gemini 2.5 Flash), GPT-Image-1 (GPT-4o), FLUX.1-Kontext-max, and Gemini 2.0 Flash—the relative ordering of human votes on Artificial Analysis matches our EdiVal rankings exactly (Table 2.5), supporting the accuracy of our methodology.

Table 2.5: **Model rankings on the overlapping set.** Relative ranks from Artificial Analysis (human votes) vs. EdiVal (ours) as of Sep 12, 2025.

Model	Artificial Analysis (Rank)	EdiVal (Rank)
Seedream 4.0	1	1
Nano Banana (Gemini 2.5 Flash)	2	2
GPT-Image-1 (GPT-4o)	3	3
FLUX.1-Kontext-max	4	4
Gemini 2.0 Flash	5	5

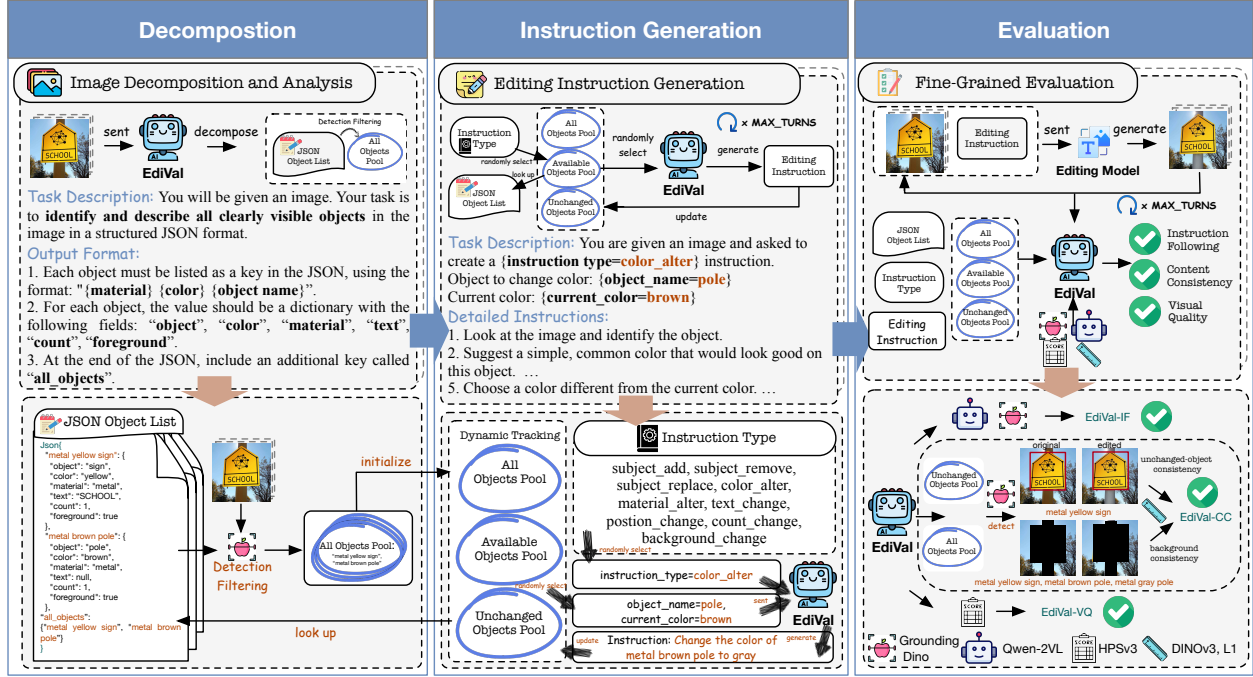


Figure 2.2: Framework of **EdiVal-Agent**. It first decomposes images into semantically meaningful objects, such as *metal yellow sign* and *metal brown pole*, and identifies their contextual relationships, e.g., they are both in *foreground*. It then generates diverse and proper editing scenarios at scale which are based on the initial analysis, e.g., *Change the color of metal brown pole to gray*. Finally, it systematically evaluates editing model outputs from multiple axes with our proposed metrics: EdiVal-IF, EdiVal-CC, and EdiVal-VQ. Our agentic pipeline is agnostic to the expert tools used and can be readily enhanced with more advanced tools in the future.

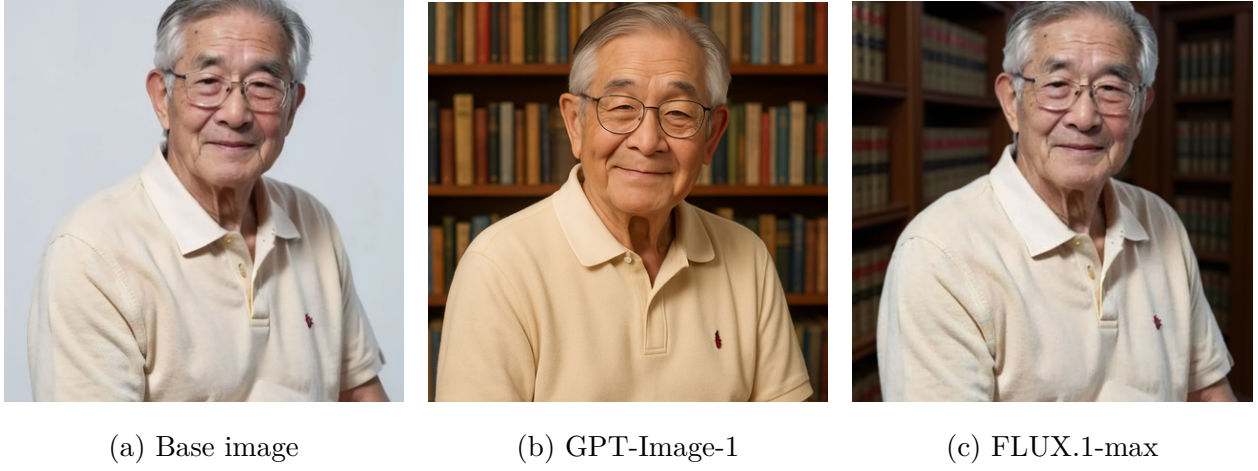


Figure 2.3: **Beautification vs. preservation** under the prompt: “*Change the background to a library.*” GPT-Image-1 tends to increase HPSv3 via beautification, while FLUX.1-Kontext-max emphasizes fidelity to the input.

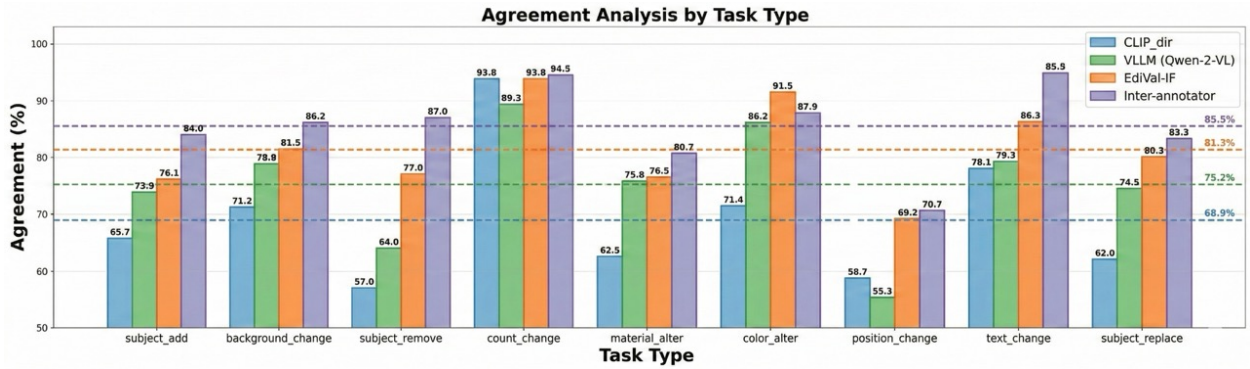


Figure 2.4: Results of human agreement. Dashed lines represent the average accuracy of each method. EdiVal-IF achieves 81.3% human agreement accuracy, significantly outperforming the VLM (Qwen2-VL) at 75.2% and thresholded CLIP_dir at 65.4%. Note that the CLIP_dir threshold is tuned separately for each task.

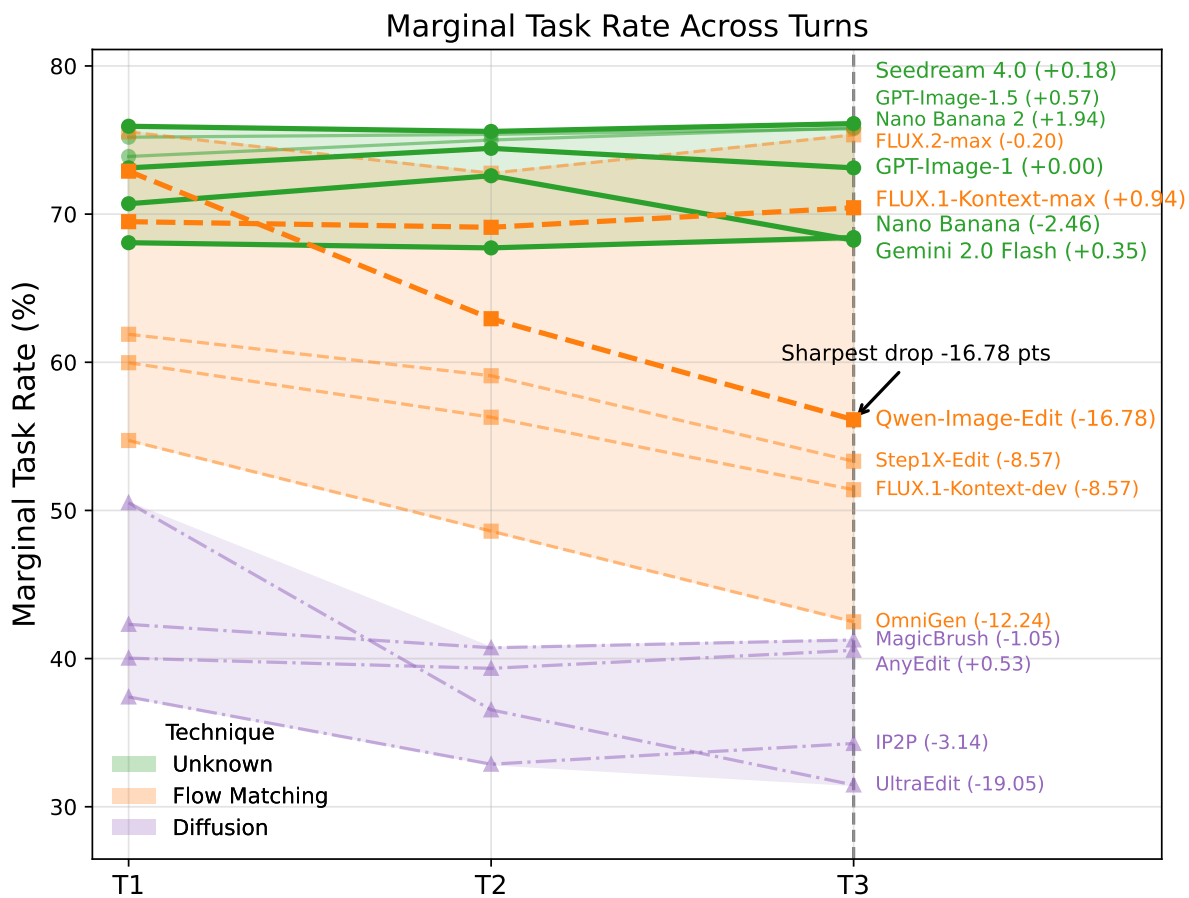


Figure 2.5: Marginal Task Success rate across turns.

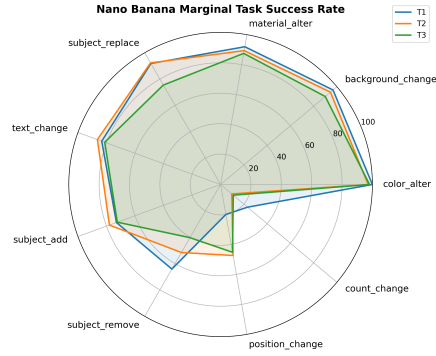


Figure 2.6: Marginal task success rate grouped by task types for Nano Banana.

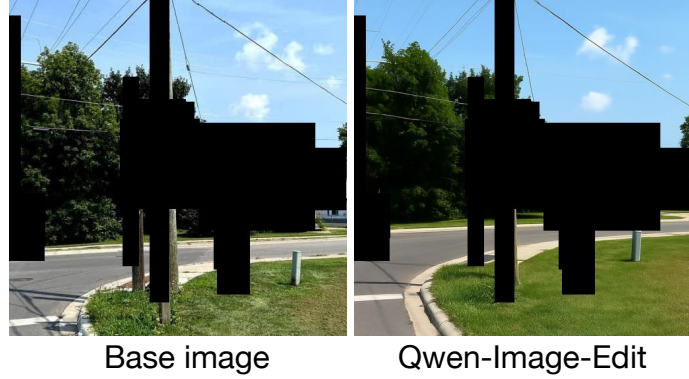


Figure 2.7: **Illustration of background consistency.** Instruction: “Remove beige brick house.” The grounding box is the union of all object regions from the raw and edited images.

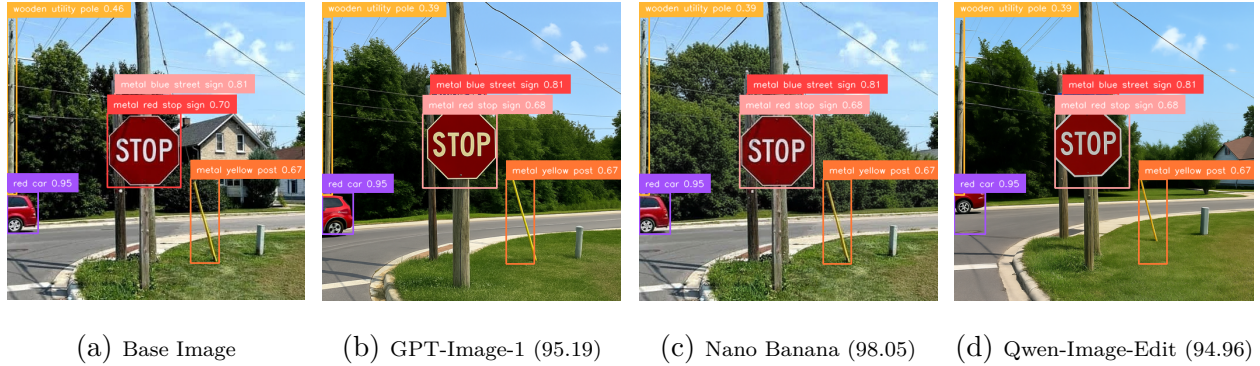


Figure 2.8: **Illustration of object consistency.** Instruction: “Remove brick beige house.” The grounding box, extracted from the raw input image, highlights the localized region used to compute unchanged-object consistency. The corresponding consistency score is shown in brackets.

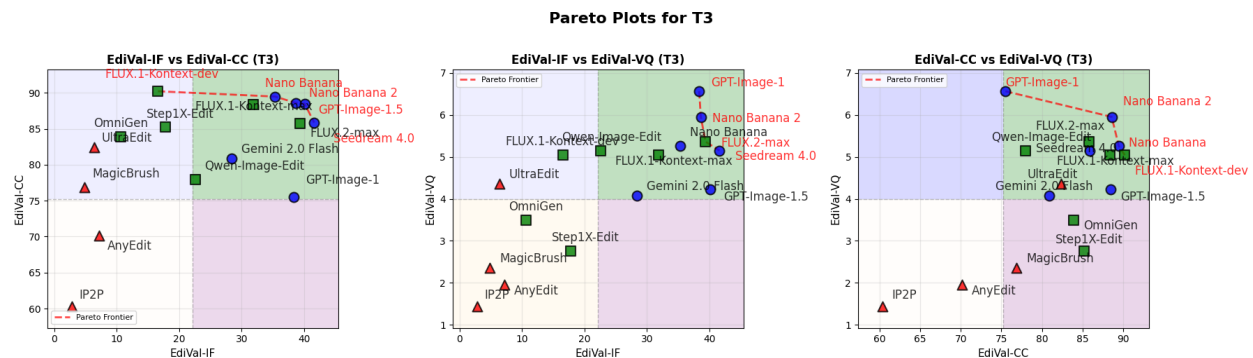


Figure 2.10: Pareto front plot for Turn 3 editing across EdiVal-IF, EdiVal-CC, and EdiVal-VQ.

Creator	Model	ELO	95% CI	Appearances
ByteDance Seed	Seedream 4.0	1,205	-20/+23	1,607
Google	Gemini 2.5 Flash	1,201	-13/+13	5,783
Black Forest Labs	FLUX.1 Kontext [pro]	1,089	-12/+12	5,993
OpenAI	GPT-4o	1,088	-12/+13	5,781
Alibaba	Qwen-Image-Edit	1,087	-12/+12	6,103
Black Forest Labs	FLUX.1 Kontext [max]	1,083	-12/+12	5,947
ByteDance seed	SeedEdit 3.0	1,076	-21/+21	1,372
HiDream	HiDream-E1.1	1,005	-13/+13	5,112
Google	Gemini 2.0 Flash Preview	1,000	+0/+0	5,521
Black Forest Labs	FLUX.1 Kontext [dev]	995	-13/+13	5,679

Figure 2.11: **Artificial Analysis leaderboard (Sep 12, 2025)**. Screenshot of the public leaderboard used for comparison in Table 2.5.

2.5.2 Failure Case

We discuss a representative failure mode of our evaluation. The most severe errors arise from false positives in *Grounding-DINO*, despite its strong open-vocabulary performance. Consider the prompt: “Replace [stone gray railing] with [wooden fence].” As shown in Fig. 2.12, Seedream 4.0 produces an edit that is visually correct. Our rule for `subject_replace` declares success if, on the edited image, the *source* object (stone gray railing) is no longer detected while the *target* object (wooden fence) is detected. However, Grounding-DINO occasionally reports both the source and target objects in the same region with high confidence, incorrectly suggesting that the source object remains and thereby degrading the measured instruction-following accuracy. Improving the reliability of open-vocabulary detection—particularly reducing false positives—would directly improve the fidelity of our evaluation.

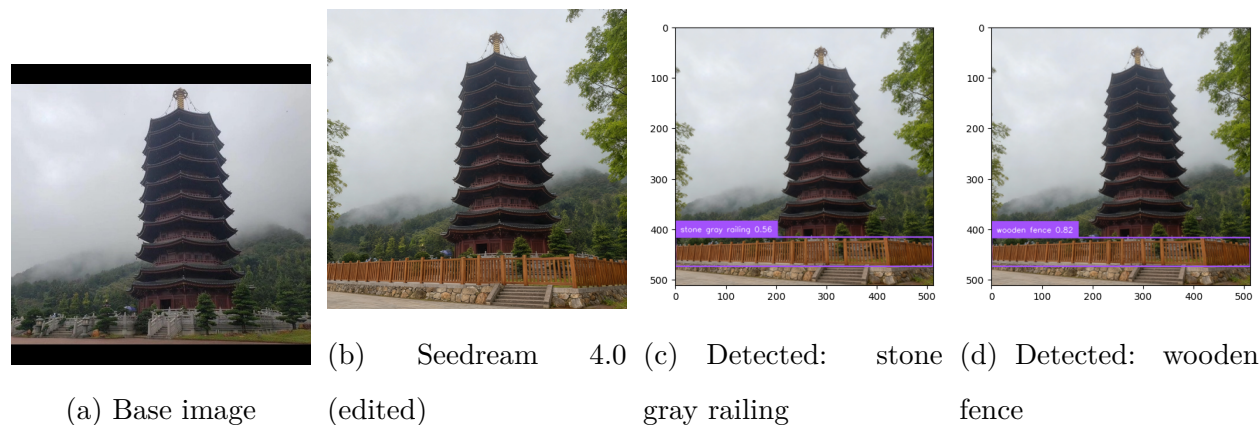


Figure 2.12: **Failure due to detector false positives.** Although the edit visually replaces the railing with a wooden fence, Grounding-DINO fires on both “stone gray railing” and “wooden fence” in overlapping regions, causing an incorrect failure in our instruction-following metric.

2.5.3 Discussion on single-shot complex editing

Figure 2.13 shows that marginal success for the final instruction remains largely stable as complex prompt length increases. Together with the multi-turn drops seen in Figure 2.5, this pattern supports an *exposure-bias* explanation: performance degradation primarily stems from error accumulation across sequential edits rather than an intrinsic inability to handle multiple instructions in a single prompt.

2.5.4 Discussion on Visual Quality

Beyond instruction following and content consistency, the perceptual *quality* of the edited image is a key dimension. We therefore report (i) a learned aesthetic score and (ii) several low-level image statistics that can surface systematic artifacts and drift in multi-turn editing pipelines.

Table 2.6: Turn-3 instruction following: Multi-turn vs. single-shot complex prompts, grouped by technique. **Bold** indicates which setting is higher for each model.

Technique	Model	Multi-turn (T3)	Complex (C3)
In-Context	Nano Banana	35.35	28.14
	GPT-Image-1	38.35	28.78
	Gemini 2.0 Flash	28.42	21.89
Flow Matching	Qwen-Image-Edit	22.55	27.62
	Step1X-Edit	17.83	15.73
	FLUX.1-Kontext-dev	16.61	19.58
	OmniGen	10.66	11.01
	AnyEdit	7.22	2.80
Diffusion	UltraEdit	6.36	8.22
	MagicBrush	4.90	4.55
	IP2P	2.80	2.80

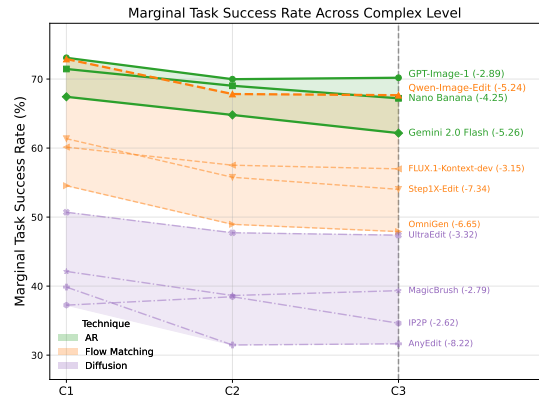


Figure 2.13: Marginal task-success rate of the *last* instruction as a function of complex prompt length (levels $C = 1, 2, 3$).

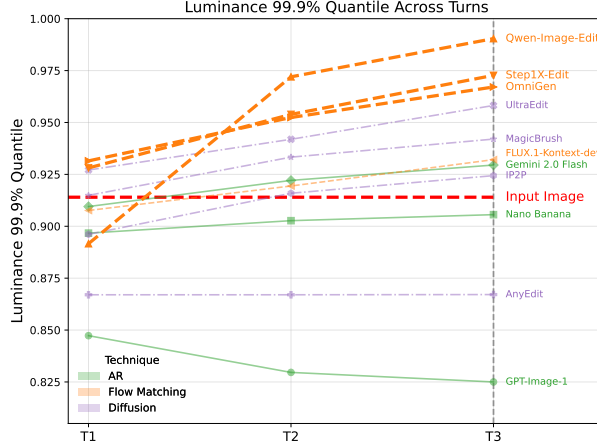


Figure 2.14: Per-image 99.9% luminance quantile across turns. Higher values indicate more extreme bright pixels and greater risk of over-exposure.

Low-level image statistics In addition to learned aesthetic scores, we compute several low-level image statistics that help reveal systematic, multi-turn editing artifacts. Concretely, we convert RGB pixels to luminance using the Rec. 709 luma coefficients: $Y = 0.2126 R + 0.7152 G + 0.0722 B$, and for each edited image we extract the **99.9% luminance quantile** (the per-image pixel value below which 99.9% of pixels fall). The 99.9% quantile is sensitive to high-exposure pixels and therefore highlights over-exposure and bright streaks while being robust to single-pixel outliers. In Figure 2.14 we plot the trend of this statistic across turns.

The measured trend shows a clear pattern: **Qwen-Image-Edit** and several other flow-matching models (with the notable exception of **FLUX.1-Kontext-dev**) exhibit a pronounced increase in the 99.9% luminance quantile over turns, indicating progressive brightening and increased risk of over-exposure. By contrast, regeneration-style editors such as **GPT-Image-1** tend to produce lower luminance values than the input (reflecting darker, more conservative reconstructions), and several models remain stable across turns.

Figure 2.15 provides qualitative examples from Qwen-Image-Edit. The edited images exhibit elevated luminance and noticeable high-frequency bright artifacts (e.g., white streaks or “line” textures) that degrade perceptual quality, with luminance quintiles increasing sub-

stantially. Correspondingly, HPS drops from 6.19 to 4.19 and 3.34, suggesting that HPS is sensitive to over-exposure to some extent. In contrast, when querying VLMs about the visual quality of these images, the returned scores do not change in the first two turns and remain consistently above 50, reflecting a *positive* evaluation under the $[0, 100]$ scale, while the T2/T3 edited images show significant artifacts.



Figure 2.15: Representative Qwen-Image-Edit examples illustrating over-exposure and bright artifact formation across turns. Although editing instructions are often satisfied, the images show elevated luminance and high-frequency bright streaks that accompany the edits (visible especially in T2/T3). Editing instructions: [Remove polyester white skirt, Change the count of tennis ball to 4, Change the color of tank top to blue]. Note that VLM gives a *positive* score to all the images.

2.5.5 VLMs failing to judge visual quality

The following is the zero-shot prompt for visual quality with VLMs. The example results are shown in Fig. 2.15.

You are an expert at evaluating image visual quality and naturalness.

I will show you an image.

Please analyze whether the image is visually pleasing and natural. Consider:

1. Is the image visually pleasing?
2. Is the image natural?
3. Does the image look natural and coherent?

Respond only with a score between 0 and 100, where 100 is the highest score.

100 means the image is visually pleasing and natural.

0 means the image is not visually pleasing and natural.

50 means the image is neutral.

2.5.5.1 Evaluation

We evaluate in two modes: (i) **Multi-turn** (each turn edits the output of the previous turn), and (ii) **Complex Editing** (compress all instructions to a single prompt).

Instruction Following. We compute a binary success per instruction with a detector combining GroundingDINO [LZR24b] and a VLM (Qwen2-VL-7B) [BCL25]. Representative details:

- Detector thresholds. Unless noted per task, GroundingDINO thresholds are 0.3–0.4; detections return normalized boxes $[x_1, y_1, x_2, y_2]$.
- Cropping and small objects. For object-level checks we crop by detected boxes; very small boxes (< 0.05 in width/height) can be enlarged before VLM queries.
- Replace. Detect old and new objects in source/target; success if both are detected and any IoU between a source box (old) and a target box (new) is > 0 . A VLM pre-check rejects obvious non-replacements. See details in Alg 1.
- Remove. Detect the object in the source; success if the object is absent in the target. See details in Alg 2.
- Position change. Detect target and reference objects and verify the requested spatial relation using object centers; also ensure the object count did not increase spuriously. See details in Alg 4.
- Count change. Use the detector to locate instances of the target object and take the number of validated detections as the count. See details in Alg 5.
- Color/material. Crop the object in the target and ask the VLM a yes/no question about the new color/material. See details in Alg 6 and Alg 7.

- Text change. If the instruction adds text anywhere, run the VLM on the whole image; if it replaces text on a specific object, first crop that object’s box, ask the VLM to extract the text, and compare it to the requested text. See details in Alg 8.
- Background change. Ask the VLM yes/no whether the requested background category is present. See details in Alg 9.

Consistency. We measure object and background stability as follows:

- Object consistency (unchanged objects): DINOv3 ViT-B/16 [SVS25] feature similarity between crops of unchanged objects in base vs. target; we also report pixel L1 consistency and average across objects per image.
- Background consistency: detect objects in `all_objects_pool` in base/target (GroundingDINO), mask them to isolate background, then compute masked L1 between backgrounds (optionally DINOv3 masked similarity). Background consistency is evaluated only when no background change occurred earlier (`bg_consistency=true`).

Perceptual Quality. We report HPSv3 [MWS25] plausibility and aesthetics, plus luminance metrics. Quality is *not* folded into the *Overall* score.

2.5.5.2 Overall Score and Aggregation Details

Let α_t be the image success rate at turn t : the fraction of images for which *all* edits up to and including turn t are successful (aggregated per task type, then averaged). Let κ denote the average content-consistency score combining object and background DINOv3 similarities when applicable.

- Overall score. We report

$$\text{Overall} = [\text{mean}_t(\alpha_t) \times \text{mean}(\kappa)]^{1/2}.$$

Algorithm 1 Evaluate Subject Replace

Require: base B , target T , old object name o , new object name n

Ensure: success flag succ

```
1:  $S \leftarrow \text{Detect}(B, o, \tau)$ ;  $T_n \leftarrow \text{Detect}(T, n, \tau)$ 
2: if  $S \neq \emptyset \wedge T_n \neq \emptyset$  then
3:    $\text{succ} \leftarrow \max_{b \in S, t \in T_n} \text{IoU}(b, t) > 0$ 
4: else
5:    $\text{succ} \leftarrow \text{false}$ 
6: end if
7: return  $\text{succ}$ 
```

Algorithm 2 Evaluate Subject Remove

Require: base B , target T , object name o

Ensure: success flag succ

```
1:  $S \leftarrow \text{Detect}(B, o, \tau)$ ;  $T_o \leftarrow \text{Detect}(T, o, \tau)$ 
2:  $\text{succ} \leftarrow (S \neq \emptyset \wedge T_o = \emptyset)$ 
3: return  $\text{succ}$ 
```

- Missing outputs across turns. For summary tables, we include only images that produce all required outputs for the evaluated mode. If a model fails to generate a later turn, that image is omitted from later-turn aggregates for that mode. Some edits will be rejected by some models since the sensitive content flag.
- No unchanged objects. If the unchanged-object list is empty, object consistency is recorded as **None** and excluded from averages; background consistency is still computed when `bg_consistency=true`.
- Turn-level reporting. We also report per-turn (T1, T2, T3) instruction-following and consistency, and per-task-type success rates $\alpha_{t,\text{type}}$. Quality metrics are reported separately and are not folded into *Overall*.

Algorithm 3 Evaluate Subject Add

Require: base B , target T , new object name n , optional reference object name r , optional position $p \in \{\text{left, right, above, below}\}$

Ensure: success flag

```
1:  $B_n \leftarrow \text{Detect}(B, n, \tau)$ ;  $T_n \leftarrow \text{Detect}(T, n, \tau)$ 
2: if  $T_n = \emptyset \vee B_n \neq \emptyset$  then
3:   return false
4: end if
5: if  $r$  and  $p$  are provided then
6:    $B_r \leftarrow \text{Detect}(B, r, \tau)$ ;  $T_r \leftarrow \text{Detect}(T, r, \tau)$ 
7:   if  $T_r = \emptyset$  then
8:     return false
9:   end if
10:  Choose max logits boxes  $t \in T_n, u \in T_r$ 
11:   $(x_t, y_t) \leftarrow \text{Center}(t)$ ;  $(x_u, y_u) \leftarrow \text{Center}(u)$ 
12:  if  $p = \text{left} \wedge x_t < x_u - \varepsilon_x$  then
13:    return true
14:  end if
15:  if  $p = \text{right} \wedge x_t > x_u + \varepsilon_x$  then
16:    return true
17:  end if
18:  if  $p = \text{above} \wedge y_t < y_u - \varepsilon_y$  then
19:    return true
20:  end if
21:  if  $p = \text{below} \wedge y_t > y_u + \varepsilon_y$  then
22:    return true
23:  end if
24:  return false
25: else
26:  return true
27: end if
```

2.5.5.3 Model generations

We evaluate a mix of closed- and open-source editors using each model’s default settings (no hyperparameter tuning):

- GPT-Image-1, Nano Banana, and Gemini 2.0 Flash: called via their APIs with default parameters.
- QWEN Image Edit: default settings from <https://huggingface.co/Qwen/Qwen-Image-Edit>.
- InstructPix2Pix (IP2P): settings from <https://github.com/timothybrooks/instruct-pix2pix>.

- Magicbrush: same settings as IP2P; model weights from <https://huggingface.co/vinesmsuic/magicbrush-jul7>.
- UltraEdit: settings from <https://github.com/HaozheZhao/UltraEdit>; we apply a black mask since no explicit mask is provided.
- AnyEdit: repository at <https://github.com/weichow23/AnySD/tree/9e7d36ef88e237b527695efc90b1abc18fa51218> with `edit_type` set to `general`.
- Step1X-Edit: repository at <https://github.com/stepfun-ai/Step1X-Edit>; weights at <https://huggingface.co/stepfun-ai/Step1X-Edit>.
- OmniGen: repository at <https://github.com/VectorSpaceLab/OmniGen>.
- FLUX: default settings from <https://huggingface.co/black-forest-labs/FLUX.1-Kontext-dev>.

Modes. For clarity in the paper: we report both **Multipass** and **Complex Editing** (renamed from *singlepass* for consistency with the rest of the paper).

Algorithm 4 Evaluate Position Change

Require: base B , target T , target object name a , reference object r , position p

Ensure: success flag

```
1:  $B_a \leftarrow \text{Detect}(B, a, \tau); \quad T_a \leftarrow \text{Detect}(T, a, \tau)$ 
2:  $B_r \leftarrow \text{Detect}(B, r, \tau); \quad T_r \leftarrow \text{Detect}(T, r, \tau)$ 
3: if  $T_a = \emptyset \vee T_r = \emptyset$  then
4:   return false
5: end if
6: if  $|T_a| > |B_a|$  then
7:   return false ▷ No count inflation
8: end if
9: Select max logits boxes  $t \in T_a, u \in T_r$ 
10:  $(x_t, y_t) \leftarrow \text{Center}(t); \quad (x_u, y_u) \leftarrow \text{Center}(u)$ 
11: if  $p = \text{left}$  then
12:   return  $x_t < x_u - \varepsilon_x$ 
13: end if
14: if  $p = \text{right}$  then
15:   return  $x_t > x_u + \varepsilon_x$ 
16: end if
17: if  $p = \text{above}$  then
18:   return  $y_t < y_u - \varepsilon_y$ 
19: end if
20: if  $p = \text{below}$  then
21:   return  $y_t > y_u + \varepsilon_y$ 
22: end if
23: return false
```

Algorithm 5 Evaluate Count Change

Require: target T , name o , requested count c^*

Ensure: success flag

- 1: $\hat{c} \leftarrow |\text{Detect}(T, o)|$
 - 2: **return** ($\hat{c} = c^*$)
-

Algorithm 6 Evaluate Color Alter

Require: target image T , object name o , color k

- 1: **return** $\text{VLMYesNo}(T, \text{"Is the } o \text{ } k\text{?"})$
-

Algorithm 7 Evaluate Material Alter

Require: target image T , object name o , material m

Ensure: success flag

- 1: **return** $\text{VLMYesNo}(T, \text{"Is the } o \text{ made of } m\text{?"})$
-

Algorithm 8 Evaluate Text Change

Require: target T , desired text t^* (optionally object name)

Ensure: success flag

- 1: $t \leftarrow \text{VLMText}(T)$
 - 2: Normalize t and t^* (case, punctuation, whitespace)
 - 3: **return** $\text{text-match}(t, t^*)$
-

Algorithm 9 Evaluate Background Change

Require: target T , category g

Ensure: success flag

- 1: **return** $\text{VLMYesNo}(T, \text{"Does the background show } g\text{?"})$
-

2.5.6 Additional Evaluation Results

In this section, we provide extended evaluation results. We separate the analysis into two modes: *multi-turn editing* and *complex editing*. Each mode is evaluated across three aspects: instruction following, consistency, and quality.

For the multi-turn editing mode, the overall instruction-following success rate is reported in Table 2.7, while success rates for individual instruction types appear in Tables 2.8 and 2.9. Consistency results are summarized in Table 2.13. We also observed that some input images are non-square after resizing, which can leave black padding on the top/bottom or left/right edges. Certain editing models, such as GPT-Image-1 and Qwen-Image-Edit, attempt to fill these areas, whereas others preserve them. To account for this, we separately report consistency for square (Table 2.14) and non-square inputs (Table 2.15). The conclusions remain consistent with the overall evaluation. Quality results for multi-turn editing are presented in Table 2.17.

For the complex editing mode, the overall instruction-following success rate is shown in Table 2.10, and per-instruction-type results are in Tables 2.11 and 2.12. Consistency and quality results are reported in Tables 2.16 and 2.18, respectively.

In consistency table, p99 means 99% quantile of luminance value, and p999 means 99.9% quantile of luminance value.

Table 2.7: Image success rates and overall task success rates across turns. (Multi-turn model)

Model	Image Success Rate			Overall Task Rate		
	T1	T2	T3	T1	T2	T3
Seedream 4.0	75.93	55.58	41.59	75.93	75.58	76.11
Nano Banana	70.70	50.66	35.35	70.70	72.59	68.24
GPT-Image-1	73.12	54.89	38.35	73.12	74.44	73.12
FLUX.1-Kontext-max	69.49	46.89	31.83	69.49	69.11	70.43
Gemini 2.0 Flash	68.07	45.96	28.42	68.07	67.72	68.42
Qwen-Image-Edit	72.90	44.06	22.55	72.90	62.94	56.12
Step1X-Edit	61.89	34.97	17.83	61.89	59.09	53.32
FLUX.1-Kontext-dev	59.97	32.69	16.61	59.97	56.29	51.40
OmniGen	54.72	24.48	10.66	54.72	48.60	42.48
UltraEdit	51.37	17.70	6.36	50.52	36.54	31.47
AnyEdit	41.07	16.32	7.22	40.03	39.34	40.56
MagicBrush	42.31	15.73	4.90	42.31	40.73	41.26
IP2P	37.41	10.66	2.80	37.41	32.87	34.27

Table 2.8: Task success rates (%) across five instruction types and three turns (multi-turn mode).

Model	Subject Replace			Subject Remove			Material Alter			Color Alter			Subject Add		
	T1	T2	T3	T1	T2	T3	T1	T2	T3	T1	T2	T3	T1	T2	T3
Seedream 4.0	90.74	91.23	88.89	68.92	47.69	50.00	95.31	96.00	95.77	100.00	98.59	100.00	83.08	89.61	81.52
Nano Banana	91.84	92.31	75.47	64.18	51.61	40.35	91.94	89.36	87.50	100.00	97.18	98.11	73.02	77.94	72.41
GPT-Image-1	84.31	94.64	85.71	70.77	55.93	47.37	96.83	95.65	87.88	100.00	97.06	100.00	80.95	72.46	72.41
FLUX.1-Kontext-max	92.31	88.89	88.00	67.16	55.56	56.36	87.30	79.07	80.30	100.00	95.65	98.04	77.05	71.23	72.73
Gemini 2.0 Flash	83.33	92.98	78.18	58.67	53.62	50.82	90.91	82.00	83.33	100.00	89.04	98.21	77.61	72.73	75.27
Qwen-Image-Edit	87.04	82.46	70.91	70.67	31.88	37.70	93.94	90.00	79.17	100.00	97.26	94.74	77.61	55.84	39.78
Step1X-Edit	90.74	96.49	67.27	53.33	30.43	21.31	95.45	80.00	87.50	100.00	100.00	91.23	64.18	57.14	45.16
FLUX.1-Kontext-dev	85.19	80.70	72.73	54.67	42.03	32.79	84.85	74.00	73.61	100.00	98.63	94.74	67.16	61.04	39.78
OmniGen	88.89	84.21	58.18	46.67	21.74	19.67	84.85	72.00	70.83	100.00	90.41	91.23	53.73	51.95	37.63
UltraEdit	88.89	63.16	38.18	26.67	5.80	6.56	87.88	66.00	63.89	98.21	80.82	78.95	38.81	23.38	9.68
AnyEdit	74.07	66.67	61.82	37.33	39.13	36.07	78.79	68.00	68.06	78.57	68.49	78.95	22.39	38.96	25.81
MagicBrush	83.33	75.44	63.64	28.00	18.84	18.03	83.33	86.00	80.56	94.64	87.67	91.23	37.31	41.56	37.63
IP2P	75.93	66.67	65.45	25.33	8.70	18.03	74.24	70.00	65.28	87.50	82.19	75.44	23.88	28.57	19.35

Table 2.9: Task success rates (%) across four instruction types and three turns (multi-turn mode).

Model	Text Change			Position Change			Count Change			Background Change		
	T1	T2	T3	T1	T2	T3	T1	T2	T3	T1	T2	T3
Seedream 4.0	95.31	97.14	96.23	39.22	39.68	48.94	18.06	19.30	20.00	98.46	96.36	93.06
Nano Banana	83.33	86.36	81.25	20.00	47.37	45.24	23.19	9.62	10.91	96.67	94.44	90.00
GPT-Image-1	88.33	97.01	97.96	31.11	40.68	50.00	11.27	18.52	18.18	98.39	94.44	91.30
FLUX.1-Kontext-max	80.36	86.15	82.69	18.75	38.18	44.44	8.82	9.09	12.28	96.88	92.59	92.54
Gemini 2.0 Flash	90.32	94.29	96.30	21.15	28.57	31.91	10.96	7.14	10.00	86.15	83.64	80.56
Qwen-Image-Edit	98.44	92.86	72.22	21.15	34.92	33.33	12.33	1.72	0.00	98.46	80.00	77.78
Step1X-Edit	60.94	51.43	44.44	13.46	31.75	27.08	0.00	1.72	1.67	87.69	87.27	83.33
FLUX.1-Kontext-dev	50.00	41.43	27.78	15.38	26.98	33.33	0.00	1.72	5.00	90.77	80.00	77.78
OmniGen	29.69	35.71	18.52	17.31	22.22	20.83	5.48	5.17	0.00	76.92	56.36	56.94
UltraEdit	28.12	15.71	7.41	21.15	36.51	35.42	5.48	6.90	5.00	75.38	38.18	43.06
AnyEdit	3.12	10.00	11.11	21.15	25.40	27.08	0.00	1.72	1.67	56.92	40.00	52.78
MagicBrush	7.81	12.86	3.70	19.23	15.87	20.83	0.00	0.00	3.33	43.08	34.55	43.06
IP2P	1.56	8.57	5.56	13.46	15.87	25.00	1.37	0.00	5.00	47.69	20.00	31.94

Table 2.10: Image rates, overall task rates, and marginal means across three turns (complex mode).

Model	Image Success Rate			Overall Task Rate			Marginal Task Rate		
	T1	T2	T3	T1	T2	T3	T1	T2	T3
GPT-Image-1	73.08	48.45	28.78	73.08	69.77	68.25	73.08	69.98	70.19
Nano Banana	71.46	46.56	28.14	71.46	68.83	67.27	71.46	69.03	67.21
Gemini 2.0 Flash	67.43	40.63	21.89	67.43	64.54	61.94	67.43	64.80	62.17
Qwen-Image-Edit	72.90	46.15	27.62	72.90	69.23	68.07	72.90	67.83	67.66
Step1X-Edit	61.36	32.34	15.73	61.36	57.69	55.01	61.36	55.77	54.02
FLUX.1-Kontext-dev	60.14	33.74	19.58	60.14	59.53	57.87	60.14	57.52	56.99
OmniGen	54.55	23.43	11.01	54.55	50.96	49.83	54.55	48.95	47.90
AnyEdit	39.86	10.31	2.80	39.86	34.79	34.27	39.86	31.47	31.64
UltraEdit	50.70	22.03	8.22	50.70	48.34	46.62	50.70	47.73	47.38
MagicBrush	42.13	14.86	4.55	42.13	38.46	38.81	42.13	38.64	39.34
IP2P	37.24	12.41	2.80	37.24	37.76	35.14	37.24	38.46	34.62

Table 2.11: Success rates (%) for five instruction types across three turns (complex mode).

Model	Subject Replace			Subject Remove			Material Alter			Color Alter			Subject Add		
	T1	T2	T3	T1	T2	T3	T1	T2	T3	T1	T2	T3	T1	T2	T3
GPT-Image-1	82.22	80.65	78.99	70.31	63.87	58.54	96.49	90.20	84.66	100.00	97.27	91.88	81.67	70.73	69.00
Nano Banana	91.67	88.89	79.33	67.16	55.83	59.88	92.59	85.15	83.33	97.87	98.15	94.19	69.84	71.21	67.77
Gemini 2.0 Flash	85.19	82.88	75.90	57.33	55.56	54.15	93.94	75.00	74.87	100.00	96.90	93.01	70.15	65.97	66.67
Qwen-Image-Edit	87.04	86.49	80.72	70.67	58.33	54.63	93.94	86.21	85.64	100.00	99.22	98.39	77.61	75.69	73.84
Step1X-Edit	90.74	84.68	76.51	52.00	41.67	42.93	93.94	82.76	79.26	100.00	93.02	90.86	64.18	59.03	54.01
FLUX.1-Kontext-dev	85.19	82.88	74.10	54.67	47.92	40.00	86.36	75.00	76.06	100.00	99.22	98.39	67.16	68.75	63.29
OmniGen	88.89	82.88	75.90	46.67	39.58	41.95	86.36	73.28	72.34	100.00	96.12	93.01	53.73	48.61	49.37
AnyEdit	66.67	58.56	49.40	33.33	20.14	22.44	77.27	76.72	70.21	78.57	68.99	72.04	28.36	22.92	21.10
UltraEdit	88.89	87.39	78.31	26.67	30.56	31.71	87.88	79.31	77.66	98.21	93.02	90.32	38.81	46.53	43.46
MagicBrush	83.33	74.77	69.28	28.00	27.78	29.27	83.33	69.83	73.94	92.86	84.50	83.87	37.31	35.42	32.91
IP2P	75.93	73.87	61.45	25.33	31.25	24.88	71.21	63.79	59.57	87.50	82.95	79.03	23.88	31.94	29.11

Table 2.12: Success rates (%) for four instruction types across three turns (complex mode).

Model	Text Change			Position Change			Count Change			Background Change		
	T1	T2	T3	T1	T2	T3	T1	T2	T3	T1	T2	T3
GPT-Image-1	87.50	92.24	93.75	30.95	22.11	25.36	13.11	10.38	13.12	98.15	96.08	93.37
Nano Banana	84.21	85.34	84.24	24.44	26.00	22.54	20.34	14.15	13.75	98.15	93.40	93.71
Gemini 2.0 Flash	85.94	86.57	84.04	17.31	23.68	16.67	11.11	10.77	6.84	90.77	84.17	80.73
Qwen-Image-Edit	98.44	97.01	93.62	21.15	22.61	24.54	12.33	3.05	3.66	98.46	95.83	93.75
Step1X-Edit	57.81	59.70	52.13	15.38	21.74	25.15	1.37	3.05	1.57	86.15	80.00	73.44
FLUX.1-Kontext-dev	50.00	48.51	47.87	15.38	26.96	27.61	0.00	1.53	3.66	90.77	90.00	88.54
OmniGen	28.12	32.84	27.13	15.38	21.74	17.18	6.85	1.53	2.09	75.38	70.00	69.79
AnyEdit	3.12	5.22	7.98	25.00	26.96	29.45	1.37	1.53	1.57	56.92	44.17	40.62
UltraEdit	29.69	16.42	11.70	21.15	26.09	22.70	5.48	2.29	3.14	75.38	65.00	64.06
MagicBrush	7.81	4.48	6.38	19.23	22.61	17.18	0.00	0.00	5.24	43.08	36.67	35.42
IP2P	3.12	5.22	6.38	13.46	20.00	20.25	1.37	2.29	1.57	47.69	37.50	38.54

Table 2.13: Consistency scores (%) across DINOv3-based and L1-based object/background metrics. (multi-turn mode)

Model	Object DINOv3 Consistency			Background DINOv3 Consistency			Object $1 - L_1$ Consistency			Background $1 - L_1$ Consistency		
	T1	T2	T3	T1	T2	T3	T1	T2	T3	T1	T2	T3
Seedream 4.0	89.50	83.68	81.18	95.52	92.38	90.54	93.31	88.76	86.22	94.30	89.04	85.00
Nano Banana	90.17	85.25	84.38	97.65	95.70	94.58	92.39	90.00	89.10	95.95	94.70	93.88
GPT-Image-1	73.26	68.80	67.21	88.74	86.76	83.78	79.65	78.32	77.60	78.07	76.53	75.79
FLUX.1-Kontext-max	90.91	86.66	83.68	96.95	95.15	93.11	94.16	91.26	89.46	95.88	93.43	91.46
Gemini 2.0 Flash	85.53	77.12	72.02	95.63	93.07	89.74	90.72	86.32	84.11	95.04	93.15	91.73
Qwen-Image-Edit	77.12	71.56	68.51	91.31	89.47	87.45	83.48	79.15	76.32	84.57	81.18	78.43
Step1X-Edit	88.17	81.65	77.33	97.34	95.40	93.09	93.92	90.64	88.80	98.24	97.10	95.73
FLUX.1-Kontext-dev	92.66	87.92	85.29	97.97	96.55	95.14	94.39	91.59	89.91	96.36	95.06	94.13
OmniGen	88.34	80.77	73.64	97.66	96.08	94.21	93.87	91.02	89.43	97.44	97.00	96.34
UltraEdit	78.81	75.11	72.24	94.80	93.89	92.57	91.86	90.47	89.65	97.12	96.62	96.19
AnyEdit	82.02	73.41	63.04	90.82	84.42	77.17	92.52	88.96	84.97	93.72	89.98	86.05
MagicBrush	79.70	70.71	65.46	94.22	91.81	88.27	91.13	87.13	85.56	96.31	94.52	92.87
IP2P	68.24	56.83	48.01	85.47	79.89	72.59	84.44	79.74	77.21	91.31	87.21	83.51

Table 2.14: Consistency and $1 - L_1$ metrics across three turns (multi-turn mode for square image).

Model	Object DINOv3 Consistency (Mean)			Background DINOv3 Consistency			Object L_1 Consistency (Mean)			Background L_1 Consistency		
	T1	T2	T3	T1	T2	T3	T1	T2	T3	T1	T2	T3
Seedream 4.0	89.04	85.39	80.74	95.76	92.43	88.27	91.66	88.02	84.43	95.34	92.45	89.69
Nano Banana	88.68	85.19	83.31	96.88	93.79	92.41	89.73	87.64	88.48	94.21	93.77	93.36
GPT-Image-1	73.62	71.27	67.27	90.00	85.72	82.16	80.33	78.80	79.88	85.69	81.86	80.85
FLUX.1-Kontext-max	91.34	89.90	86.04	95.99	94.18	90.10	93.81	92.80	90.58	97.27	95.77	93.94
Gemini 2.0 Flash	84.16	79.91	71.13	91.50	91.83	87.07	88.98	87.07	85.13	95.19	93.35	93.34
Qwen-Image-Edit	75.36	72.68	68.85	88.50	88.98	82.71	81.66	78.47	77.04	91.16	87.85	85.78
Step1X-Edit	87.40	84.03	78.27	97.43	92.42	88.15	93.14	91.57	89.40	98.10	97.12	96.01
FLUX.1-Kontext-dev	92.55	88.92	84.04	96.49	93.57	92.19	92.83	90.61	88.34	96.91	95.74	94.73
OmniGen	89.41	84.51	77.77	97.34	93.13	86.64	93.26	91.48	89.53	95.79	96.44	95.36
UltraEdit	79.43	76.51	71.54	92.83	89.88	86.54	91.74	90.16	89.24	95.87	95.14	94.63
AnyEdit	82.25	72.40	53.59	86.12	78.60	70.09	92.33	88.00	82.10	94.36	91.68	87.55
MagicBrush	79.02	75.60	70.07	92.07	87.33	81.08	89.67	87.53	86.90	95.29	93.88	92.38
IP2P	76.38	66.12	54.94	85.29	81.46	65.50	86.90	82.25	77.79	92.45	88.01	84.51

Table 2.15: Consistency and L_1 metrics across three turns (multi-turn model for unsquared image).

Model	Object DINOv3 Consistency (Mean)			Background DINOv3 Consistency			Object $1 - L_1$ Consistency (Mean)			Background $1 - L_1$ Consistency		
	T1	T2	T3	T1	T2	T3	T1	T2	T3	T1	T2	T3
Seedream 4.0	89.55	83.48	81.23	95.49	92.38	90.86	93.50	88.85	86.44	94.17	88.62	84.35
Nano Banana	90.34	85.26	84.50	97.74	95.94	94.88	92.69	90.29	89.17	96.16	94.81	93.95
GPT-Image-1	73.22	68.53	67.20	88.60	86.88	83.99	79.57	78.27	77.35	77.20	75.91	75.14
FLUX.1-Kontext-max	90.86	86.31	83.43	97.06	95.27	93.51	94.20	91.09	89.34	95.73	93.16	91.14
Gemini 2.0 Flash	85.69	76.79	72.12	96.14	93.23	90.10	90.91	86.23	83.99	95.02	93.12	91.51
Qwen-Image-Edit	77.32	71.43	68.47	91.65	89.53	88.10	83.68	79.23	76.23	83.78	80.37	77.43
Step1X-Edit	88.26	81.37	77.22	97.33	95.77	93.77	94.01	90.53	88.73	98.26	97.09	95.70
FLUX.1-Kontext-dev	92.67	87.80	85.44	98.15	96.92	95.54	94.57	91.70	90.10	96.30	94.98	94.05
OmniGen	88.22	80.33	73.14	97.70	96.45	95.25	93.94	90.97	89.42	97.64	97.07	96.47
UltraEdit	78.74	74.95	72.32	95.04	94.39	93.40	91.87	90.51	89.70	97.26	96.80	96.40
AnyEdit	82.00	73.53	64.18	91.39	85.14	78.14	92.54	89.07	85.31	93.64	89.78	85.84
MagicBrush	79.78	70.14	64.90	94.48	92.37	89.27	91.30	87.08	85.40	96.44	94.59	92.94
IP2P	67.32	55.75	47.18	85.50	79.69	73.57	84.17	79.45	77.14	91.18	87.12	83.38

Table 2.16: Consistency scores (%) across object/background DINOv3 and L_1 metrics (complex mode).

Model	Object DINOv3			Object $1 - L_1$			Background DINOv3			Background $1 - L_1$		
	T1	T2	T3	T1	T2	T3	T1	T2	T3	T1	T2	T3
GPT-Image-1	73.23	70.02	67.57	79.52	78.03	77.29	88.72	86.77	84.79	77.96	77.38	76.51
Nano Banana	89.23	87.20	86.46	92.15	91.08	90.40	97.39	96.69	95.38	96.39	95.75	95.33
Gemini 2.0 Flash	85.41	80.37	77.38	90.60	88.45	86.53	96.03	94.18	92.83	95.03	94.77	93.46
Qwen-Image-Edit	77.12	76.09	76.69	83.48	83.08	83.11	91.31	91.32	90.51	84.57	84.93	85.35
Step1X-Edit	88.14	85.31	84.38	93.93	92.34	92.11	97.34	96.37	95.44	98.24	98.02	98.04
FLUX.1-Kontext-dev	92.66	90.30	89.19	94.39	92.79	91.61	97.97	96.74	95.40	96.36	95.57	94.04
OmniGen	88.37	85.15	83.14	93.88	92.46	91.06	97.62	97.10	96.07	97.45	97.58	97.40
AnyEdit	81.90	82.94	84.92	92.34	92.43	93.72	90.97	92.63	93.78	94.15	95.11	95.87
UltraEdit	78.81	72.75	71.67	91.86	89.51	88.99	94.80	93.01	92.03	97.12	96.35	96.02
MagicBrush	79.70	75.64	75.53	91.13	89.23	88.69	94.22	94.34	93.13	96.31	96.14	95.83
IP2P	68.24	67.31	69.49	84.44	82.93	83.72	85.47	85.88	86.45	91.31	89.94	90.19

Table 2.17: Human preference scores, p999, and p99 across three turns (multi-turn mode).

Model	Human Preference Score			p999			p99		
	T1	T2	T3	T1	T2	T3	T1	T2	T3
GPT-Image-1	6.6519	6.5898	6.5609	84.73	82.96	82.50	74.38	71.91	70.54
Nano Banana	4.9431	5.1179	5.2638	89.67	90.27	90.56	81.31	82.01	82.09
Gemini 2.0 Flash	4.4386	4.2332	4.0677	90.95	92.21	92.95	83.08	84.79	86.24
Qwen-Image-Edit	5.8591	5.7198	5.1502	89.16	97.20	99.04	79.60	90.51	95.28
Step1X-Edit	4.0577	3.3443	2.7569	92.81	95.39	97.27	85.10	88.46	91.21
FLUX.1-Kontext-dev	5.1192	5.0701	5.0354	90.76	91.94	93.21	82.05	83.03	84.58
OmniGen	4.6099	4.0743	3.4958	93.15	95.24	96.71	85.55	88.24	90.50
AnyEdit	3.6609	2.8017	1.9457	86.70	86.70	86.71	77.54	76.53	75.82
UltraEdit	4.7934	4.6806	4.3598	92.71	94.19	95.82	85.42	86.76	88.34
MagicBrush	3.8465	3.0805	2.3606	91.49	93.33	94.20	83.42	84.70	85.32
IP2P	3.2020	2.3779	1.4418	89.61	91.59	92.44	81.79	83.78	84.77

Table 2.18: Updated human preference scores, p999 scores, and p99 scores across three turns (complex mode).

Model	Human Preference Score			p999			p99		
	T1	T2	T3	T1	T2	T3	T1	T2	T3
GPT-Image-1	6.6328	6.8428	6.9655	85.33	84.14	84.44	74.73	73.92	73.07
Nano Banana	4.9444	5.1700	5.3632	89.65	90.67	91.79	81.02	81.93	82.75
Gemini 2.0 Flash	4.4511	4.5428	4.5732	91.27	92.66	93.48	83.85	85.69	86.75
Qwen-Image-Edit	5.8591	5.8769	5.9155	89.16	90.92	92.23	79.60	81.36	82.62
Step1X-Edit	4.0534	3.9063	3.8648	92.82	93.55	94.05	85.11	85.49	86.01
FLUX.1-Kontext-dev	5.1192	5.2446	5.4645	90.76	91.01	91.29	82.05	81.53	81.55
OmniGen	4.5976	4.3070	3.8122	93.15	93.74	95.65	85.56	86.28	88.57
AnyEdit	3.7020	3.7601	3.8382	86.47	87.16	87.43	77.82	78.60	79.12
UltraEdit	4.7934	4.7647	4.8117	92.71	93.06	93.24	85.42	86.09	86.47
MagicBrush	3.8465	3.6029	3.5523	91.49	91.52	91.64	83.42	83.31	83.07
IP2P	3.2020	3.3552	3.5640	89.61	90.46	90.86	81.79	82.57	82.71

2.5.7 Related Work

Instruction-based editing models. InstructPix2Pix (IP2P) [BHE23] introduced a two-stage recipe that converts a text-to-image diffusion model [RBL22a, ZYW25] into an editor: (i) synthesize paired editing data using Stable Diffusion [RBL22a] and training-free techniques such as Prompt-to-Prompt [HMT23a]; (ii) fine-tune the diffusion model on these pairs. Subsequent systems—MagicBrush [ZMC23], UltraEdit [ZMC24], and AnyEdit [YCY25]—scale this paradigm to large, fine-grained real-image editing. More recent work (e.g., OmniGen [XWZ25, WZY25], Step1X-Edit [LHX25], FLUX.1 Kontext [LBB25], and Qwen-Image-Edit [WLZ25], Seedream [GGG25]) adopts task-aware architectures and increasingly leverages flow matching [LGL22a, LCB22, ZYZ24].

A complementary line explores autoregressive (AR) editors such as Gemini 2.0 Flash Image [Gem25], Gemini 2.5 Flash Image (“Nano Banana”) [Dee25], and GPT-Image-1 [Ope25]. These models enable **in-context multi-turn editing**: users iteratively refine an image within a conversational interface, with the model maintaining a coherent editing history. To our knowledge, we provide the first systematic comparison of in-context multi-turn AR editing versus context-free multi-turn editing with non-AR models across instruction following, content consistency, and visual quality.

Editing evaluation. Early evaluations (e.g., [BHE23]) rely on CLIP-based similarity [RKH21a], including directional variants [GPM22], to approximate editing quality. However, CLIP emphasizes semantic alignment and is less sensitive to fine, pixel-level changes. When ground-truth edited images exist, it is natural to compare model outputs against references using pixel distances (L_1) and semantic similarities (DINO [CTM21], CLIP [RKH21a]) [ZMC23, ZMC24, YCY25, SPS24]. Yet references are imperfect: the space of valid edits is inherently multimodal, while a single reference captures only one realization; moreover, many references are themselves synthesized by prior editors (e.g., Prompt-to-Prompt [HMT23a], SDXL [PEL24], DALLÉ-2 [RDN22a]), importing their biases into evaluation.

Recent work relies exclusively on VLMs as interpretable judges—e.g., VIEScore [KJW23], HQ-Edit [HYZ24], and Complex-Edit [YHZ25]—by querying models such as GPT-4o [Ope25] about specific aspects of an edit. While VLMs offer holistic, language-mediated assessments, they are insufficient on their own: they are notoriously poor at spatial reasoning [ZHL25, CYF24, CXK24, QGF24, CZF25] and are prone to object hallucinations in existence, category, attributes, and relations [BWX24]; they have limited sensitivity to pixel-level changes and frequently miss subtle, localized modifications [VNT25] (e.g., fine structures, small attribute shifts, etc.), which are crucial for evaluating content consistency; they are miscalibrated for artifacts and aesthetics [LHL24, XLW23, MWS25], which humans are sensitive to. Our approach, **EdiVal-Agent**, addresses these gaps by integrating VLM-based reasoning with grounding tools, symbolic, object-centric pixel- and semantic-level tools, and human preference models, yielding a precise and interpretable evaluation of instruction-based editing.

Editing tasks. We consider three settings: (i) **Single-turn vs. multi-turn.** Multi-turn editing [ZMC23, ZMC24] is more demanding than single-turn, as the model must maintain consistency across sequential instructions. In contrast to *context-free* multi-turn pipelines (each turn consumes the previous image and the next instruction), AR models [Gem25, Dee25, Ope25] support *in-context* multi-turn editing by conditioning on the full conversational history. (ii) **Complex single-shot vs. multi-turn.** Following [YHZ25], a sequence of edits can be concatenated into a single complex prompt and executed in one pass; we compare this setting to genuine multi-turn editing. (iii) **Other tasks.** We focus on instruction-based editing, the most common regime; other scenarios (e.g., prompt-to-prompt/caption-to-caption [HMT23a]) are outside our scope. To the best of our knowledge, this paper offers the first comprehensive comparison covering single-turn, multi-turn, and complex single-shot editing within a unified framework.

2.5.8 Limitation and Discussion

Given the object-centric evaluations conducted in this study, several limitations warrant consideration. First, our instruction types are limited to object-centric prompts, which may not capture the full range of creative editing requests typical in real-world scenarios. Future research should explore a broader spectrum of instructions, including those involving stylistic changes or complex narrative elements. Additionally, while our work provides a reliable and comprehensive evaluation framework for multi-turn editing, it does not apply the evaluation results to improve the editing models themselves. A straightforward extension would be to use evaluation scores for Best-of-N selection to improve inference-time performance. Future work could also explore post-training methods such as reinforcement learning, treating the evaluation scores as reward signals.

Please note that we have included performance evaluations for GPT-Image-1.5, Nano Banana 2, and FLUX.2-max in this camera-ready version. These models were released following the initial submission to ICLR 2026. While the main text has been updated to provide a self-contained analysis of these models, please be advised that the appendix may not reflect exhaustive comparative analysis for these three newly additions due to space and timing constraints.

2.5.9 Pareto Plot

In addition to presenting a leaderboard in Table 2.5, we also visualize the trade-offs among the three EdiVal dimensions—EdiVal-IF, EdiVal-CC, and EdiVal-VQ—using Pareto plots. These plots illustrate the performance frontier for each turn, providing a more concrete view of the trade-off and allowing users to select the most suitable model according to their preference.

We present the Pareto plots for Turn 1 in Figure 2.16, Turn 2 in Figure 2.17, and Turn 3 in Figure 2.10.

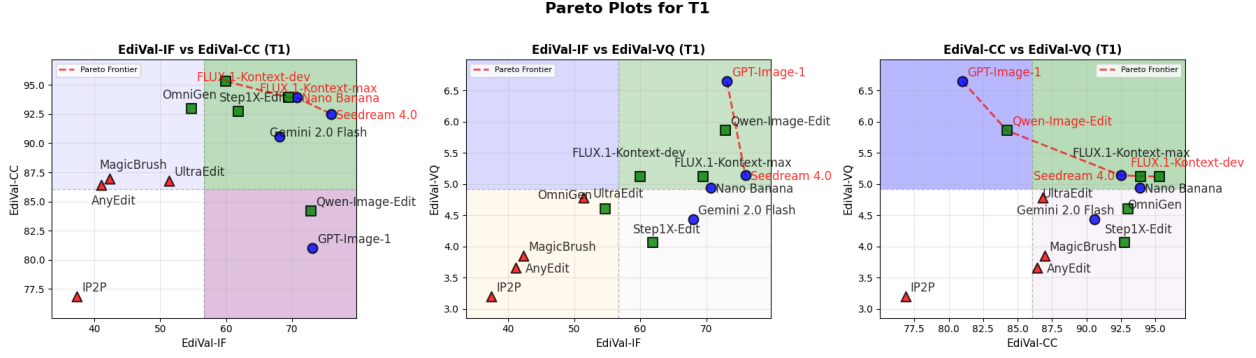


Figure 2.16: Pareto plot for Turn 1.

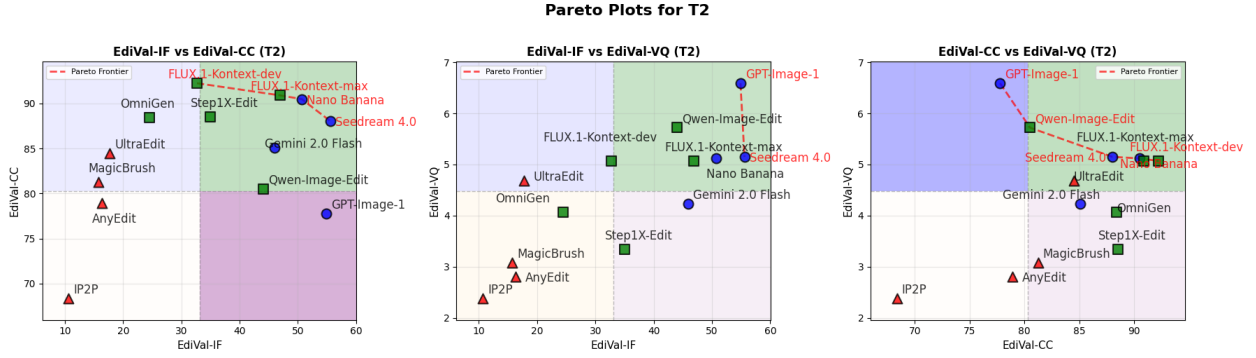


Figure 2.17: Pareto plot for Turn 2.

2.5.10 Ablation Study on Complex Editing Compression

We conduct an ablation study on how to compress three-turn instructions into a single “complex edit” prompt. In the previous experiment, we adopt the simplest concatenation strategy: {prompt T1}. {prompt T2}. {prompt T3}. We further evaluate three alternative variants on Qwen-Image-Edit: 1) *Random shuffle*: randomly shuffle the three per-turn prompts before concatenation. 2) *Sequential connector*: explicitly encode the order as: first, {prompt T1}, then, {prompt T2}, last, {prompt T3}. 3) *Keep-unchanged objects*: append an explicit constraint: {prompt T1}. {prompt T2}. {prompt T3}. Keep {unchanged objects} unchanged. Table 2.19 reports the resulting complex editing success rate(%) at $C = 3$. The results show these compression variants have only a very mild effect on performance.

Table 2.19: Complex Editing Performance for compression variants.

Connector Variant	Complex(C3) (%)
Default	27.62
Random shuffle	27.10
Sequential connector	26.92
Keep-unchanged	25.87

Table 2.20: Tool swap analysis (correlations).

Type	Default	Variant	Metric	Pearson	Spearman
VLM	Qwen2-7B-VL	Qwen2.5-7B-VL	IF	0.9544	0.9298
		Qwen2.5-32B-VL	IF	0.9790	0.9544
		InternVL3-8B	IF	0.9660	0.9228
Threshold	[0.3, 0.4]	Threshold = 0.4	IF	0.9817	0.9860
		Threshold = 0.3	IF	0.9772	0.9457
Filter	Has filter	No filter	IF	0.9982	0.9930
Detector	Grounding DINO	OWL-ViT	IF	0.8157	0.7929
Feature	DINOv3	DINOv2	CC	0.9987	1.0000

2.5.11 Tool Swap Analysis

We analyze the effect of swapping individual components in our evaluation stack, including the VLM, detector, detector threshold, and image feature extractor. We find that modifying the VLM (Appendix 2.5.11.1), adjusting the detector threshold (Appendix 2.5.11.2), changing the detector (Appendix 2.5.11.3), or changing the image feature extractor (Appendix 2.5.11.4) has only a minor impact on the final evaluation outcomes. The summary statistics are shown in Tab. 2.20, which demonstrates that for each configuration replacement, our evaluation results remain highly correlated with those obtained under the default stack. As for **EdiVal-VQ**, we found that HPSv3 is the only human preference model trained on images generated after SD3.5. This suggests that other preference models, such as HPSv2, cannot provide reliable assessments of recent, more advanced generations, as they are only sensitive to earlier-stage generations that are significantly lower in quality.

However, we note that replacing Grounding DINO with OWL-ViT [MGS22] (Appendix 2.5.11.3) fails to filter out failure cases originally caused by Grounding DINO and also reduces agreement with human annotations. This highlights that detector accuracy should be prioritized when considering detector substitutions.

VLM swap. We replace the default VLM in our stack (Qwen2-7B-VL) with several alternatives: Qwen2.5-7B-VL, Qwen2.5-32B-VL, and InternVL3-8B. For each configuration,

we recompute the per-turn instruction-following success rates for four representative editors: GPT-Image-1, Seedream 4.0, Nano Banana, and FLUX.1-Kontext-max.

Across all VLM choices, the ranking of these editors remain the same. While the absolute numbers shift slightly (e.g., the Qwen2.5- and InternVL3-based stacks tend to assign somewhat lower scores than the default Qwen2-7B-VL stack). When we compare per-turn success rates between the default Qwen2-7B-VL stack and each swapped VLM, both Pearson and Spearman correlations remain high over all (model, turn) pairs. Concretely, we show the results in Tab 2.21.

2.5.11.1 VLM Swap

To assess how our evaluation results depend on the choice of VLM, we swap the default VLM in our stack (Qwen2.5-7B-VL) with several alternatives: Qwen2-7B-VL, Qwen2.5-32B-VL, and InternVL3-8B. For each configuration, we recompute the per-turn instruction-following success rates for four representative editors: GPT-Image-1, Seedream 4.0, Nano Banana, and FLUX.1-Kontext-max.

Across all VLM choices, the relative ranking of these editors remains unchanged. While the absolute scores shift slightly (e.g., the Qwen2.5- and InternVL-based stacks tend to assign somewhat lower scores than the Qwen2-7B-VL stack), the overall ordering is stable. When we compare per-turn/marginal success rates between the default Qwen2.5-7B-VL stack and each swapped VLM, both Pearson and Spearman correlations remain high over all (model, turn) pairs.

Concretely, Table 2.21 reports correlations between the per-turn success rates under our default VLM (Qwen2.5-7B-VL) and those obtained with each alternative VLM. All Pearson correlations exceed 0.95 and all Spearman correlations exceed 0.92, indicating that changing the VLM has only a modest effect on the absolute scores and largely preserves the relative ordering of the four models. In particular, on the EdiVal-IF leaderboard we consistently

observe

Seedream 4.0 > GPT-Image-1 > Nano Banana > FLUX.1-Kontext-max.

Table 2.21: Correlations between per-turn instruction-following success rates under the default VLM (Qwen2.5-7B-VL) and three strong vision-language baselines. All correlations are computed over 12 (model, turn) points.

Baseline Model	Pearson	Spearman
Qwen2-7B-VL	0.9544	0.9298
Qwen2.5-32B-VL	0.9790	0.9544
InternVL3-8B	0.9660	0.9228

2.5.11.2 Grounding DINO Threshold and Large-Box Filter Swap

In this section, we ablate the effect of changing Grounding DINO’s detection threshold and disabling the large-box filter. By default, we use a threshold of 0.30 for `subject_remove`, 0.40 for `position_change`, and 0.35 for all other tasks. In addition, during detection we discard any box whose normalized height *and* width are both larger than 0.98, in order to filter out degenerate predictions that cover almost the entire image.

To investigate sensitivity to these choices, we re-run the evaluation under three alternative settings: (1) a global threshold of 0.30, (2) a global threshold of 0.40, and (3) the default thresholds but with the large-box filter disabled. The results are reported in Table 2.22. Across all settings, the instruction-following metrics (image success, task success, and overall scores) change only slightly, and the ranking of models remains unchanged. This suggests that our conclusions are robust to reasonable variations in the detector threshold and the large-box filtering heuristic.

Table 2.22: Per-turn image success, task success, and overall scores under different evaluation settings (ranked).

Config	Model	Img. Success			Task Success			Overall			Rank
		T1	T2	T3	T1	T2	T3	T1	T2	T3	
Default	Seedream 4.0	75.930	55.580	41.590	75.930	75.580	76.110	83.811	69.948	59.757	1
	Nano Banana	70.700	50.660	35.350	70.700	72.590	68.240	81.483	67.703	56.242	2
	GPT-Image-1	73.120	54.890	37.970	73.120	74.440	72.740	76.959	65.340	53.542	3
	FLUX.1-Kontext-max	69.490	46.890	31.830	69.490	69.110	70.430	80.791	65.286	53.045	4
Threshold 0.3	Seedream 4.0	73.450	53.270	38.940	73.450	74.870	74.870	82.431	68.479	57.822	1
	Nano Banana	69.940	49.340	33.460	69.940	71.640	67.490	81.044	66.815	54.717	2
	GPT-Image-1	71.430	52.820	37.030	71.430	74.060	72.180	76.065	64.096	52.875	3
	FLUX.1-Kontext-max	68.170	45.570	30.510	68.170	68.360	68.930	80.020	64.361	51.933	4
Threshold 0.4	Seedream 4.0	73.980	52.570	38.940	73.980	73.450	74.510	82.728	68.027	57.822	1
	Nano Banana	68.810	47.260	31.950	68.810	69.570	64.460	80.386	65.392	53.469	2
	GPT-Image-1	70.110	50.190	35.150	70.110	72.560	71.240	75.359	62.480	51.515	3
	FLUX.1-Kontext-max	66.290	43.690	28.630	66.290	67.230	67.420	78.909	63.019	50.308	4
No Large-Box Filter	Seedream 4.0	75.930	55.220	41.240	75.930	75.220	76.110	83.811	69.721	59.505	1
	Nano Banana	70.700	50.660	35.350	70.700	72.590	68.240	81.483	67.703	56.242	2
	GPT-Image-1	72.930	54.890	38.350	72.930	74.440	73.120	76.859	65.340	53.809	3
	FLUX.1-Kontext-max	69.490	46.890	31.830	69.490	69.110	70.430	80.791	65.286	53.045	4

2.5.11.3 Detector Swap

We also perform an ablation study by swapping the open-vocabulary detector in our pipeline from Grounding DINO to alternative detectors, such as OWL-ViT [MGS22] and GLIP [LZZ22]. For this swap, detector accuracy is the primary factor we must prioritize, since a more accurate detector directly translates to higher agreement with human annotations.

Table 2.23 summarizes the performance of several popular open-vocabulary detectors on standard detection benchmarks. In the open-set setting ODinW (object detection in the wild), which most closely resembles our scenario, Grounding DINO outperforms GLIP. It also achieves the best AP on COCO, a widely used benchmark with 80 common objects. LVIS is a challenging benchmark with more than 1k categories spanning rare, common, and frequent objects; on LVIS^{val}, OWL-ViT attains strong performance, but Grounding DINO

still provides a better trade-off for our open-world editing setting, particularly when ODinW performance is considered.

We do not adopt other available detectors such as Grounding DINO 1.5, because they are closed-source and thus unsuitable for a community benchmark where users may need to run evaluation many times or adapt the pipeline to their own models. Moreover, our current configuration already achieves better alignment with human judgment (81.3%) than strong zero-shot VLM baselines, so switching to a closed-source detector would reduce reproducibility without a clear benefit.

When we swap Grounding DINO for OWL-ViT in our EdiVal-IF pipeline, the *relative* ranking of models (based on Img.Success at Turn 3) remains unchanged, but the *absolute* scores drop. Nevertheless, the per-turn/marginal success rates still exhibit high correlation with those under Grounding DINO (Pearson 0.82, Spearman 0.79). OWL-ViT frequently fails to detect objects that are clearly present in the image, which leads to many spurious “reject” decisions and thus lower Task.Success and Img.Success across all models. This bias is largely consistent across models, so EdiVal-IF preserves the same ordering, but the absolute values are shifted downward. A qualitative example is shown in Figure 2.19, where OWL-ViT fails to detect objects that are clearly present in the image.

We therefore do not recommend replacing Grounding DINO with OWL-ViT, GLIP, or other weaker open-vocabulary detectors in our framework. Grounding DINO remains a widely adopted state-of-the-art open-vocabulary detector, and weaker detectors not only reduce absolute performance scores but also harm agreement with human judgments. For example, when swapping Grounding DINO for OWL-ViT, the human agreement of EdiVal-IF drops from 81.30% to 53.67%. In general, upgrading to a stronger open-vocabulary detector should preserve the relative EdiVal-IF ranking while improving absolute scores and human agreement, whereas downgrading to significantly weaker detectors has the opposite effect and is therefore undesirable.

Method	Backbone	Pre-training data	COCO	LVIS ^{minival}					LVIS ^{val}				ODinW35	ODinW13
			AP _{all}	AP _{all}	AP _r	AP _c	AP _f		AP _{all}	AP _r	AP _c	AP _f	AP _{avg}	AP _{avg}
OWL-ViT	ViT-L	O365, OID, VG, LiT	42.2	-	-	-	-	34.6	31.2	-	-	-	-	-
GLIP	Swin-L	FourODs, GoldG, Cap24M	49.8	37.3	28.2	34.3	41.5	26.9	17.1	23.3	35.4	-	-	52.1
Grounding DINO	Swin-L	O365, OID, GoldG	52.5	-	-	-	-	-	-	-	-	-	26.1	56.9

Table 2.23: Performance of popular detectors including OWL-ViT, GLIP, and Grounding DINO (Swin-L). Numbers are copied from Table 1 of Grounding DINO 1.5 [RJL24]. In the open-set setting ODinW, which is most similar to our scenario, Grounding DINO outperforms GLIP. It also achieves the best AP on COCO. LVIS is a large-scale benchmark with over 1k categories spanning rare, common, and frequent objects.

Can detector swapping avoid failure cases? No. We also examine the same failure cases of Grounding DINO under OWL-ViT and GLIP. In these examples, OWL-ViT behaves differently from Grounding DINO but does not fix the underlying issue: regardless of whether the object truly exists in the image, OWL-ViT often detects nothing at all. GLIP, on the other hand, exhibits similar false-positive behavior to Grounding DINO, but with denser and less precise bounding boxes. Thus, switching to OWL-ViT or GLIP does not eliminate such failure cases; it merely changes them into systematic missed detections or more cluttered false positives.

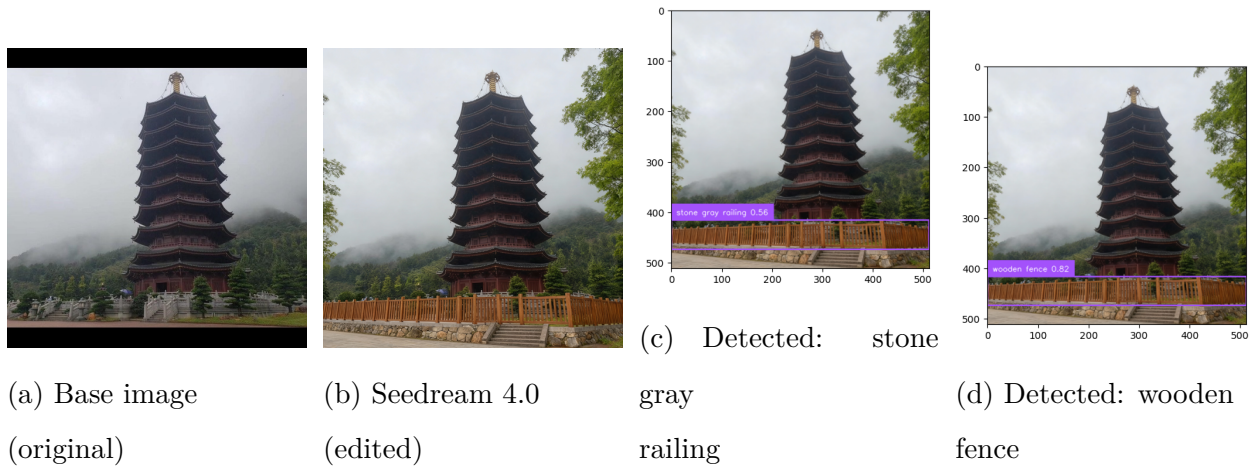


Figure 2.18: **Failure due to detector false positives.** Although the edit visually replaces the railing with a wooden fence, Grounding DINO fires on both “stone gray railing” and “wooden fence” in overlapping regions, causing an incorrect failure in our instruction-following metric.



Figure 2.19: OWL-ViT fails to detect the queried objects regardless of whether they are actually present in the image, and therefore does not resolve this failure case.

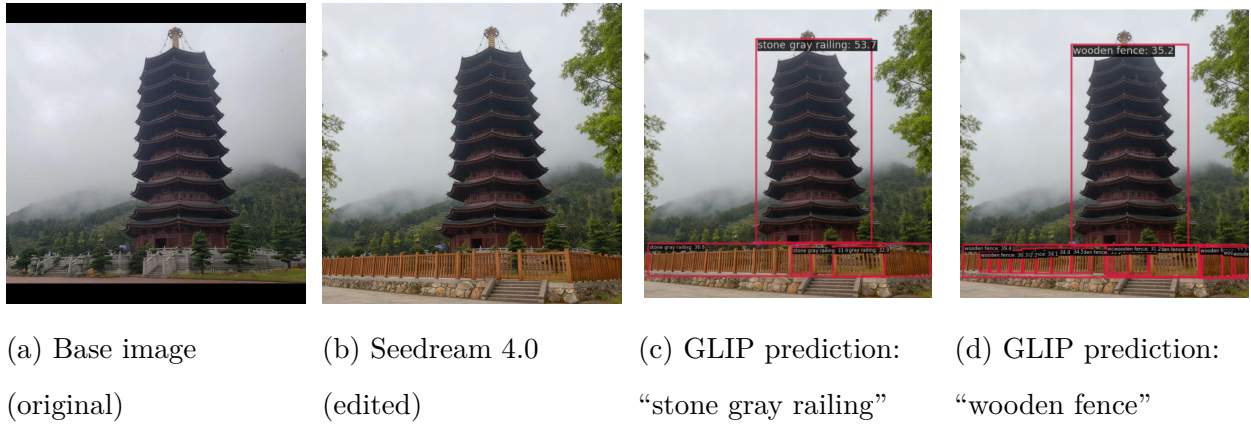


Figure 2.20: GLIP produces substantially more false positives than Grounding DINO, hallucinating objects such as a “stone gray railing” and misclassifying the tower region as a “wooden fence” or “stone gray railing”.

2.5.11.4 Image Feature Extractor Swap: DINOv3 to DINOv2

We further ablate the image feature extractor by swapping the default backbone (DINOv3) with DINOv2. As with our other component swaps (VLM, detector, and threshold), we observe the same qualitative behavior: the *relative* ranking of models remains unchanged, while the *absolute* consistency scores shift.

Concretely, DINOv2 systematically produces lower EdiVal-CC values than DINOv3, but the ordering across models is identical. When we compare per-turn consistency scores across the two backbones over all (model, turn) pairs, we obtain Pearson correlation 0.9987 and Spearman correlation 1.0000, indicating that DINOv2 and DINOv3 induce essentially the same ranking and very similar relative differences between models, despite the shift in absolute scale.

Table 2.24: DINO-v2 feature backbone results with EdiVal-CC Rank.

Model	Turn 1	Turn 2	Turn 3	EdiVal-CC Rank
Seedream 4.0	89.455	84.185	81.120	3
Nano Banana	90.820	86.900	85.070	1
GPT-Image-1	76.360	72.555	69.845	4
FLUX.1-Kontext-max	91.530	87.535	84.460	2

Table 2.25: DINO-v3 feature backbone results with EdiVal-CC Rank.

Model	Turn 1	Turn 2	Turn 3	EdiVal-CC Rank
Seedream 4.0	92.51	88.03	85.86	3
Nano Banana	93.91	90.48	89.48	1
GPT-Image-1	81.00	77.78	75.50	4
FLUX.1-Kontext-max	93.93	90.90	88.40	2

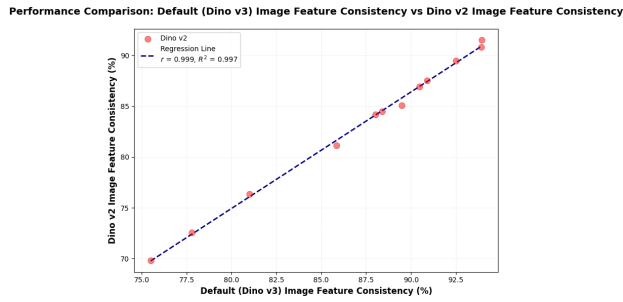


Figure 2.21: Per-turn EdiVal-CC scores under DINOv3 (x-axis) vs. DINOv2 (y-axis) for the four representative models. Each point is a (model, turn) pair; Pearson correlation is 0.9987 and Spearman correlation is 1.0000, confirming that the feature-extractor swap only shifts the absolute scale while preserving the ranking.

CHAPTER 3

MT-EditFlow: Reinforcement Learning for Multi-Turn Image Editing with Flow Matching

3.1 Introduction

Recent advancements in instruction-based image editing [Dee25, Ope25, GGG25, Lab25, LBB25, WZY25, XWZ25, LHX25, WLZ25] have enabled models to handle complex, real-world requests. However, a significant gap remains: while humans naturally edit images through iterative, multi-turn dialogues [YHL25, YHZ25, CZZ26], existing models are primarily trained for “one-shot” interactions. When applied to sequential editing, these models suffer severe degradation in instruction adherence and visual consistency. This degradation stems from two primary factors (see Fig. 3.1 for qualitative illustrations):

- **The “All-or-Nothing” Requirement:** Multi-turn editing demands sequential perfection. A single failed turn (e.g., a sleep failing to be removed) compromises the entire sequence, leaving the user dissatisfied even if all other edits succeed.
- **Error Propagation:** Existing models struggle when conditioned on their own imperfect outputs, known as exposure bias [Sch19, CZZ26]. Errors compound over multiple steps, derailing subsequent turns. For instance, if the model mistakenly transforms a white horse into a deer in turn two, a third edit intended for the horse is effectively doomed.

Recent post-training methods for flow-matching models [LGL22a, LCB22], including GRPO

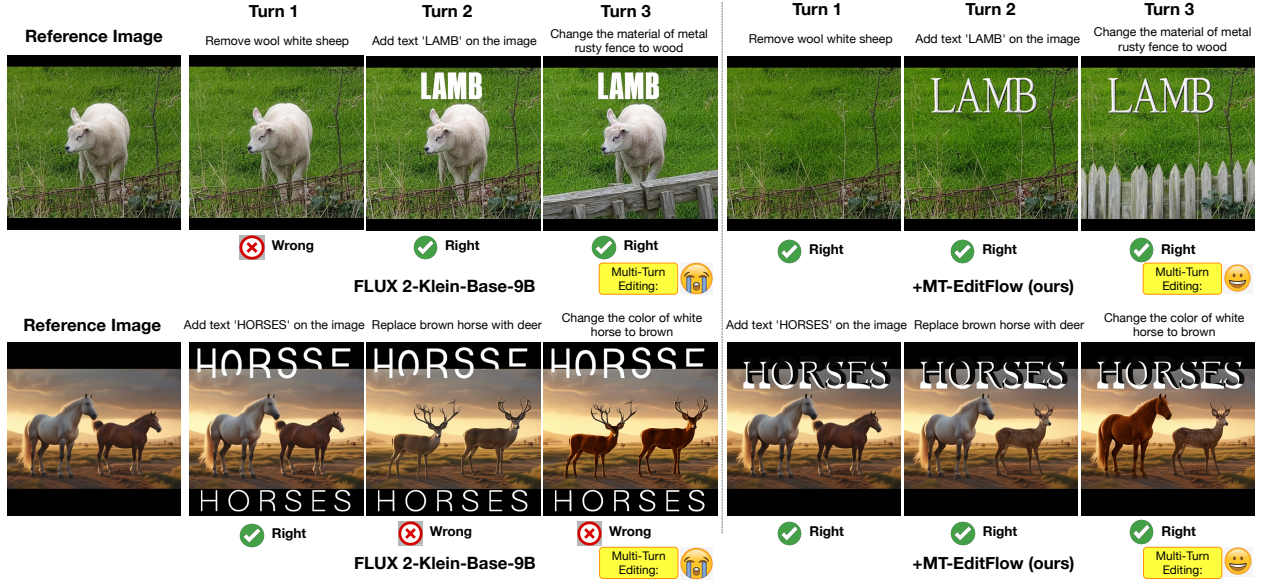


Figure 3.1: **Qualitative results of MT-EditFlow (Ours).** In multi-turn image editing, a single failed turn compromises the entire sequence, even if other turns succeed (e.g., a sheep failing to be removed). Furthermore, *error accumulation* is a common pitfall: if an intermediate edit is incorrect, subsequent turns are often derailed. For instance, using the FLUX.2 base model, if a white horse is mistakenly transformed into a deer during the second turn, the third edit intended for the horse is effectively doomed. By contrast, MT-EditFlow enables the model to execute robust, successful edits across multiple consecutive turns, effectively preventing error propagation.

[LLL25, XWG25] and DiffusionNFT [ZCY26], demonstrate that reinforcement learning (RL) [SB99] can effectively align generative models [RBL22a, Lab25, WLZ25] with reward signals. Nevertheless, these approaches strictly target single-shot image generation. Because they operate on a single-turn, single-reward paradigm, they do not explicitly model the temporal dependencies of multi-turn interactions, nor do they address the quality drift that arises during chained edits.

To address these challenges, we introduce **MT-EditFlow**, a flow-matching reinforcement learning framework that tailors reward signal design for sequential image editing by integrating a *multi-turn* perspective with a *multi-reward* formulation, ultimately providing a unified reward structure applicable to both GRPO- and NFT-based RL methods. We focus on algorithms based on flow-matching models due to their superior performance, scalability, and fast inference efficiency, as demonstrated by recent open-source SOTA editing models such as Qwen-Image-Edit [WLZ25], the FLUX series [Lab25, LBB25], and Step1X-Edit [LHX25]. Prior work implements editing evaluation by zero-shot prompting a SOTA VLM (usually closed-source) [YHL25, LHX25, LLZ25], utilizing a fine-tuned VLM for more specialized evaluation [LWW26, WJK26], or employing a combination of VLMs and expert tools in an agentic framework [CZZ26]. For fast RL training, MT-EditFlow adopts zero-shot prompting of SOTA VLMs (i.e., Qwen3-VL-8B [YLY25]) to evaluate Instruction Following (IF), paired with EdiVal-CC [CZZ26] to precisely measure Content Consistency (CC) in terms of semantic-level distance in unedited regions. Crucially, we comprehensively study and optimize reward signal design—an area notably underexplored in previous work—by systematically analyzing:

- **Scoring Strategies:** How to effectively aggregate multiple turn-level rewards.
- **Evaluator Modes:** The trade-off between reward bias and variance when comparing “thinking” versus “non-thinking” VLM modes.
- **Advantage Fusion Levels:** How to balance competing reward aspects (IF vs. CC)

to avoid reward hacking.

To support this framework, we constructed $\sim 2.5\text{K}$ high-quality multi-turn prompt chains with GPT-4o [AAA23a] using the EdiVal-Agent pipeline [CZZ26]. We then applied MT-EditFlow to extend both GRPO-based and NFT-based diffusion RL methods to the sequential editing setting. Our results demonstrate that broadcasting the aggregated advantage across the entire multi-turn trajectory effectively aligns local planning decisions with global task success, ensuring more cohesive sequential edits. Extensive experiments demonstrate that MT-EditFlow significantly improves sequential editing performance across diverse base models. Notably, it boosts FLUX.1-Kontext-dev [LBB25] by 6.85 points in turn-3 overall performance, surpassing state-of-the-art open-source Qwen-Image-Edit [WLZ25] models, and improves FLUX.2-klein-base [Lab25] by 2.90 points, pushing the boundary of current open-source models. By maintaining high marginal editing success rates, MT-EditFlow consistently reduces exposure bias over successive turns. Ultimately, our results show that MT-EditFlow provides a unified reward signal design applicable to both GRPO and NFT-style RL algorithms, effectively unlocking robust, multi-turn capabilities in flow-based image editors.

3.2 Related Work

3.2.1 Instruction-Based Image Editing

InstructPix2Pix (IP2P) [BHE23] proposed a two-stage procedure for turning a text-to-image diffusion model [RBL22a, ZYW25] into an image editor: (i) generate paired editing examples using Stable Diffusion [RBL22a] together with training-free methods such as Prompt-to-Prompt [HMT23a]; and (ii) fine-tune the diffusion model on these synthetic pairs. Follow-up diffusion-based systems, including MagicBrush [ZMC23], UltraEdit [ZMC24], and AnyEdit [YCY25], extend this approach to large-scale, fine-grained editing of real images. More recent approaches (e.g., OmniGen [XWZ25, WZY25], Step1X-Edit [LHX25], FLUX.1 Kontext

[LBB25], Qwen-Image-Edit [WLZ25], and Seedream [GGG25]) introduce task-specialized architectures and increasingly adopt flow-matching frameworks [LGL22a, LCB22, ZYZ24]. In parallel, a separate research direction has focused on LLM-native editors (typically closed-source), such as Gemini 2.0 Flash Image [Gem25], Nano Banana 1/2 [Dee25], and GPT-Image-1/1.5 [Ope25]. These systems support in-context multi-turn editing: users iteratively adjust an image through dialogue, while the model tracks and leverages a consistent editing history—an interaction mode shown to be effective for multi-step image refinement.

To the best of our knowledge, we present the first RL-based framework aimed at narrowing the gap between current open-source editors and such multi-turn editing capabilities, with a particular focus on a unified reward signal design.

3.2.2 Image Editing Evaluation and Benchmarks

Recent work relies exclusively on vision-language models (VLMs) as interpretable judges—e.g., VIEScore [KJW23], GEdit-Bench [LHX25], I²EBench [MJY24], HQ-Edit [HYZ24], Complex-Edit [YHZ25], and ImgEdit [YHL25]—by querying models such as GPT-4o [Ope25] about specific aspects of an edit. While VLMs provide holistic, language-mediated assessments, they fall short in several ways: they are poor at spatial reasoning [ZHL25, CYF24, CXK24, QGF24, CZF25], prone to hallucinating objects and attributes [BWX24], insensitive to subtle pixel-level changes [VNT25] (e.g., fine structures or small attribute shifts), and miscalibrated for artifacts and aesthetics [LHL24, XLW23, MWS25]. EdiVal-Agent [CZZ26] mitigates these limitations by integrating VLM reasoning with grounding tools, symbolic and object-centric pixel- and semantic-level metrics, and human preference models, producing precise and interpretable evaluation for instruction-based editing.

In this paper, we evaluate MT-EditFlow primarily on the multi-turn benchmark EdiVal-Bench [CZZ26], with supplementary evaluations on widely-used single-turn benchmarks, ImgEdit [YHL25].

3.2.3 Reinforcement Learning with Diffusion and Flow Matching

A central challenge in extending RL [SB99] from discrete autoregressive models [BMR20, AAA23a] to diffusion models [HJA20a, SSK20] lies in the intractability of exact likelihoods in diffusion [SDM21], which undermines many standard RL algorithms. Early PPO-style [SWD17] adaptations, such as DDPO [BJD23] and DPOK [FWD23], factorize trajectory likelihoods step-by-step along the reverse process, but ignore forward consistency. Subsequent GRPO-based [GYZ25] extensions, including Flow-GRPO [LLL25] and DanceGRPO [XWG25], discretize reverse diffusion to enable gradient optimization and demonstrate improved scalability. However, they tightly couple the training objective with specific SDE solvers, leading to forward–reverse inconsistency, solver restrictions, and limited efficiency. MixGRPO [LCH25] enhances efficiency by blending SDE and ODE formulations, yet retains these structural constraints. More recently, Diffusion Negative-aware FineTuning (DiffusionNFT) [ZCY26] reframes diffusion RL through the forward process via flow matching, contrasting positive and negative samples to define implicit policy improvement without explicit likelihood estimation or reliance on particular samplers. This decoupled formulation substantially improves efficiency over Flow-GRPO and even eliminates the need for classifier-free guidance [HS22a].

In our work, MT-EditFlow extends both Flow-GRPO and DiffusionNFT with MT-Reward to enable effective RL for multi-turn image editing, and demonstrates consistent improvements across different base models.

3.3 Background

3.3.1 Flow Matching

Flow Matching (FM) [LGL22a, LCB22] defines a continuous-time transformation between a noise distribution p_1 and a data distribution p_0 . Let $x_0 \sim p_0$ be a data sample and

$x_1 \sim p_1$ be standard Gaussian noise. A probability path $p_t(x)$ is constructed to bridge these distributions. In the case of *Rectified Flow* [LGL22a, SVS25], the latent x_t is defined by the linear interpolation: $x_t = (1 - t)x_0 + tx_1$. The model is trained to learn a time-dependent vector field $v_\theta(x_t, t)$ that predicts the velocity $v_t = \frac{dx_t}{dt} = x_1 - x_0$. The flow matching objective is minimized as:

$$\mathcal{L}_{FM}(\theta) = \mathbb{E}_{t \sim \mathcal{U}[0,1], x_0 \sim p_0, x_1 \sim p_1} [\|v_\theta(x_t, t) - (x_1 - x_0)\|^2]. \quad (3.1)$$

After training is done, inference is usually performed by numerically solving the ODE $dx_t = v_\theta(x_t, t)dt$ from $t = 1$ to $t = 0$ using an ODE solver.

3.3.2 Reinforcement Learning for Image Editing

Let $\mathcal{D}$ denote the training set, (x_f, c) be the pair of reference image and editing instruction in the training dataset, and π_θ is the policy induced by the flow matching editing model. Fundamentally, the objective of reinforcement learning [SB99] for single-turn image editing is to maximize the reward [LLZ25, LWW26]:

$$\mathcal{J}(\theta) = \mathbb{E}_{(x_f, c) \sim \mathcal{D}, x_0 \sim \pi_\theta(\cdot | x_f, c)} [r(x_0, x_f, c)], \quad (3.2)$$

where the reward function r evaluates editing success. While r may comprise evaluations across multiple dimensions of an edit, we aggregate them into a single, unified function r for conciseness.

Extending Reinforcement Learning (RL) [SB99] from discrete autoregressive models [BMR20, AAA23a] to diffusion and flow models [HJA20a, LGL22a] is challenging due to the intractability of exact likelihoods.

3.3.2.1 Flow-GRPO

Flow-GRPO [LLL25] addresses this by approximating the likelihood via a stochastic denoising process. Using an SDE sampler [SSK20] that shares the same marginal distribution

as the deterministic flow, Flow-GRPO enables gradient-based optimization through Euler-Maruyama discretization, where each sampling step is modeled as a Gaussian distribution. For a given reference image x_f and editing instruction c , the policy π_θ generates a group of G outputs $\{x_{0,i}\}_{i=1}^G$. Each sample $x_{0,i}$ is produced through a T -step stochastic process $\{x_{t,i}\}$ starting from Gaussian noise $x_{1,i} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and terminating at data $x_{0,i}$. The advantage A_i is computed by normalizing the rewards $R_i = r(x_{0,i}, x_f, c)$ within the group:

$$A_i := \text{Adv}(R_i) = \frac{R_i - \frac{1}{G} \sum_{j=1}^G R_j}{\text{std}(\{R_j\}_{j=1}^G)}. \quad (3.3)$$

The Flow-GRPO objective is optimized by maximizing (we ignore the expectation condition for brevity):

$$\mathcal{J}_{\text{Flow-GRPO}}(\theta) = \mathbb{E} \left[\frac{1}{G} \sum_{i=1}^G \left(\frac{1}{T} \sum_{t=1}^T (\mathcal{L}_{\text{clip}}^{\text{GRPO}}(\rho_{t,i}(\theta), A_i) - \beta \mathbb{D}_{KL}(\pi_\theta \| \pi_{\text{ref}})) \right) \right], \quad (3.4)$$

where the clipped surrogate $\mathcal{L}_{\text{clip}}^{\text{GRPO}}(\rho, A) = \min(\rho A, \text{clip}(\rho, 1 - \epsilon, 1 + \epsilon)A)$, the step-wise importance ratio $\rho_{t,i}(\theta) = \frac{p_\theta(x_{t-\Delta t,i} | x_{t,i}, x_f, c)}{p_{\theta_{\text{old}}}(x_{t-\Delta t,i} | x_{t,i}, x_f, c)}$, and the π_{ref} is the reference policy used for regularization.

3.3.2.2 DiffusionNFT

Another line of work, DiffusionNFT [ZCY26], performs implicit policy improvement by contrasting positive and negative samples, without explicit likelihood estimation or reliance on specific samplers. Specifically, the DiffusionNFT objective is defined as (we ignore the expectation condition for brevity):

$$\mathcal{J}_{\text{DiffusionNFT}}(\theta) = \mathbb{E} \left[\frac{1}{G} \sum_{i=1}^G \left(\frac{1}{T} \sum_{t=1}^T (\mathcal{L}_{\text{clip}}^{\text{NFT}}(A_i) \| v_\theta^+ - v \|_2^2 + (1 - \mathcal{L}_{\text{clip}}^{\text{NFT}}(A_i)) \| v_\theta^- - v \|_2^2) \right) \right]. \quad (3.5)$$

Here $v_\theta^+(x_{t,i}, t)$ and $v_\theta^-(x_{t,i}, t)$ denote the positive and negative velocities parameterized by θ with inputs $(x_{t,i}, t, x_f, c)$, and A_i is computed in the same way as in Eq. 3.3. In practice, the clipped reward function is defined as $\mathcal{L}_{\text{clip}}^{\text{NFT}}(A) := \frac{1}{2} + \frac{1}{2} \text{clip}(A/z_{\text{max}}, -1, 1)$.

3.4 MT-EditFlow: RL for Multi-Turn Image Editing

We note that existing RL methods for flow-matching models typically consider only a single reward (excluding the KL divergence with a reference model) and operate in a single-turn setting. In contrast, MT-EditFlow extends this framework from two perspectives: a *multi-reward* structure and a *multi-turn* formulation. Specifically, we introduce rewards for both instruction following (IF) and content consistency (CC), which are the two most important aspects of image editing. Our goal is to provide a unified reward signal design for multi-turn editing that is applicable to both GRPO-style and NFT-style RL algorithms.

We first formulate the multi-turn editing setting. We then identify three key factors that significantly affect the reward signal’s effectiveness. In particular, we study: (1) the score aggregation strategy across turns for effective RL optimization, (2) the fusion level of multiple rewards to avoid reward hacking, and (3) the use of thinking-mode reward evaluation to trade off reward bias and variance and boost overall performance.

3.4.1 MT-EditFlow Reward Objective

Suppose we have a multi-turn editing task with K turns. The training dataset $\mathcal{D}$ consists of pairs $(x_f, \mathcal{C})$, where $\mathcal{C} = [c_1, c_2, \dots, c_K]$ is a sequence of editing instructions applied to the same reference image x_f in order. As a result, the model produces a sequence of edited images $\mathcal{X} = [x_0^{(1)}, x_0^{(2)}, \dots, x_0^{(K)}]$. We denote $\mathcal{X}_{\text{int}} = [x_0^{(1)}, x_0^{(2)}, \dots, x_0^{(K-1)}]$ as the intermediate edits, which are typically not directly evaluated by users. Strictly speaking, the multi-turn editing task can be defined solely based on the final result $(x_0^{(K)}, x_f, \mathcal{C})$, without explicitly considering the intermediate edits, as illustrated in Fig. 3.1. In the following, we first discuss two key aspects of editing quality at each turn, and then describe how these two aspects are fused at the reward level.

3.4.1.1 Instruction Following (IF)

Marginal instruction-following success at turn k is defined over $(x_0^{(k)}, x_0^{(k-1)}, c_k)$. For notation consistency, we define $x_0^{(0)} := x_f$. This follows the standard single-turn editing formulation. Prior work evaluates editing performance by zero-shot prompting a state-of-the-art (SOTA) VLM (often closed-source) [YHL25, LHX25, LLZ25], using fine-tuned VLMs for more specialized evaluation [LWW26, WJK26], or employing a combination of VLMs and expert tools in an agentic framework [CZZ26]. In this work, for RL training efficiency and simple reward evaluation, we implement the instruction-following (IF) reward by prompting the open-source VLM Qwen3-VL-8B. Specifically, we query a VLM reward model with the inputs $\{(x_0^{(k)}, x_0^{(k-1)}, c_k)\}_{k=1}^K$ and obtain the IF score $r_{\text{IF}}^{(k)} = \Phi(x_0^{(k)}, x_0^{(k-1)}, c_k)$ for each turn k , where Φ denotes the VLM evaluator. Under the strict multi-turn definition, overall success can be defined as $r_{\text{IF}}^{\text{mt}} = \bigwedge_{k=1}^K r_{\text{IF}}^{(k)}$, where $\bigwedge$ denotes the logical AND operator. However, in our experiments we find that this strict formulation does not provide an effective learning signal for RL. We therefore also consider an alternative aggregation that averages the turn-level rewards: $r_{\text{IF}}^{\text{avg}} = \frac{1}{K} \sum_{k=1}^K r_{\text{IF}}^{(k)}$.

3.4.1.2 Content Consistency (CC)

Prior work typically computes pixel-level or semantic distances directly between the reference image and the edited image [ZMC23, ZMC24, YCY25, SPS24]. This approach ignores the fact that regions intended to be changed should not be included in the consistency evaluation. To more precisely measure content consistency (CC) and capture the essence of a successful edit, we adopt EdiVal-CC, which is proposed in [CZZ26]. Formally, we denote our CC reward as $r_{\text{CC}}^{(k)} = \text{EdiVal-CC}(x_0^{(k)}, x_0^{(0)}, \Omega)$, where Ω is provided by EdiVal-CC and indicates the regions that should remain unchanged, excluding areas intended for editing. Unlike instruction-following, $r_{\text{CC}}^{(k)}$ already aggregates consistency across the editing chain by comparing the reference image with the k -turn image. Note that evaluating consistency be-

tween the $(k-1)$ -turn and k -turn images is not meaningful, as the $(k-1)$ -turn may already have deviated significantly from the reference image even if the k -turn appears consistent.

3.4.1.3 Discussion on Visual Quality (VQ)

As discussed in GIE-Bench [QLF25] and EdiVal-Bench [CZZ26], editing models such as GPT-Image-1 that prioritize visual quality often exhibit poor fidelity to the original image’s content and style. Consequently, using a pure VQ reward, or computing VQ differences between the reference and edited images, remains inconclusive. Therefore, we do not include VQ in our reward structure.

3.4.1.4 Reward-Level Fusion

Straightforwardly, we obtain the final reward by a weighted sum of the IF and CC rewards (at index i within the group), extending the single-turn reward R_i to the multi-turn setting:

$$R_i^{\text{MT-EditFlow}}(\mathcal{X}_i, x_f, \mathcal{C}) = \lambda_{\text{IF}} r_{i,\text{IF}} + \lambda_{\text{CC}} r_{i,\text{CC}}^{(K)}, \quad (3.6)$$

where $r_{i,\text{IF}}$ has 3 different variants as described below. We fix $\lambda_{\text{IF}} = 1$ and sweep λ_{CC} in our main experiments.

3.4.2 Scoring Strategy: Granularity and Sparsity

We prompt the VLM with different scoring rubrics. Specifically, we consider two levels of granularity: a binary success indicator $y_i^{(k)} \in \{0, 1\}$ for turn k of sample i , and a fine-grained Likert score $s_i^{(k)} \in \{1, \dots, 5\}$. We further study different operators for aggregating turn-level scores across the trajectory. Combining both dimensions, we consider the following three scoring strategies:

- **Binary MT ($r_{i,\text{IF}}^{\text{mt}}$):** A strict indicator where $r_{i,\text{IF}}^{\text{mt}} = \bigwedge_{k=1}^K y_i^{(k)}$, yielding 1 only if all turns succeed.

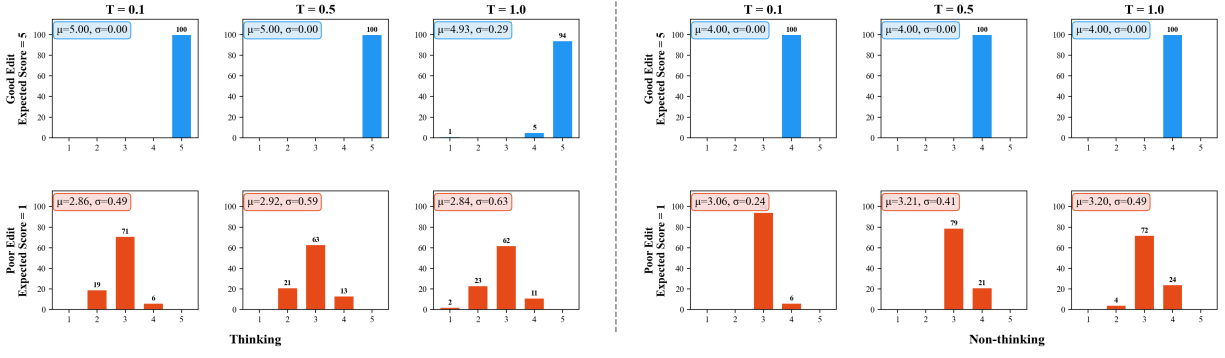


Figure 3.2: **Illustration of the reward bias and variance trade-off between thinking and non-thinking modes.** Thinking mode consistently leads to reduced reward bias and increased variance across different temperatures. For example, the mean evaluation score under thinking mode for poor edits is closer to 1, the lowest score.

- **Binary Average** ($r_{i,IF}^{\text{avg}}$): The mean of per-turn binary success, $r_{i,IF}^{\text{avg}} = \frac{1}{K} \sum_{k=1}^K y_i^{(k)}$.
- **Fine-grained Average** ($r_{i,IF}^{\text{fg}}$): The mean of fine-grained scores, $r_{i,IF}^{\text{fg}} = \frac{1}{K} \sum_{k=1}^K s_i^{(k)}$.

While $r_{i,IF}^{\text{mt}}$ aligns with the strict definition of multi-turn success, it suffers from extreme reward sparsity. For a per-turn success probability p , the expected number of positive samples in a group of size G is Gp^K . As K increases, the probability of a null gradient or high-variance advantage increases. In contrast, the fine-grained reward $r_{i,IF}^{\text{fg}}$ provides partial credit, preserving a dense learning signal even when most of turns fail.

3.4.3 Evaluator Mode: Thinking vs. Non-thinking

The reliability of the VLM reward model Φ is influenced by the prompting structure. We compare a **Non-thinking** mode (direct score output) with a **Thinking** mode, where the VLM is prompted to generate a Chain-of-Thought (CoT) analysis of the editing trajectory $\mathcal{X}$ relative to x_f and $\mathcal{C}$ before producing $r_{IF}^{(k)}$. Empirically, the thinking mode is better aligned with human judgments: the non-thinking mode can mis-score obvious good edits and occasionally assign overly high scores to poor edits, while the thinking mode produces

a more accurate score distribution, albeit with higher variance (Fig. 3.2). We provide the prompt details in the Appendix.

3.4.4 Fusion Level: Reward vs. Advantage

When jointly optimizing Instruction Following (IF) and Content Consistency (CC), we compare two fusion levels. **Reward-Level Fusion** (Eq. 3.6) sums the raw signals before computing a single group-relative advantage A_i . However, because r_{IF} and r_{CC} operate on different scales and noise distributions, this can hurt the policy optimization. Alternatively, we propose **Advantage-Level Fusion**, where we compute independent advantages $A_{i,\text{IF}}$ and $A_{i,\text{CC}}$ for each objective and then aggregate them:

$$A_i^{\text{MT-EditFlow}} = \lambda_{\text{IF}} A_{i,\text{IF}} + \lambda_{\text{CC}} A_{i,\text{CC}}, \quad (3.7)$$

where $A_{i,\text{IF}} := \text{Adv}(R_{i,\text{IF}}) = \text{Adv}(r_{i,\text{IF}}(\mathcal{X}_i, x_f, \mathcal{C}))$, and similarly for $A_{i,\text{CC}}$. By standardizing the signals within the group before fusion, we prevent one reward component from dominating the gradient direction due to scale imbalances.

3.4.5 Trajectory-Level Advantage Broadcasting

Given the multi-turn reward defined in Eqs. 3.6 and 3.7, we broadcast the same trajectory-level advantage to every turn in the editing chain:

$$A_i^{(k)} := \text{Adv}(R_i^{\text{MT-EditFlow}}) \quad \text{or} \quad A_i^{\text{MT-EditFlow}}, \quad k = 1, \dots, K. \quad (3.8)$$

For both Flow-GRPO and DiffusionNFT, this mechanism propagates a single trajectory-level reward signal to all denoising steps and editing turns. As a result, the optimization objective aligns local planning decisions with the global success of the multi-turn editing trajectory.

Table 3.1: **Results of multi-turn editing on EdiVal-Bench.** EdiVal-IF (Instruction Following), EdiVal-CC (Content Consistency), and EdiVal-O (Overall) across three sequential editing turns.

Model	EdiVal-IF			EdiVal-CC			EdiVal-O		
	T1	T2	T3	T1	T2	T3	T1	T2	T3
<i>Closed-Source Models</i>									
Seedream 4.0	75.93	55.58	41.59	92.51	88.03	85.86	83.81	69.95	59.76
GPT-Image-1.5	75.19	55.92	40.08	94.49	91.20	88.49	84.29	71.41	59.55
Nano Banana 2	73.89	54.17	38.61	93.54	90.52	88.61	83.14	70.02	58.49
FLUX.2-max	75.55	55.27	39.36	92.91	88.30	85.78	83.78	69.86	58.10
Nano Banana	70.70	50.66	35.35	93.91	90.48	89.48	81.48	67.70	56.24
GPT-Image-1	73.12	54.89	38.35	81.00	77.78	75.50	76.96	65.34	53.81
FLUX.1-Kontext-max	69.49	46.89	31.83	93.93	90.90	88.40	80.79	65.29	53.04
Gemini 2.0 Flash	68.07	45.96	28.42	90.58	85.10	80.88	78.52	62.54	47.94
<i>Open-Source Models</i>									
IP2P	37.41	10.66	2.80	76.85	68.36	60.30	53.62	26.99	12.99
MagicBrush	42.31	15.73	4.90	86.96	81.26	76.86	60.66	35.75	19.41
AnyEdit	41.07	16.32	7.22	86.42	78.91	70.10	59.58	35.89	22.50
UltraEdit	51.37	17.70	6.36	86.80	84.50	82.40	66.78	38.67	22.89
OmniGen	54.72	24.48	10.66	93.00	88.42	83.92	71.34	46.52	29.91
Step1X-Edit	61.89	34.97	17.83	92.76	88.52	85.21	75.77	55.64	38.98
Qwen-Image-Edit	72.90	44.06	22.55	84.22	80.52	77.98	78.36	59.56	41.93
FLUX.1-Kontext-dev	59.97	32.69	16.61	95.32	92.24	90.22	75.61	54.91	38.71
+ MT-EditFlow-NFT	62.24	38.99	22.90	95.75	92.47	90.62	77.20	60.04	45.56
Δ	+2.27	+6.30	+6.29	+0.43	+0.23	+0.40	+1.59	+5.13	+6.85
FLUX.2-Klein-base-9B	67.83	45.63	28.67	94.57	90.52	87.25	80.09	64.27	50.01
+ MT-EditFlow-NFT	70.28	48.08	31.47	95.18	91.53	88.95	81.79	66.34	52.91
Δ	+2.45	+2.45	+2.80	+0.61	+1.01	+1.70	+1.70	+2.07	+2.90

3.5 Experiments

Our training and evaluation data are built on the Pico-Banana-400K dataset [QBS25], whose images originate from OpenImages [KRA20]. To construct multi-turn RL trajectories, we use the EdiVal-Agent pipeline [CZZ26] to generate object-centric and context-consistent instruction sequences with $K=3$ turns per reference image. In total, the dataset contains 2,319 reference images with planned 3-turn editing trajectories across 9 editing types and 12 semantic object categories, yielding **6,957 turn-level steps** for online RL fine-tuning.

For evaluation, we mainly utilize EdiVal-Bench [CZZ26], which contains completely *different* reference images and instruction chains from training to assess the performance of our

model in multi-turn editing setting, with supplementary evaluations on widely-used single-turn benchmarks, ImgEdit [YHL25].

3.5.1 Training Details

We use FLUX.1-Kontext-dev [LBB25] and FLUX.2-klein-base-9B [Lab25] as our primary base models and fine-tune them with Low-Rank Adaptation (LoRA) [HSW22]. For efficiency, we employ Fully Sharded Data Parallelism (FSDP) [ZGV23] and DeepSpeed ZeRO [RRR20] alongside gradient checkpointing, and serve VLM-based rewards via a vLLM endpoint during online rollouts.

We train MT-EditFlow combined with either Flow-GRPO [LLL25] or Diffusion-NFT [ZCY26] depending on the setting. Unless otherwise specified, we keep the training recipe fixed within each backbone/objective combination and vary only the factor of interest; we study reward signal design in Sec. 3.5.3 and provide other ablations in Sec. 3.5.4. We fix $\lambda_{IF}=1$ and sweep λ_{CC} where indicated.

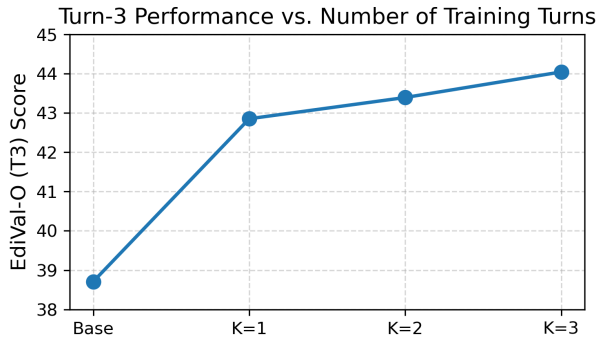


Figure 3.3: Multi-turn editing performance increases with the number of training turns K used.

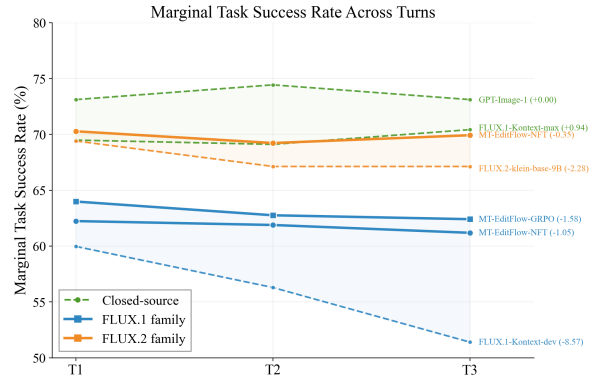


Figure 3.4: Marginal task success across turns. With MT-EditFlow, performance remains stable, reducing exposure bias.

3.5.2 Main Results

Our experiments reveal several important insights regarding MT-EditFlow:

Table 3.2: Single-turn editing results of MT-EditFlow on ImgEdit-Bench. GPT-4.1 is used for evaluation. Our multi-turn RL algorithm naturally leads to improved single-turn performance.

Model	Add	Adjust	Extract	Replace	Remove	Background	Style	Hybrid	Action	Overall↑
ICEdit	3.58	3.39	1.73	3.15	2.93	3.08	3.84	2.04	3.68	3.05
Step1X-Edit	3.88	3.14	1.76	3.40	2.41	3.16	4.63	2.64	2.52	3.06
UniWorld-V1	3.82	3.64	2.27	3.47	3.24	2.99	4.21	2.96	2.74	3.26
BAGEL	3.81	3.59	1.58	3.85	3.16	3.39	4.51	2.67	4.25	3.42
OmniGen2	3.57	3.06	1.77	3.74	3.20	3.57	4.81	2.52	4.68	3.44
GPT-Image-1	4.61	4.33	2.90	4.35	3.66	4.57	4.93	3.96	4.89	4.20
UniWorld-V2	4.29	4.44	4.32	4.69	4.72	4.41	4.91	3.83	4.83	4.49
FLUX.2-Klein-base-9B	4.42	4.23	2.30	4.42	3.86	4.23	4.94	3.43	4.72	4.06
+ MT-EditFlow-NFT	4.42	4.23	2.33	4.50	4.04	4.39	4.94	3.62	4.81	4.14
Δ	–	–	+0.03	+0.08	+0.18	+0.16	–	+0.19	+0.09	+0.08

Table 3.3: Comparison of MT-EditFlow variants on FLUX.1-Kontext-dev.

Method	EdiVal-IF			EdiVal-CC			EdiVal-O		
	T1	T2	T3	T1	T2	T3	T1	T2	T3
MT-EditFlow-GRPO	60.66	35.49	21.50	95.35	92.26	90.24	76.04	57.22	44.04
MT-EditFlow-NFT	62.24	38.99	22.90	95.75	92.47	90.62	77.20	60.04	45.56

1. **Substantial Multi-turn Improvement:** As shown in Tab. 3.1, MT-EditFlow significantly improves overall multi-turn performance (EdiVal-O) over the corresponding open-source backbones, narrowing the gap to strong closed-source systems. The gains are particularly pronounced at later turns, with a 6.85-point increase for FLUX.1-Kontext-dev and a 2.90-point increase for FLUX.2-klein-base-9B.
2. **Single-turn Performance:** MT-EditFlow also enhances single-turn editing performance, as shown in Tab. 3.2. These results demonstrate that our RL approach *generalizes* well to different reference image sources and benchmarks.
3. **Effect of Training Turns:** Fig. 3.3 further shows that the EdiVal-O (T3) score

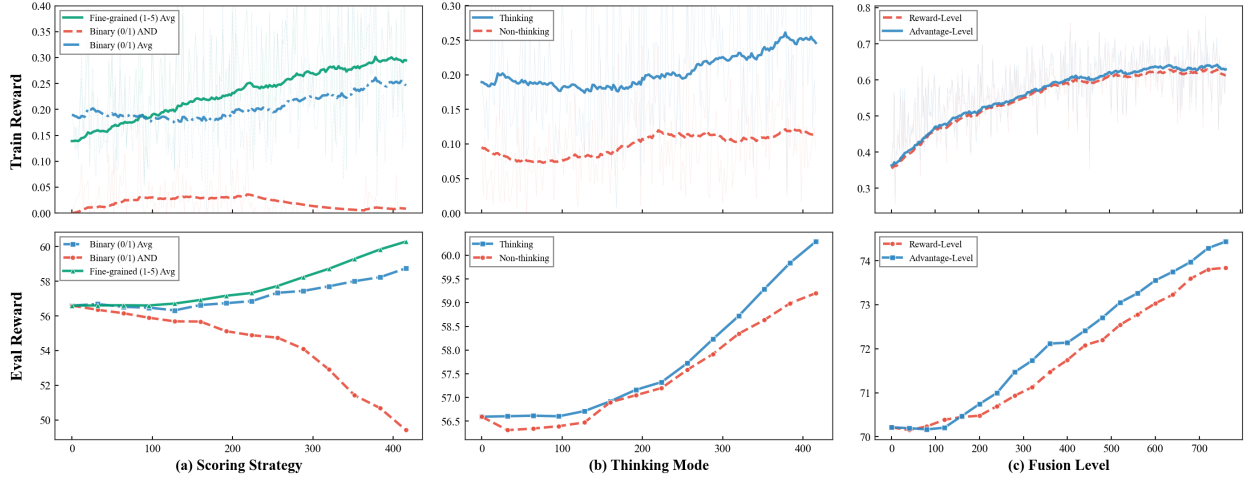


Figure 3.5: **Ablation on Reward Signal Design.** The top row shows training reward, and the bottom row shows held-out evaluation reward. Fine-grained 1–5 scoring provides a more effective learning signal than binary variants, thinking mode produces better performance with more accurate but higher-variance rewards, and advantage-level fusion consistently outperforms reward-level fusion.

increases significantly even with only one training turn ($K = 1$), indicating that single-turn training alone can be effective for multi-turn tasks. However, to achieve optimal multi-turn performance, additional RL training turns are required.

4. **Reduced Exposure Bias:** Fig. 3.4 shows that MT-EditFlow maintains a much flatter marginal task success across turns compared to open-source baselines, approaching the trajectory-level reliability of strong closed-source models. This indicates that MT-EditFlow improves performance across the entire editing trajectory rather than merely boosting early-turn results.

3.5.3 Analysis on Reward Signal Design

We study reward signal design along three axes: scoring granularity, evaluator prompting, and fusion level for joint IF+CC optimization.

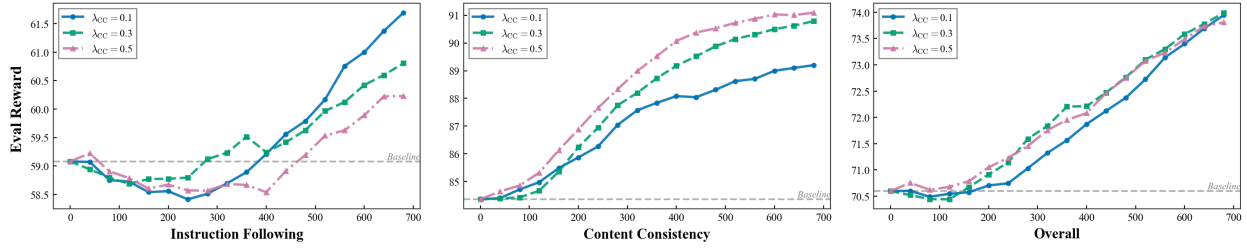


Figure 3.6: **Ablation on λ_{CC} .** Higher λ_{CC} improves CC at the cost of IF; $\lambda_{CC} = 0.3$ achieves the best overall balance.

3.5.3.1 Scoring strategy: granularity and sparsity

We compare three turn-wise aggregation methods for $K = 3$ turns: **Binary Avg** (per-turn 0/1 mean), **Binary AND** (1 only if all turns succeed), and **Fine-grained Avg** (per-turn 1–5 scores). Binary AND is extremely sparse: a trajectory is positive with probability p^K , yielding an ineffective learning signal. Fine-grained scoring provides partial credit, reduces sparsity, and stabilizes updates. Fig. 3.5(a) shows it yields the strongest and most stable improvement; binary averaging is weaker, and AND degrades due to sparse rewards.

3.5.3.2 Thinking vs. non-thinking mode

We compare a **thinking** evaluator (structured per-turn analysis) with a **non-thinking** evaluator (direct score output). Fig. 3.5(b) shows thinking mode improves the performance. All main experiments use fine-grained 1–5 scoring with thinking prompting.

3.5.3.3 Fusion level: reward vs. advantage

Fig. 3.5(c) shows advantage-level fusion outperforms reward-level fusion, with the gap widening during training. Standardizing advantages prevents IF or CC from dominating due to scale differences.

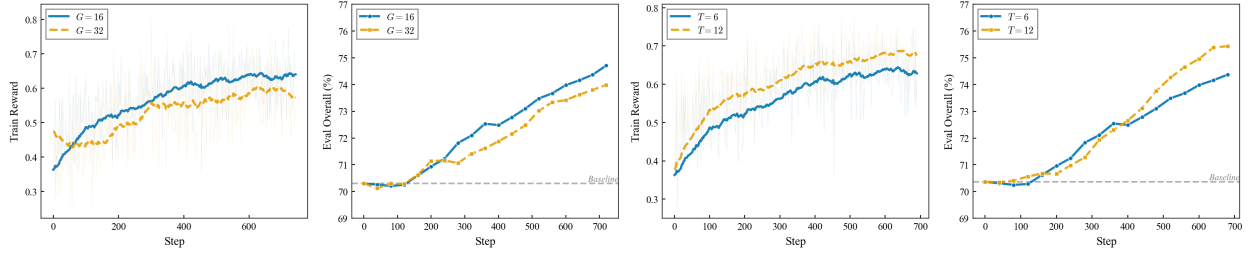


Figure 3.7: **Ablation on group size G and discretization steps T .** We found that a smaller group size $G = 16$ and a larger discretization step $T = 12$ lead to better performance.

3.5.4 Ablation Studies

3.5.4.1 Ablations on different RL algorithms

Tab. 3.3 shows that NFT outperforms GRPO on FLUX.1. In our implementation, Flow-GRPO fails on FLUX.2 because its likelihood estimation becomes inaccurate when the CFG guidance scale is embedded in the Diffusion Transformer. In contrast, NFT is more robust to different guidance designs with flow matching, making it the preferred choice.

3.5.4.2 Impact of Content Consistency Weight λ_{CC}

Fig. 3.6 visualizes this trade-off on FLUX.2 with DiffusionNFT: increasing λ_{CC} improves CC in both training and held-out evaluation, but suppresses IF, and the overall curve favors moderate weights. In sum, λ_{CC} provides an effective way to prevent reward hacking.

3.5.4.3 Group Size G and Discretization Steps T

For MT-EditFlow-NFT, Fig. 3.7 shows that the best performance is achieved with a moderate group size $G = 16$ and finer discretization $T = 12$.

3.6 Limitations

First, computational overhead is a concern, as using high-capacity VLMs for zero-shot rewards increases RL training latency compared to smaller reward models. Second, evaluator bias persists despite chain-of-thought prompting, as VLM hallucinations can introduce noise into the RL signal. Third, actor-critic RL methods could be explored for multi-turn editing, but they would likely incur additional computational cost due to the need for training a critic model, making them more expensive than trajectory-level advantage broadcasting. Finally, trajectory scaling remains an open question, as current evaluations focus on three-turn sequences and performance on longer interaction chains remains unexplored.

3.7 Conclusion

In this paper, we presented **MT-EditFlow**, a reinforcement learning framework specifically designed to address the challenges of multi-turn image editing within the flow-matching paradigm. By shifting from a single-turn, single-reward perspective to a multi-turn, multi-reward formulation, our approach effectively mitigates the all-or-nothing requirement and the exposure bias that typically causes sequential edits to fail. Through a systematic analysis of reward signal design, MT-EditFlow, applicable to both GRPO and NFT-based methods, achieves superior performance in multi-turn editing. By enabling robust, iterative refinement, MT-EditFlow provides a foundation for more reliable and natural human-AI collaboration in visual content creation.

Part II

Alignment

CHAPTER 4

REAL: Regression-Aware Reinforcement Learning for LLM-as-a-Judge

4.1 Introduction

Large Language Models (LLMs) [BMR20, AAA23b] have evolved beyond mere content generators to become sophisticated evaluators, a paradigm known as LLM-as-a-Judge [GJS24, CL23]. This role is now central to diverse applications, from assessing text quality [OWJ22] and instruction following [ZWW25] to safety alignment [ZZC24] and preference modeling [KSC23]. In these contexts, the model must produce a numeric score that accurately reflects quality, correctness, or preference intensity.

Despite the regression nature of these tasks, standard practices—exemplified by Prometheus 1 and 2 [KSC23, KSL24]—rely on traditional SFT via cross-entropy loss. By treating numeric scores as categorical tokens, these methods ignore the inherent ordinal structure of the data. Recent advancements, such as Regression-Aware Fine-Tuning (RAFT) [LMN25, LNM24], have begun to bridge this gap by optimizing models using regression losses over expected numerical predictions. Extensions like TRACT [CLL25] further incorporate Chain-of-Thought (CoT) supervision to improve reasoning faithfulness. However, these approaches remain confined to the SFT regime: they depend on static ground-truth trajectories and lack a principled mechanism for exploring model-generated reasoning pathways.

Transitioning from regression-aware SFT to Reinforcement Learning (RL) is, therefore, a natural progression to enable exploration over CoT pathways. This shift allows the model

to search for optimal reasoning trajectories guided by downstream numerical objectives. However, standard RL post-training frameworks [SWD17, GYZ25, ACG24] typically rely on rule-based verifiers to produce binary rewards [GYZ25] (e.g., 0–1 accuracy), creating a mismatch between the LLM-as-a-Judge task’s regression nature and categorical RL optimization. For instance, standard RL fails to recognize that predicting a 4 is significantly better than predicting a 1 when the ground truth is 5. Empirically, we also find that standard RL leads to suboptimal learning behavior, specifically a collapse in correlation metrics (see Fig. 4.2 and Tab. 4.2).

To address this challenge, we introduce Regression-Aware Reinforcement Learning (REAL), a principled framework designed specifically for LLM-as-a-Judge and pointwise evaluation tasks, as presented in Fig. 4.1. However, integrating regression-aware objectives into RL introduces a significant theoretical challenge: the resulting reward functions depend explicitly on the policy parameters themselves. To resolve the resulting policy-dependency reward issue, we employ the generalized policy gradient [SHW15] that explicitly accounts for parameter-dependent rewards. This derivation decomposes the learning process into two complementary components: (1) A CoT policy-gradient term that encourages exploration over reasoning trajectories based on downstream regression-aware rewards; and (2) A prediction refinement term that provides regression-aware supervision to the final numerical prediction via standard backpropagation. This decomposition clarifies how reasoning exploration and numerical accuracy can be jointly optimized within a unified framework.

Theoretically, Sec. 4.3 proves that minimizing the regression loss leads to optimal optimization in terms of correlation metrics. Empirically, REAL consistently outperforms prior regression-aware SFT methods [LMN25, CLL25] across multiple benchmarks (See Fig. 4.3). Notably, REAL exhibits superior generalization to out-of-domain tasks, highlighting the importance of modeling regression structures directly within RL. While concurrent works like J1 [WWY25] also apply RL to the LLM-as-a-Judge setting, they fail to exploit the ordinal nature of numeric scoring. To the best of our knowledge, we are the first to successfully inte-

grate a regression-aware objective directly into the RL training pipeline for LLM evaluators.

In summary, our contributions are threefold:

1. We propose REAL, a principled regression-aware RL framework that bridges the gap between standard RL and regression-based tasks by introducing a policy-dependent regression loss, as presented in Fig. 4.1. This formulation explicitly models the ordinal structure of evaluation tasks, addressing the inherent limitations of binary rewards (See Fig. 4.2). Furthermore, we provide a theoretical proof demonstrating that minimizing this regression loss also optimizes correlation metrics, as presented in Sec. 4.3.
2. We employ a generalized policy gradient estimator that enables mathematically sound optimization of policy-dependent objectives. This derivation reveals a functional decomposition that decouples the exploration of reasoning trajectories (CoT) from the optimization of final numerical predictions.
3. Experiments across model scales (8B to 32B) demonstrate that REAL consistently outperforms both regression-aware SFT baselines and standard RL methods. REAL exhibits superior generalization on out-of-domain benchmarks; specifically, on Qwen3-32B, we achieve gains of +8.40 Pearson and +7.20 Spearman correlation over the SFT baseline, and +18.30/+11.20 over the base model, as shown in Fig. 4.3.

4.2 Background

4.2.1 LLMs and Reinforcement Learning

LLMs are typically trained in phases, starting from pre-training [KMH20, BMR20], following with multiple stages of post-training [OWJ22], most notably an alignment phase where an LLM is trained to improve its reasoning capabilities via Reinforcement Learning (RL) [GYZ25]. Let x denote the input prompt, c the chain-of-thought (CoT), and y the final

answer. We define $\mathcal{D}$ as the training set, which, depending on the context, may consist of prompts $\{x\}$, prompt-target pairs $\{(x, y^*)\}$, or triplets $\{(x, c^*, y^*)\}$. Here, the superscript $*$ denotes ground truth, as opposed to a predicted value from the model. Furthermore, let $\pi_\theta(\cdot | x)$ represent the distribution over token sequences, typically the CoT reasoning c followed by the final answer y , as generated by a language model with learnable parameters θ . Fundamentally, the objective of reinforcement learning [SB99] is to maximize a reward:

$$\mathcal{L}(\theta) = \mathbb{E}_{x \sim \mathcal{D}, (c, y) \sim \pi_\theta(\cdot | x)} [r(x, y)], \quad (4.1)$$

where the reward function r is commonly an accuracy-based reward in the form of $\mathbf{1}(y = y^*)$ if the ground-truth label y^* is available in the training dataset.

Since the reward $r(x, y)$ in LLM applications is typically only available after the entire sequence is generated, it is common to treat the full completion as a single action, resulting in a bandit-style formulation. This enables the use of the REINFORCE gradient estimator [Wil92, NDB17]. Under this formulation, the policy gradient can be written as:

$$\nabla_\theta \mathcal{L}(\theta) = \mathbb{E}_{x \sim \mathcal{D}, (c, y) \sim \pi_\theta(\cdot | x)} [r(x, y) \nabla_\theta \log \pi_\theta(y, c | x)]. \quad (4.2)$$

In this formulation, the reward $r(x, y)$ serves as a scalar weighting factor on the policy gradient, amplifying updates for high-reward generations while attenuating those associated with low rewards. We note that the derivation of the gradient relies on the assumption that the reward does not explicitly depend on the policy parameters, i.e., $\nabla_\theta r(x, y) = 0$.

4.2.2 Regression-Aware Inference and Finetuning in LLMs

Standard supervised fine-tuning (SFT) for LLMs typically relies on cross-entropy loss [BMR20]. This objective is suboptimal for regression tasks as it treats all incorrect numerical tokens equally, failing to account for their metric distance from the target value. To address this, several methods have been proposed to align LLM outputs with the inherent structure of numerical targets.

Regression-Aware Fine-Tuning (RAFT) [LMN25] replaces token-level classification with a squared error loss. This loss is computed between the ground-truth score and an expected-value predictor, which applies a Bayes-optimal decision rule to the model’s output distribution. Specifically, given an input x and ground-truth reasoning c^* and label y^* from a dataset $\mathcal{D}$, the RAFT objective to *minimize* is defined as:

$$\mathcal{L}_{\text{RAFT}}(\theta) = \mathbb{E}_{(x, c^*, y^*) \sim \mathcal{D}} \left[\underbrace{(\hat{y}_\theta(x, c^*) - y^*)^2}_{\text{regression loss}} - \underbrace{\lambda \log \pi_\theta(c^*, y^* | x)}_{\text{next token prediction loss}} \right], \quad (4.3)$$

where $\hat{y}_\theta(x, c) = \sum_{k \in \mathcal{K}} k \cdot \pi_\theta(k | x, c)$ denotes the expected value of the numerical prediction, originally proposed as the **RAIL predictor** (see Alg. 10) for regression-aware inference [LNM24]. Here, $\mathcal{K} = \{0, 1, \dots, 9\}$ represents the set of single-digit numerical tokens, and λ is a hyperparameter balancing the regression objective with the standard next-token prediction (NTP) loss. The RAIL predictor has been shown to offer “free-lunch” improvements in Pearson and Spearman correlation [LMN25, CLL25], a finding we corroborate in Tab. 4.2.

Algorithm 10 RAIL predictor [LNM24]

Require: Input prompt x , LLM π_θ

- 1: Generate chain of thought $c \sim \pi_\theta(\cdot | x)$
 - 2: *# Compute expected value over digit token set $\mathcal{K}$*
 - 3: $\hat{y}_\theta \leftarrow \sum_{k \in \mathcal{K}} k \cdot \pi_\theta(k | x, c)$
 - 4: **return** RAIL predictor $\hat{y}_\theta$
-

Building on this foundation, **TRACT** [CLL25] extends the RAFT framework by incorporating self-generated reasoning paths. TRACT utilizes a two-stage SFT pipeline: initially, a temporary model π_{temp} is trained via the RAFT objective (Eq. 4.3). Subsequently, a refined dataset is synthesized by pairing the original inputs x with reasoning trajectories c sampled from the temporary policy, $c \sim \pi_{\text{temp}}(\cdot | x)$, while retaining the ground-truth labels y^* . The final model is then trained on this augmented triplet dataset $\{(x, c, y^*)\}$ using the same

RAFT objective. While this self-generation strategy significantly improves performance, it remains fundamentally a static SFT procedure. It lacks a principled mechanism for continuous, active exploration of the reasoning space—a limitation we overcome by transitioning to a reinforcement learning framework.

4.3 Optimal Decision Rule for LLM-as-a-Judge Evaluation Metrics

Standard approaches for LLM post-training often frame reinforcement learning using a binary accuracy reward (i.e., the indicator function $\mathbf{1}\{y = y^*\}$), strictly optimizing for prediction correctness while performing inference via standard stochastic decoding. We argue that this paradigm is theoretically suboptimal for benchmarks primarily evaluated on correlation metrics, such as LLM-as-a-Judge [GJS24, CL23, ZCS23a, LIX23, ZZC24, OWJ22, ZWW25]. Prior works on regression-aware LLMs showed that other decision rules are optimal for regression loss [LMN25, CLL25], but an optimal rule for correlation has not been discussed in these works, despite evaluating on correlation.

A fundamental challenge in optimizing for correlation is that it is inherently a population statistic defined over a set of predictions, rather than a single data point. Consequently, correlation cannot be directly utilized as a standard per-sample reward signal in reinforcement learning. We formally demonstrate that this limitation can be bypassed by identifying a shared optimal estimator: the squared error loss, a tractable per-sample objective, serves as the natural proxy for maximizing Pearson correlation.

Lemma 1 (Optimality of Squared Error for Pearson Correlation). *Consider the following distributions: the distribution over input prompts $x \sim P_{\mathcal{D}}(\cdot)$, the distribution over chains of thought (CoTs) $c \sim \pi_{\theta}(\cdot | x)$ generated by the LLM policy π_{θ} , and the distribution over targets conditioned on the inputs, $y^* \sim P(\cdot | x)$. In this case, we assume the conditional independence $c \perp y^* | x$, yielding the joint distribution: $P_{\theta}(x, c, y^*) = P_{\mathcal{D}}(x)\pi_{\theta}(c | x)P(y^* |$*

x). All expectations $\mathbb{E}[\cdot]$ below are taken with respect to $P_\theta(x, c, y^*)$. Define the posterior mean $\mu(x, c) \triangleq \mathbb{E}[y^* \mid x, c]$ and the squared-error risk $\mathcal{R}(\hat{y}) \triangleq \mathbb{E}\left[(\hat{y}(x, c) - y^*)^2\right]$. Assume $\text{Var}(y^*) > 0$ and $\text{Var}(\mu(x, c)) > 0$. The risk $\mathcal{R}(\hat{y})$ is minimized (a.s.) by $\hat{y}^*(x, c) = \mu(x, c)$, and the Pearson correlation $\rho(\hat{y}(x, c), y^*)$ is maximized by any positive affine transform of μ : $\hat{y}(x, c) = a\mu(x, c) + b$, $a > 0$, $b \in \mathbb{R}$. In particular, the squared-error minimizer $\hat{y}^*(x, c) = \mathbb{E}[y^* \mid x, c]$ is also optimal for maximizing Pearson correlation. In other words, by minimizing the squared error, we implicitly train the predictor to be the optimal estimator for the Pearson correlation metric.

The proof, provided in Appendix 4.8.2, establishes that the Pearson correlation is maximized by any positive linear transformation of the conditional expectation (Lemma 3), and that the squared error loss is uniquely minimized by that same conditional expectation (Lemma 4). By minimizing the squared loss, we implicitly train the predictor to be the optimal estimator for the Pearson correlation metric. This result aligns with empirical findings in RAFT [LMN25] and TRACT [CLL25], which demonstrate that regression-aware objectives are effective for both squared error and correlation-based evaluation metrics.

4.4 REAL: Regression-Aware Reinforcement Learning

In this section, we introduce Regression-Aware Reinforcement Learning (REAL), a framework that leverages regression-based rewards to optimize LLM evaluators. As theoretically demonstrated in Sec. 4.3, this formulation is optimal for maximizing correlation metrics.

We first note that REAL extends the principles of regression-aware supervised fine-tuning—specifically RAFT [LMN25] and TRACT [CLL25]—into the reinforcement learning regime. By transitioning to RL, the model is no longer confined to static, ground-truth reasoning but is instead able to explore and refine its own self-generated reasoning trajectories. A key distinction in our framework is that the resulting regression reward is policy-dependent, which necessitates the use of a generalized policy gradient to ensure valid optimization. This

estimator naturally decomposes the training process into two distinct components: the exploration of reasoning pathways and the refinement of numerical predictions.

4.4.1 REAL Objective

We define the REAL objective to *minimize* $\mathcal{L}_{\text{REAL}}(\theta)$ as:

$$\begin{aligned} \mathcal{L}_{\text{REAL}}(\theta) = & \mathbb{E}_{(x, y^*) \sim \mathcal{D}, c \sim \pi_{\theta}(\cdot|x)} \\ & [(\hat{y}_{\theta}(x, c) - y^*)^2 - \lambda \log \pi_{\theta}(y^*|x, c)], \end{aligned} \quad (4.4)$$

In other words, we are augmenting the regression loss with an auxiliary log-likelihood term on the final answer tokens. When setting $\lambda = 0$ we recover the exact regression loss, which is of key importance for endowing the model with better numerical prediction abilities. Augmenting with the log-likelihood term is also in line with the SFT objective design in TRACT [CLL25].

Consequently, the implicit reward function in our framework is defined as:

$$r_{\text{REAL}}(\theta, x, c) = -(\hat{y}_{\theta}(x, c) - y^*)^2 + \lambda \log \pi_{\theta}(y^*|c, x). \quad (4.5)$$

Crucially, unlike standard RL post training where the reward is provided by a fixed external model, our reward function r_{REAL} **explicitly depends on the policy parameters** θ (via the estimator $\hat{y}_{\theta}$ and probability π_{θ}).

4.4.2 Generalized Policy Gradient with Policy-Dependent Reward Functions

In standard LLM post-training methodologies, the reward function is typically modeled as an external signal derived from the environment or a fixed preference model, independent of the current policy π_{θ} (i.e., $\nabla_{\theta} r = 0$). However, our framework introduces a reward function $r_{\text{REAL}}(\theta, x, c)$ (Eq.4.5) that explicitly depends on the policy parameters θ . Consequently, the standard policy gradient formulation is insufficient. In this section, we formalize the generalized policy gradient [SHW15] for this class of policy-dependent reward functions.

Lemma 2 (Generalized Policy Gradient with Policy-Dependent Rewards for Regression).

Let the objective be $\mathcal{L}(\theta) = \mathbb{E}_{x \sim \mathcal{D}} \mathbb{E}_{c \sim \pi_\theta(\cdot|x)} [r(\theta, x, c)]$. The gradient $\nabla_\theta \mathcal{L}(\theta)$ is given by:

$$\nabla_\theta \mathcal{L}(\theta) = \mathbb{E}_{(x, y^*) \sim \mathcal{D}, c \sim \pi_\theta(\cdot|x)} \left[\underbrace{r(\theta, x, c) \nabla_\theta \log \pi_\theta(c | x)}_{\text{Term 1: CoT Update}} + \underbrace{\nabla_\theta r(\theta, x, c)}_{\text{Term 2: Prediction Update}} \right]. \quad (4.6)$$

The first term optimizes the CoT generation, weighted by the reward; the second term provides supervision for final answer prediction through standard backpropagation. Specifically, by substituting the REAL reward function (Eq. 4.5) into the general gradient expression (Eq. 4.6), we obtain the gradient for the REAL objective (Eq. 4.4). Specifically, Term 2 can be expanded as:

$$\begin{aligned} \text{Term 2} &= -2(\hat{y}_\theta(x, c) - y^*) \nabla_\theta \hat{y}_\theta(x, c) \\ &\quad + \lambda \nabla_\theta \log \pi_\theta(y^* | x, c) \end{aligned} \quad (4.7)$$

where the gradient of the predicted value is $\nabla_\theta \hat{y}_\theta(x, c) = \sum_{k \in \mathcal{K}} k \cdot \nabla_\theta \pi_\theta(k | x, c)$.

Distinguishing Reasoning Exploration from Prediction Refinement The REAL gradient (i.e., Eq. 4.6) explicitly decomposes optimization into two distinct parts: exploring reasoning trajectories (Term 1) and refining score prediction (Term 2). This separation is critical because standard RL approaches like GRPO [SWZ24] treat the reasoning chain c and final answer y as a homogeneous sequence, applying a uniform update rule. In contrast, our framework addresses their structural differences: the CoT update allows exploration via policy gradients, whereas the final score targets a known ground truth, allowing for direct correction via standard backpropagation.

4.4.3 Stabilization and Implementation

To enhance the stability of the REAL objective during training, we utilize the RLOO estimator [KHW19] for the first term, which governs reasoning exploration. Other stabilization

approaches [SWD17, GYZ25] can be explored in future work. We note that this is not central to the core contribution of our paper, i.e., the regression-aware RL objective and algorithm.

For a given input (x, y^*) , we sample K independent CoT trajectories $\{c_1, c_2, \dots, c_K\}$ from the current policy $\pi_\theta(\cdot|x)$. Let $r^{(i)} = r_{\text{REAL}}(\theta, x, c_i)$ denote the reward associated with the i -th trajectory. The advantage $A^{(i)}$ is computed as: $A^{(i)} = r^{(i)} - b^{(i)}$; $b^{(i)} = \frac{1}{K-1} \sum_{j \neq i} r^{(j)}$. We further compute the standardized advantage $\tilde{A}^{(i)} = \text{clip}\left(\frac{A^{(i)}}{\sigma(A) + \epsilon}, -1, 1\right)$ for better training stability, where $\sigma(A)$ denotes the standard deviation of the advantages within the sampled group and $\epsilon = 10^{-8}$. Then from Eq. 4.6, replacing reward by RLOO advantages in Term 1 and add β on Term 2 controls the strength of the prediction updates, we have the stabilized gradient estimator defined as

$$\nabla_\theta \mathcal{L}_{\text{REAL}}^{\text{RLOO}}(\theta) \approx \frac{1}{K} \sum_{i=1}^K \left[\underbrace{\tilde{A}_{\text{REAL}}^{(i)} \nabla_\theta \log \pi_\theta(c_i|x)}_{\text{CoT Update}} + \beta \underbrace{\nabla_\theta r_{\text{REAL}}(\theta, x, c_i)}_{\text{Prediction Update}} \right] \quad (4.8)$$

Following [TWM25] we introduce a prediction weight β to balance Term 1 and Term 2. Note that Eq. 4.6 implies an exact weighting of $\beta = 1.0$, which we find to already bring good results, as shown in Tab. 4.9.

4.4.4 Relations with Prior Methods

We provide a complete conceptual comparison with prior methods in Tab. 4.1.

Comparison to TRACT: Active vs. Static Reasoning We identify both a conceptual connection and a critical limitation in TRACT [CLL25]. While TRACT introduces reasoning traces to the regression setting, it remains confined to the SFT paradigm. Specifically, TRACT treats all self-generated Chain-of-Thought (CoT) trajectories as ground truth, failing to evaluate their intermediate quality. In contrast, REAL leverages a regression-aware reward to actively explore and rank CoT trajectories during RL. Mathematically, TRACT’s

Table 4.1: Conceptual comparison of selected SFT and RL training methods on non-ordinal and ordinal targets. $r(\theta, x, c) = -(\hat{y}_\theta(x, c) - y^*)^2 + \lambda \log \pi_\theta(y^*|x, c)$ represents the regression-aware reward. REAL is the first framework to combine a regression-aware reward with reinforcement learning for reasoning exploration. Note that JEPO further approximates the gradient with the multi-sample Jensen’s Bound.

Method	Paradigm	CoT Exploration	Ordinal Awareness	Gradient
SFT	SFT	✗	✗	$\mathbb{E}_{(x, y^*) \sim \mathcal{D}} [\nabla_\theta \log \pi_\theta(y^* x)]$
RAFT/TRACT	SFT	✗	✓	$\mathbb{E}_{(x, c, y^*) \sim \mathcal{D}} [\nabla_\theta r(\theta, x, c)]$
Standard RL	RL	✓	✗	$\mathbb{E}_{x \sim \mathcal{D}, (c, y) \sim \pi_\theta(\cdot x)} [r(x, y) \nabla_\theta \log \pi_\theta(y, c x)]$
JEPO [TWM25]	RL	✓	✗	$\mathbb{E}_{x \sim \mathcal{D}, c \sim \pi_\theta(\cdot x)} [\log \pi_\theta(y^* x, c) \nabla_\theta \log \pi_\theta(c x) + \nabla_\theta \log \pi_\theta(y^* x, c)]$
REAL (ours)	RL	✓	✓	$\mathbb{E}_{(x, y^*) \sim \mathcal{D}, c \sim \pi_\theta(\cdot x)} [r(\theta, x, c) \nabla_\theta \log \pi_\theta(c x) + \nabla_\theta r(\theta, x, c)]$

update is equivalent to the *prediction refinement* term (Term 2) of our objective, but it lacks the *trajectory exploration* component (Term 1) necessary for true reinforcement learning, as can be seen when comparing the respective rows in Tab. 4.1. Thus, TRACT can be viewed as an offline, supervised subset of the more general REAL framework.

Regression-Aware vs. Standard RL Standard RL typically relies on coarse binary rewards (e.g., 0/1 accuracy), which ignores the ordinal structure inherent in evaluation tasks. As illustrated in Fig. 4.2, REAL utilizes the policy’s full probability distribution to shape a fine-grained, regression-aware reward. This allows the model to distinguish ”near-miss” predictions from total failures, leading to significantly higher alignment with human judgment as demonstrated by the correlation metrics in Tab. 4.2.

Comparison to JEPO: JEPO [TWM25] optimizes the marginal log-likelihood $\log p(y^*|x)$ using Jensen’s bound to handle unverifiable data. While effective for tasks like mathematical proofs, JEPO lacks the ordinal awareness required for numeric scoring in LLM-as-a-Judge settings. Our REAL objective equation 4.4 is explicitly regression-aware, providing a more

principled approach for tasks where the distance between scores matters. We contrast JEPO and our objectives in Tab. 4.1. Since JEPO does not release an official codebase, we reproduced their results in our setting. The empirical comparisons in Tab. 4.12 in the Appendix confirm that REAL consistently outperforms JEPO across all regression metrics.

4.5 Experiments

We mainly follow the experimental setup in [CLL25, KSC23, KSL24] for evaluation, and implement our RL algorithm using the codebase of verl [SZY24].

Models We implement REAL on three LLMs with different architectures, *tokenizers*, and sizes to demonstrate robustness across backbones: 1) **Mistral2-7B** [JSM23], for a fair comparison with TRACT and Prometheus-2, which both use it as their base model; 2) **Qwen3-8B** [YLY25], which uses a different tokenizer than Mistral and has recently shown strong reasoning and instruction-following abilities; and 3) **Qwen3-32B** [YLY25], to which we scale up performance to demonstrate the scalability of REAL.

Datasets We implement training on **Feedback Collection** [KSC23], which consists of approximately 100K pointwise samples together with 1K fine-grained score rubrics, 20K instructions. Following [CLL25, KSC23, KSL24], we evaluate pointwise LLM-as-a-Judge performance on four widely used benchmarks: 1) **Feedback Bench**, the official test set for Feedback Collection, includes 1K *non-overlapping* score rubrics and 200 instructions; 2) **FLASK**, a fine-grained evaluation benchmark, consists of 200 prompts, 12 score rubrics, and 2K responses generated by Alpaca-7B [TGZ23], Vicuna-13B [CLL23], Bard [Goo23], and GPT-3.5-Turbo-0613 [Ope23]; 3) **Vicuna Bench**, a single-turn dialogue dataset, includes 80 test prompts, 80 hand-crafted score rubrics, and 320 responses generated by WizardLM-13B [XSZ23], Vicuna-13B [CLL23], Llama-2Chat-13B [TMS23], and GPT-3.5-Turbo-0613 [Ope23]; and 4) **MT Bench**, a multi-turn chat benchmark, contains 80 test prompts, 80

hand-crafted score rubrics, and 320 responses generated by the same models as above. Together, these datasets cover a diverse range of prompts, score rubrics, response styles, and difficulty levels. In addition, we note that Feedback Bench can be regarded as more of an *in-domain* test dataset, while the other three are *out-of-distribution*.

Evaluation Metrics We report the Pearson correlation coefficient r [Pea95], Spearman’s rank correlation ρ [Spe61], and Kendall’s τ [Ken38] with ground-truth scores. Additionally, we provide RMSE and MAE metrics in the ablation study for reference.

Baselines We compare REAL against zero-shot, SFT, and RL baselines:

- (i) **Zero-shot** inference using the base model;
- (ii) **RAFT** [LMN25], which applies regression-aware supervised fine-tuning with Eq. 4.3;
- (iii) **TRACT** [CLL25], which performs two-stage regression-aware supervised fine-tuning. Note that TRACT simply treats all self-generated CoTs as ground truth for the second stage, without any CoT update. In contrast, our method conducts CoT policy gradient updates weighted by the REAL reward;
- (iv) **Standard RL (accuracy reward)** [KHW19], a baseline that uses accuracy as the reward, where $r_{\text{acc}} = \mathbf{1}(y = y^*)$, and updates via $A_{\text{acc}} \nabla_{\theta} \pi(c, y \mid x)$, where A_{acc} is the RLOO advantage for r_{acc} ;
- (v) **Other baseline reward models:** 1) zero-shot GPT3.5-Turbo [Ope23]; 2) Prometheus-1-7B [KSC23]; 3) Prometheus-1-13B [KSC23]; and 4) Prometheus-2-7B [KSL24].

Following the findings in [CLL25, LNM24], we mainly report RAIL inference results, as RAIL has been shown to offer *free-lunch* improvements under Pearson and Spearman correlation metrics at inference.

Configuration We conduct experiments using 8 NVIDIA A100 GPUs for Mistral2-7B and Qwen3-8B, and 2×8 A100 GPUs for Qwen3-32B. All experiments are performed using full-parameter fine-tuning. The learning rate is set to 5×10^{-8} for the Mistral models and 1×10^{-6} for the Qwen series. The *max_response_length* is set to 1024. Unless otherwise specified, other hyperparameters are shared across all three evaluated LLMs. Ablation studies are carried out to analyze the impact of different hyperparameter settings on model performance.

4.5.1 Main Results

See Tab. 4.2 for all the results. Across all benchmarks and model backbones, REAL consistently improves correlation with ground-truth scores compared to the zero-shot inference, regression-aware supervised fine-tuning, and the standard RL methods. We note that REAL performs significantly better on *out-of-domain* benchmarks such as Vicuna, MT, and Flask. Qualitative examples are provided in the supplementary material of the source paper.

Average-of-N results at inference We average the RAIL prediction $\hat{y}_\theta$ over N generations for each test prompt as the final output and then calculate the metrics shown in Tab. 4.3. Scaling N to 4 or 10 slightly improves performance, which indicates that REAL already achieves strong results at $N = 1$. This demonstrates that the model learns a robust policy capable of efficiently identifying high-reward outputs. This single-generation capability ensures peak sampling efficiency, making REAL highly suitable for low-latency, real-world deployment by eliminating the computational overhead of multi-sample averaging.

4.5.2 On Regression-Aware Policy-Independent Rewards

A natural question is whether the gains of REAL mainly come from the regression-aware reward itself, rather than from our generalized policy gradient formulation. To investigate this, we additionally evaluate a policy-independent regression reward under standard RL training. Specifically, Standard RL above uses an accuracy reward $r_{\text{acc}} = \mathbf{1}(y = y^*)$.

We further construct a regression-aware but policy-independent reward using squared error, $r_{\text{reg}} = -(y - y^*)^2$ where $y = \arg \max_k \pi_\theta(k|x, c)$, and train it with standard RL, denoted as **Standard RL(reg)**. Tab. 4.4 shows that Standard RL(reg) behaves very similarly to Standard RL and quickly collapses during RL training, whereas REAL consistently improves performance.

Intuitively, the key issue is that standard RL methods fail to evaluate the true quality of sampled trajectories. For example, suppose the ground-truth score is $y^* = 5$. Since our REAL model is initialized from a strong SFT checkpoint, many sampled trajectories already correctly decode to $y = 5$. Under standard RL methods such as RLOO, PPO, or GRPO, all such trajectories receive identical rewards and advantages, regardless of whether $\pi_\theta(k = 5|x, c) = 0.9$ or 0.8 . Even with the regression-aware reward r_{reg} , the advantage remains identical as long as the decoded prediction y is unchanged.

REAL circumvents this limitation by defining the reward on the predictive distribution, which explicitly depends on the policy parameters. Consequently, two trajectories that both decode to 5 may still yield different rewards (e.g., 4.99 vs. 4.70), providing a substantially richer learning signal. These results suggest that the gains of REAL do not simply come from replacing classification rewards with regression-style rewards. Instead, the key ingredient is combining a policy-dependent regression-aware reward with the generalized policy gradient required to optimize it.

4.5.3 Ablation Studies

Rows shaded in gray indicate the default setting used in our experiments.

Group size K : RLOO variance reduction results in superior performance Tab. 4.5 indicates a clear positive correlation between the number of sampled responses and the quality of the learned model. Increasing the group size from $K = 4$ to $K = 16$ yields consistent improvements across all correlation metrics, with the Pearson correlation r rising

from 63.7 to 67.9, indicating the effectiveness of RLOO variance reduction.

Log-likelihood weight λ : regression-aware reward alone excels in performance; additional log-likelihood term gives further benefits. Tab. 4.6 shows that using the regression-aware reward alone ($\lambda = 0$) already significantly outperforms the TRACT baseline. Moreover, introducing the log-likelihood objective ($\lambda = 1/5/10$) provides further enhancement. This suggests that the log-likelihood component serves as a beneficial complement to the core regression objective. $\lambda = 1$ is chosen by default.

Dynamic Sampling Strategy: Partially Correct Prompts Provide the Most Informative Learning Signal. As shown in Tab. 4.7, we evaluate four training settings based on different dynamic sampling [YZZ25, XSF25] strategies using *group accuracy* as the primary metric.

Let $\{y^{(k)}\}_{k=1}^K$ denote a group of K generated responses for a given prompt x , and let y^* be the ground-truth label. We define the group accuracy $\text{acc}(x)$ as the proportion of correct responses within the group: $\text{acc}(x) = \frac{1}{K} \sum_{k=1}^K \mathbf{1}\{y^{(k)} = y^*\}$. Aligned with [YZZ25], our results demonstrate that training exclusively on the *partially correct* set ($0 < \text{acc} < 1$) consistently yields the best performance. This indicates that these prompts provide a rich learning signal while avoiding the gradient noise of entirely incorrect examples ($\text{acc} = 0$) and the vanishing supervision of already mastered ones ($\text{acc} = 1$).

Initialization checkpoints, Prediction weight β , and Temperature T . Tab. 4.8 demonstrates that REAL is robust to different starting configurations, yielding consistent improvements whether initialized from a base model or a high-quality SFT checkpoint. Tab. 4.9 indicates $\beta = 0.01$ brings the best performance regardless of initialization checkpoint types. $T = 1$ is chosen by default as supported by Tab. 4.10.

4.5.4 Generalizability to Other Settings

Extension to continuous-valued scores. REAL supports continuous-valued prediction during both training and inference. Specifically, REAL outputs a continuous regression prediction $\hat{y}_\theta$, and the ground-truth labels y^* can be arbitrary real values without needing to be integers. Our experiments include continuous-valued benchmarks such as FLASK, MT-Bench, and Vicuna, where the ground-truth targets are averages of multiple integer ratings from humans or GPT-4.

Extension to wider score ranges. For datasets with score ranges exceeding our default setting (e.g. $[1, 100]$), labels can simply be normalized into a smaller range during training, such as $[0, 9]$. This scales the targets to match the single-digit token space used by the expected-value predictor, which can then be mapped back through de-normalization at inference time.

Extension to pairwise difference modeling. REAL is applicable to pairwise scoring settings as well. For example, the HuggingFace `prometheus-eval/Preference_Collection` dataset provides scores for pairs of responses, denoting the preference of the first response over the second response. In case such scores span a wide range (e.g. negative values), they can be easily shifted into a non-negative range, after which REAL can be trained to predict the shifted difference score directly, as shown in Appendix 4.8.1.

4.6 Limitations

REAL relies on the quality and calibration of the underlying score supervision. If training labels are noisy, inconsistent, or systematically biased, the regression-aware objective may optimize toward imperfect numerical targets. In addition, REAL relies on self-generated Chain-of-Thought reasoning during evaluation, which may inherit biases or systematic errors

from the underlying base models.

4.7 Conclusion

We introduced REAL, a regression-aware reinforcement learning framework that employs the generalized policy gradient methods to solve regression-aware objectives. Our practical implementation combines RLOO stabilization with regression-aware prediction updates. Empirically, REAL achieves significant and consistent improvements across multiple LLM-as-a-Judge benchmarks over zero-shot and SFT baselines, specifically for out-of-domain evaluation. We believe this work provides a foundation for future research on RL algorithms for LLM-as-a-Judge.

4.8 Appendix

4.8.1 Extension to pairwise difference modeling

In this section, we demonstrate that REAL can be applied pairwise datasets. In particular, we conducted an additional experiment using Qwen3-1.7B and the HuggingFace `prometheus-eval/Preference_Collection`¹ dataset, which provides scores for Responses A and B, both in the range $[1, 5]$. We construct the pairwise annotation by taking a difference between the scores, and shifting the result by +5 to map the final scores to $[1, 9]$. We then prompt the LLM to predict this difference.

In Tab. 4.11, we evaluate the model on the in-domain Preference-Bench [KSL24] (using correlation metrics) and the out-of-domain HHH Alignment [ABC21]/ MT-Bench Human Judgement [ZCS23b] / Auto-J [LSY24] benchmarks (using accuracy with threshold 5, since they do not provide ground-truth score differences). REAL consistently outperforms the

¹<https://huggingface.co/datasets/prometheus-eval/Preference-Collection>

SFT baseline on out-of-domain tasks while maintaining strong in-domain performance.

4.8.2 Proofs of the theoretical results

First, to prove Lemma 1, we restate and utilize the following two results from statistical theory: 1) We show that this conditional expectation is the optimal estimator for maximizing Pearson correlation (Lemma 3); 2) We establish that minimizing the regression-aware loss (MSE) recovers the conditional expectation (Lemma 4).

4.8.2.1 Additional Lemmas

Lemma 3 (Optimality of the Posterior Mean for Pearson Correlation). ***Measure space and generative graph.** Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space on the tuple (x, c, y^*) with the following data-generating process:*

$$x \sim P_{\mathcal{D}} \quad \longrightarrow \quad \begin{cases} c \sim \pi_{\theta}(\cdot \mid x), \\ y^* \sim P(\cdot \mid x), \end{cases}$$

i.e., $c \perp y^ \mid x$.*

This induces the joint distribution

$$P_{\theta}(x, c, y^*) = P_{\mathcal{D}}(x) \pi_{\theta}(c \mid x) P(y^* \mid x), \quad (4.9)$$

and expectations $\mathbb{E}[\cdot]$ are taken w.r.t. $P_{\theta}(x, c, y^)$ unless stated otherwise.*

Conditional mean. *Define the conditional mean (posterior mean) of the label given (x, c) as*

$$\mu(x, c) \triangleq \mathbb{E}[y^* \mid x, c]. \quad (4.10)$$

Under the conditional independence $c \perp y^ \mid x$, we have*

$$P(y^* \mid x, c) = P(y^* \mid x) \quad \text{and thus} \quad \mu(x, c) = \mathbb{E}[y^* \mid x, c] = \mathbb{E}[y^* \mid x]. \quad (4.11)$$

Pearson correlation objective. Let $\hat{y}(x, c) \in \mathbb{R}$ be any measurable estimator with finite second moment. Assume Pearson correlation is well-defined and non-degenerate:

$$\text{Var}(y^*) > 0, \quad \text{Var}(\hat{y}(x, c)) > 0, \quad \text{Var}(\mu(x, c)) > 0. \quad (4.12)$$

Define Pearson correlation as

$$\rho(U, V) \triangleq \frac{\text{Cov}(U, V)}{\sqrt{\text{Var}(U) \text{Var}(V)}}. \quad (4.13)$$

Claim. The set of estimators that maximize the Pearson correlation $\rho(\hat{y}(x, c), y^*)$ (over all such $\hat{y}$) is exactly the set of positive affine transformations of $\mu(x, c)$:

$$\hat{y}^*(x, c) \in \arg \max_{\hat{y}} \rho(\hat{y}(x, c), y^*) \iff \hat{y}^*(x, c) = a \mu(x, c) + b, \quad \forall a > 0, b \in \mathbb{R}. \quad (4.14)$$

Remark (invariance and conditional-independence). Pearson correlation is invariant to positive scaling and shifting: $\rho(\hat{y}, y^*) = \rho(a\hat{y} + b, y^*)$ for any $a > 0$ and $b \in \mathbb{R}$, hence the maximizer is not unique. Moreover, since $c \perp y^* \mid x$ in equation 4.9, $\mu(x, c)$ reduces to $\mathbb{E}[y^* \mid x]$ as in equation 4.11; we retain the (x, c) notation to match the policy-generated reasoning variable used in post-training. This result follows from Cauchy–Schwarz arguments; see, e.g., [BKL22].

Lemma 4 (Optimality of Regression Objective). *The unique global minimizer of the squared error loss $\mathcal{R}(\theta) = \mathbb{E}[(\hat{y}_\theta(x, c) - y^*)^2]$ is the conditional expectation:*

$$\hat{y}_\theta(x, c) = \mathbb{E}[y^* \mid x, c]. \quad (4.15)$$

This is a standard result in Bayesian decision theory (see [BN06]).

4.8.2.2 Proof of Lemma 1

By Lemma 4, the estimator that minimizes the squared error loss is $\hat{y}^* = \mathbb{E}[y^* \mid x, c]$. By setting $a = 1$ and $b = 0$ in Lemma 3, it follows that this specific estimator is also a maximizer of the Pearson correlation. Thus, the regression objective is a sufficient condition for achieving the optimal correlation estimator. $\square$

4.8.2.3 Proof of Lemma 3

Setup and computation graph. Let (X, C, Y^*) be random variables on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with the following data-generating process:

$$X \sim P_{\mathcal{D}}, \quad C \sim \pi_{\theta}(\cdot | X), \quad Y^* \sim P(\cdot | X).$$

Equivalently, the conditional independence relation is

$$C \perp Y^* | X, \tag{4.16}$$

which corresponds to the computation graph $C \leftarrow X \rightarrow Y^*$. This induces the joint distribution

$$P_{\theta}(x, c, y^*) = P_{\mathcal{D}}(x) \pi_{\theta}(c | x) P(y^* | x). \tag{4.17}$$

All expectations $\mathbb{E}[\cdot]$ below are with respect to $P_{\theta}(x, c, y^*)$ unless stated otherwise.

Conditional mean. Define the conditional mean (posterior mean)

$$\mu(x, c) \triangleq \mathbb{E}[Y^* | X = x, C = c]. \tag{4.18}$$

Under equation 4.16, we have $P(y^* | x, c) = P(y^* | x)$ and hence

$$\mu(x, c) = \mathbb{E}[Y^* | X = x, C = c] = \mathbb{E}[Y^* | X = x]. \tag{4.19}$$

We keep the notation $\mu(x, c)$ because our estimator is explicitly a function of both (x, c) .

Remark (role of the reasoning variable c). In the supervised dataset, only (x, y^*) is observed; the reasoning chain c is *not* a supervised variable and is instead generated by the model/policy via $\pi_{\theta}(c | x)$ during training and inference. From the statistical model equation 4.16, conditioning on c does not reveal additional information about the label beyond x , so the Bayes-optimal conditional mean reduces as in equation 4.19. Nevertheless,

in our post-training procedure the predictor we optimize is of the form $\hat{y}_\theta(x, c)$, i.e., it is evaluated on realizations of (x, c) sampled from $P_{\mathcal{D}}(x)\pi_\theta(c \mid x)$. The lemma therefore characterizes the correlation-optimal form *within the space of functions of (x, c)* (and the correlation is taken over the induced joint distribution over (x, c, y^*)).

Goal. Let $\hat{y}(x, c)$ be any measurable real-valued estimator with finite second moment. Assume Pearson correlation is well-defined and non-degenerate:

$$\text{Var}(Y^*) > 0, \quad \text{Var}(\hat{y}(X, C)) > 0, \quad \text{Var}(\mu(X, C)) > 0. \quad (4.20)$$

We show that any maximizer of $\rho(\hat{y}(X, C), Y^*)$ must be of the form $\hat{y}^*(x, c) = a\mu(x, c) + b$ with $a > 0$.

Step 1: Covariance with Y^* reduces to covariance with $\mu(X, C)$. Recall the Pearson correlation definition

$$\rho(U, V) \triangleq \frac{\text{Cov}(U, V)}{\sqrt{\text{Var}(U)\text{Var}(V)}}. \quad (4.21)$$

We start from the covariance term:

$$\text{Cov}(\hat{y}(X, C), Y^*) = \mathbb{E}[\hat{y}(X, C)Y^*] - \mathbb{E}[\hat{y}(X, C)] \mathbb{E}[Y^*]. \quad (4.22)$$

Using the law of iterated expectations and conditioning on (X, C) ,

$$\mathbb{E}[\hat{y}(X, C)Y^*] = \mathbb{E}[\mathbb{E}[\hat{y}(X, C)Y^* \mid X, C]]. \quad (4.23)$$

At this point, it is important to note that $\hat{y}(X, C)$ is a (measurable) function of (X, C) , hence it is (X, C) -measurable and acts as a constant inside the conditional expectation given (X, C) . Therefore,

$$\mathbb{E}[\hat{y}(X, C)Y^* \mid X, C] = \hat{y}(X, C) \mathbb{E}[Y^* \mid X, C] = \hat{y}(X, C) \mu(X, C), \quad (4.24)$$

and substituting back into equation 4.23 yields

$$\mathbb{E}[\hat{y}(X, C)Y^*] = \mathbb{E}[\hat{y}(X, C)\mu(X, C)]. \quad (4.25)$$

Similarly,

$$\mathbb{E}[Y^*] = \mathbb{E}[\mathbb{E}[Y^* \mid X, C]] = \mathbb{E}[\mu(X, C)]. \quad (4.26)$$

Plugging equation 4.25–equation 4.26 into equation 4.22 gives

$$\begin{aligned} \text{Cov}(\hat{y}(X, C), Y^*) &= \mathbb{E}[\hat{y}(X, C)\mu(X, C)] - \mathbb{E}[\hat{y}(X, C)] \mathbb{E}[\mu(X, C)] \\ &= \text{Cov}(\hat{y}(X, C), \mu(X, C)). \end{aligned} \quad (4.27)$$

This identity holds irrespective of whether Y^* depends on C ; in our setting, $\mu(X, C)$ further reduces to $\mathbb{E}[Y^* \mid X]$ by equation 4.19.

Step 2: Maximizing correlation with Y^* is equivalent to maximizing correlation with $\mu(X, C)$. Using equation 4.27,

$$\begin{aligned} \rho(\hat{y}(X, C), Y^*) &= \frac{\text{Cov}(\hat{y}(X, C), \mu(X, C))}{\sigma_{\hat{y}}\sigma_{Y^*}} \\ &= \frac{\text{Cov}(\hat{y}(X, C), \mu(X, C))}{\sigma_{\hat{y}}\sigma_{\mu}} \cdot \frac{\sigma_{\mu}}{\sigma_{Y^*}} \\ &= \rho(\hat{y}(X, C), \mu(X, C)) \cdot \frac{\sigma_{\mu}}{\sigma_{Y^*}}, \end{aligned} \quad (4.28)$$

where $\sigma_{\hat{y}} = \sqrt{\text{Var}(\hat{y}(X, C))}$ and $\sigma_{\mu} = \sqrt{\text{Var}(\mu(X, C))}$. By equation 4.20, the factor $\sigma_{\mu}/\sigma_{Y^*}$ is a positive constant that does not depend on the choice of $\hat{y}$. Hence, maximizing $\rho(\hat{y}(X, C), Y^*)$ is equivalent to maximizing $\rho(\hat{y}(X, C), \mu(X, C))$.

Step 3: Correlation is maximized by a positive affine transformation of $\mu(X, C)$.

Let $\tilde{A} \triangleq A - \mathbb{E}[A]$ denote the centered version of a random variable. Then

$$\text{Cov}(\hat{y}(X, C), \mu(X, C)) = \mathbb{E}[\tilde{\hat{y}} \tilde{\mu}]. \quad (4.29)$$

By the Cauchy–Schwarz inequality in L_2 ,

$$\mathbb{E}[\tilde{y} \tilde{\mu}] \leq \sqrt{\mathbb{E}[\tilde{y}^2]} \sqrt{\mathbb{E}[\tilde{\mu}^2]} = \sigma_{\tilde{y}} \sigma_{\mu}. \quad (4.30)$$

Dividing both sides by $\sigma_{\tilde{y}} \sigma_{\mu}$ shows

$$\rho(\hat{y}(X, C), \mu(X, C)) \leq 1, \quad (4.31)$$

with equality if and only if $\tilde{y} = a \tilde{\mu}$ almost surely for some $a \geq 0$. Equivalently,

$$\hat{y}(X, C) = a \mu(X, C) + b \quad \text{a.s.} \quad (4.32)$$

for some $a \geq 0$ and $b \in \mathbb{R}$. If $a = 0$ then $\text{Var}(\hat{y}(X, C)) = 0$, which is excluded by equation 4.20; thus $a > 0$. This proves Lemma 3.

Connection to regression objectives. Since the Bayes-optimal minimizer of expected squared loss is $\mu(x, c) = \mathbb{E}[Y^* \mid x, c]$ (and reduces to $\mathbb{E}[Y^* \mid x]$ under equation 4.16), a regression-aware objective that targets the conditional mean yields an estimator that is also optimal for Pearson correlation, up to a positive affine transformation. A closely related formulation appears in Bottai et al. [BKL22].

4.8.2.4 Proof of Lemma 4

Optimality of MSE for Conditional Expectation. Let the risk function for the regression-aware loss be defined over the joint distribution of inputs x , generated chains $c \sim \pi_{\theta}(\cdot \mid x)$, and ground-truth labels $y^* \sim P(\cdot \mid x)$:

$$\mathcal{R}(\hat{y}) = \mathbb{E}_{x, c, y^*} [(\hat{y}(x, c) - y^*)^2]. \quad (4.33)$$

Let $\mu(x, c) = \mathbb{E}[y^* \mid x, c]$ denote the conditional expectation of the label given the input and reasoning chain. We can decompose the squared error term by adding and subtracting

$\mu(x, c)$:

$$\begin{aligned}
\mathbb{E} [(\hat{y} - y^*)^2 \mid x, c] &= \mathbb{E} [(\hat{y} - \mu(x, c) + \mu(x, c) - y^*)^2 \mid x, c] \\
&= \mathbb{E} [(\hat{y} - \mu(x, c))^2 \mid x, c] + \mathbb{E} [(\mu(x, c) - y^*)^2 \mid x, c] \\
&\quad + 2(\hat{y} - \mu(x, c)) \underbrace{\mathbb{E} [\mu(x, c) - y^* \mid x, c]}_{=0}.
\end{aligned} \tag{4.34}$$

The cross-term vanishes because $\mathbb{E}[y^* \mid x, c] = \mu(x, c)$ by definition. The total risk is thus:

$$\mathcal{R}(\hat{y}) = \mathbb{E}_{x,c} [(\hat{y}(x, c) - \mu(x, c))^2] + \text{Var}(y^* \mid x, c). \tag{4.35}$$

The second term, $\text{Var}(y^* \mid x, c)$, represents irreducible noise and is independent of the estimator $\hat{y}$. Therefore, minimizing the risk $\mathcal{R}(\hat{y})$ is equivalent to minimizing the first term, which is strictly non-negative and equals zero if and only if:

$$\hat{y}(x, c) = \mu(x, c) = \mathbb{E}[y^* \mid x, c]. \tag{4.36}$$

Thus, the global minimizer of the regression-aware loss is the conditional expectation.

4.8.2.5 Proof of Lemma 2

To derive the gradient, we expand the inner expectation as an integral over the reasoning trajectories $c \sim \pi_\theta(c \mid x)$:

$$\mathcal{L}(\theta) = \mathbb{E}_{(x, y^*) \sim \mathcal{D}} \int_c \pi_\theta(c \mid x) r(\theta, x, c) dc.$$

Applying the gradient operator ∇_θ and utilizing the product rule for differentiation:

$$\begin{aligned}
\nabla_\theta \mathcal{L}(\theta) &= \mathbb{E}_{(x, y^*) \sim \mathcal{D}} \int_c \nabla_\theta [\pi_\theta(c \mid x) r(\theta, x, c)] dc \\
&= \mathbb{E}_{(x, y^*) \sim \mathcal{D}} \int_c [(\nabla_\theta \pi_\theta(c \mid x)) r(\theta, x, c) + \pi_\theta(c \mid x) (\nabla_\theta r(\theta, x, c))] dc.
\end{aligned}$$

Term 1: CoT Policy Update. Using the log-derivative trick, $\nabla_{\theta}\pi_{\theta}(c \mid x) = \pi_{\theta}(c \mid x)\nabla_{\theta}\log\pi_{\theta}(c \mid x)$, the first part of the sum becomes:

$$\mathbb{E}_{(x,y^*)\sim\mathcal{D}}\int_c\pi_{\theta}(c \mid x)[r(\theta,x,c)\nabla_{\theta}\log\pi_{\theta}(c \mid x)]dc = \mathbb{E}_{(x,y^*)\sim\mathcal{D},c\sim\pi_{\theta}}[r(\theta,x,c)\nabla_{\theta}\log\pi_{\theta}(c \mid x)].$$

This term performs reinforcement learning on the reasoning process, shifting the policy’s mass toward trajectories that yield higher rewards.

Term 2: Prediction Update. The second part accounts for the explicit dependence of the reward on θ . Within this integral, the trajectory c is a sample from the current policy (effectively a *stopped-gradient* c):

$$\mathbb{E}_{(x,y^*)\sim\mathcal{D}}\int_c\pi_{\theta}(c \mid x)[\nabla_{\theta}r(\theta,x,c)]dc = \mathbb{E}_{(x,y^*)\sim\mathcal{D},c\sim\pi_{\theta}}[\nabla_{\theta}r(\theta,x,c)].$$

This term enables direct backpropagation through the differentiable components of the reward, mapping the predicted $\hat{y}_{\theta}(x,c)$ to the target value y^* .

4.8.3 Algorithm and Experiments

4.8.3.1 Response Length and Entropy

We note that the response length increases, and entropy decreases throughout REAL training, as shown in Fig. 4.4.

4.8.3.2 Comparison with JEPO

We conduct a direct comparison with JEPO as shown in Tab. 4.12. The JEPO does not perform as effectively as our proposed approach in regression-aware tasks.

4.8.3.3 REAL Algorithm

See Algo. 11 for detailed implementation.

Algorithm 11 REAL: Regression-Aware Reinforcement Learning

Require: dataset $\mathcal{D}$, policy π_θ , log-likelihood weight λ , prediction weight β , group size K , learning rate η .

```
1: while not converged do
2:   Sample batch  $B = \{(x, y^*)\} \sim \mathcal{D}$ .
3:   /* Step 1: Sample  $K$  generations with CoTs */
4:   Sample  $K$  reasoning chains  $\{c_1, \dots, c_K\} \sim \pi_\theta(\cdot|x)$ .
5:   Compute rewards  $r_{\text{REAL}}(\theta, x, c_i)$  via Eq. 4.5.
6:   /* Step 2: Compute CoT Update term and Prediction Update term */
7:   Compute Advantage  $\tilde{A}_{\text{REAL}}^{(i)}$ 
8:   Forward the generation again to acquire the gradient for both CoT and Prediction
   tokens.
9:   Aggregate terms to estimate  $\nabla_\theta \mathcal{L}_{\text{REAL}}^{\text{RLOO}}$  ( Eq. 4.8).
10:  /* Step 3: Optimization */
11:  Update parameters:  $\theta \leftarrow \theta + \eta \cdot \frac{1}{|B|} \sum \nabla_\theta \mathcal{L}_{\text{REAL}}^{\text{RLOO}}$ .
12: end while
```

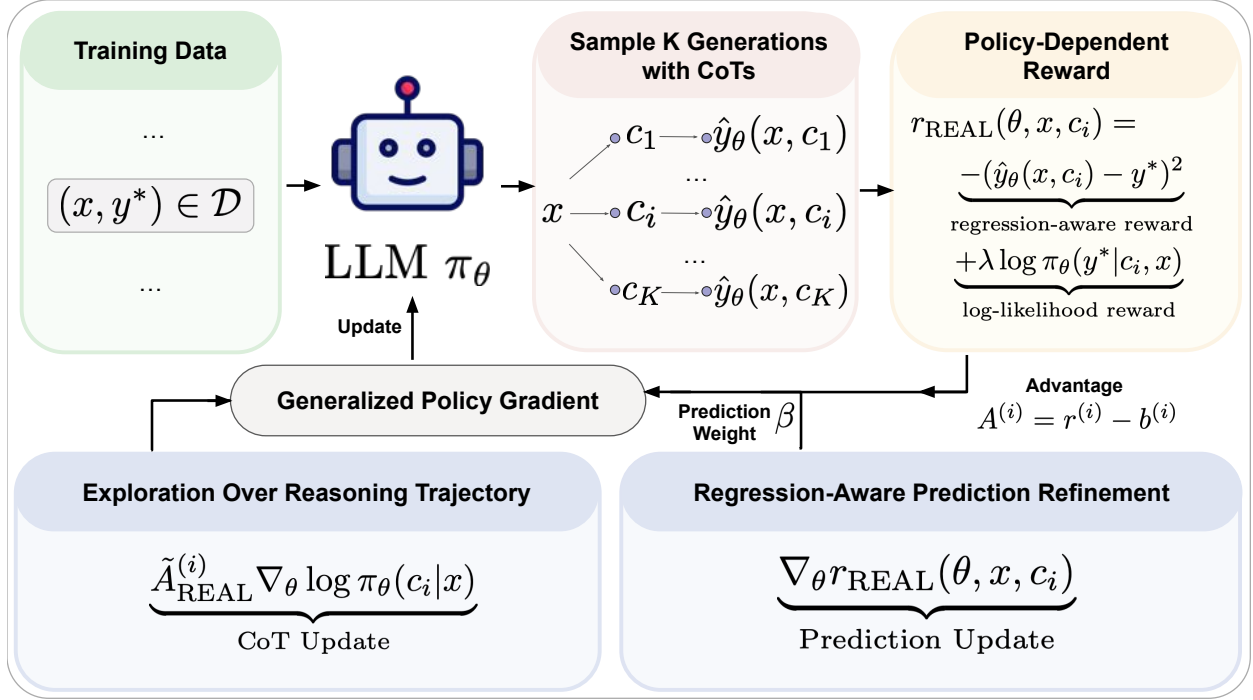


Figure 4.1: **Overview of the REAL framework.** REAL addresses the limitations of standard RL in LLM-as-a-Judge tasks by optimizing a policy-dependent regression reward. The framework employs a generalized policy gradient that leads to a gradient update that decomposes into two terms: (1) **Exploration Over Reasoning Trajectory**; and (2) **Regression-Aware Prediction Refinement**. This enables principled optimization of ordinal structures that standard RL with binary rewards typically ignores. The full algorithm is in Alg. 11.

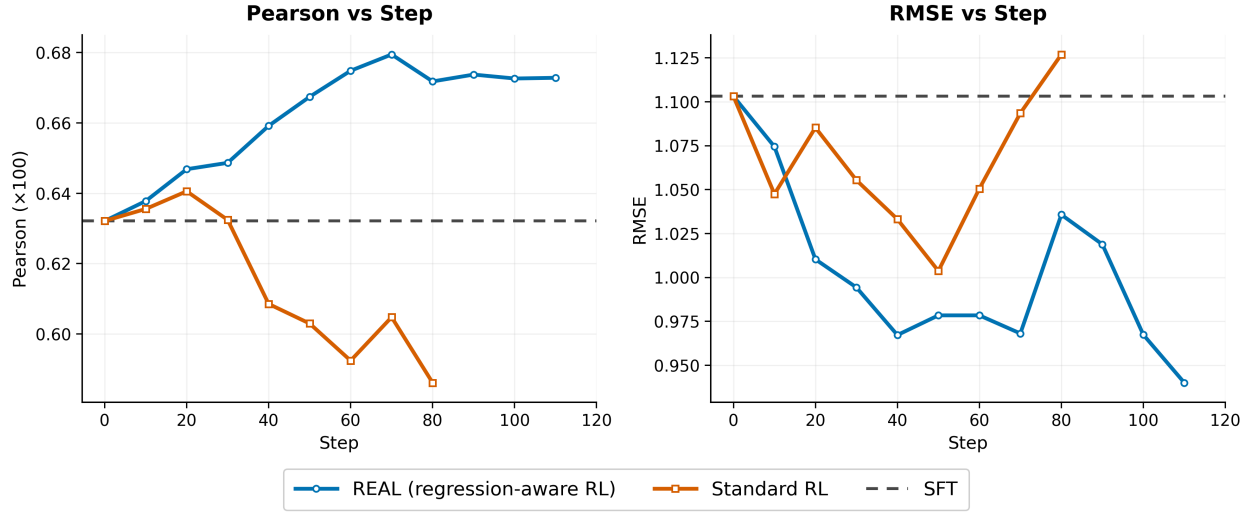


Figure 4.2: Evaluation performance during RL training. Both standard RL with binary reward (i.e., $r_{\text{acc}} = \mathbf{1}(y = y^*)$) and REAL with regression-aware reward (i.e., Eq. 4.5) were initialized from the SOTA SFT checkpoint, i.e., TRACT [CLL25]. Standard RL results in suboptimal performance in correlation metrics compared to our proposed approach: REAL.

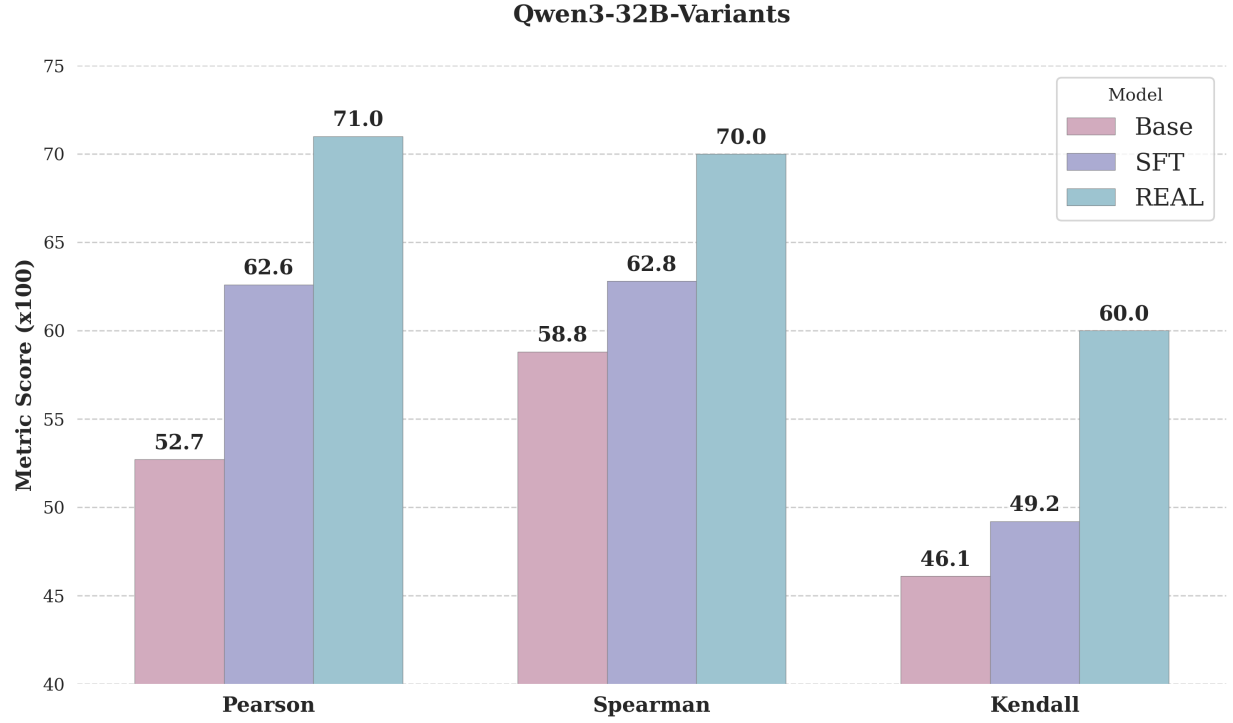


Figure 4.3: Performance gains on Qwen3-32B. Our method achieves an average increase of +8.40 Pearson and +7.20 Spearman correlation over the SFT baseline, and +18.30/+11.20 over the base model.

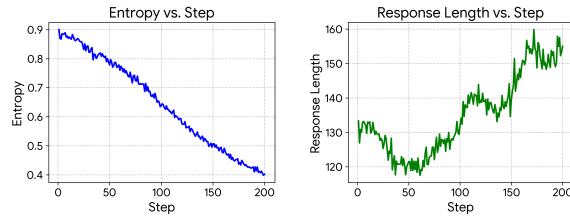


Figure 4.4: Response length and entropy during REAL training. Response length increases, and the per-token entropy of the policy model decreases steadily.

Table 4.2: **Evaluation results ($\times 100$) across benchmarks using Pearson (r), Spearman (ρ), and Kendall (τ).** All baseline reward model scores are taken from TRACT [CLL25] and Prometheus [KSC23]. Results for the Mistral and Qwen models are obtained from our own runs. Underline indicates the initialization checkpoint used by REAL. We find that TRACT saturates on the Qwen3 series models, so we choose RAFT as the initialization checkpoint. Mistral2-7B’s base model failed to follow instructions and respond in the correct format, so we applied a 100-step warmup. Due to computational constraints we run only RAFT and REAL for Qwen3-32B; we include all baselines for other models.

Method	Training	Inference	FB Bench			FLASK			Vic. Bench			MT Bench			Average		
			r	ρ	τ	r	ρ	τ	r	ρ	τ	r	ρ	τ	r	ρ	τ
Baseline Reward Models																	
GPT-3.5-Turbo-0613	None	Standard	56.3	52.1	45.3	27.0	23.2	18.7	27.5	26.7	20.2	42.2	37.1	29.9	38.3	34.8	28.5
Prometheus-1-7B	SFT	Standard	84.7	84.9	76.7	45.7	45.7	36.5	29.3	29.5	21.6	36.7	37.1	28.5	49.1	49.3	40.8
Prometheus-1-13B	SFT	Standard	86.0	85.8	77.1	46.6	42.9	34.6	47.3	45.1	34.1	46.7	45.5	34.5	56.7	54.8	45.1
Prometheus-2-7B	SFT	Standard	84.5	84.7	76.5	51.2	49.3	40.5	48.8	48.0	41.1	51.9	48.3	39.2	59.1	57.6	49.3
Prometheus-2-7B	SFT	RAIL	85.3	85.3	72.9	52.5	51.4	39.2	51.0	51.3	40.3	53.8	51.1	38.9	60.7	59.8	47.8
Mistral2-7B-Variants																	
Base (w/ warmup)	None	Standard	83.1	83.3	74.8	41.5	41.9	34.1	49.2	42.4	36.1	30.9	31.8	25.3	51.2	49.8	42.6
Base (w/ warmup)	None	RAIL	83.7	84.3	70.2	42.5	43.7	32.3	50.3	45.4	34.0	32.0	29.9	21.4	52.1	50.8	39.5
RAFT	SFT	RAIL	87.9	88.0	76.3	41.8	41.9	31.5	52.8	51.3	40.9	39.9	41.8	30.7	55.6	55.8	44.8
TRACT	SFT	RAIL	93.9	93.7	82.9	50.7	50.0	37.2	56.2	54.8	42.6	52.1	50.1	36.6	63.2	62.2	49.8
Standard RL	RL	RAIL	93.7	93.7	82.8	51.6	50.5	37.9	58.0	56.0	43.4	52.9	50.7	37.1	64.1	62.7	50.3
REAL (ours)	RL	RAIL	93.2	93.4	82.5	56.0	54.1	41.1	63.3	60.2	46.3	59.3	56.9	42.2	67.9	66.2	53.0
Qwen3-8B-Variants																	
Base	None	Standard	56.6	62.7	53.9	44.8	48.0	40.1	37.3	46.5	40.8	35.5	32.7	26.5	43.6	47.4	40.3
Base	None	RAIL	56.7	65.4	54.1	45.0	48.3	38.5	37.3	46.1	38.6	35.9	35.5	27.0	43.7	48.8	39.5
RAFT	SFT	RAIL	84.3	85.5	73.0	49.2	50.1	37.5	59.9	57.2	44.0	54.1	51.7	38.5	61.9	61.1	48.3
TRACT	SFT	RAIL	94.9	94.7	84.3	51.4	51.3	38.5	50.5	50.4	38.3	55.8	58.6	43.6	63.1	63.8	51.2
Standard RL	RL	RAIL	94.0	94.2	83.5	52.3	52.4	39.5	57.1	56.1	43.0	49.5	53.5	39.7	63.2	64.0	51.4
REAL (ours)	RL	RAIL	92.0	92.1	85.7	53.8	53.9	43.1	60.5	57.8	48.2	61.7	60.8	47.1	67.0	66.0	56.1
Qwen3-32B-Variants																	
Base	None	RAIL	63.4	70.8	56.7	54.3	60.4	47.2	50.8	57.4	45.0	42.5	46.8	35.3	52.7	58.8	46.1
RAFT	SFT	RAIL	85.4	86.5	72.9	52.1	52.9	39.9	51.9	52.0	39.9	61.1	59.6	43.9	62.6	62.8	49.2
REAL (ours)	RL	RAIL	91.1	91.7	85.9	58.9	58.6	47.4	65.1	60.7	51.2	68.9	69.1	55.2	71.0	70.0	60.0

Table 4.3: **Average-of-N results at inference.** Averaging the results of N generations slightly improves performance, although REAL already enjoys efficiency and strong performance at $N = 1$.

Setting	$r \uparrow$	$\rho \uparrow$	$\tau \uparrow$	RMSE $\downarrow$	MAE $\downarrow$
$N = 1$	67.9	66.2	53.0	0.968	0.697
$N = 4$	68.1	66.2	53.0	0.965	0.695
$N = 10$	68.1	66.2	53.0	0.964	0.695

Table 4.4: **Comparison between standard RL baselines and REAL on Mistral2-7B.** Standard RL(reg) with regression-aware policy-independent rewards also fail to improve upon the SFT checkpoint.

Method	Type	r	ρ	τ
TRACT	SFT	63.2	62.2	49.8
Standard RL(acc)	RL	64.1	62.7	50.3
Standard RL(reg)	RL	63.4	62.6	50.1
REAL	RL	67.9	66.2	53.0

Table 4.5: **Ablation on group size K .** Larger values of K generally result in better correlation performance.

Setting	$r \uparrow$	$\rho \uparrow$	$\tau \uparrow$	RMSE $\downarrow$	MAE $\downarrow$
$K = 4$	63.7	62.6	50.2	1.063	0.775
$K = 8$	67.2	65.0	52.0	0.943	0.679
$K = 16$	67.9	66.2	53.0	0.968	0.697

Table 4.6: **Ablation on log-likelihood weight λ .** Using regression-aware reward alone ($\lambda = 0$) already yields excellent results, while extra log-likelihood reward further improves.

Setting	$r \uparrow$	$\rho \uparrow$	$\tau \uparrow$	RMSE $\downarrow$	MAE $\downarrow$
TRACT	63.2	62.2	49.8	1.103	0.813
$\lambda = 0$	66.0	64.6	51.8	1.027	0.736
$\lambda = 1$	67.9	66.2	53.0	0.968	0.697
$\lambda = 5$	68.1	65.8	52.7	0.955	0.719
$\lambda = 10$	68.1	66.1	52.9	0.928	0.699

Table 4.7: **Ablation on dynamic sampling when group size $K = 8$.** Partially correct prompts provide the best learning signal.

Setting	$r \uparrow$	$\rho \uparrow$	$\tau \uparrow$	RMSE $\downarrow$	MAE $\downarrow$
$0 < \text{acc} < 1$	67.2	65.0	52.0	0.943	0.679
$0 < \text{acc} \leq 1$	65.6	64.1	51.3	1.000	0.711
$0 \leq \text{acc} < 1$	65.3	63.4	50.9	1.001	0.728
$0 \leq \text{acc} \leq 1$	65.5	63.9	51.1	0.990	0.715

Table 4.8: **Initialization from different checkpoints.** Initializing from a high-quality SFT checkpoint can lead to better SOTA performance. Our algorithm works with different checkpoint types.

Setting	$r \uparrow$	$\rho \uparrow$	$\tau \uparrow$	RMSE $\downarrow$	MAE $\downarrow$
base	52.1	50.8	39.5	1.225	0.901
init. from base	58.8	56.8	45.1	1.155	0.793
Δ	+6.7	+6.0	+5.6	-0.070	-0.108
SFT	63.2	62.2	49.8	1.103	0.813
init. from SFT	67.9	66.2	53.0	0.968	0.697
Δ	+4.7	+4.0	+3.2	-0.135	-0.116

Table 4.9: **Ablation on prediction weight β .** We choose $\beta = 0.01$ (See Eq. 4.8) for initialization from both base and SFT checkpoints.

Setting	$r \uparrow$	$\rho \uparrow$	$\tau \uparrow$	RMSE $\downarrow$	MAE $\downarrow$
<i>Init. from Base</i>					
$\beta = 0$	57.7	56.9	45.1	1.200	0.831
$\beta = 0.01$	58.8	56.8	45.1	1.155	0.793
$\beta = 1$	57.9	56.4	44.4	1.184	0.822
<i>Init. from SFT</i>					
$\beta = 0$	67.3	65.9	52.5	0.980	0.701
$\beta = 0.01$	67.9	66.2	53.0	0.968	0.697
$\beta = 1$	66.7	64.5	51.6	0.935	0.677

Table 4.10: **Ablation on temperature T .** $T = 1$ is chosen by default. The temperature parameter is applied in both the training and inference stages, for both CoT generation and score prediction.

Setting	$r \uparrow$	$\rho \uparrow$	$\tau \uparrow$	RMSE $\downarrow$	MAE $\downarrow$
$T = 0.8$	66.8	64.7	51.9	0.973	0.725
$T = 1$	67.9	66.2	53.0	0.968	0.697

Table 4.11: Results on score-difference prediction using the Preference_Collection dataset. REAL achieves better out-of-domain generalization while maintaining competitive in-domain performance.

	Preference-Bench			HHH	MT-Bench	Auto-J
Method	Pearson $\uparrow$	Spearman $\uparrow$	Acc $\uparrow$	Acc $\uparrow$	Acc $\uparrow$	Acc $\uparrow$
Base	9.30	17.6	12.2	50.7%	64.8%	64.7%
TRACT	93.9	92.8	82.3	75.6%	75.6%	79.2%
REAL	93.4	92.8	84.1	76.9%	76.3%	79.7%

Table 4.12: **Comparison with JEPO with Mistral2-7B.**

Method	FB Bench			FLASK			Vic. Bench			MT Bench			Average		
	r	ρ	τ	r	ρ	τ	r	ρ	τ	r	ρ	τ	r	ρ	τ
JEPO	92.5	92.9	81.6	54.3	52.6	39.6	61.7	59.7	45.9	55.7	54.9	39.8	66.3	64.8	51.8
REAL	93.2	93.4	82.5	56.0	54.1	41.1	63.3	60.2	46.3	59.3	56.9	42.2	67.9	66.2	53.0

CHAPTER 5

Object-Conditioned Energy-Based Attention Map Alignment in Text-to-Image Diffusion Models

5.1 Introduction

Recently, large-scale text-to-image diffusion models [RBL22b, NDR21, RDN22b, SCS22, BNH22, HHS23] have showcased remarkable capabilities in producing diverse, imaginative, high-resolution visual content based on free-form text prompts. Despite their revolutionary progress, however, these models may not consistently capture and convey the full semantic meaning of the provided text prompts [CU22, RRG22]. Some well-known issues include omission, hallucination, or duplication of details [YXK22], semantic leakage of attributes between entities [RRG22], and miscomprehension of intricate textual descriptions [SCS22].

Many previous works have focused on addressing the semantic misalignment issues, particularly concerning multiple-object generation and attribute binding. Composable Diffusion (CD) [LLD22] composes multiple output noises guided by different objects in a text prompt during the generation process. However, this approach often results in a blend of objects, failing to distinctly separate them. Prompt-to-Prompt (PtP) [HMT23b] observes a strong correlation between cross-attention maps and the layout of an image. Building on this, Structured Diffusion (StrD) [FHF23] experiments with averaging attention maps generated by different noun phrases for the same queried image latent representation. Yet, a simple

⁰Accepted to European Conference on Computer Vision (ECCV) 2024

average is inadequate for consistently generating images with multiple objects possessing complex attributes. Attend-and-Excite (AnE) [CAV23] proposes a novel approach of maximizing the attention map scores of object tokens by updating the latent at each sampling step. However, we note that artifacts and incorrect attribute binding are likely when AnE maximizes the attention weights of object tokens without any concerns on attributes. Similarly, A-star (A*) [AKJ23] aims to minimize the intersection of different objects’ attention maps. In response, SynGen (SG) [RHG23] proposes an attribute-object pair-centric objective, aiming to minimize the distribution distance within the pair while maximizing it from other tokens, based on the assumption that normlized attention maps follow a multinomial distribution. Our findings (see Figs. 5.1 and 5.4-5.6) indicate that this approach still struggles with object neglect due to its pair-centric nature. We argue that multiple-object generation is more critical than attribute binding, as attributes cannot manifest without the presence of objects. Furthermore, in scenarios with multiple objects and no explicit attributes in a prompt, SG is degraded to standard Stable Diffusion Models (SD) [RBL22b]. Diverging from these methods, Energy-Based Cross Attention (EBCA) [PKK23] introduces an Energy-Based Model (EBM) framework [XLZ16a, YXM21, YXM22, YZH24, YZX24] for queries and keys within cross-attention mechanisms, proposing updates to text embeddings instead of latent noise representations.

A closer look at both the fluctuations of attention intensities and the attention distributions of attribute-object pairs in these methods shed light on the root cause of the misalignment issues. As illustrated in Fig. 5.1, alignment in attribute-object attention maps (e.g., ‘purple crown’ in SG) encourages attribute binding. However, attention map alignment alone does not guarantee complete semantic alignment, as the intensity levels of object attention maps are crucial in determining the presence of an object in the final image. For example, in the image generated by SG, the crown is notably absent. Conversely, despite successful generation of both objects with strong intensities, AnE binds the attribute ‘purple’ incorrectly to the suitcase resulting from misaligned attention distribu-

tions of attribute-object pairs. Motivated by these key observations, we introduce a novel *object-conditioned* Energy-Based Attention Map Alignment (EBAMA) method to hopefully address both the incorrect attribute binding and the catastrophic object neglect problems in a unified framework. Notably, we show that approximately maximizing the log-likelihood for a z -parameterized EBM effectively leads to optimizing an object-centric binding loss, which emphasizes both the object attention map intensity levels and the attribute-object attention map alignment. We further develop an object-centric intensity regularizer to prevent excessive shifts of objects towards their attributes, providing an extra degree of freedom balancing the trade-off between correct attribute binding and the necessary presence of objects.

We summarize our **contributions** as follows: i) we introduce a novel object-conditioned EBAMA method to address both the incorrect attribute binding and the catastrophic object neglect problems in text-controlled image generation; ii) extensive qualitative and quantitative experiments, including human evaluation, on several challenging benchmarks demonstrate the superior performance of our method over strong previous approaches. iii) We showcase that our approach has great promise in further enhancing the text-controlled image editing ability of diffusion models.

5.2 Related Work

EBM Framework for Attention Mechanisms Recent advancements in the theoretical exploration of attention mechanisms have increasingly embraced the EBM framework [Hop82, MPR87, RSL20]. Modern Hopfield Networks [RSL20] showcases that one of the proposed energy minima is equivalent to the attention mechanism. Building on this groundwork, Energy Transformer [HLP23] designs an engineered energy function to extract the relationships between tokens. Furthering this approach, EBCA [PKK23] first formulates EBMs of query values conditioned on key values in each cross-attention layer. Similarly, our method seeks to exploit the theoretical potential of EBMs, focusing on the unique formulation of

object-conditioned EBMs for attention maps.

Text-to-Image Diffusion Models Most large-scale text-to-image diffusion models [RBL22b, NDR21, RDN22b, SCS22, BNH22, HHS23] utilize classifier-free guidance [HS22b] for improved conditional synthesis results. However, due to its strong linguistic and visual priors injected from the training dataset, these models suffer from diverse semantic misalignment issues related to the objects in the provided text prompts and their attribute(s) [CU22, SCS22, RRG22, YXK22, ZYZ24]. Our approach better aligns images with its provided texts by mitigating the issues of object neglect and incorrect attribute binding without fine-tuning the diffusion models or additional training datasets.

Attention-Based Enhancement PtP [HMT23b] identifies a correlation between cross-attention maps and image layout. Expanding on this, StrD [FHF23] experiments with averaging attention maps from different noun phrases to mitigate object neglect and attribute leakage. AnE [CAV23] introduces a method to enhance object presence by maximizing attention map weights for object tokens. SG [RHG23] proposes minimizing distribution distances of the attention maps within attribute-object pairs and maximizing the distances between the pairs and the other tokens. Different from the previous approaches, EBCA [PKK23] adopts an EBM framework, focusing on updating text embeddings within cross-attention mechanisms. Our work also introduces an energy-inspired attention map alignment objective, while our objective has a specific emphasis on the object tokens.

5.3 Background

5.3.1 Stable Diffusion Models

For fair comparison with previous approaches, we also conduct experiments on open-sourced state-of-the-art Stable Diffusion Models (SD) [RBL22b]. SD first encodes an image x into

the latent space using a pretrained encoder [ERO21], i.e., $z = \mathcal{E}(x)$. Given a text prompt y , SD optimizes the conditional denoising autoencoder ϵ_θ by minimizing the objective

$$\mathcal{L}_\theta = \mathbb{E}_{t, \epsilon \sim \mathcal{N}(0,1), z \sim \mathcal{E}(x)} \|\epsilon - \epsilon_\theta(z_t, t, \phi(y))\|^2, \quad (5.1)$$

where ϕ is a frozen CLIP text encoder [RKH21b], z_t is a noised version of the latent z , and the time step t is uniformly sampled from $\{1, \dots, T\}$. During sampling, z_T is randomly sampled from standard Gaussian and denoised iteratively by the denoising autoencoder ϵ_θ from time T to 0. Finally, a decoder $\mathcal{D}$ reconstructs the image as $\tilde{x} = \mathcal{D}(z_0)$.

5.3.2 Cross-Attention Mechanism

In the cross-attention mechanism, K is the linear projections of W_y , the CLIP-encoded text embeddings of text prompt y . Q is the linear projection of the intermediate image representation parameterized by latent variables z . Given a set of queries Q and keys K , the (unnormalized) attention features and (softmax-normalized) scores between these two matrices are

$$A = \frac{QK^T}{\sqrt{m}}, \quad \tilde{A} = \text{softmax}\left(\frac{QK^T}{\sqrt{m}}\right), \quad (5.2)$$

where m is the feature dimension. We consider both attention features and scores for our modeling here, which we denote as A_s and $\tilde{A}_s$ for token s , respectively.

5.4 Method

The key idea of the proposed method derives from the *object-oriented structure* that underlies most prompts for text-to-image generation. To be specific, syntactically the majority of prompts can be parsed as the modifiers and the entity-nouns, i.e., the nouns that correspond to *objects* in the generated image, such as “A red metal crown”, and “A girl in red”, etc. Based on the observation, we propose to exploit the object-oriented structure by employing

an ensemble of object-centric cross-attention losses for inference-time optimization. Our aim is to address the semantic misalignment issues including both the attribute binding (e.g., semantic leakage and attribute neglect [RHG23]) and catastrophic object neglect [CAV23], with the principally derived and deliberately designed optimization objective. We then discuss the key components of our method as follows.

5.4.1 Extraction of the Object-Oriented Structure

Following the pre-processing step in [RHG23], we parse the prompt using Spacy’s [HM17] transformer-based dependency parser to extract the object-oriented structure. We identify a set S of object tokens s from the prompt, whose tag is either NOUN (noun) such as ‘backpack’ or PROP (proper noun) such as ‘Tesla company’ using the parser; we exclude nouns that serve as direct modifiers of other nouns. The remaining modifiers are grouped by their corresponding object tokens, denoted as the modifier sets for each object token s , i.e., $\mathcal{M}(s)$. Note that $\mathcal{M}(s) = \emptyset$ if there are no modifiers corresponding to the object token s . We refer to supplementary material for more details about the parsing process.

5.4.2 Object-Conditioned Energy-Based Model

We assume that the distribution of the modifier tokens $l \in \bigcup_s \mathcal{M}(s)$ given the object token s is

$$p_z(l|s) = \frac{1}{Z(s)} \exp(f(A_l, A_s)), \quad (5.3)$$

where $Z(s) = \sum_l \exp(f(A_l, A_s))$ is the normalizing constant and f is the negative energy function. How to choose the energy function for attention maps remains an interesting and open problem. Prior works [RSL20, PKK23] utilize a log-sum-exp term to model the exponential interaction and alignment between state patterns (query) and stored patterns (key). However, the alignment measurement for attentions maps across different tokens is unclear. To bridge the gap, we propose the application of a non-crafted yet effective

energy function — cosine similarity defined as $f(A_l, A_s) = \langle A_l, A_s \rangle / (\|A_l\| \cdot \|A_s\|)$. The efficacy of this choice of energy function is validated in the experimental section. Eqn. (5.3) therefore defines a multinomial token distribution as a z -parameterized conditional energy-based model, where z is the latent variables of SD. The inference-time optimization over the latent variables z is then equivalently maximizing the log-likelihood of this EBM, which increases the probabilities of the syntactically related modifier tokens of the given object s . To be specific, it can be shown that (see the supplementary material)

$$\nabla_z \log p_z(l|s) = \nabla_z f(A_l, A_s) - \mathbb{E}_{p_z(l|s)} [\nabla_z f(A_l, A_s)]. \quad (5.4)$$

Since the vocabulary size of modifier tokens can be large in practice (in the order of 10^4), we consider resorting to negative sampling [MSC13] for the approximation of the expectation term, where we uniformly sample tokens unrelated to the object token and calculate the Monte Carlo average. This particular implementation choice of Eqn. (5.4) then leads to the object-centric attribute binding loss below.

5.4.3 Object-Conditioned Energy-Based Attention Map Alignment

For each object token $s \in S$, we design the following two components that consist of the object-centric attention loss:

Object-centric attribute binding First, instead of operating on the noun-modifier normalized attention score pairs as in [RHG23], we focus on optimizing the log-likelihood of the object-conditioned EBM using negative sampling. This gives us the attribute binding loss:

$$L_b^{(s)} = -\frac{1}{|\mathcal{M}(s)|} \sum_{l \in \mathcal{M}(s)} f(A_s, A_l) + \frac{1}{N - |\mathcal{M}(s)| - 1} \sum_{l \notin \mathcal{M}(s), l \neq s} f(A_s, A_l), \quad (5.5)$$

whose negative gradient w.r.t. z could be seen as the Monte Carlo approximation of Eqn. (5.4). The goal of $L_b^{(s)}$ is to: i) maximize the cosine similarity between the given object s and its syntactically-related modifier tokens, while ii) enforcing the repulsion of grammatically

unrelated ones in the feature space. Note that the loss above only applies to the cases where $\mathcal{M}(s)$ is a non-empty set. For the case where $\mathcal{M}(s) = \emptyset$, only the **repulsive term** of Eqn. (5.5) is used.

Object-centric intensity regularizer Although the proposed attribute binding loss mitigates the catastrophic object neglect problem (see $\lambda = 0$ entries in Tab. 5.1), we observe that the object-related attention feature can still be overly shifted when there are multiple modifier tokens in the $\mathcal{M}(s)$ or multiple object tokens in a prompt; this could again potentially leads to the object neglect phenomenon. To address this issue, we follow [CAV23] and propose an object-centric intensity regularizer to maintain the attention intensity level of object s :

$$L_n^{(s)} = -\|\mathcal{K}(\tilde{A}_s)\|_\infty, \quad (5.6)$$

where $\mathcal{K}$ is a 3x3 Gaussian kernel, and $\|\cdot\|_\infty$ denotes the maximum value of a vector. We use the attention scores in Eqn. (5.2) as its input. We refer to $\|\mathcal{K}(\tilde{A}_s)\|_\infty$ as the **intensity level** of the object token s .

The final object-centric attention loss L is the linear combination of the binding loss and the regularizer, i.e.

$$L = \sum_{s \in S} L^{(s)} = \sum_{s \in S} L_b^{(s)} + \lambda L_n^{(s)}, \quad (5.7)$$

where intensity weight λ is a hyper-parameter to specify. $\lambda > 0$ enforces the presence of object s , but excessively intensified object attention can hinder the attribute binding performance and lower visual image quality. We provide empirical analysis on how to tune the weight in practice (see Section 5.5.4 and supplementary material for details).

5.4.4 Workflow

Our workflow is illustrated in Fig. 5.2. To begin, at each time step t , we aggregate the attention map features denoted as A at a resolution of 16x16. This aggregation is performed

Table 5.1: **Comparison of Full Sim., Min. Sim., and T-C Sim. across different methods on the AnE dataset.** Note that the performance of SG on Animal-Animal is degraded to SD, as the prompts do not contain any attribute-object pairs. The best and second-best performances are marked in bold numbers and underlines, respectively; tables henceforth follows this format.

Method	Animal-Animal			Animal-Object			Object-Object		
	Full Sim.	Min. Sim.	T-C Sim.	Full Sim.	Min. Sim.	T-C Sim.	Full Sim.	Min. Sim.	T-C Sim.
SD[RBL22b]	0.311	0.213	0.767	0.340	0.246	0.793	0.335	0.235	0.765
CD[LLD22]	0.284	0.232	0.692	0.336	0.252	0.769	0.349	0.265	0.759
StrD[FHF23]	0.306	0.210	0.761	0.336	0.242	0.781	0.332	0.234	0.762
EBCA[PKK23]	0.291	0.215	0.722	0.317	0.229	0.732	0.321	0.231	0.726
AnE[CAV23]	0.332	0.248	0.806	0.353	0.265	0.830	0.360	0.270	0.811
SG[RHG23]	0.311	0.213	0.767	0.355	0.264	0.830	0.355	0.262	0.811
Ours($\lambda = 0$)	0.340	<u>0.255</u>	<u>0.814</u>	<u>0.362</u>	0.271	0.851	<u>0.360</u>	<u>0.270</u>	<u>0.823</u>
Ours	0.340	0.256	0.817	0.362	<u>0.270</u>	0.851	0.366	0.274	0.836

after one step of propagating z_t through the denoising model. Subsequently, we calculate the object-centric attention loss, as described in Eqn. (5.7). Finally, we backpropagate the computed loss and update z_t for each time step, following the formula $z'_t \leftarrow z_t - \alpha \nabla_{z_t} L$, where α represents the step size. In our experimental setup, we set $\alpha = 20$. Note that we only perform updates on z_t during the first half of the sampling steps, which corresponds to 25 steps since we use a DDIM sampler with a total of 50 steps. More details about the workflow can be found in the supplementary material.

5.5 Experiments

We compare our generation results with SD, CD, StrD, EBCA, AnE, and SG on two artificial datasets, AnE dataset [CAV23] and DVMP [RHG23], and one natural-language dataset, ABC-6K [FHF23]. We refer to supplementary material for implementation details and computational efficiency comparison.

Datasets The AnE dataset [CAV23] comprises three benchmarks: Animal-Animal, Animal-Object, and Object-Object. Each benchmark varies in complexity and incorporates a combination of potentially colored animals and objects. The prompt patterns for these benchmarks include two unattributed animals, one unattributed animal and one attributed object, and two attributed objects, respectively. The DVMP dataset [RHG23] features a diverse set of objects (e.g., daily objects, animals, fruits, etc.) and diverse modifiers including colors, textures and so on. It features more than three attribute descriptor per prompt. The ABC-6K dataset [FHF23], derived from natural MSCOCO [LMB14] captions, includes prompts with at least two color words modifying different objects. The first two datasets are artificial, while ABC-6K is composed of natural language captions.

5.5.1 Quantitative Comparison

Following the setting of [CAV23], we compare the Text-Image Full Similarity (Full Sim.), Text-Image Min Similarity (Min. Sim.), and Text-Caption Similarity (T-C Sim.) on the AnE dataset. Additionally, we present the Full Sim. results on the DVMP and ABC-6K datasets.

Full Sim. is the CLIP [RKH21b] cosine similarity score between the text prompt and the generated image. Furthermore, we assess CLIP similarity for the most neglected object independently from the full text by computing the CLIP similarity scores between each sub-prompt and the generated image. The smaller score is denoted as Min. Sim.. T-C Sim. is

the average CLIP similarity between the prompt and all captions generated by a pre-trained BLIP image-captioning model [LLX22] with the generated image as input.

We generate 64 images for each prompt using the same seed across all methods and compute the average score between each prompt and its corresponding images. Our method consistently demonstrates superior performance across all datasets, as shown in Tab. 5.1. We stress the following advantages of our method: (1) Our method distinguishes itself from SG by its adaptability to the Animal-Animal dataset, even when the prompts lack specific attributes; (2) Our method with $\lambda = 0$ surpasses AnE and SG in all cases, underscoring the effectiveness of our object-centric attribute binding loss; (3) As the dataset becomes more complicated, our method with hyper-picked λ gains a more significant advantage over that with $\lambda = 0$.

In Fig. 5.3, despite SG’s deliberate design for multi-attribute prompts, our method consistently surpasses SG. Furthermore, in the ABC-6K dataset, AnE and SG exhibit performance levels similar to that of SD, while our method consistently achieves superior results. These advantages are further confirmed by our human evaluation in 5.5.3.

5.5.2 Qualitative Comparison

In Figs. 5.4-5.6, we identify recurrent failure modes in SG and AnE, attributable to the ineffectiveness of their objective design. AnE frequently struggles with incorrect attribute association, whereas SG often fails to generate multiple objects simultaneously. In contrast, our method attains high-quality semantic alignment with deliberately designed optimization objective. It also exhibits more stable performance across different random seed selections.

Object Omission SG, due to its pair-centric approach, frequently omits objects, as evidenced by missing items like cars, apples, and crowns in Fig. 5.4, tomatoes, crowns, strawberries in Fig. 5.5, and earrings, ties, etc. in Fig. 5.6.

Attribute Omission Due to a lack of concern for attribute tokens, AnE fails to overcome the strong visual priors over objects. e.g., the green apple in Fig. 5.4, the non-spotted dog in Fig. 5.5, and the brown cat in Fig. 5.6.

Attribute Leakage In the case of SG, examples include purple on the wall, blue spilled on the plants, and purple on the suitcase in Fig. 5.4, illustrating how attributes emerge as leakage when the respective object is absent. Additional examples include the tomato’s color spilling onto the dog and red metal leaking onto the chair in Fig. 5.5, as well as blue color leaking and forming artifacts in Fig. 5.6. AnE, with its sole focus on intensity, also suffers significantly from attribute leakage, evident in the purple backpack and blue suitcase in Fig. 5.4, the red metal chair in Fig. 5.5, and the blue glasses and earrings, and red towel in Fig. 5.6.

We argue that addressing object neglect or attribute binding in isolation is insufficient, as these issues are intrinsically interconnected. Our method adeptly balances these two concerns with a chosen intensity weight λ , demonstrating its success in addressing the challenges above.

5.5.3 Human Evaluation

Recent work [YBK22, CZF24] has found that large Vision-and-Language Models (VLMs) [RKH21b, SHG22, LLX22, ZZL21] demonstrate a significant lack of compositional understanding, failing to reflect human preferences accurately. Given this, we conducted human evaluations across all three datasets to rigorously assess our model’s performance.

Raters were enlisted online, with the requirement that each participant possessed an educational level of a bachelor’s degree or higher. In the process of evaluation, they were presented with 2-way multiple choice problems consisting of a text prompt and two images generated by our method and one of four baselines, including SD, AnE, SG, and our method with $\lambda = 0$. For each dataset, 100 prompts were randomly sampled for evaluation. The

effectiveness of prompt-image alignment was assessed by asking raters, "Which image better matches the given description?". More details are provided in supplementary material.

The human evaluation results are shown in Fig. 5.7. We observe that: 1) our method consistently surpasses SD, AnE, and SG aligned with quantitative results in Tab. 5.1 and Fig. 5.3; 2) our method shows a more pronounced advantage over other methods on the natural-language ABC-6K dataset. We argue that our object-centric objective, in harmony with the object-oriented patterns prevalent in naturally occurring prompts, exhibits superior efficacy in handling complex real-world, natural-language-based prompts.

5.5.4 Ablation Study

Repulsive Term Tab. 5.2 presents the results of ours($\lambda = 0$) w/o and w/ the repulsive term in rows 1 and 2, and similarly, ours w/o and w/ this term in rows 3 and 4, under the same settings as Tab. 5.1. Row 2/4 demonstrates a significant performance increase than Row 1/3 due to the repulsive term, validating the effectiveness of negative sampling approximation.

Intensity Weight We demonstrate the impact of different choices for the intensity weight λ , which plays a role in enhancing the intensity level. In Fig. 5.8, we present some representative examples where the model needs to generate multiple objects with certain modifiers. When $\lambda = 0.5$, the generation is balanced. However, when $\lambda = 0.0$, all images more or less suffer from object neglect. Conversely, when $\lambda = 1.0$, artifacts are likely to appear and attribute binding becomes less effective.

Object-Conditioned Perspective Besides being able to handle more flexible prompts where no attribute-object pairs exist, our method is object-centric and thus, in the repulsive term, we only calculate the energy of objects and non-modifiers. The way SG handles it is to treat objects and modifiers equivalently when faced with non-modifiers. In the row ‘-

obj cond.’ of Tab. 5.3, we replace the energy of object and non-modifier $f(A_s, A_l)$ with the average energy of $f(A_s, A_l)$ and $f(A_s, A_m)$, where s, l, m represent object, non-modifier, and modifier, respectively. Note that as no modifiers exist in A-A, the results remain the same. In the other datasets, our object-conditioned perspective plays a vital role in the success of our method as the performance significantly decreases.

Energy function Choice To calculate KL div., SG assumes attention maps follow a multinomial distribution. Yet, cosine similarity does not pose any assumption on the distribution and achieves superior performance. In the row ‘- cos sim.’ of Tab. 5.3, we replace cosine similarity with the average KL div..

5.5.5 Augmented Attribute Editing

PtP [HMT23b] allows local editing with word swap, adding new phrase, or re-weighting by replacing the attention weights of unchanged tokens, or re-weighting the attention maps of target tokens. This heavily relies on the semantic coupling between tokens and their attention maps. In Fig. 5.9, we categorize the failure cases in PtP shown in the left panel into four situations: (a) ineffective editing with aligned text-to-image generation; (b) ineffective editing with incorrect attribute binding, e.g. the semantic leakage of ‘pink’; (c) ineffective editing with object neglect, e.g. the ‘apple’; and (d) insignificant editing with aligned text-to-image generation, e.g. the property ‘metal’ for the drum. In contrast, our method effectively enhances the semantic distribution of attention maps, allowing PtP with our approach to apply effective and significant local attribute editing to the original images.

5.6 Conclusion

We introduce an object-conditioned EBAMA framework to address the alignment issues in text-to-image diffusion models. We propose an object-centric attribute binding loss that

maximizes the log-likelihood of the object-conditioned EBM in the attention feature space. An intensity regularizer is further designed to provide an extra degree of freedom balancing the trade-off between correct attribute binding and the necessary presence of objects. Extensive quantitative and qualitative comparisons demonstrate the superiority of our method in aligned text-to-image generation. This advancement promises great improvements in text-controlled attention-based image editing with semantically aligned attention maps.

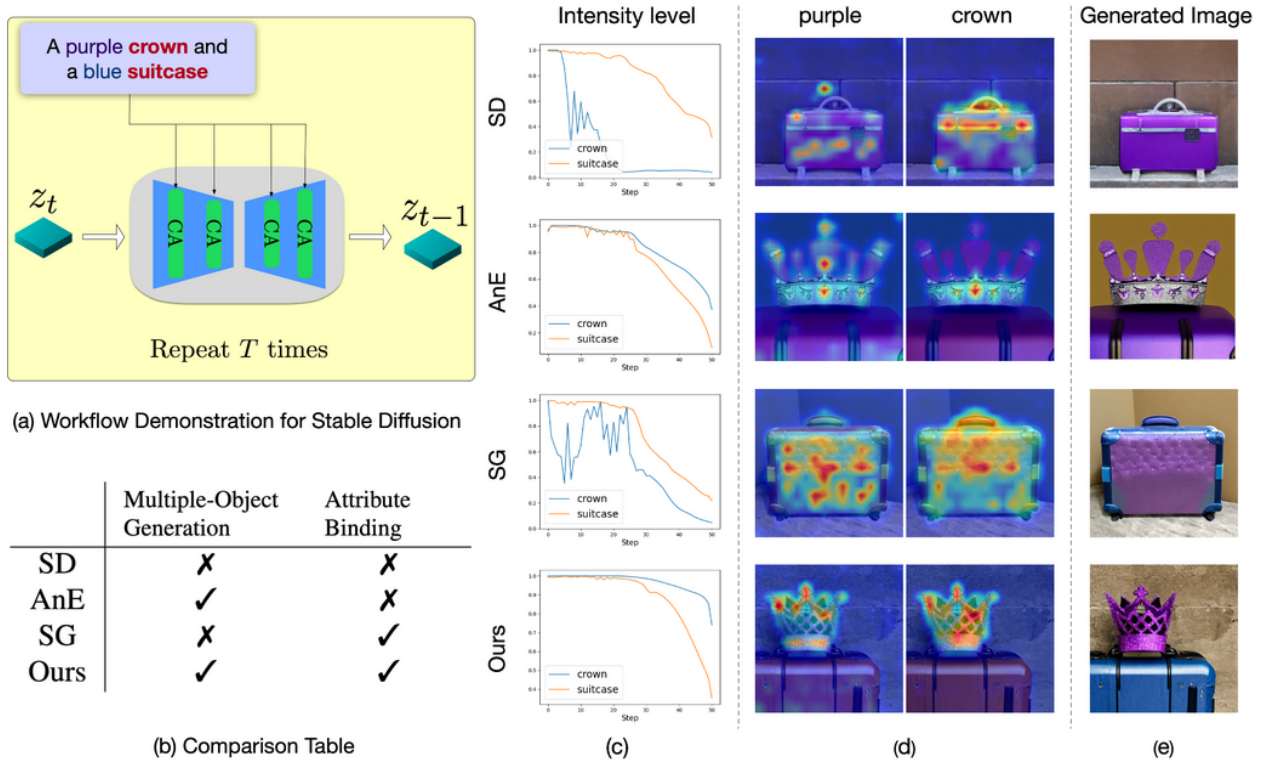


Figure 5.1: **Key observations of the generation process of diffusion models.** The given prompt is “a purple crown and a blue suitcase”. In panel (c), we hypothesize that if the intensity level of any object in the prompt does not remain high during the first half of the denoising process, e.g. the crown in SD and SG, the model would fail to generate the object in the final image. The panel (d) suggests that if the attention map distributions of any attribute-object pair are not aligned, the model would struggle to correctly bind attributes to their respective objects, e.g. ‘purple’ and ‘crown’ in SD and AnE. The generated images are displayed in the panel (e). All methods share the same random seed.

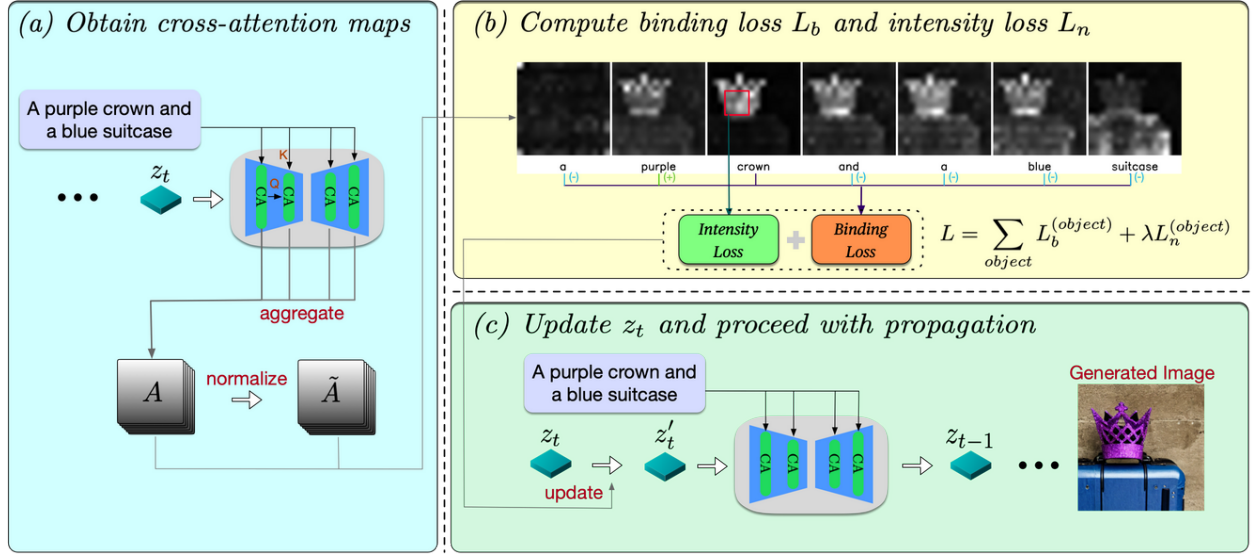


Figure 5.2: **An overview of our workflow for optimizing diffusion models.** It includes aggregation of attention maps, computation of object-centric attention loss, and updates to z_t .

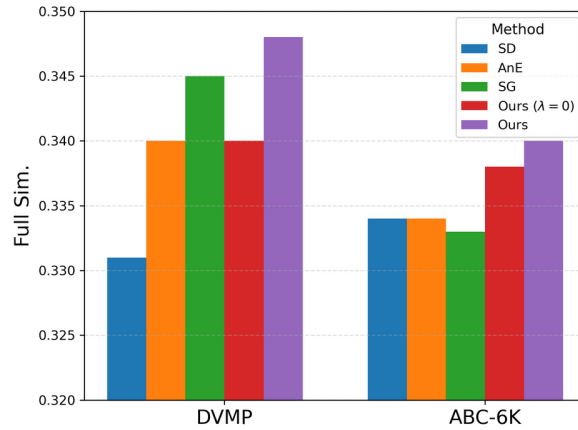


Figure 5.3: **Full Sim. results on DVMP and ABC-6K datasets.** We randomly sample 200 prompts from each dataset and generate 4 images for each prompt.

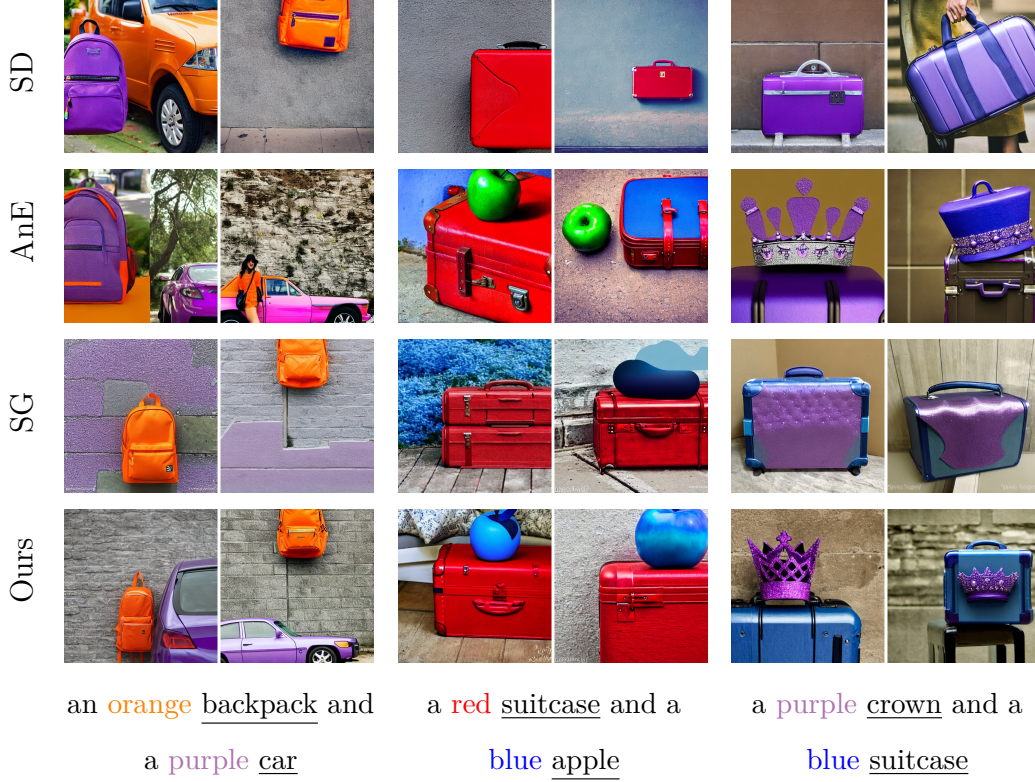
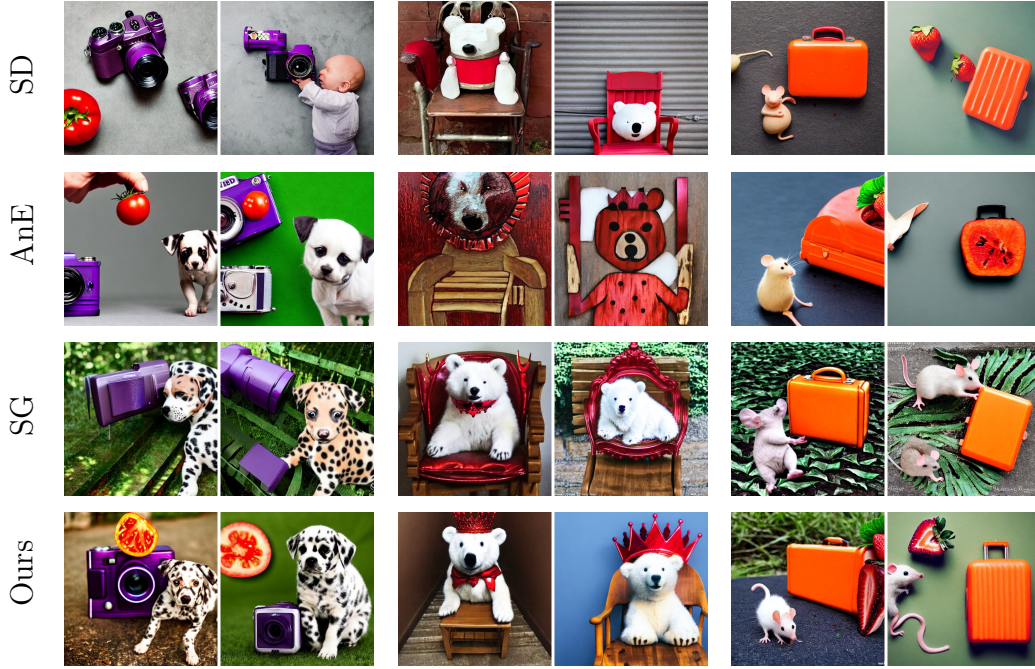


Figure 5.4: **Qualitative comparison on the AnE dataset.** Each column shares the same random seed.

Table 5.2: **Ablation results on repulsive term.** Both Ours and Ours($\lambda = 0$) benefit from the repulsive term as defined in Eqn. (5.5).

Method	Repul.	Animal-Animal			Animal-Object			Object-Object		
		Full Sim.	Min. Sim.	T-C Sim.	Full Sim.	Min. Sim.	T-C Sim.	Full Sim.	Min. Sim.	T-C Sim.
Ours($\lambda = 0$)	✗	0.311	0.213	0.767	0.343	0.246	0.794	0.334	0.237	0.765
	✓	0.340	<u>0.255</u>	<u>0.814</u>	<u>0.362</u>	0.271	0.851	<u>0.360</u>	<u>0.270</u>	<u>0.823</u>
Ours	✗	0.338	0.250	0.810	0.360	0.267	0.841	0.359	0.269	0.819
	✓	0.340	0.256	0.817	0.362	<u>0.270</u>	0.851	0.366	0.274	0.836



a purple modern camera a red metal crown and a an orange suitcase and a
 and a spotted baby dog white bear and a sliced strawberry and a
 and a sliced tomato wooden chair baby mouse

Figure 5.5: **Qualitative comparison on the DVMP dataset.** Each column shares the same random seed.

Table 5.3: **Ablations on object-conditioned perspective and energy function choice.**

	A-A	A-O	O-O
ours($\lambda = 0$)	0.814	0.851	0.823
- obj cond.	0.814 (-0.0%)	0.832 (-2.2%)	0.813 (-1.2%)
- cos sim.	0.812 (-0.2%)	0.846 (-0.6%)	0.817 (-0.7%)
SG	0.767 (-5.8%)	0.830 (-2.5%)	0.811 (-1.5%)

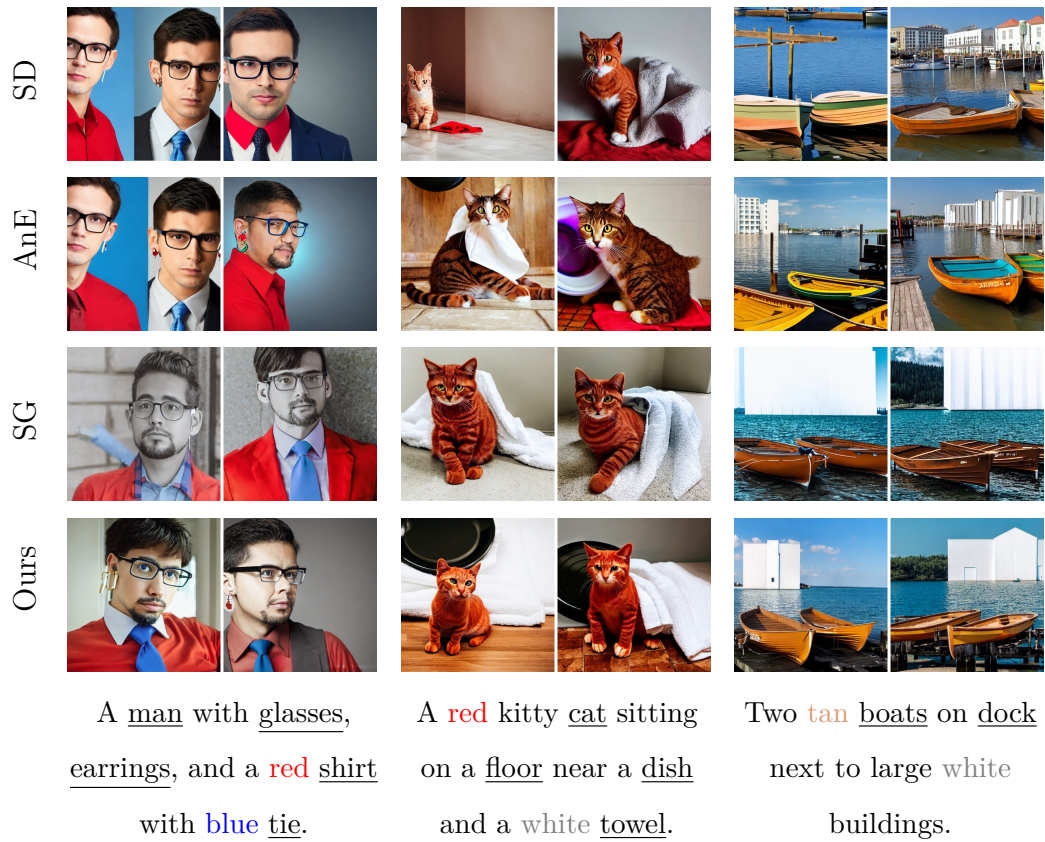


Figure 5.6: **Qualitative comparison on the ABC-6K dataset.** Each column shares the same random seed.

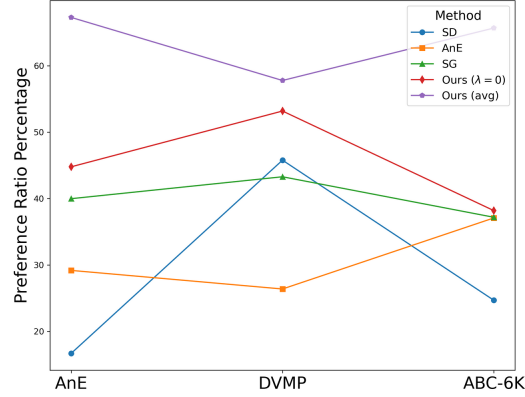


Figure 5.7: **Preference ratio percentage on text-image alignment by human evaluation.** Ours(avg) represents the average preference ratio of our method compared with the other four methods.

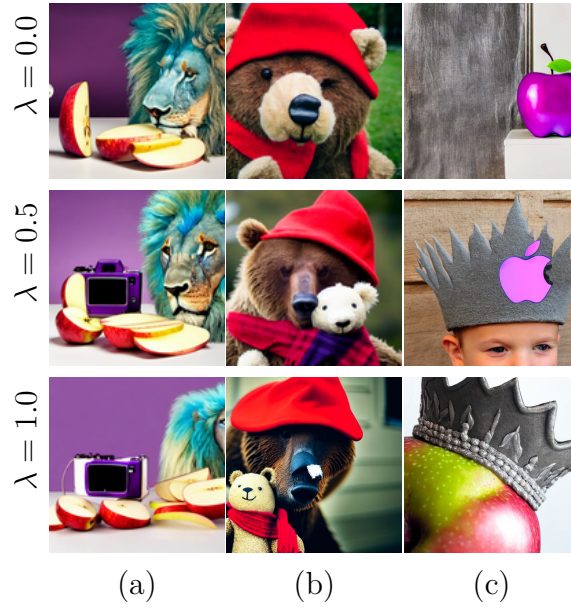
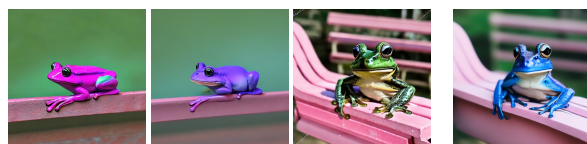


Figure 5.8: **Ablation demonstration for intensity weight λ .** (a) a sliced apple and a purple camera and a teal lion; (b) a brown bear with red hat and scarf and a small stuffed bear; (c) a gray crown and a purple apple. We have selected one prompt from each dataset to showcase the stability of our method. Each column shares the same random seed.



(a) a yellow(→ green) guitar



(b) a (+ blue) frog on a pink bench



(c) a green backpack and a black(→ purple) apple



(d) a dog is playing a leather(→ metal) drum on the beach

Figure 5.9: **Augmented attribute editing with our method.** The left two columns demonstrates the attribute editing results from PtP, while the right two columns demonstrates the results from PtP w. ours.

Part III

Generation

CHAPTER 6

Score Distillation Beyond Acceleration: Generative Modeling from Corrupted Data

6.1 Introduction

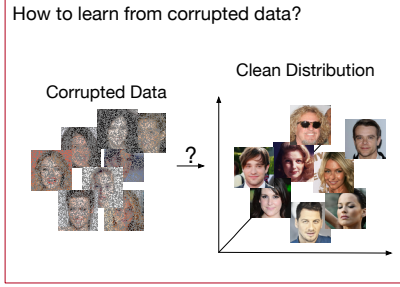
Learning from corrupted data is central to many scientific and engineering domains where clean observations are scarce or costly, including astronomy [Rod88, LGB24], medical imaging [RKA21, JAD21], and seismology [Nol08, RFS14]. For instance, fully sampled MRI acquisitions are time-consuming and uncomfortable for patients [KZS20, ZKS18], motivating methods that recover the structure of the underlying clean distribution from corrupted measurements alone.

Problem Statement. We study *generative modeling from corrupted observations*. Let $x \in \mathbb{R}^d$ be drawn from an unknown clean distribution p_X . We observe only

$$y = \mathcal{A}(x) + \sigma\epsilon, \quad \epsilon \sim \mathcal{N}(0, I_m), \quad (6.1)$$

where $\mathcal{A} : \mathbb{R}^d \rightarrow \mathbb{R}^m$ is a (known) non-invertible corruption operator and σ is the noise level. The operator may be identity (i.e., denoising), a deterministic linear map (blur, downsampling), a random mask (inpainting), or a Fourier-domain undersampling pattern (MRI). Our goal is to learn a generator whose samples follow p_X using only a dataset of N corrupted datapoints $\{y^{(i)}\}_{i=1}^N$ of the form Eq 6.1.

Problem Statement



Our Method: Restoration Score Distillation

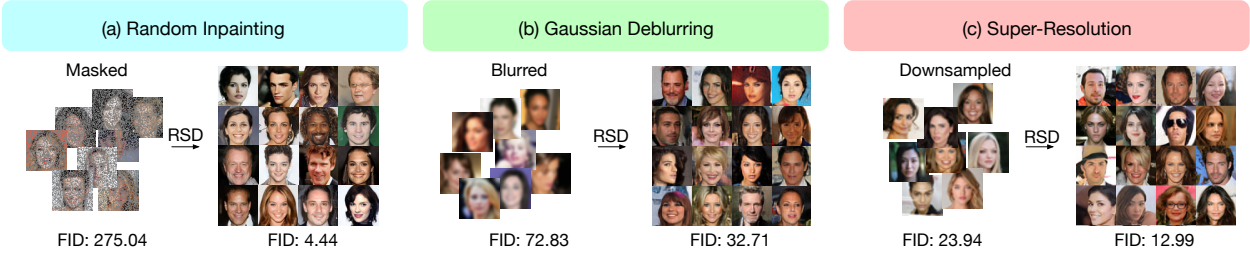
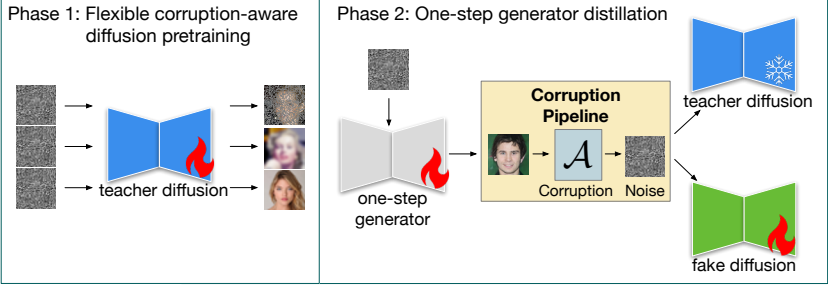


Figure 6.1: Overview of *Restoration Score Distillation* (RSD) and some qualitative results across diverse operators: Gaussian deblurring, random inpainting, and super-resolution.

Background and limitations. Diffusion models [SWM15, HJA20a], also known as score-based generative models [SE19, SSK21], achieve state-of-the-art results in high-dimensional image synthesis [DN21, HSC22, RDN22b, RBL22c, PX23, ZWY24, ZYW25, CZF25, CZZ25]. When only measurements are available, *corruption-aware* training adapts diffusion objectives to the forward operator: Ambient Diffusion for masking [DSD23], Ambient Tweedie for additive noise [DCD25], and Fourier-space variants for MRI [ADL25]. These methods, however, inherit the sampling cost of multi-step reverse processes. EM-Diffusion [BWC24] offers broader operator coverage but requires a few clean images, is computationally expensive, and can struggle under severe corruption.

Distillation beyond acceleration. Score distillation transfers a pretrained diffusion teacher into a one-step generator while largely preserving fidelity [PJB22, WLW24, LHZ24, YGZ24, ZZW24, XXK24, XNV25, YGP24]. Recent reports indicate that distilled generators can even outperform their diffusion teachers [ZZW24], though gains under clean-data

training are typically modest. In contrast, under *corrupted-data-only* training, we observe substantially larger improvements over the teacher (Sec. 6.4), underscoring distillation’s particular advantage in challenging regimes.

Restoration Score Distillation (RSD). We introduce *RSD*, a unified framework for learning high-fidelity one-step generators directly and only from corrupted observations, with denoising ($\mathcal{A}=I$) as a special case. RSD proceeds in two stages: (i) *corruption-aware diffusion pretraining*, where a teacher is trained on measurements using an objective matched to the forward operator $\mathcal{A}$ (Sec. 6.3.1); and (ii) *score distillation*, which transfers the teacher into a single-step generator while explicitly *respecting the measurement operator* during training (Sec. 6.3.2). Concretely, we synthesize measurements by applying the same corruption pipeline to generator outputs and then align the induced generator scores with the teacher’s scores under a divergence (e.g., Fisher or KL). The distillation phase, which includes a corruption-respecting procedure, consistently improves generation quality upon the diffusion teacher across both denoising and more general operators (Fig. 6.1).

Contributions. Our work makes three primary contributions. **Unified framework for diverse corruptions:** We propose *RSD*, a unified approach that learns generators directly from diverse corrupted measurements $y = \mathcal{A}(x) + \sigma\epsilon$. This formulation encompasses denoising ($\mathcal{A} = I$) as well as more general operators, including blur, downsampling, random masking, and Fourier undersampling, under both noisy and noiseless regimes, and achieves state-of-the-art performance across these settings. **Modular training:** Our training pipeline is organized into two phases. Phase I accommodates a variety of corruption-aware techniques, such as standard diffusion, diffusion for denoising, random inpainting, and masked Fourier-space (F.S.) transformations. Phase II distills the teacher model into a one-step generator while retaining the corruption pipeline used during training. This modularity makes it straightforward to plug in new forward operators or training objectives. We also provide theoretical analysis that explains why the distillation phase can enhance generation qual-

ity. **Extensive experiments:** We conduct comprehensive evaluations on natural-image benchmarks (CIFAR-10, CelebA-HQ, FFHQ, AFHQ-v2), restoration tasks (denoising with $\sigma \in \{0.1, 0.2, 0.4\}$, Gaussian deblurring, random inpainting with $p \in \{0.6, 0.8, 0.9\}$, and $2\times$ super-resolution), and multi-coil MRI with acceleration factors $R \in \{4, 6, 8\}$. Across all settings, RSD consistently improves FID over corruption-aware diffusion teachers while offering substantial speedups via one-step generation (Sec. 6.4). Additional ablations (Sec. 6.4.5) on unknown corruption types and different data size further underscore the robustness of our framework. Moreover, we demonstrate that the learned clean-image prior (the generator) can be directly leveraged for downstream conditional inverse problems, achieving good performance (Sec. 6.4.7).

6.2 Background

6.2.1 Diffusion Models

Diffusion models [SWM15, HJA20a], also known as score-based generative models [SE19, SSK21], consist of a forward process that gradually perturbs data with noise and a reverse process that denoises this signal to recover the data distribution $p_X(x)$. Specifically, the forward process defines a family of conditional distributions over noise levels $t \in (0, 1]$, given by $q_t(x_t | x) = \mathcal{N}(\alpha_t x, \sigma_t^2 I)$, with marginals $q_t(x_t)$. We adopt a variance-exploding (VE) process [SE19] by setting $\alpha_t = 1$, yielding the simple form $x_t = x + \sigma_t \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, I_d)$. To model the reverse denoising process, one typically trains a time-dependent denoising autoencoder (DAE) $f_\phi(\cdot, t) : \mathbb{R}^d \times [0, 1] \rightarrow \mathbb{R}^d$ [Vin11], parameterized by a neural network, to approximate the posterior mean $\mathbb{E}[x | x_t]$. This is achieved by minimizing the following standard diffusion loss:

$$\mathcal{L}_{\text{SD}}(\phi; \{x^{(i)}\}_{i=1}^N) := \mathbb{E}_{x, \sigma_t, \varepsilon} [\lambda(t) \|f_\phi(x_t, t) - x\|_2^2] \quad (6.2)$$

where $x_t = x + \sigma_t \varepsilon$ and $\{x^{(i)}\}_{i=1}^N$ denotes the dataset. One can also apply the loss to

Table 6.1: Summary of corruption-aware diffusion objectives used for pretraining. Our framework can be seamlessly integrated with existing advanced corruption-aware diffusion objectives.

Suitable Scenario	Algorithm	Operator	Domain	Pretrain Objective	Notation
Noiseless Corruption	Alg(13)	$\mathcal{A}(x)$	Image	$\mathcal{L}_{\text{SD}} = \mathbb{E}[\lambda(t)\ f_\phi(y + \sigma_t \varepsilon, t) - y\ _2^2]$	y : corrupted data
Noisy Corruption	Alg(14)	$\mathcal{A}(x) + \sigma \epsilon$	Image	$\mathcal{L}_{\text{N}} = \mathbb{E}\left[\left\ \frac{\tilde{\sigma}_t^2 - \sigma^2}{\tilde{\sigma}_t^2} f_\phi(y_t, t) + \frac{\sigma^2}{\tilde{\sigma}_t^2} y_t - y\right\ _2^2\right]$	σ : data noise level, $\tilde{\sigma}_t = \max\{\sigma_t, \sigma\}$
Random Inpainting	Alg(16)	Mx	Image	$\mathcal{L}_{\text{RI}} = \mathbb{E}[\ M(f_\phi(\tilde{M}, \tilde{M}y_t, t) - y)\ _2^2]$	$\tilde{M}$: further corrupted mask M
Masked F.S.	Alg(17)	$M\mathcal{F}x$	Fourier	$\mathcal{L}_{\text{FS}} = \mathbb{E}[\ \mathcal{A}f_\phi(\tilde{M}, \tilde{y}_t, t) - y\ _2^2]$	$\tilde{y}_t$: further corrupted y_t , $\mathcal{A}$ defined in Appx.6.6.6.4

the corrupted data $\{y^{(i)}\}_{i=1}^N$ to directly learn the distribution $p_Y(y)$. For clarity, we omit the diffusion training schedule $\lambda(t)$, $p(\sigma_t)$ and the noise term $\varepsilon \sim \mathcal{N}(0, I_m)$; full details are deferred to Appendix 6.6.6.1, Alg 13. The corresponding objectives are summarized in Table 6.1, with additional discussion below.

6.2.2 Diffusion Models for Corruptions

Recent advances have extended diffusion models to address diverse forms of data corruption. We present several variants of the diffusion loss tailored to different corruption settings, including objectives for noisy data, random inpainting in the image domain and Fourier domain.

Diffusion for Noisy Corruptions. [DCD25] generalizes score matching to noisy observations $y = x + \sigma \varepsilon$, $\varepsilon \sim \mathcal{N}(0, I)$. Here, $\sigma > 0$ is a known noise level in the measurements. One can incorporate the known corruption by minimizing the following diffusion for noisy corruptions

$$\mathcal{L}_{\text{N}}(\phi; \{y^{(i)}\}_{i=1}^N) = \mathbb{E}_{y, \tilde{\sigma}_t, \epsilon} \left[\lambda(t) \left\| \frac{\tilde{\sigma}_t^2 - \sigma^2}{\tilde{\sigma}_t^2} f_\phi(y_t, t) + \frac{\sigma^2}{\tilde{\sigma}_t^2} y_t - y \right\|_2^2 \right], \quad (6.3)$$

where $\tilde{\sigma}_t = \max\{\sigma_t, \sigma\}$, $y_t = y + \tilde{\sigma}_t \epsilon$, and $p(\sigma_t)$ and $\lambda(t)$ arise from the diffusion training schedule. This formula models the distribution of the $x_t := x + \sigma_t \varepsilon$ where $\sigma_t \geq \sigma$. Full

details are deferred to Appendix 6.6.6.2, the training algorithm in Algorithm 14 and the sampling algorithm in Algorithm 15.

Diffusion for Image-Space Random Masked Corruptions. Daras et al. [DSD23] learn $p_X(x)$ from randomly inpainted measurements by incorporating mask into the diffusion objective. Given observations $y = Mx$ for a binary mask M , the teacher is trained with a random-inpainting loss

$$\mathcal{L}_{\text{RI}}(\phi; \{y^{(i)}\}_{i=1}^N) = \mathbb{E}_{(y,M), \tilde{M}, \sigma_t, \varepsilon} \left[\left\| M \left(f_\phi(\tilde{M}, \tilde{M}y_t, t) - y \right) \right\|_2^2 \right], \quad (6.4)$$

where $y_t = y + \sigma_t \varepsilon$ is from diffusion schedule, $\varepsilon \sim \mathcal{N}(0, I)$, and $\tilde{M}$ is a secondary mask that further erases pixels based on M . Details of the diffusion schedule $\lambda(t)$, $p(\sigma_t)$, and the full training procedure are provided in Appendix 6.6.6.3, Algorithm 16.

Diffusion for Fourier-Space Random Masked Corruptions. [ADL25] extends diffusion models to handle frequency-domain measurements with random masking, which are used in scientific imaging (e.g., MRI). Specifically, observations take the form $y = M\mathcal{F}x$, where $\mathcal{F}$ denotes the Fourier transform operator and M is a sampling mask. Formal definitions of the measurement process and the corresponding training algorithm are provided in Appendix 6.6.6.4 and Algorithm 17.

These specialized frameworks enable generative modeling from realistic scenarios such as noisy, missing data, or frequency-domain degradation. All objective functions are summarized in Table 6.1.

6.2.3 Score Distillation for Generative Modeling

Score distillation compresses multi-step diffusion models into efficient one-step generators. Originally proposed for text-to-3D generation [PJB22, WLW24] and later extended to image synthesis [LHZ24, YGZ24, ZZW24, XXK24], it transfers knowledge from a pretrained

diffusion teacher f_ϕ to a generator $G_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$. To bridge the two, a fake diffusion model f_ψ approximates the distribution induced by $G_\theta(\cdot)$ across diffusion noise levels. Training encourages consistency between f_ψ and f_ϕ over time:

$$\mathcal{L}_{\text{distill}}(\theta) = \mathbb{E}_{\sigma_t, z \sim \mathcal{N}(0, I_d)} \mathbb{E}_{x_\theta = G_\theta(z)} [\|f_\phi(x_t, t) - f_\psi(x_t, t)\|_2^2], \quad (6.5)$$

where $x_t = x_\theta + \sigma_t \epsilon$ and the loss corresponds to Fisher divergence, following SiD [ZZW24]. This divergence of true and fake distribution can be replaced by KL divergence; ablation results are reported in Section 6.4.5 and Appendix 6.6.3.5. Intuitively, the fixed teacher f_ϕ represents the true data distribution, while f_ψ captures the generator’s induced distribution. Updating G_θ to minimize $\mathcal{L}_{\text{distill}}(\theta)$ aligns generator samples with the true data. Notably, distilled generators can even outperform their teachers: for instance, [ZZW24] reports that on FFHQ the teacher achieves an FID of 2.39, whereas its distilled generator attains 1.55. In our setting, we find that such improvements are further amplified when distillation is performed under corrupted-data training. Details of the distillation training schedules are deferred to Appendix 6.6.7.

6.3 Restoration Score Distillation

Problem Statement. Suppose we are given a finite corrupted dataset of size N , denoted by $\{y^{(i)}\}_{i=1}^N$. Each corrupted observation is generated as $y^{(i)} = \mathcal{A}(x^{(i)}) + \sigma \varepsilon^{(i)}$, where σ is a known noise level (with extensions available for the unknown- σ setting, see Section 6.4.5), and $\varepsilon^{(i)} \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_m)$. Crucially, the clean data $\{x^{(i)}\}_{i=1}^N$ is never accessible. In certain scenarios, such as random inpainting, the corruption operator $\mathcal{A}$ may vary across samples, drawn from a common distribution. In this case, each corrupted observation takes the form $y^{(i)} = \mathcal{A}^{(i)}(x^{(i)}) + \sigma \varepsilon^{(i)}$.

RSD is a two-phase framework for learning generative models solely from corrupted observations. It decouples the process into: (1) a flexible corruption-aware diffusion pretraining stage, and (2) a distillation stage that compresses the pretrained model into a single-step

generator and further improves generation quality. The overall procedure is summarized in Algorithm 12. We summarize different types of pretrained diffusion models in Table 6.1, and then describe how they can be seamlessly integrated into RSD to further enhance performance. Notably, even when a suitable corruption-aware pretrained model is unavailable, employing a standard diffusion model (Tab 6.1) to learn the corrupted data distribution still yields substantial improvements after distillation (Tab 6.4).

6.3.1 Phase I: Flexible Corruption-Aware Pretraining

The first phase trains a teacher diffusion model $f_\phi(\cdot, t)$ directly on corrupted observations $y = \mathcal{A}(x) + \sigma\epsilon$, where $\mathcal{A}$ is a potentially non-invertible corruption operator. A straightforward approach is to train on corrupted data y using the standard diffusion loss (Eq. 6.2). In practice, we find that this naive strategy already yields strong performance. To further improve performance across diverse corruption operators, we incorporate existing corruption-aware diffusion training methods, as summarized in Table 6.1. Our choice of pretraining objectives for different corruption scenarios is as follows: 1) **Standard Diffusion** is effective under **noiseless corruptions**, directly modeling the distribution of $y = \mathcal{A}(x)$. 2) **Noisy Corruption** [DCD25] targets additive noise settings, learning the distribution of $\mathcal{A}(x)$ from noisy observations $y = \mathcal{A}(x) + \sigma\epsilon$, with denoising as a special case when $\mathcal{A} = I$. 3) **Random Inpainting** [DSD23] addresses learning from partial observations Mx , where M is a random inpainting mask, and learns the distribution of x . 4) **Masked F.S** [ADL25] extends inpainting to Fourier-domain corruptions, well-suited for frequency-based degradation such as accelerated MRI, learning from masked Fourier observations $M\mathcal{F}x$.

6.3.2 Phase II: One-Step Generator Distillation

After pretraining, we distill the teacher diffusion model $f_\phi(\cdot, t)$ into a one-step generator G_θ that maps latent noise $z \sim \mathcal{N}(0, I_d)$ directly to clean samples. To facilitate this distillation, we introduce an auxiliary fake diffusion model $f_\psi(\cdot, t)$, initialized from f_ϕ and trained on

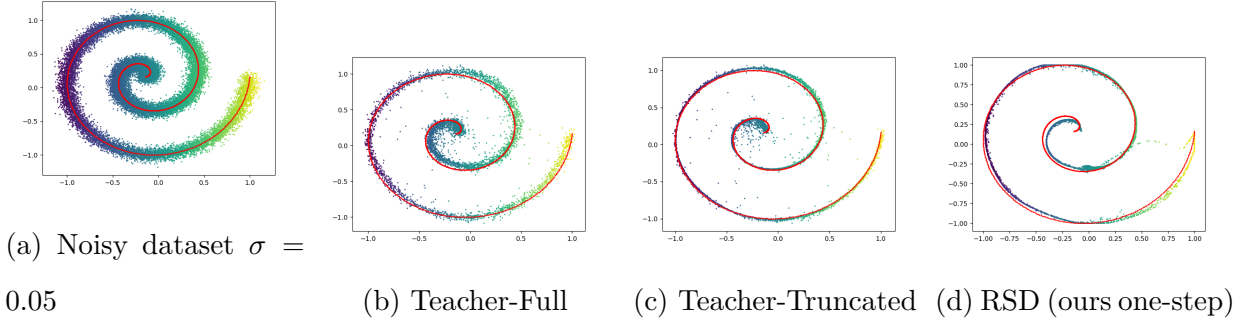


Figure 6.2: **A toy example of learning from a noisy dataset with $\sigma = 0.05$.**

corrupted output of $G_\theta(z)$, which is also initialized from f_ϕ . Note that this is common practice in the distillation literature to facilitate and stabilize training. In this case, G_θ, f_ϕ, f_ψ share the same network structure; see Appendix 6.6.7 for further discussion. By encouraging f_ψ to match the teacher’s time-dependent dynamics (e.g., score fields parameterized by f_ϕ), we align the behavior of G_θ with the data distribution learned by f_ϕ , thereby narrowing the gap between the generated and clean data distributions.

We adopt the SiD [ZZW24] framework for distillation and optimize the distillation loss $\mathcal{L}_{\text{distill}}(\theta)$ (Eq. 6.5). An ablation of loss choices is provided in Sec. 6.4.5. Beyond standard score distillation, we further corrupt the generated samples $x_g = G_\theta(z)$ into $\tilde{y}$ using the same corruption pipeline as the training data (Algorithm 12, Line 6). These corrupted samples are then used to train a fake diffusion model $f_\psi(\cdot, t)$. In Algorithm 12, Lines 2 and 7 allow any of the pretraining objectives summarized in Table 6.1 (or new objectives tailored to specific corruptions), but *the same* objective must be used for both lines; mixing objectives destabilizes training and can lead to divergence. To complement the empirical results in Section 6.4, Appendix 6.6.5 offers a theoretical analysis establishing conditions under which distillation yields improved sample quality.

Algorithm 12 Restoration Score Distillation (RSD)

```

1: procedure Restoration Score Distillation( $\{y^{(i)}\}_{i=1}^N$ )
2:    $f_\phi \leftarrow \text{DiffusionTraining}(\{y^{(i)}\}_{i=1}^N, \text{Obj})$   $\triangleright$  Diffusion training with  $\text{Obj} \in \text{Tab. 6.1}$ 
3:   Initialize  $f_\psi \leftarrow f_\phi, G_\theta \leftarrow f_\phi$ 
4:   for  $j = 1, \dots, K$  do
5:      $x_g \leftarrow G_\theta(z), \quad z \sim \mathcal{N}(0, I_d)$   $\triangleright$  Generate fake clean images
6:      $\tilde{y} = \mathcal{A}(\text{stopgrad}(x_g)) + \sigma \varepsilon, \varepsilon \sim \mathcal{N}(0, I_d)$   $\triangleright$  Corrupt  $x_g$  same way as observation
7:     # If corruption contains noise, choose an Obj for  $f_\phi, f_\psi$  that is denoising-aware
8:      $f_\psi \leftarrow \text{DiffusionTraining}(\{\tilde{y}^{(i)}\}_{i=1}^N, \text{Obj})$   $\triangleright$  Same Obj as above
9:     # Do NOT inject additional noise during distillation when updating  $\theta$ 
10:     $y_\theta \leftarrow \mathcal{A}(G_\theta(z)), \quad z \sim \mathcal{N}(0, I_d)$ 
11:    Update  $\theta$  by distillation loss  $\mathcal{L}_{\text{distill}}(\theta)$  with generated  $x_\theta = y_\theta$   $\triangleright$  Eq. 6.5
12:  end for
13: end procedure

```

6.4 Experiments

We conduct comprehensive experiments to evaluate the effectiveness, flexibility, and generality of our framework, RSD, across both natural and scientific imaging domains. We first demonstrate its performance through a 2D toy task and qualitative visualizations, providing intuition into the underlying mechanism. Then we consider four challenging image restoration tasks—denoising, Gaussian deblurring, random inpainting, and super-resolution—on CIFAR-10 (32×32), FFHQ (64×64), and CelebA-HQ (64×64) [LLW15a, KAL17a], under both noiseless ($\sigma = 0.0$) and noisy ($\sigma = 0.2$) regimes. We further show that RSD can be distilled directly from pretrained teacher models trained with corruption-aware objectives, including diffusion for denoising [DCD25], diffusion for random inpainting [DSD23], and diffusion for Fourier-space inpainting [ADL25], highlighting the framework’s modularity and flexibility. Sec. 6.4.2 benchmarks denoising across multiple datasets and noise levels. Sec. 6.4.3 evaluates general corruption operators in both noiseless and noisy settings, including a sweep over random-inpainting missing rates $p \in \{0.6, 0.8, 0.9\}$.

To assess real-world applicability, we evaluate RSD on multi-coil Magnetic Resonance Imaging (MRI) using the FastMRI dataset [ZKS18, KZS20], a setting where fully sampled ground truth is often unavailable. Across tasks and corruption regimes, RSD consistently surpasses teacher diffusion models in generative quality as measured by FID [HRU17] without access to clean data (Sec. 6.4.4).

Finally, RSD exhibits strong *distillation–data efficiency* (Sec. 6.4.6)—it quickly surpasses the teacher—and offers substantial inference-time gains from its one-step design (up to $30\times$; see Sec. 6.4.6). Our ablations (Sec. 6.4.5) show that the framework accommodates unknown noise levels σ and is robust with respect to the training-data size; we also examine alternative distillation losses in an ablation study. Separately, Sec. 6.4.7 demonstrates that the trained generator achieves superior performance on downstream conditional inverse problems, indicating broad applicability.

6.4.1 A 2D Toy Example

In Fig. 6.2, the observation is a noisy spiral dataset with $\sigma = 0.05$. The core issue with teacher diffusion models trained on noisy datasets, i.e., (b) Teacher-Full and (c) Teacher-Truncated ([DDD24], Algorithm 15), is that they force the approximating distribution to spread its probability mass across all regions, making the learned density overly diffuse. In contrast, (d) RSD excels at denoising the original dataset, producing a narrow, concentrated, and sharp approximation. This effect occurs *without* providing the clean data itself during training.

6.4.2 Denoising as a Special Case of Restoration Score Distillation

We begin with the denoising special case $y = x + \sigma\epsilon$ where the forward operator $\mathcal{A} = I$. We compare RSD against three groups of baselines (Table 6.2). **(1) Teacher diffusion models.** A teacher trained with the noisy corruption loss in Eq. 6.3 serves as a strong generative baseline. We adopt the two sampling schedules of [DDD24]—**Teacher-Full** and **Teacher-Truncated** (Algorithm 15)—and additionally evaluate the **Teacher-consistency** variant [DCD25]. There may exist better sampling schedules for denoising; however, to the best of our knowledge, the schedules we adopt achieve strongest performance reported in prior work. **(2) Few-shot methods.** **EM-Diffusion** [BWC24] alternates DPS-based reconstructions (E-step) with model refinement (M-step), while **SFBD** [LWY25] casts the problem as density deconvolution; both use 50 clean images for initialization. **(3) Clean-data diffusion (upper bound).** **DDPM** [HJA20a], **DDIM** [SME21], and **EDM** [KAA22] are trained on clean data ($\sigma=0$) and serve as upper bounds for any method trained purely on corrupted observations. Across CIFAR-10 and CelebA-HQ at $\sigma=0.2$, RSD (one-step) outperforms all zero-shot and few-shot baselines and improves over its teachers (Table. 6.2). For FFHQ and AFHQ-v2 where prior few-shot results are not reported, RSD also surpasses teacher models (Table 6.3). Finally, we sweep noise levels $\sigma \in \{0.1, 0.2, 0.4\}$ on CIFAR-10 and

Table 6.2: **Denoising results on CIFAR-10 and CelebA-HQ at $\sigma = 0.2$.** Rows with $\sigma=0.0$ are clean-data upper bounds. Few-shot methods use 50 clean images. Baseline numbers are from the original papers or [BWC24, LWY25]. The distilled student (RSD, one-step) improves over teacher models (Full/Truncated/Consistency) and surpasses all few-shot baselines.

Methods	CIFAR-10 (32×32)			CelebA-HQ (64×64)		
	σ	Type	FID	σ	Type	FID
DDPM [HJA20a]	0.0	Full-Shot	4.04	0.0	Full-Shot	3.26
DDIM [SME21]	0.0	Full-Shot	4.16	0.0	Full-Shot	6.53
EDM [KAA22]	0.0	Full-Shot	1.97	–	–	–
SURE-Score [AAK23]	0.2	Few-Shot	132.61	–	–	–
Ambient Diffusion [DSD23]	0.2	Few-Shot	114.13	–	–	–
EM-Diffusion [BWC24]	0.2	Few-Shot	86.47	–	–	–
TweedieDiff [DDD24]	0.2	Few-Shot	65.21	0.2	Few-Shot	58.52
SFBD [LWY25]	0.2	Few-Shot	13.53	0.2	Few-Shot	6.49
TweedieDiff [DDD24]	0.2	Zero-Shot	167.23	0.2	Zero-Shot	246.95
Teacher-Full [DCD25]	0.2	Zero-Shot	60.73	0.2	Zero-Shot	61.14
Teacher-Truncated [DCD25]	0.2	Zero-Shot	12.21	0.2	Zero-Shot	13.90
Teacher-Consistency [DCD25]	0.2	Zero-Shot	11.93	0.2	Zero-Shot	12.97
RSD (Ours, One-Step)	0.2	Zero-Shot	4.77	0.2	Zero-Shot	6.48

observe consistent gains over teachers. Quality examples are provided in the supplementary material of the source paper.

6.4.3 General corruptions mixed with noise

We now move beyond denoising to the general measurement model $y = \mathcal{A}(x) + \sigma\epsilon$, where $\mathcal{A}$ may be non-invertible. Using CelebA-HQ as a running example, we study both noiseless ($\sigma = 0.0$) and noisy ($\sigma = 0.2$) regimes and instantiate $\mathcal{A}$ as (i) Gaussian deblurring, (ii)

Table 6.3: **RSD vs. teachers on CIFAR-10/FFHQ/AFHQ-v2**. RSD (distilled) consistently surpasses teacher diffusion (*Full*, *Truncated*) across datasets and noise levels. (FID).

Methods	CIFAR-10			FFHQ	AFHQ-v2
Data noise	$\sigma=0.1$	$\sigma=0.2$	$\sigma=0.4$	$\sigma=0.2$	$\sigma=0.2$
Observation	73.74	127.22	205.52	110.83	51.51
Teacher-Full	25.55	60.73	124.28	41.52	17.93
Teacher-Truncated	7.55	12.21	22.12	14.67	9.82
RSD (Distilled)	3.98	4.77	21.63	6.29	5.42

Table 6.4: **CelebA-HQ restoration under noiseless/noisy settings**. Baselines are taken from original papers when available, otherwise reproduced. EM-Diffusion uses 50 clean images for initialization, while RSD uses none. Best results are highlighted.

Methods	σ	Gaussian Deblurring	Random Inpainting ($p = 0.9$)	Super-Resolution ($\times 2$)
Observation	0.0	72.83	396.14	23.94
<u>Teacher Diffusion</u>		<u>94.40</u>	<u>25.53</u>	<u>23.28</u>
EM-Diffusion (<i>Few-Shot</i>)		56.69	104.68	58.99
RSD (Ours, One-Step)		31.90	16.86	12.99
Observation	0.2	264.37	419.92	200.04
<u>Teacher Diffusion</u>		<u>99.19</u>	<u>319.34</u>	<u>23.92</u>
EM-Diffusion (<i>Few-Shot</i>)		51.33	165.60	57.31
RSD (Ours, One-Step)		76.98	79.48	22.00

random inpainting, and (iii) $2\times$ super-resolution. Aggregate results across tasks and noise levels are reported in Table 6.4. For random inpainting, we further sweep the missing rate $p \in \{0.6, 0.8, 0.9\}$; see Table 6.5. Across settings, RSD—trained *without* clean images—consistently improves over its teacher diffusion models and is competitive with, or exceeds, few-shot methods that rely on clean initialization. Qualitative examples are provided in the supplementary material of the source paper.

Table 6.5: **CelebA-HQ random in-painting vs. missing rate p .** Teacher diffusion is trained as in [DSD23].

Method	$p = 0.6$	$p = 0.8$	$p = 0.9$
Observation	275.04	383.82	396.14
Teacher Diffusion	6.08	11.19	25.53
RSD	4.44	7.10	16.86

Table 6.6: **FID across acceleration levels R in multi-coil MRI.** Teacher diffusion is trained as in [ADL24].

Method	$R = 2$	$R = 4$	$R = 6$	$R = 8$
L1-EDM	18.55	27.64	51.43	102.98
Teacher Diffusion	30.34	32.31	31.50	48.15
RSD	12.95	10.71	14.64	22.51

Table 6.7: Comparison of RSD with teacher diffusion and baselines on CIFAR-10. Metrics for DDPM and Rectified Flow are from [LGL22b].

Method	σ	FID↓	IS↑	Prec.↑	Rec.↑	KID↓
Teacher	0.2	12.21	8.31	0.59	0.41	.0059
RSD	0.4	21.63	7.93	0.53	0.38	.0127
RSD	0.2	4.77	9.16	0.65	0.56	.0025
RSD	0.1	3.98	9.34	0.64	0.57	.0015
DDPM	0.0	3.21	9.46	N/A	0.57	N/A
RF (ODE)	0.0	2.58	9.60	N/A	0.57	N/A

6.4.4 Beyond natural images: Multi-coil Magnetic Resonance Imaging (MRI)

We next apply RSD to a practical medical-imaging setting where fully sampled data are often unavailable due to time and cost constraints [KZS20, ZKS18, TDT23, DSR22]. This case study demonstrates: (i) the flexibility of RSD to integrate with advanced corruption-aware diffusion techniques, (ii) robustness across acceleration factors R , and (iii) an extension from real- to complex-valued signals, $x \in \mathbb{C}^d$. Table 6.6 reports FID versus acceleration $R \in \{2, 4, 6, 8\}$ comparing RSD, a teacher trained via **Fourier-Space Ambient Diffusion** [ADL25], and a baseline **L1-EDM** that trains EDM [KAA22] on L1-Wavelet reconstructions [LDP07]. Across all acceleration levels, RSD improves over the teacher; notably, it also outperforms L1-EDM at low acceleration ($R=2$), suggesting that distillation-based

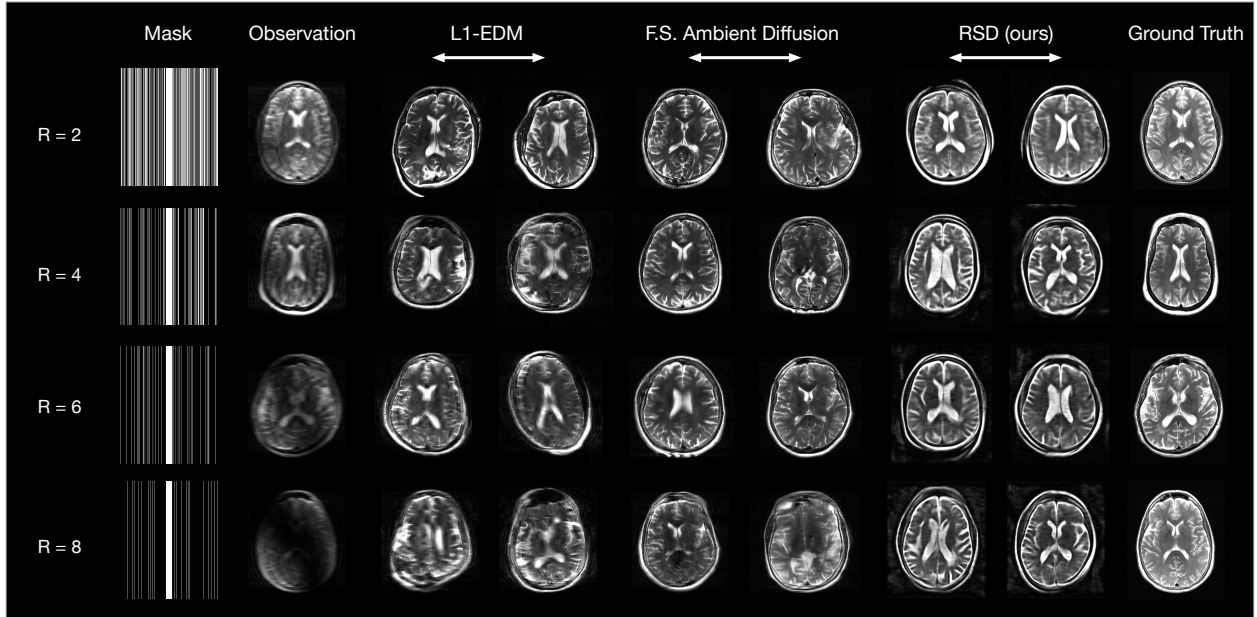


Figure 6.3: **Multi-coil MRI:** qualitative reconstructions from **RSD** across acceleration levels R . See Tab. 6.6 for corresponding FID trends.

Table 6.8: **Data-size ablation on CelebA-HQ.**

Data Size	Teacher-Full	Teacher-Trunc.	RSD
10%	62.25	14.36	10.53
50%	56.09	17.19	9.76
100%	61.14	13.90	6.48

regularization is more effective than handcrafted L1 priors in the wavelet domain. Qualitative examples are shown in Fig. 6.3, with full algorithmic details in Appendix 6.6.6.4 and 6.6.7.

6.4.5 Ablations: Data Scale, Distillation Loss Choice, and More

Data scale. We next vary the training set size on CelebA-HQ to test data efficiency. RSD maintains strong performance even with 10% of the data and improves as more data become available, outperforming both Teacher-Full and Teacher-Truncated across settings.

The results are shown in Table 6.8.

Comprehensive data quality metrics. While FID is the primary metric for assessing data quality in image generation, we incorporate additional metrics for a more holistic evaluation, as detailed in Table 6.7. The results demonstrate that our distilled RSD generator achieves both superior density modeling (low FID) and robust mode coverage (high Recall), even when compared to generative models trained on clean datasets.

Choice of distillation loss. We compare several popular distillation *losses* used to compress diffusion teachers into one-step generators, including SDS [PJB22], DMD [YGZ24], and SiD [ZZW24]. Unless otherwise noted, we use the *default* hyperparameters from the original papers (and public repos) without task-specific tuning. Under these settings, SiD consistently yields the strongest FID across our datasets, while SDS and DMD underperform. We report the full per-dataset and per- σ breakdown in Table 6.18 (Appendix 6.6.3.5). We emphasize that we did not perform a hyperparameter sweep; consequently, better-tuned configurations of SDS or DMD may close the gap to SiD. Empirically, we find SiD’s default hyperparameters at different setting is a stable choice in our RSD pipeline.

More ablations can be found in the Sec. 6.6.3 of the Appendix.

6.4.6 Sampling and Training Efficiency for Distillation

After distillation, RSD attains markedly higher inference throughput than its diffusion teacher. On CIFAR-10 at $\sigma = 0.2$, producing 50,000 samples drops from **10 minutes** (teacher) to **20 seconds** (generator), a $\sim 30\times$ reduction in wall-clock time. Training is likewise efficient: the *Phase II* distillation stage surpasses the teacher’s FID in roughly **4 hours**, compared to **48 hours** for *Phase I* teacher pretraining. Thus, once the teacher is available, running RSD adds only a minor computational budget yet yields substantial quality and speed gains. Figure 6.4 summarizes both sampling and training efficiency. All wall-clock

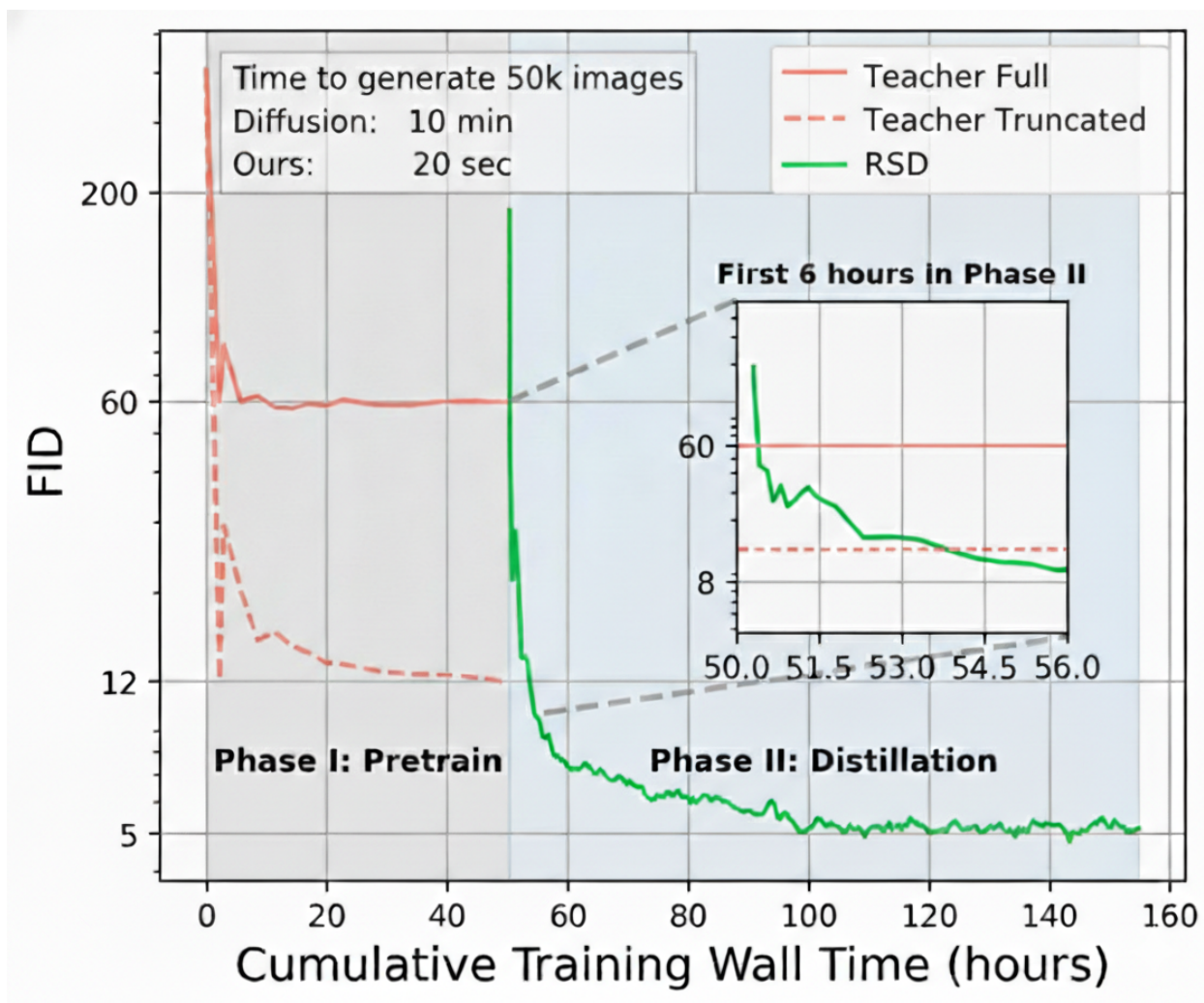


Figure 6.4: Efficiency of RSD (CIFAR-10, $\sigma = 0.2$).

Table 6.9: Conditional inverse problem results of denoising on CIFAR10 at $\sigma = 0.2$. Results for the baselines are taken from [BWC24]. We follow [BWC24] and sample 250 test images and compute the average PSNR and LPIPS.

Method	Type	PSNR $\uparrow$	LPIPS $\downarrow$
Observations		18.05	0.047
DPS w/ clean prior [CKM22]	Full-Shot	25.91	0.010
SURE-Score [AAK23]	Few-Shot	22.42	0.138
AmbientDiffusion [DSD23]	Few-Shot	21.37	0.033
EM-Diffusion [BWC24]	Few-Shot	23.16	0.022
Noise2Self [BR19]	Zero-Shot	21.32	0.227
RSD (ours)	Zero-Shot	24.11	0.025

times were measured on $8\times$ RTX A6000 (32 GB). Additional quantitative statistics and efficiency examples are provided in Appendix 6.6.2 and Table 6.12.

6.4.7 After Distillation: Solving Conditional Inverse Problems

Because our framework yields a high-quality clean image generator—which naturally serves as a prior in conditional inverse solvers [CKM22, ZYZ24, ZDZ24, ZL25, HWD25]—a direct extension is to evaluate its utility on downstream inverse problems. We report a denoising task performance in Table 6.9, where RSD substantially outperforms prior methods and achieves performance comparable to few-shot approaches such as EM-Diffusion. In our implementation, we solve $\min_z \|\mathcal{A}(G_\theta(z)) - y\|_2^2$ for 1000 steps using Adam [Kin14] with a learning rate of 0.05. Exploring alternative strategies for inverse problems with one-step generators is an exciting direction. Additional results are provided in Appendix 6.6.3.3.

6.4.8 Model Selection Criterion under Corruption: Proximal FID

Prior baselines [DSD23, BWC24] report FID scores yet do not specify how to perform *model selection* under corruption during training. To address this gap, we introduce *Proximal FID*,

a model-selection metric tailored to corrupted-data regimes. Concretely, we generate 50k clean samples $\{x^{(i)}\}_{i=1}^{50k}$ from the current generator, corrupt them to match the training noise and operator—yielding $\{\mathcal{A}(x^{(i)}) + \sigma \epsilon^{(i)}\}_{i=1}^{50k}$, and compute FID against the corrupted training set $\{y^{(i)}\}_{i=1}^n$. As shown on CIFAR-10 in Fig. 6.5 and Tab. 6.10, Proximal FID tracks the true FID closely throughout distillation and attains near-optimal *true* FID across datasets. More results can be found in the Appendix. Taken together, these results support Proximal FID as a practical and reliable proxy for model selection when clean data are unavailable.

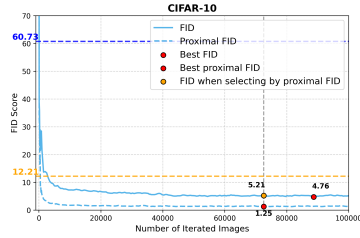


Figure 6.5: **CIFAR-10 (denoising, $\sigma = 0.2$):** Proximal FID closely tracks true FID and selects a near-optimal checkpoint.

Table 6.10: **Model selection by Proximal FID.** Best true FID obtained by selecting checkpoints via Proximal FID vs. selecting via true FID (oracle). The red (+·) indicates the gap to the best achievable true FID during training.

Dataset	Proximal FID (selected)	Best true FID (oracle)
CIFAR-10	5.21 (+0.45)	4.76
FFHQ	6.12 (+0.04)	6.08
CelebA-HQ	6.90 (+0.54)	6.36
AFHQ-v2	5.45 (+0.06)	5.39

6.4.9 Why Score Distillation Beyond Acceleration?

Score distillation is traditionally viewed as a means of *accelerating* sampling by compressing a multi-step diffusion process into a single forward pass. However, our paper aims to reveal a more fundamental **conceptual shift**: In corrupted or low-quality data regimes, score distillation serves as *a principled mechanism* for improving the sample quality of teacher diffusion models.

Intuitive Understanding: Diffusion models are trained using the ELBO objective for maximum likelihood, which minimizes the *forward* KL divergence, a mode-covering objective. As a result, teacher diffusions are incentivized to place probability mass on *all* plausible

locations suggested by noisy data, even if these regions do not correspond to the true clean distribution.

In contrast, the *reverse* distillation objective (Eq. 6.5) takes its expectation over the *student generator’s* distribution, which naturally induces *mode-seeking* behavior: the generator does not need to cover all regions where the teacher assigns non-zero mass. Instead, it is encouraged to place probability only where it will actually sample from. Importantly, we stress that *encouraging mode seeking does not imply mode collapse*. This is also evidenced by the FID and Recall in Tab. 6.7, qualitative results in Fig. 6.2, and extensive qualitative examples in Appendix 2.5.4. This change in objective allows the one-step generator to concentrate its probability mass on the *high-density, well-supported regions* implied by the teacher’s score field, while discarding diffuse or low-density regions that the teacher includes.

Theoretical Support: Detailed analysis is provided in the appendix. Sec. 6.6.5.1 first focuses on a linear Gaussian setting, providing a direct quantitative analysis and error bounds (Eq. 6.9). These findings are extended to general linear corruption in Sec. 6.6.5.2. Sec. 6.6.5.3 derives performance bounds for the distilled generator under a more general setting. We posit that a deeper theoretical analysis of the surprising effects of score distillation under corruption is a promising research direction.

6.5 Conclusion

In this work, we introduced Restoration Score Distillation (RSD) to learn clean data distribution from a broad class of corruption types. Our empirical results on natural images and scientific MRI datasets show consistent improvements over existing baselines. Moreover, beyond standard diffusion objectives, the RSD framework is compatible with several corruption-aware training techniques, enabling flexible integration with recent advances in diffusion modeling. Together, our contributions highlight the potential of score distillation

as a powerful mechanism for robust generative learning in real-world settings where clean data are scarce or unavailable. A detailed discussion of limitations is provided in Appendix 6.6.4.

6.6 Appendix

6.6.1 Proximal FID

When only corrupted data are available, model selection must proceed without clean references. In standard image generation, FID is the de facto criterion, but in our setting computing FID against clean ground-truth images is infeasible.

Prior baselines—Ambient Diffusion [DSD23] and EM-Diffusion [BWC24]—report FID scores yet do not specify how to perform *model selection* under corruption during training. To address this gap, we introduce *Proximal FID*, a model-selection metric tailored to corrupted-data regimes. Concretely, we generate 50k clean samples $\{x^{(i)}\}_{i=1}^{50k}$ from the current generator, corrupt them to match the training noise and operator—yielding

$$\{\mathcal{A}(x^{(i)}) + \sigma \epsilon^{(i)}\}_{i=1}^{50k},$$

and compute FID against the corrupted training set $\{y^{(i)}\}_{i=1}^n$. As shown on CIFAR-10 in Fig. 6.5, Proximal FID tracks the true FID closely throughout distillation. Quantitatively, Table 6.10 shows that the model chosen by Proximal FID attains near-optimal *true* FID across datasets (e.g., 6.12 vs. best 6.08 on FFHQ) for denoising task, and Table 6.11 for general corruption task. We further visualize the dynamics on FFHQ, CelebA-HQ, and AFHQ-v2 in Fig. 6.6, and extend the analysis to multiple corruption operators in Fig. 6.7. Taken together, these results support Proximal FID as a practical and reliable proxy for model selection when clean data are unavailable.

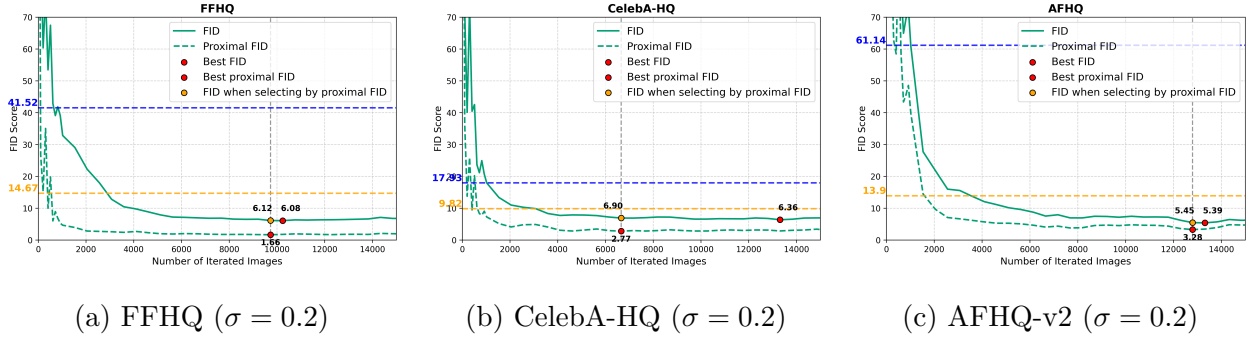


Figure 6.6: **Across datasets:** True FID vs. *Proximal FID* during distillation on FFHQ, CelebA-HQ, and AFHQ-v2 (denoising, $\sigma = 0.2$). In all cases, Proximal FID reliably identifies checkpoints with near-optimal true FID.

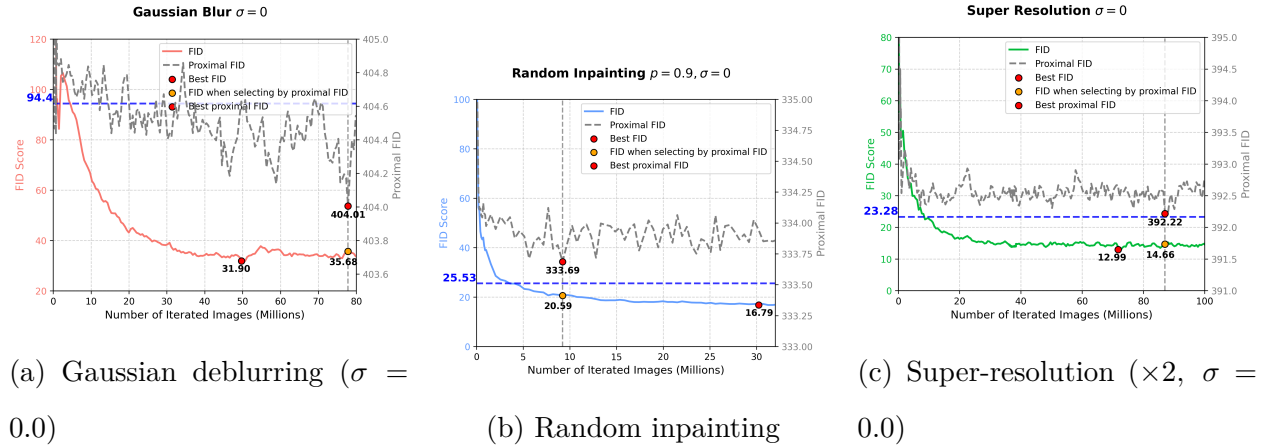


Figure 6.7: **Across corruption operators:** True FID vs. *Proximal FID* during training for Gaussian deblurring, random inpainting, and super-resolution. Proximal FID consistently tracks the true FID and supports reliable model selection across operators.

Table 6.11: Comparison between the best true FID and the FID of the model selected by proximal FID for the general corruption task.

Noise (σ)	Task	Best FID	FID Selected by Proximal FID
$\sigma = 0.0$	Gaussian Deblurring	31.90	35.68
	Random Inpainting ($p = 0.9$)	16.79	20.59
	Super Resolution ($\times 2$)	12.99	14.66
$\sigma = 0.2$	Gaussian Deblurring	76.98	88.29
	Random Inpainting ($p = 0.9$)	79.48	83.98
	Super Resolution ($\times 2$)	22.00	27.42

6.6.2 Training and Inference Efficiency

We use the denoising setting with $\sigma = 0.2$ as a running example to quantify efficiency. Our approach improves not only accuracy but also end-to-end efficiency in both training and inference. All experiments were conducted on a Linux system with 8×NVIDIA RTX A6000 GPUs unless otherwise stated.

Training. The additional distillation phase introduces only a minor overhead: the FID of the one-step generator rapidly decreases and *surpasses* the Teacher (Teacher-Truncated) within **4 hours**. Representative wall-clock times across datasets are summarized in Table 6.12. During distillation, we employ early stopping when the validation FID begins to diverge.

Inference. Our one-step generator produces 50k images in ~ 20 s on 4×NVIDIA RTX A6000 GPUs with batch size 1024, compared to ~ 10 min for the diffusion teacher—yielding a $30\times$ speedup. Inference wall-clock measurements are reported in the rightmost columns of Table 6.12.

Table 6.12: **Training and inference efficiency of our method.** During training, the additional distillation phase introduces only a minor overhead, as FID decreases rapidly and surpasses the teacher diffusion model, Teacher-Truncated, within just 4 hours. For inference, our one-step generator enables the generation of 50k images in only 20 seconds, achieving a $30\times$ speedup.

Datasets	Pretraining Time	Distillation Time to Achieve the Same FID as			Time to Generate 50k Images	
		Teacher-Full	Teacher-Truncated	Best	Diffusion	RSD
CIFAR-10	~2 days	7 minutes	~3 hours	~3 days	10 minutes	20 seconds
FFHQ		56 minutes	~3 hours	~9 hours	15 minutes	30 seconds
CelebA-HQ		34 minutes	~2 hours	~13 hours		
AFHQ-v2		80 minutes	~3 hours	~13 hours		

6.6.3 Additional Experiments

6.6.3.1 More Metrics

Table 6.13: Comparison of RSD with teacher diffusion and baselines on CIFAR-10. *Metrics for DDPM and Rectified Flow (ODE) are copied from the Rectified Flow [LGL22b] paper’s Table 1.

Method	σ	FID ($\downarrow$)	IS ($\uparrow$)	Precision ($\uparrow$)	Recall ($\uparrow$)	KID ($\downarrow$)
Teacher	0.2	12.21	8.312	0.595	0.416	0.00593
RSD	0.4	21.63	7.934	0.536	0.384	0.01270
RSD	0.2	4.77	9.165	0.650	0.564	0.00252
RSD	0.1	3.98	9.346	0.643	0.578	0.00157
DDPM	0.0	3.21	9.46	N/A	0.57	N/A
Rectified Flow (ODE)	0.0	2.58	9.60	N/A	0.57	N/A

The detailed results show that our distilled RSD generator achieves both strong density modeling (low FID) and robust mode coverage (high Recall).

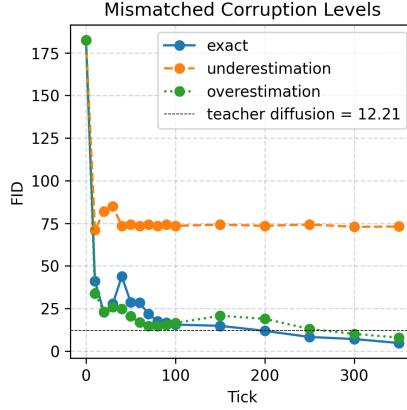


Figure 6.8: **Mismatched Corruption Levels during Distillation.** Our method remains robust even when the assumed corruption level is mismatched during distillation.

6.6.3.2 Unknown σ .

When σ is unavailable at training time, we adopt a simple strategy: (1) estimate per-image noise level using the off-the-shelf estimator implemented via `skimage.restoration.estimate_sigma` [DJ94]; (2) for reliability, select the *maximum* estimated $\hat{\sigma}$ over a small calibration set. Concretely, we sample 100 noisy CIFAR-10 images, compute $\hat{\sigma}$ per image across 100 independent trials, and use the trial-wise maximum as the working $\hat{\sigma}$. As shown in Table 6.14, this deliberately *slightly overestimates* the noise, a regime where RSD remains stable (see experiments in Table 6.15) and, in line with blind denoising results [ZZZ18, ZZC17, DFK07], helps avoid under-regularization.

Effect of misspecified σ . To isolate the impact of noise misspecification, we fix the ground-truth level at $\sigma = 0.2$ on CIFAR-10 and evaluate three settings: (i) *underestimation*, $\hat{\sigma} = 0.15$; (ii) *correct*, $\hat{\sigma} = 0.20$; and (iii) *overestimation*, $\hat{\sigma} = 0.25$. The same $\hat{\sigma}$ is used consistently in both Phase I pretraining (noisy corruption objective) and Phase II distillation. As shown in Table 6.15 and Fig. 6.8, RSD attains its best accuracy under correct specification and remains competitive under mild overestimation, whereas underestimation is substantially

Table 6.14: **Unknown noise: true σ vs. $\hat{\sigma}$ on CIFAR-10.**

σ	$\hat{\sigma}$	95% CI
0.05	0.07	[0.073, 0.074]
0.10	0.12	[0.119, 0.121]
0.20	0.22	[0.215, 0.216]
0.30	0.31	[0.311, 0.313]

Table 6.15: **Misspecification study on CIFAR-10.** RSD is robust to slight overestimation when the true $\sigma = 0.2$.

	Teacher-Full	Teacher-Trunc.	RSD
$\hat{\sigma} = 0.15$ (under)	80.60	49.78	103.55
$\hat{\sigma} = 0.20$ (true)	60.73	12.21	4.77
$\hat{\sigma} = 0.25$ (over)	42.99	88.11	16.07

more harmful—echoing observations in blind denoising [DFK07, ZZZ18, ZYC17]. Thus, when σ is unknown, slight overestimation is a robust practical choice, consistent with the strategy shown in Table 6.14. We further include an ablation where the noise level is drawn from a distribution, i.e., $\sigma \sim p(\sigma)$. Our method remains robust under this setting as well; see Appendix 6.6.3.4 for details.

6.6.3.3 More Conditional Inverse Problem Results

Our primary goal is to learn a strong generative *prior* solely from corrupted data. Once such a prior is obtained via RSD, it is agnostic to the forward operator, enabling conditional generation under arbitrary measurement models. A natural approach with our one-step generator is to solve

$$\min_z \|\mathcal{A}(G_\theta(z)) - y\|_2^2, \quad (6.6)$$

which enforces data consistency through the measurement process.

Setup. Beyond the denoising task discussed in Section 6.4.5, we consider an additional conditional inverse problem. Specifically, we use the generator trained on the $\times 2$ super-resolution task and evaluate it in a conditional inverse setting where $\mathcal{A}$ corresponds to a $\times 2$ down-sampling operator. We conduct experiments on 100 CelebA-HQ images and compare RSD against EM-Diffusion (few-shot) and the Teacher Diffusion prior. For a fair comparison,

Table 6.16: $\times 2$ **super-resolution on CelebA-HQ (100 images)**. We compare a pseudo-inverse baseline ($\mathcal{A}^\dagger y$), EM-Diffusion (few-shot) with DPS, Teacher Diffusion with DPS, and our RSD prior with latent optimization (Eq. 6.6). Higher PSNR/SSIM and lower LPIPS/FID are better.

Method	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	Prior FID $\downarrow$
$\mathcal{A}^\dagger y$	24.065	0.816	0.125	23.94
EM-Diffusion (few-shot, DPS)	25.678	0.839	0.056	58.99
Teacher Diffusion (DPS)	23.810	0.778	0.056	23.28
RSD (ours, latent opt.)	27.803	0.909	0.047	12.99

RSD optimizes Eq. 6.6 for 1000 steps using Adam with a learning rate of 0.1, while both EM-Diffusion and Teacher Diffusion adopt DPS [CKM22], a classical solver for diffusion priors, with 1000 diffusion sampling steps. LPIPS is computed using AlexNet.

Results. Table 6.16 reports PSNR, SSIM, LPIPS, and the prior’s FID. RSD attains the best performance across all metrics (e.g., PSNR = 27.803, LPIPS = 0.047), and its learned prior achieves a strong FID of 12.99, highlighting the benefit of high-quality priors for conditional generation.

6.6.3.4 Random Noise Ablation

We conducted an additional experiment in a more challenging setting where each sample is corrupted with a different noise level. Specifically, we adopt the corruption model:

$$y = x + \sigma\epsilon, \quad \sigma \sim p(\sigma)$$

where $p(\sigma) = \text{Uniform}[0.15, 0.25]$. This naturally introduces variability across samples both in terms of noise level and corruption behavior.

Table 6.17: **Comparison of FID scores.** Lower is better.

Method	FID
Teacher-Full	50.29
Teacher-Truncated ($\sigma = 0.25$)	16.21
Teacher-Truncated ($\sigma = 0.15$)	11.80
RSD (ours)	5.67

To accommodate this setup, we made a minimal modification to Algorithm 12—replacing Lines 6 from the fixed-noise setting ($\tilde{y} = x_g + \sigma\epsilon$) to the sample-dependent corruption ($\tilde{y} = x_g + \sigma\epsilon, \sigma \sim p(\sigma)$). With this simple change, RSD can be applied directly to heterogeneous corruption settings.

We tested this on the CIFAR-10 dataset, and the results are reported below in Table 6.17.

6.6.3.5 Choice of Distillation Loss

In our experiments (Sec. 6.4), the distillation phase adopts the SiD loss (Eq. 6.5) by default. Other distillation objectives are also applicable—e.g., KL-based variants such as SDS [PJB22], DMD [YGZ24] (also referred to as Diff-Instruct [LHZ23] or VSD [WLW24]), and SiD [ZZW24]. For completeness, we report their generator-level results below and defer implementation details (e.g., time-scheduling and hyperparameters) to the original papers.

We ablate this design choice on the CIFAR-10 denoising task; results are summarized in Table 6.18.

Notes. All distillation variants use the *default* hyperparameters from their official repositories; we did not tune hyperparameters. The relatively weaker performance of D-SDS and D-DMD in Table 6.18 may therefore reflect suboptimal default settings for the corrupted-data regime rather than intrinsic limitations of the losses. Empirically, SiD is robust under

Table 6.18: **Ablation of distillation losses on CIFAR-10 denoising** ($\sigma \in \{0.1, 0.2, 0.4\}$; FID)

Method	$\sigma=0.1$	$\sigma=0.2$	$\sigma=0.4$
SDS	> 200	> 200	> 200
DMD	12.52 $\pm$ 0.04	7.48 $\pm$ 0.06	30.09 $\pm$ 0.23
SiD	3.98 $\pm$ 0.04	4.77 $\pm$ 0.03	21.63 $\pm$ 0.03

defaults and already yields strong results, hence our choice to use SiD for all main experiments (Sec. 6.4). Importantly, the distillation stage in our framework is *modular* and can be replaced by more advanced objectives; future work may close the gap—or even surpass SiD—via principled hyperparameter tuning and improved losses.

6.6.4 Discussions and Limitations

Solving inverse problems. We demonstrate how to use the distilled generator as a learned prior for inverse problems in Section 6.4.7 and Appendix 6.6.3.3. Concretely, given measurements $y = \mathcal{A}(x) + \text{noise}$, we recover a plausible x by optimizing over the latent z so that the synthesized sample matches the observations, e.g.,

$$\min_z \|\mathcal{A}(G_\theta(z)) - y\|_2^2 \quad (\text{optionally with regularization or priors on } z).$$

A broader treatment—including principled conditioning, data-consistency guidance, and plug-and-play/score-based solvers—is a promising direction [CKM22, ZYZ24, ZDZ24, ZL25, CWZ24], with applications in scientific and engineering domains where reconstructing clean signals from measurements is critical.

Applications in scientific discovery. Our approach is particularly well-suited for scientific discovery, where clean observational data are often scarce or fundamentally unobtainable. Extending our method to datasets across diverse scientific domains is a promising avenue for future research. For instance, ground-truth black hole images are inherently

unobservable, yet large collections of corrupted telescope measurements are available, as demonstrated by the Event Horizon Telescope (EHT) observations [AAA19]. EHT relies on Very Long Baseline Interferometry (VLBI), where the measurement process is modeled as a 2D Fourier transform of the sky brightness distribution. Specifically, the forward model can be expressed as

$$V(u, v) = \iint I(x, y) e^{-2\pi i(ux+vy)} dx dy,$$

as in Eq. 2 of [AAA19], or in practice as

$$V_{(a,b)}^t = g_a^t g_b^t e^{-i(\phi_a^t - \phi_b^t)} \mathcal{I}_{(a,b)}^t(z) + \eta_{(a,b)}^t,$$

as described in Eq. 16 of [ZCZ25].

Unknown corruption operator. We first note that in many scientific applications such as backhole imaging and multi-coil MRI, the corruption operator is **known**. In settings where the true corruption operator is unknown, off-the-shelf estimators can provide usable approximations of the corruption process, as illustrated in Section 6.4.5. Crucially, our method does not require an explicit closed-form specification of the operator: it only assumes access to its *forward propagation* (i.e., the ability to apply the corruption). This black-box requirement confers a key advantage—our approach can recover salient properties of the underlying data distribution without explicit knowledge of how the corruption is parameterized.

6.6.5 Theoretical Results and Proofs

We provide theoretical insights to support our empirical results. In Section 6.6.5.1, we first focus on a simpler, linear Gaussian setting where we explicitly analyze the optimization landscape of score distillation given noisy samples, yielding quantitative error bounds and characterizations of global minimizers. Then, we offer several extensions of this theory in Section 6.6.5.2, including handling linear corruption and multiple noise levels. Finally, in Section 6.6.5.3, we provide a more general analysis asking when can the distilled student

achieve a strictly smaller Fisher divergence to the clean distribution than the teacher. This regime will operate under distributional assumptions on the data and corruption along with capacity and optimization assumptions on the teacher and generator.

6.6.5.1 Quantitative analysis in a linear setting

To begin, we first analyze the performance of score distillation in a stylized setting where the underlying data distribution is Gaussian. In particular, we will assume that our data follows a low-rank linear model for our analysis.

Assumption 1 (Linear Low-Rank Data Distribution). Suppose our underlying data distribution is given by a low-rank linear model $x = Ez \sim p_X$ and $z \sim \mathcal{N}(0, I_r)$, where $E \in \mathbb{R}^{d \times r}$ with $r < d$ and with orthonormal columns (i.e., $E^T E = I_r$).

Assumption 1 is equivalent to $p_X := \mathcal{N}(0, EE^T)$. For a fixed corruption noise level $\sigma > 0$, consider the setting we only have access to the noisy distribution $y = x + \sigma\epsilon$, where $x \sim p_X$ and $\epsilon \sim \mathcal{N}(0, I_d)$. In other words, $p_{Y,\sigma} := \mathcal{N}(0, EE^T + \sigma^2 I_d)$. In our setting we assume that we have perfectly learned the noisy score:

Assumption 2 (Perfect Score Estimation). Suppose we can estimate the score function of corrupted data y perfectly:

$$\nabla \log p_{Y,\sigma}(x) = -(EE^T + \sigma^2 I_d)^{-1} x.$$

Our goal is to distill this distribution into a distribution $p_{G_\theta} := (G_\theta)_\#(\mathcal{N}(0, I_d))$ given by the push-forward of $\mathcal{N}(0, I_d)$ by a generative network $G_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$. To model a U-Net [RFB15a] style architecture with bottleneck structure, we assume G_θ satisfies the following low-rank linear structure detailed in Assumption 3.

Assumption 3 (Low-Rank Linear Generator). Assume the generator is a low-rank linear mapping, where G_θ is parameterized by $\theta = (U, V)$ where $U, V \in \mathbb{R}^{d \times r}$ with $r < d$ and has

the form:

$$G_\theta(z) := UV^T z.$$

Note that G_θ induces a degenerate low-rank Gaussian distribution $p_{G_\theta} := \mathcal{N}(0, UV^T V U^T)$. Consider a bounded noise schedule $(\sigma_t) \subseteq [\sigma_{\min}, \sigma_{\max}]$ for some $0 < \sigma_{\min} < \sigma_{\max} < \infty$ and perturbed data points $x_t = x + \sigma_t \epsilon$ where $\epsilon \sim \mathcal{N}(0, I_d)$ and $x \sim p_{G_\theta}$. Then $x_t \sim p_{G_\theta}^{\sigma_t} := \mathcal{N}(0, UV^T V U^T + \sigma_t^2 I_d)$. To distill the noisy distribution, we minimize the score-based loss (or Fisher divergence) as in [ZZW24]:

$$\mathcal{L}(\theta) := \mathbb{E}_{t \sim \text{Unif}(0,1)} \mathbb{E}_{x_t \sim p_{G_\theta}^{\sigma_t}} \left[\left\| s_{\sigma, \sigma_t}(x_t) - \nabla \log p_{G_\theta}^{\sigma_t}(x_t) \right\|_2^2 \right]. \quad (6.7)$$

Here, $s_{\sigma, \sigma_t}(x) := -(EE^T + (\sigma^2 + \sigma_t^2)I_d)^{-1}x$. Note this objective is similar to Eq. 6.5, but with the real score in place of the fake score. This is also considered the idealized distillation loss (see Eq. (8) in [ZZW24]). In Theorem 1, we show that minimizing Eq. 6.7 over a certain family of non-degenerate parameters finds a distilled distribution with **smaller** Wasserstein-2 distance to the underlying clean distribution.

Theorem 1. *Fix $\sigma > 0$. Under Assumptions 1, 2, and 3, consider the family of parameters $\theta = (U, V)$ such that*

$$\theta \in \Theta := \{(U, V) : U^T U = I_r, V^T V \succ 0\}.$$

For any bounded noise schedule $(\sigma_t) \subseteq [\sigma_{\min}, \sigma_{\max}]$, the global minimizers of $\mathcal{L}$ (6.7) over Θ , denoted by $\theta_\sigma^ := (U^*, V_\sigma^*)$, satisfy the following:*

$$U^* = EQ \text{ for some orthogonal matrix } Q \text{ and } (V_\sigma^*)^T V_\sigma^* = (1 + \sigma^2)I_r. \quad (6.8)$$

For any such θ_σ^ , the induced generator distribution $p_{G_{\theta_\sigma^*}} = \mathcal{N}(0, (1 + \sigma^2)EE^T)$ satisfies*

$$W_2^2(p_{G_{\theta_\sigma^*}}, p_X) = W_2^2(p_{Y, \sigma}, p_X) - (d - r)\sigma^2 < W_2^2(p_{Y, \sigma}, p_X). \quad (6.9)$$

Discussion. This result shows that global minimizers of the distillation loss over a family of “non-degenerate” parameters induces a distribution close to the ground truth. Moreover,

we can precisely quantify the distance to the underlying distribution due to the fact that all distributions are now Gaussians and the Wasserstein-2 distance has a closed form. We further explore what the unconstrained minimizers are in Theorem 3 for the rank-one case.

Regarding our assumptions, we note that making either Gaussian or potentially more complex Gaussian mixture model assumptions is common in the generative modeling literature [CHZ23, CKV23, WV24]. We further note that this result focuses on the setting where the underlying generator has low-rank structure. While it is common to make simplifying assumptions on the network architecture to understand score-based models [CHZ23, CKS24], there is also recent work [WZZ24] that has shown when trained on data of low intrinsic dimensionality, score-based models can exhibit low-rank structures. Empirically, we find that neural-network-based distilled models can find such low-dimensional structures through noisy data. An interesting future direction of this work is to understand the influence of neural-network-based parameterizations of the score function along with analyzing the fake score setting.

Before we dive into the proof, we provide the following lemmas.

Lemma 5. [*Generalized Woodbury Matrix Identity [Hig02]*]

Given an invertible square matrix $A \in \mathbb{R}^{n \times n}$, along with matrices $U \in \mathbb{R}^{n \times k}$ and $V \in \mathbb{R}^{k \times n}$, define the perturbed matrix: $B = A + UV$. If $(I_k + VA^{-1}U)$ is invertible, then the inverse of B is given by:

$$B^{-1} = A^{-1} - A^{-1}U(I_k + VA^{-1}U)^{-1}VA^{-1}.$$

Lemma 6. *The Wasserstein-2 distance between two mean-zero Gaussians $\mathcal{N}(0, \Sigma_1)$ and $\mathcal{N}(0, \Sigma_2)$ whose covariance matrices commute, i.e., $\Sigma_1\Sigma_2 = \Sigma_2\Sigma_1$, is given by*

$$W_2^2(\mathcal{N}(0, \Sigma_1), \mathcal{N}(0, \Sigma_2)) = \sum_{i=1}^d \lambda_i(\Sigma_1) + \lambda_i(\Sigma_2) - 2\sqrt{\lambda_i(\Sigma_1)\lambda_i(\Sigma_2)}.$$

Lemma 7 ([Mir75]). *Suppose A and B are $d \times d$ complex matrices with singular values $\sigma_1(A) \geq \sigma_2(A) \geq \dots \geq \sigma_d(A) \geq 0$ and $\sigma_1(B) \geq \sigma_2(B) \geq \dots \geq \sigma_d(B) \geq 0$, respectively.*

Then

$$|\operatorname{tr}(AB)| \leq \sum_{i=1}^d \sigma_i(A) \sigma_i(B).$$

Lemma 8. Let $E \in \mathbb{R}^{d \times r}$ with $r < d$ have orthonormal columns and $\Sigma \in \mathbb{R}^{r \times r}$ be symmetric positive definite. Then

$$\operatorname{argmax}_{U^T U = I_r} \operatorname{tr}(EE^T U \Sigma U^T) = \{EQ : Q \text{ orthogonal}\}.$$

Proof of Lemma 8. Observe that by the von Neumann trace inequality (Lemma 7), we have that for any feasible U ,

$$\operatorname{tr}(EE^T U \Sigma U^T) = \operatorname{tr}(U^T EE^T U \Sigma) \leq \sum_{i=1}^r \lambda_i(U^T EE^T U) \lambda_i(\Sigma) \leq \sum_{i=1}^r \lambda_i(\Sigma)$$

where the last line we used the fact that $\lambda_i(U^T EE^T U) \leq 1$ for each $i \in [r]$. Hence, to maximize $U \mapsto \operatorname{tr}(EE^T U \Sigma U^T)$ over $\{U : U^T U = I_r\}$, we want U^* to satisfy $\operatorname{tr}(EE^T U^* \Sigma (U^*)^T) = \sum_{i=1}^r \lambda_i(\Sigma)$.

We claim that this occurs if and only if $U^* = EQ$ for some orthogonal Q . If $U^* = EQ$, then $(U^*)^T EE^T U^* = Q^T E^T EE^T EQ = I$ so

$$\operatorname{tr}(EE^T U^* \Sigma (U^*)^T) = \operatorname{tr}((U^*)^T EE^T U^* \Sigma) = \operatorname{tr}(\Sigma) = \sum_{i=1}^r \lambda_i(\Sigma).$$

For the other direction, suppose U^* maximizes the objective. Then

$$\operatorname{tr}((U^*)^T EE^T U^* \Sigma) = \operatorname{tr}(\Sigma) \iff \operatorname{tr}(((U^*)^T EE^T U^* - I_r) \Sigma) = 0.$$

Set $Q := E^T U^*$. Note that the eigenvalues of $Q^T Q$ are bounded by 1 so $Q^T Q - I_r$ is negative semi-definite while Σ is positive definite. But if $\operatorname{tr}((Q^T Q - I_r) \Sigma) = 0$, by positive definiteness of Σ , we must have $Q^T Q - I_r = 0$, i.e., $Q^T Q = I_r$. This means Q is orthogonal. Since Q is orthogonal and $Q = E^T U^* \implies U^* = EQ$, as desired. $\square$

Lemma 9. Fix $\sigma > 0$ and consider a noise schedule $\sigma_t > 0$ for $t \in (0, 1)$ such that $(\sigma_t) \subseteq [\sigma_{\min}, \sigma_{\max}]$ for some $0 < \sigma_{\min} < \sigma_{\max} < \infty$. Define the function $f_\sigma : (0, \infty) \rightarrow \mathbb{R}$ by

$$f_\sigma(u) := \mathbb{E}_{t \sim \operatorname{Unif}(0,1)} \left[\frac{u}{(\sigma^2 + \sigma_t^2 + 1)^2} - \frac{u}{\sigma_t^2(u + \sigma_t^2)} \right].$$

Then f_σ is strictly convex and has a unique minimizer at $u^* = \sigma^2 + 1$ which is the unique solution to the equation

$$\mathbb{E}_{t \sim \text{Unif}(0,1)} \left[\frac{1}{(\sigma^2 + \sigma_t^2 + 1)^2} \right] = \mathbb{E}_{t \sim \text{Unif}(0,1)} \left[\frac{1}{(u^* + \sigma_t^2)^2} \right].$$

Proof of Lemma 9. First, note that the conditions on σ_t ensure that all of the following expectations are finite. By direct calculation, we have the derivatives of f_σ are

$$f'_\sigma(u) = \mathbb{E}_t \left[\frac{1}{(\sigma^2 + \sigma_t^2 + 1)^2} \right] - \mathbb{E}_t \left[\frac{1}{(\sigma_t^2 + u)^2} \right] \text{ and } f''_\sigma(u) = \mathbb{E}_t \left[\frac{2}{(\sigma_t^2 + u)^3} \right].$$

Hence $f''_\sigma(u) > 0$ for all $u > 0$ so f_σ is strictly convex. To find its minimizer u^* , setting the derivative equal to 0 yields u^* must satisfy

$$\mathbb{E}_t \left[\frac{1}{(\sigma^2 + \sigma_t^2 + 1)^2} \right] = \mathbb{E}_t \left[\frac{1}{(\sigma_t^2 + u^*)^2} \right].$$

Note that the point $u^* = 1 + \sigma^2$ clearly satisfies the critical point equation. Uniqueness follows due to strict convexity. □

Proof of Theorem 1 We break down the proof of Theorem 1 into three key steps. First, we show that minimizing the objective (Eq. 6.7) is equivalent to minimizing a simpler objective. Then, we show that we can derive exact analytical expressions for the global minimizers of this simpler objective, which are then global minimizers of the original score-based loss. Finally, we will directly compute the Wasserstein distance between our learned distilled distribution to the clean distribution and compare this to the noisy distribution.

Reduction of objective function: For $\sigma_t > 0$, define $p_{G_\theta}^{\sigma_t} := \mathcal{N}(0, UV^T V U^T + \sigma_t^2 I_d)$ and $s_{\sigma, \sigma_t}(x) := -(EE^T + (\sigma^2 + \sigma_t^2)I_d)^{-1}x$. For the proof, we will assume our parameters $\theta = (U, V) \in \Theta$ so that $U^T U = I_r$ and $V^T V \succ 0$. We consider minimizing the loss

$$\mathcal{L}(\theta) := \mathbb{E}_{t \sim \text{Unif}(0,1)} \mathbb{E}_{x_t \sim p_{G_\theta}^{\sigma_t}} \left[\left\| s_{\sigma, \sigma_t}(x_t) - \nabla \log p_{G_\theta}^{\sigma_t}(x_t) \right\|_2^2 \right].$$

For $t \in (0, 1)$, consider the inner expectation of the loss

$$\tilde{\mathcal{L}}_t(\theta) := \mathbb{E}_{x_t \sim p_{G_\theta}^{\sigma_t}} \left[\|s_{\sigma, \sigma_t}(x_t) - \nabla \log p_{G_\theta}^{\sigma_t}(x_t)\|_2^2 \right].$$

For notational convenience, set $\Sigma_{\sigma, t} := EE^T + (\sigma^2 + \sigma_t^2)I_d$ and $\Sigma_{\theta, t} := UV^TVU^T + \sigma_t^2 I_d$. Then $s_{\sigma, \sigma_t}(x) := -\Sigma_{\sigma, t}^{-1}x$ and $\nabla \log p_{G_\theta}^{\sigma_t}(x) := -\Sigma_{\theta, t}^{-1}x$. First, recall that for $x_t \sim p_{G_\theta}^{\sigma_t}$ and any matrix Σ , $\mathbb{E}_{x \sim p_{G_\theta}^{\sigma_t}} [\|\Sigma x_t\|_2^2] = \|\Sigma \Sigma_{\theta, t}^{1/2}\|_F^2$. Using this, we can compute the loss as follows:

$$\begin{aligned} \tilde{\mathcal{L}}_t(\theta) &= \mathbb{E}_{x_t \sim p_{G_\theta}^{\sigma_t}} [\|(\Sigma_{\sigma, t}^{-1} - \Sigma_{\theta, t}^{-1})x_t\|_2^2] \\ &= \|(\Sigma_{\sigma, t}^{-1} - \Sigma_{\theta, t}^{-1})\Sigma_{\theta, t}^{1/2}\|_F^2 \\ &= \text{tr} \left(\Sigma_{\theta, t}^{1/2}(\Sigma_{\sigma, t}^{-1} - \Sigma_{\theta, t}^{-1})(\Sigma_{\sigma, t}^{-1} - \Sigma_{\theta, t}^{-1})\Sigma_{\theta, t}^{1/2} \right) \\ &= \text{tr} \left(\Sigma_{\theta, t}(\Sigma_{\sigma, t}^{-1} - \Sigma_{\theta, t}^{-1})(\Sigma_{\sigma, t}^{-1} - \Sigma_{\theta, t}^{-1}) \right) \\ &= \text{tr} \left((\Sigma_{\theta, t}\Sigma_{\sigma, t}^{-1} - I_d)(\Sigma_{\sigma, t}^{-1} - \Sigma_{\theta, t}^{-1}) \right) \\ &= \text{tr} \left(\Sigma_{\theta, t}\Sigma_{\sigma, t}^{-2} - \Sigma_{\theta, t}\Sigma_{\sigma, t}^{-1}\Sigma_{\theta, t}^{-1} - \Sigma_{\sigma, t}^{-1} + \Sigma_{\theta, t}^{-1} \right) \\ &= \text{tr} \left(\Sigma_{\theta, t}\Sigma_{\sigma, t}^{-2} \right) - \text{tr} \left(\Sigma_{\theta, t}\Sigma_{\sigma, t}^{-1}\Sigma_{\theta, t}^{-1} \right) - \text{tr} \left(\Sigma_{\sigma, t}^{-1} \right) + \text{tr} \left(\Sigma_{\theta, t}^{-1} \right) \\ &= \text{tr} \left(\Sigma_{\sigma, t}^{-2}\Sigma_{\theta, t} \right) - 2\text{tr} \left(\Sigma_{\sigma, t}^{-1} \right) + \text{tr} \left(\Sigma_{\theta, t}^{-1} \right) \\ &=: C_{\sigma, t} + \text{tr} \left(\Sigma_{\sigma, t}^{-2}\Sigma_{\theta, t} \right) + \text{tr} \left(\Sigma_{\theta, t}^{-1} \right). \end{aligned}$$

Using Lemma 6, it is straightforward to see that

$$\begin{aligned} \Sigma_{\sigma, t}^{-1} &= \frac{1}{\sigma^2 + \sigma_t^2} I_d - \frac{1}{(\sigma^2 + \sigma_t^2)^2(\sigma^2 + \sigma_t^2 + 1)} EE^T \text{ and} \\ \Sigma_{\theta, t}^{-1} &= \sigma_t^{-2} I_d - \sigma_t^{-4} U \left((V^T V)^{-1} + \sigma_t^{-2} I_r \right)^{-1} U^T \end{aligned}$$

Hence the third term in $\tilde{\mathcal{L}}_t$ is given by

$$\text{tr}(\Sigma_{\theta, t}^{-1}) = \text{tr} \left(\sigma_t^{-2} I_d - \sigma_t^{-4} U \left((V^T V)^{-1} + \sigma_t^{-2} I_r \right)^{-1} U^T \right) =: C_{\sigma, t} - \sigma_t^{-4} \text{tr} \left(\left((V^T V)^{-1} + \sigma_t^{-2} I_r \right)^{-1} \right)$$

where we used the cyclic property of the trace and $U^T U = I_r$ in the last equality. For the second term, let $\beta_t^2 := \sigma^2 + \sigma_t^2$ and $\gamma_{\sigma,t} := \frac{1}{\beta_t^2(\beta_t^2+1)}$. Then we have by direct computation,

$$\begin{aligned}
\text{tr}(\Sigma_{\sigma,t}^{-2} \Sigma_{\theta,t}) &= \text{tr}((\beta_t^{-2} I_d - \gamma_{\sigma,t} E E^T)(\beta_t^{-2} I_d - \gamma_{\sigma,t} E E^T)(U V^T V U^T + \sigma_t^2 I_d)) \\
&= \text{tr}((\beta_t^{-4} I_d - 2\beta_t^{-2} \gamma_{\sigma,t} E E^T + \gamma_{\sigma,t}^2 E E^T)(U V^T V U^T + \sigma_t^2 I_d)) \\
&= \text{tr}(\beta_t^{-4} U V^T V U^T - \sigma_t^2 \beta_t^{-4} I_d + (\gamma_{\sigma,t}^2 - 2\beta_t^{-2} \gamma_{\sigma,t}) E E^T U V^T V U^T) \\
&\quad - \text{tr}(2\beta_t^{-2} \sigma_t^2 E E^T + \gamma_{\sigma,t}^2 \sigma_t^2 I_d) \\
&=: \tilde{C}_{\sigma,t} + \beta_t^{-4} \text{tr}(U V^T V U^T) + (\gamma_{\sigma,t}^2 - 2\beta_t^{-2} \gamma_{\sigma,t}) \cdot \text{tr}(E E^T U V^T V U^T) \\
&= \tilde{C}_{\sigma,t} + \beta_t^{-4} \text{tr}(V^T V) + (\gamma_{\sigma,t}^2 - 2\beta_t^{-2} \gamma_{\sigma,t}) \cdot \text{tr}(E E^T U V^T V U^T)
\end{aligned}$$

where we used the cyclic property of trace and orthogonality of U in the final line. Combining the above displays, we get that there exists a constant $C_{\sigma,\sigma_t} := C_{\sigma,t} + C_{\sigma_t} + \tilde{C}_{\sigma,t}$ such that

$$\begin{aligned}
\tilde{\mathcal{L}}_t(\theta) &= C_{\sigma,\sigma_t} + \left(\frac{1}{\beta_t^4(\beta_t^2+1)^2} - \frac{2}{\beta_t^4(\beta_t^2+1)} \right) \cdot \text{tr}(E E^T U V^T V U^T) \\
&\quad + \beta_t^{-4} \text{tr}(V^T V) - \sigma_t^{-4} \text{tr}\left(\left((V^T V)^{-1} + \sigma_t^{-2} I_r\right)^{-1}\right) \\
&=: C_{\sigma,\sigma_t} + B_t(U, V) + R_t(V)
\end{aligned}$$

where we have defined the quantities

$$\begin{aligned}
B_t(U, V) &:= \left(\frac{1}{\beta_t^4(\beta_t^2+1)^2} - \frac{2}{\beta_t^4(\beta_t^2+1)} \right) \cdot \text{tr}(E E^T U V^T V U^T) \text{ and} \\
R_t(V) &:= \beta_t^{-4} \text{tr}(V^T V) - \sigma_t^{-4} \text{tr}\left(\left((V^T V)^{-1} + \sigma_t^{-2} I_r\right)^{-1}\right).
\end{aligned}$$

Recalling the definition of $\mathcal{L}(\cdot)$, we have that

$$\mathcal{L}(\theta) = \mathbb{E}_{t \sim \text{Unif}(0,1)} [\tilde{\mathcal{L}}_t(\theta)] = \mathbb{E}_{t \sim \text{Unif}(0,1)} [C_{\sigma,\sigma_t} + B_t(U, V) + R_t(U, V)].$$

Hence we have the equivalence

$$\underset{\theta \in \Theta}{\text{argmin}} \mathcal{L}(\theta) = \underset{\theta \in \Theta}{\text{argmin}} \mathbb{E}_{t \sim \text{Unif}(0,1)} [B_t(U, V)] + \mathbb{E}_{t \sim \text{Unif}(0,1)} [R_t(V)].$$

Form of minimizers: We use the shorthand notation $\mathbb{E}_t[\cdot] := \mathbb{E}_{t \sim \text{Unif}(0,1)}[\cdot]$. First, note that we can first minimize $\mathbb{E}_t[B_t(U, V)]$ over feasible U . But note that

$$\mathbb{E}_t[B_t(U, V)] = \mathbb{E}_t \left[\underbrace{\frac{1}{\beta_t^4(\beta_t^2 + 1)^2} - \frac{2}{\beta_t^4(\beta_t^2 + 1)}}_{<0} \right] \text{tr}(EE^T UV^T VU^T)$$

since for any t , $\frac{1}{(\beta_t^2 + 1)^2} < \frac{2}{\beta_t^2 + 1}$. Hence minimizing $\mathbb{E}_t[B_t(U, V)]$ is equivalent to maximizing $\text{tr}(EE^T UV^T VU^T)$. Taking $\Sigma = V^T V$ in Lemma 8, we have that the minimizer of $\mathbb{E}_t[B_t(U, V)]$ is given by

$$U^* = EQ \text{ for some orthogonal } Q.$$

Moreover, the proof of Lemma 8 shows that $\text{tr}(EE^T U^* V^T V (U^*)^T) = \text{tr}(V^T V)$. This gives

$$\mathbb{E}_t[B_t(U^*, V)] = \mathbb{E}_t \left(\frac{1}{\beta_t^4(\beta_t^2 + 1)^2} - \frac{2}{\beta_t^4(\beta_t^2 + 1)} \right) \text{tr}(V^T V).$$

In summary, we now must minimize the following with respect to invertible V :

$$\begin{aligned} \mathbb{E}_t[B_t(U^*, V)] + \mathbb{E}_t[R_t(V)] &= \mathbb{E}_t \left(\frac{1}{\beta_t^4(\beta_t^2 + 1)^2} - \frac{2}{\beta_t^4(\beta_t^2 + 1)} + \frac{1}{\beta_t^4} \right) \text{tr}(V^T V) \\ &\quad - \mathbb{E}_t \left[\sigma_t^{-4} \text{tr} \left(((V^T V)^{-1} + \sigma_t^{-2} I_r)^{-1} \right) \right] \\ &= \mathbb{E}_t \left(\frac{1}{\beta_t^4} \left(\frac{1}{\beta_t^2 + 1} - 1 \right)^2 \right) \text{tr}(V^T V) - \mathbb{E}_t \left[\sigma_t^{-4} \text{tr} \left(((V^T V)^{-1} + \sigma_t^{-2} I_r)^{-1} \right) \right] \\ &= \mathbb{E}_t \left(\frac{1}{\beta_t^4} \left(\frac{\beta_t^2}{\beta_t^2 + 1} \right)^2 \right) \text{tr}(V^T V) - \mathbb{E}_t \left[\sigma_t^{-4} \text{tr} \left(((V^T V)^{-1} + \sigma_t^{-2} I_r)^{-1} \right) \right] \\ &= \mathbb{E}_t \left(\frac{1}{(\beta_t^2 + 1)^2} \right) \text{tr}(V^T V) - \mathbb{E}_t \left[\sigma_t^{-4} \text{tr} \left(((V^T V)^{-1} + \sigma_t^{-2} I_r)^{-1} \right) \right] \end{aligned}$$

where in the second equality, we completed the square.

We now claim that $\mathbb{E}_t[B_t(U^*, V)] + \mathbb{E}_t[R_t(V)]$ solely depends on the eigenvalues of $V^T V$. In particular, for invertible V , note that $V^T V \succ 0$ so it admits the decomposition $V^T V = Q\Lambda Q^T$ where $Q^T Q = Q Q^T = I_r$ and Λ is a diagonal matrix with positive entries $\Lambda_{ii} = \lambda_i(V^T V) > 0$. Hence $\text{tr}(V^T V) = \text{tr}(Q\Lambda Q^T) = \text{tr}(Q^T Q \Lambda) = \text{tr}(\Lambda) = \sum_{i=1}^r \lambda_i(V^T V)$. Likewise, we have using

the orthogonality of Q that for any $\varepsilon > 0$,

$$\begin{aligned}
\text{tr} \left(((V^T V)^{-1} + \varepsilon^{-2} I_r)^{-1} \right) &= \text{tr} \left(((Q \Lambda Q^T)^{-1} + \varepsilon^{-2} I_r)^{-1} \right) \\
&= \text{tr} \left((Q \Lambda^{-1} Q^T + \varepsilon^{-2} Q Q^T)^{-1} \right) \\
&= \text{tr} \left((Q (\Lambda^{-1} + \varepsilon^{-2} I_r) Q^T)^{-1} \right) \\
&= \text{tr} \left(Q (\Lambda^{-1} + \varepsilon^{-2} I_r)^{-1} Q^T \right) \\
&= \text{tr} \left((\Lambda^{-1} + \varepsilon^{-2} I_r)^{-1} \right) \\
&= \sum_{i=1}^r \frac{1}{\lambda_i (V^T V)^{-1} + \varepsilon^{-2}} \\
&= \sum_{i=1}^r \frac{\lambda_i (V^T V) \cdot \varepsilon^2}{\lambda_i (V^T V) + \varepsilon^2}.
\end{aligned}$$

In sum, the final objective is a particular function of the eigenvalues of $V^T V$:

$$\begin{aligned}
\mathbb{E}_t[B_t(U^*, V)] + \mathbb{E}_t[R_t(V)] &= \sum_{i=1}^r \mathbb{E}_t \left[\frac{\lambda_i(V^T V)}{(\beta_t^2 + 1)^2} - \frac{\lambda_i(V^T V)}{\sigma_t^2(\lambda_i(V^T V) + \sigma_t^2)} \right] \\
&= \sum_{i=1}^r \mathbb{E}_t \left[\frac{\lambda_i(V^T V)}{(\sigma^2 + \sigma_t^2 + 1)^2} - \frac{\lambda_i(V^T V)}{\sigma_t^2(\lambda_i(V^T V) + \sigma_t^2)} \right] \\
&=: \sum_{i=1}^r f_\sigma(\lambda_i(V^T V)).
\end{aligned}$$

In Lemma 9, we show that the function $u \mapsto f_\sigma(u)$ is strictly convex on $(0, \infty)$ with a unique minimizer at $1 + \sigma^2$. Thus $V \mapsto B(U^*, V) + R(V)$ for invertible V is minimized when the gram matrix of V_σ^* has equal eigenvalues $\lambda_i((V_\sigma^*)^T V_\sigma^*) = 1 + \sigma^2$ for all $i \in [r]$. Since all of its eigenvalues are the same, by the Spectral Theorem, we must have that $(V_\sigma^*)^T V_\sigma^* = (1 + \sigma^2) I_r$.

Wasserstein bound: We now show the Wasserstein error bound. Note that $\theta_\sigma^* = (U^*, V_\sigma^*)$ induces the distribution $p_{G_{\theta_\sigma^*}}$ defined by

$$x = G_{\theta_\sigma^*}(z), \quad z \sim \mathcal{N}(0, I_d) \iff x \sim p_{G_{\theta_\sigma^*}} := \mathcal{N}(0, E Q (V_\sigma^*)^T V_\sigma^* Q^T E^T) = \mathcal{N}(0, (1 + \sigma^2) E E^T).$$

Then by Lemma 6, we have

$$\begin{aligned} W_2^2(p_{Y,\sigma}, p_X) &= r \left(1 + \sigma^2 + 1 - 2\sqrt{1 + \sigma^2} \right) + (d - r)\sigma^2, \\ W_2^2(p_{G_{\theta_\sigma^*}}, p_X) &= r \left(1 + \sigma^2 + 1 - 2\sqrt{1 + \sigma^2} \right). \end{aligned}$$

This gives

$$W_2^2(p_{G_{\theta_\sigma^*}}, p_X) = W_2^2(p_{Y,\sigma}, p_X) - (d - r)\sigma^2 < W_2^2(p_{Y,\sigma}, p_X).$$

6.6.5.2 Extensions of the theory in Section 6.6.5.1

We now discuss three extensions of Theorem 1: 1) we allow for additional corruption in y , 2) characterize the full optimization landscape in the rank-one case, and 3) analyze the global minimizers when we may have varying noise levels in the training data.

Additional measurement corruption We will now consider the case when the data is not simply noisy, but also exhibits more general corruption. For a fixed corruption noise level $\sigma > 0$, consider the setting we only have access to the noisy distribution $y = \mathcal{A}(x) + \sigma\epsilon$, where $\epsilon \sim \mathcal{N}(0, I_m)$ and $\mathcal{A}(x) = Ax$ with $A \in \mathbb{R}^{m \times d}$ is a linear corruption operator. By our assumption on p_X (see Assumption 1) and the noise, $p_{Y,\sigma} := \mathcal{N}(0, AEE^T A^T + \sigma^2 I_m)$. In order to get rid of error in estimating score, we assume that we have perfectly learned the noisy score:

Assumption 4 (Perfect Score Estimation with $\mathcal{A}$). Suppose $A \in \mathbb{R}^{m \times d}$ with $\text{rank}(A) = m$ and we can estimate the score function of corrupted data y perfectly:

$$s_{A,\sigma^2}(x) := \nabla \log p_{Y,\sigma^2}(x) = -(AEE^T A^T + \sigma^2 I_m)^{-1}x.$$

Our goal is to match the corrupted noise distribution's score with the score of p_{G_θ} under Assumption 3 corrupted by A over a series of noise schedules (σ_t) , which is given by

$$\tilde{p}_{G_\theta}^{\sigma_t}(y) = \mathcal{N}(0, AUV^T VU^T A^T + \sigma_t^2 I_m).$$

To distill the noisy distribution, we minimize the score-based loss (or Fisher divergence) as in [ZZW24]:

$$\mathcal{L}(\theta) := \mathbb{E}_{t \sim \text{Unif}[0,1]} \mathbb{E}_{y_t \sim \tilde{p}_{G_\theta}^{\sigma_t}} [\|s_{A, \sigma^2 + \sigma_t^2}(y_t) - \nabla \log \tilde{p}_{G_\theta}^{\sigma_t}(y_t)\|_2^2]. \quad (6.10)$$

We show that we can characterize the global minimizers of this loss, which correspond to a noise-dependent scaling of the *true* eigenspace of p_X plus perturbations in the kernel of A . If we penalize the norm of our solution, we can nearly recover the true data distribution in Wasserstein-2 distance up to the noise in our measurements. We consider the rank- $r = 1$ case for simplicity.

Theorem 2. *Fix $\sigma > 0$ and consider $e \in \mathbb{R}^d$ with unit norm and $Ae \neq 0$. Under Assumptions 1, 4, and 3 and any bounded noise schedule $(\sigma_t) \subseteq [\sigma_{\min}, \sigma_{\max}]$, the set of global minimizers of the loss 6.10 under the parameterization $u \mapsto \theta(u) = (u, u/\|u\|)$ is given by*

$$\Theta_* := \left\{ \pm \sqrt{1 + \frac{\sigma^2}{\|Ae\|^2}} \cdot e \right\} + \ker(A).$$

If $e \in \text{Im}(A^T)$, we have that $\theta_ = \theta_*(u_*)$ with the minimum norm solution $u_* \in \arg\min_{u \in \Theta_*} \|u\|$ satisfies*

$$W_2^2(p_X, p_{G_{\theta_*}}) = \left(\sqrt{1 + \frac{\sigma^2}{\|Ae\|^2}} - 1 \right)^2.$$

Discussion. Theorem 2 aims to provide a quantitative bound of the Wasserstein distance between the distribution learned via distillation and the target clean distribution. To do this, we characterize the global minimizers of the loss, which correspond to scalings of the true principal component e plus perturbations in the kernel of A . The scaling depends on an effective signal-to-noise ratio $SNR := \|Ae\|^2/\sigma^2$. Note that terms involving perturbations in the kernel of A are expected since the corruption compresses the data, leaving many plausible images that could give rise to the same measurements (a fundamental part of the ill-posedness in inverse problems). Furthermore, we show that if we penalize the norm of

our parameter, the Wasserstein distance simplifies into a more interpretable quantity. A more general bound for all elements in the set of global minimizers is shown at the end of the proof. The intuition for the Theorem is that if we encourage finding a “simple” model (i.e., one with low-norm), the Wasserstein distance decreases and is effectively inversely proportional to the SNR . The distance between our learned distilled distribution and the true distribution goes to zero whenever 1) the noise goes to zero or 2) the signal strength increases. This result recovers the rank-1 version of Theorem 2 when $A = I$, showing that the learned distilled distribution’s Wasserstein distance to the true distribution is less than the noisy distribution’s Wasserstein distance to the true distribution subtracted by a factor of $(d - 1)\sigma^2$, showing a clear improvement in distribution learning. Finally, we note that the condition $e \in \text{Im}(A^T)$ we use is akin to assumptions in the compressed sensing literature on stable recovery via the construction of dual certificates [FR13].

To prove the Theorem, we first show that the objective under the parameterization $\theta = (u, u/\|u\|)$ simplifies into a form that we directly analyze.

Lemma 10. *Consider the setting of Theorem 2. For $t \in [0, 1]$, let $a_t := \sigma_t^2$, $c_t := a_t + \sigma^2$, and $\eta_t := \frac{2c_t + \|Ae\|^2}{c_t^2(c_t + \|Ae\|^2)}$. Then the objective $\mathcal{L}(\theta)$ with $\theta = (u, u/\|u\|)$ satisfies the following: there exists a constant C independent of $u \in \mathbb{R}^d$ such that*

$$\mathcal{L}(\theta) = C + L(u)$$

where

$$L(u) := \mathbb{E}_{t \sim \text{Unif}[0,1]} \left[\frac{1}{c_t^2} \|Au\|^2 - \eta_t (e^T A^T Au)^2 - \frac{\|Au\|^2}{a_t(a_t + \|Au\|^2)} \right].$$

Proof of Lemma 10. Consider the loss function $\mathcal{L}(\theta)$ under the parameterization $\theta = (u, u/\|u\|)$ for $u \neq 0$:

$$\mathcal{L}(\theta) := \mathbb{E}_{t \sim \text{Unif}[0,1]} \mathbb{E}_{y_t \sim \tilde{p}_{G_\theta}^{\sigma_t}} \left[\left\| (\Sigma_{t,e,A}^{-1} - \Sigma_{t,\theta,A})^{-1} y_t \right\|_2^2 \right]$$

where

$$\Sigma_{t,e,A} := Aee^T A^T + c_t I_m \text{ and } \Sigma_{t,\theta,A} := Auu^T A^T + a_t I_m.$$

Using a similar reduction in the proof of Theorem 1, we get that the above loss equals

$$\mathcal{L}(\theta) = \mathbb{E}_{t \sim \text{Unif}[0,1]} [C_{\sigma,t}] + \mathbb{E}_{t \sim \text{Unif}[0,1]} [\text{tr}(\Sigma_{t,e,A}^{-2} \Sigma_{t,\theta,A})] + \mathbb{E}_{t \sim \text{Unif}[0,1]} [\text{tr}(\Sigma_{t,\theta,A}^{-1})]$$

where $C_{\sigma,t}$ is a constant that depends only on time and σ and not on θ . Note that using the Woodbury matrix identity in Lemma 5, we get

$$\Sigma_{t,e,A}^{-2} = (Aee^T A^T + c_t I_m)^{-2} = c_t^{-2} I_m - \frac{2c_t + \|Ae\|^2}{c_t^2(c_t + \|Ae\|^2)^2} Aee^T A^T$$

which implies

$$\begin{aligned} \Sigma_{t,e,A}^{-2} \Sigma_{t,\theta,A} &= \left(c_t^{-2} I_m - \frac{2c_t + \|Ae\|^2}{c_t^2(c_t + \|Ae\|^2)^2} Aee^T A^T \right) Auu^T A^T \\ &\quad + c_t^{-2} a_t I_m - a_t \frac{2c_t + \|Ae\|^2}{c_t^2(c_t + \|Ae\|^2)^2} Aee^T A^T \\ &= c_t^{-2} Auu^T A^T - \eta_t Aee^T A^T Auu^T A^T + c_t^{-2} a_t I_m - a_t \eta_t Aee^T A^T. \end{aligned}$$

Also, we have that

$$\Sigma_{t,\theta,A}^{-1} = (Auu^T A^T + a_t I_m)^{-1} = a_t^{-1} I_m - \frac{Auu^T A^T}{a_t(a_t + \|Au\|^2)}.$$

Thus computing the trace, taking an expectation, and collecting terms that only involve u , we see that there exists a constant C depending on (σ_t) , σ , A , and e (independent of u) such that

$$\begin{aligned} \mathcal{L}(\theta) &= C + \mathbb{E}_t[c_t^{-2}] \text{tr}(Auu^T A^T) - \mathbb{E}_t[\eta_t] \text{tr}(Aee^T A^T Auu^T A^T) - \mathbb{E}_t \left[\frac{\text{tr}(Auu^T A^T)}{a_t(a_t + \|Au\|^2)} \right] \\ &= C + \mathbb{E}_t[c_t^{-2}] \|Au\|^2 - \mathbb{E}_t[\eta_t] (e^T A^T Au)^2 - \mathbb{E}_t \left[\frac{\|Au\|^2}{a_t(a_t + \|Au\|^2)} \right] \\ &=: C + L(u). \end{aligned}$$

□

Proof of Theorem 2. Note that Lemma 10 shows that we have the simpler formula

$$\mathcal{L}(\theta) = C + L(u)$$

where

$$L(u) := \mathbb{E}_t \left[\frac{1}{c_t^2} \|Au\|^2 - \eta_t (e^T A^T Au)^2 - \frac{\|Au\|^2}{a_t(a_t + \|Au\|^2)} \right]$$

For notational simplicity, let $G = A^T A$, $g(u) = u^T Gu$, $h(u) = e^T Gu$, $\phi_t(g) = g/(a_t(a_t + g))$, and $\eta_t := (2c_t + \|Ae\|^2)/(c_t^2(c_t + \|Ae\|^2)^2)$. Then our objective results in

$$L(u) := \mathbb{E}_t[c_t^{-2}]g(u) - \mathbb{E}_t[\eta_t]h(u)^2 - \mathbb{E}_t[\phi_t(g(u))] =: \tilde{c}g(u) - \tilde{\eta}h(u)^2 - \mathbb{E}_t[\phi_t(g(u))].$$

We claim that any global minimizer of L is of the form $u_* = \pm \lambda_* e + q$ for some constant λ_* to be defined and $q \in \ker(A)$. First, note that we can lower bound $L(u)$ by another function $\Psi(g(u))$ as follows: since G is PSD, note that we have the generalized Cauchy Schwarz inequality:

$$h(u)^2 = (e^T Gu)^2 \leq (e^T Ge)(u^T Gu) = g(e)g(u).$$

Moreover, this holds with equality if and only if Au and Ae are collinear, i.e., $Au = \lambda Ae$ or equivalently $u = \lambda e + v$ for $v \in \ker(A)$. Then we get the lower bound

$$L(u) = \tilde{c}g(u) - \tilde{\eta}h(u)^2 - \mathbb{E}_t[\phi_t(g(u))] \geq (\tilde{c} - \tilde{\eta}\|Ae\|^2)g(u) - \mathbb{E}_t[\phi_t(g(u))] =: \Psi(g(u))$$

where we have defined the new function $\Psi(g) := (\tilde{c} - \tilde{\eta}\|Ae\|^2)g - \mathbb{E}_t[\phi_t(g)]$ for $g \geq 0$. Since $g(u) \geq 0$ for every u , we have that

$$L(u) \geq \min_{g \geq 0} \Psi(g).$$

We first analyze the minimizers of Ψ and then construct u_* such that $L(u_*) = \Psi(g_*)$. First let $\Delta = (\tilde{c} - \tilde{\eta}\|Ae\|^2)$ which is given by

$$\Delta = \mathbb{E}_t [c_t^{-2} - \eta_t \|Ae\|^2] = \mathbb{E}_t \left[\frac{1}{c_t^2} - \frac{(2c_t + \|Ae\|^2)\|Ae\|^2}{c_t^2(c_t + \|Ae\|^2)^2} \right].$$

We claim that $\Delta > 0$. Indeed, note that for any $g > 0$,

$$\frac{g(2c_t + g)}{(c_t + g)^2} = \frac{g(2c_t + g)g}{c_t^2 + (2c_t + g)g} = \frac{g(2c_t + g) + c_t^2 - c_t^2}{c_t^2 + (2c_t + g)g} = 1 - \frac{c_t^2}{(c_t + g)^2}.$$

Applying this to Δ with $g = \|Ae\|^2$ yields

$$\begin{aligned} \Delta &= \mathbb{E}_t \left[\frac{1}{c_t^2} - \frac{(2c_t + \|Ae\|^2)\|Ae\|^2}{c_t^2(c_t + \|Ae\|^2)^2} \right] \\ &= \mathbb{E}_t \left[\frac{1}{c_t^2} \left(1 - \frac{(2c_t + \|Ae\|^2)\|Ae\|^2}{(c_t + \|Ae\|^2)^2} \right) \right] \\ &= \mathbb{E}_t \left[\frac{1}{c_t^2} \left(1 - 1 + \frac{c_t^2}{(c_t + \|Ae\|^2)^2} \right) \right] \\ &= \mathbb{E}_t \left[\frac{1}{(c_t + \|Ae\|^2)^2} \right] > 0 \end{aligned}$$

as desired. This gives

$$\Psi(g) = \Delta g - \mathbb{E}_t[\phi_t(g)].$$

Observe that

$$\Psi'(g) = \Delta - \frac{\partial}{\partial g} \int_0^1 \frac{g}{a_t^2 + a_t g} dt = \Delta - \int_0^1 \frac{a_t^2}{(a_t^2 + a_t g)^2} dt = \Delta - \mathbb{E}_t \left[\frac{1}{(a_t + g)^2} \right]$$

and

$$\Psi''(g) = -\partial/\partial g \mathbb{E}_t[(a_t + g)^{-2}] = 2\mathbb{E}_t[(a_t + g)^{-3}] > 0 \quad \forall g.$$

Hence the function Ψ is strictly convex. Moreover, a sign analysis reveals that

$$\Psi'(0) = \Delta - \mathbb{E}_t[1/a_t^2] = \mathbb{E}_t \left[\frac{1}{(c_t + \|Ae\|^2)^2} - \frac{1}{a_t^2} \right] = \mathbb{E}_t \left[\frac{1}{(\sigma^2 + a_t + \|Ae\|^2)^2} - \frac{1}{a_t^2} \right] < 0$$

while for $g > \sigma^2 + \|Ae\|^2$, we have

$$\Psi'(g) = \mathbb{E}_t \left[\frac{1}{(\sigma^2 + a_t + \|Ae\|^2)^2} - \frac{1}{(a_t + g)^2} \right] > 0$$

so there must exist a unique root g_* such that $\Psi'(g_*) = 0$. In fact, $g_* = \sigma^2 + \|Ae\|^2$ achieves $\Psi'(g_*) = 0$. Finally, we have that

$$L(u) \geq \min_{g \geq 0} \Psi(g) = \Psi(g_*) = \Delta(\sigma^2 + \|Ae\|^2) - \mathbb{E}_t[\phi_t(\sigma^2 + \|Ae\|^2)].$$

We claim that $u_* = \pm\sqrt{g_*/\|Ae\|^2}e$ achieves $L(u_*) = \Psi(g_*)$. Indeed, note that $g(u_*) = (u_*)^T Gu_* = g_*/\|Ae\|^2 e^T Ge = g_*/\|Ae\|^2 \cdot \|Ae\|^2 = g_*$ and

$$h(u_*)^2 = (e^T Gu_*)^2 = \frac{g_*}{\|Ae\|^2} (e^T Ge) = g_* \|Ae\|^2.$$

Hence

$$\begin{aligned} L(u_*) &= \tilde{c}g(u_*) - \tilde{\eta}h(u_*) - \mathbb{E}_t[\phi_t(g(u_*))] \\ &= \tilde{c}g_* - \tilde{\eta}g_*\|Ae\|^2 - \mathbb{E}_t[\phi_t(g_*)] \\ &= (\tilde{c} - \tilde{\eta}\|Ae\|^2)g_* - \mathbb{E}_t[\phi_t(g_*)] \\ &= \Delta g_* - \mathbb{E}_t[\phi_t(g_*)] \\ &= \Psi(g_*). \end{aligned}$$

The only other cases include $h_* = u_* + v$ for $v \in \ker(A)$ since these are precisely the equality cases of the generalized Cauchy-Schwarz inequality. Note that for such h_* ,

$$g(h_*) = (h_*)^T Gh_* = (u_* + v)^T G(u_* + v) = g(u_*) + 2v^T Gu_* + v^T Gv = g_*$$

and $h(u_* + v) = e^T G(u_* + v) = h(u_*)$. These cases are also global minimizers, which gives us the total set of globally optimal solutions:

$$\Theta_* := \left\{ \pm \sqrt{\frac{\sigma^2 + \|Ae\|^2}{\|Ae\|^2}} e \right\} + \ker(A).$$

Finally, we consider the Wasserstein distance bound. Note that for any $u \neq 0$, we have that

$$\begin{aligned} W_2^2(\mathcal{N}(0, ee^T), \mathcal{N}(0, uu^T)) &= \text{tr} \left(ee^T + uu^T - 2((uu^T)^{1/2} ee^T (uu^T)^{1/2})^{1/2} \right) \\ &= 1 + \|u\|^2 - 2\text{tr} \left[\left(\frac{uu^T}{\|u\|} ee^T \frac{uu^T}{\|u\|} \right)^{1/2} \right] \\ &= 1 + \|u\|^2 - 2\text{tr} \left[\left(\frac{(e^T u)^2}{\|u\|^2} uu^T \right)^{1/2} \right] \\ &= 1 + \|u\|^2 - 2\text{tr} \left[\frac{|e^T u|}{\|u\|^2} uu^T \right] \\ &= 1 + \|u\|^2 - 2|e^T u|. \end{aligned}$$

In our case, note that $u_* = q \pm \lambda_* e$ where $q \in \ker(A)$ and $\lambda_* = \sqrt{1 + \sigma^2/\|Ae\|^2} > 1$. Since $e \in \text{Im}(A^T)$, $e^T q = 0$. Hence $|e^T u_*| = \lambda_*$ and $\|u_*\|^2 = \|q\|^2 + \lambda_*^2$ where we used $\|e\| = 1$ in both equalities. Using the previous result and these properties of u_* , we see that for any $u_* \in \Theta_*$,

$$W_2^2(\mathcal{N}(0, ee^T), \mathcal{N}(0, u_* u_*^T)) = 1 + \|q\|^2 + \lambda_*^2 - 2\lambda_* = \|q\|^2 + (\lambda_* - 1)^2.$$

The minimum norm element of Θ_* is precisely given by either $\lambda_* e$ or $-\lambda_* e$. This is because e and q are orthogonal for any $q \in \ker(A)$, so for any $u_* \in \Theta_*$, $\|u_*\|^2 = \lambda_*^2 \|e\|^2 + \|q\|^2 \geq \lambda_*^2 = \|\lambda_* e\|^2$. Taking $q = 0$ minimizes the norm of $u_* \in \Theta_*$. $\square$

Strict saddle property A natural question is whether the objective landscape is well-behaved for first-order optimization algorithms. We now analyze the rank-one case and show that the objective landscape has a strict saddle property, namely that all critical points are either global minimizers or strict saddles, i.e., points for which the gradient is zero, but the Hessian exhibits a negative eigenvalue, hence a descent direction.

Theorem 3. *Consider the setting of Theorem 1 with $r = 1$ and $E = e \in \mathbb{R}^d$ with unit norm. Then the objective $\mathcal{L}$ under the parameterization $\theta(u) = (u, u/\|u\|)$ satisfies the following: the set of critical points for $u \neq 0$ is precisely given by*

$$\Omega := \{\sqrt{1 + \sigma^2}e, -\sqrt{1 + \sigma^2}e\} \cup \Omega^\perp \text{ where } \Omega^\perp := \{u \in \mathbb{R}^d : \langle u, e \rangle = 0, \|u\| = \sigma\}.$$

Each point in $\{\sqrt{1 + \sigma^2}e, -\sqrt{1 + \sigma^2}e\}$ is a global minimizer. Moreover, each point in $\Omega^\perp$ is a strict saddle, meaning that for any $u \in \Omega^\perp$, the Hessian $\nabla^2 \mathcal{L}(u)$ is a strictly negative eigenvalue. Hence all local minima are global minima.

Proof of Theorem 3. By Lemma 10, we have the simpler formula

$$\mathcal{L}(\theta) = C + L(u)$$

where

$$L(u) := \mathbb{E}_t \left[\frac{1}{c_t^2} \|u\|^2 - \eta_t (e^T u)^2 - \frac{\|u\|^2}{a_t(a_t + \|u\|^2)} \right]$$

We will use the notation $a_l = \sigma_l^2$, $c_l = a_l + \sigma^2$, and $\eta_l = (2c_l + \|e\|^2)/(c_l^2(c_l + \|e\|^2)^2)$ for $l \sim \text{Unif}[0, 1]$. Furthermore, let $\tilde{c} := \mathbb{E}_l[c_l^{-2}]$, $\tilde{\eta} := \mathbb{E}_l[\eta_l]$ and $\phi_l(g) := \frac{g}{a_l(a_l+g)^2}$ for $g > 0$. Further setting $g(u) = u^T u$, and $h(u) = e^T u$ our objective can be written as

$$L(u) := \tilde{c} u^T u - \tilde{\eta} (e^T u)^2 - \mathbb{E}_l[\phi_l(u^T u)].$$

An elementary calculation shows that $\phi'_l(g) = \frac{1}{(a_l+g)^2}$ and $\phi''_l(g) = -2(a_l+g)^{-3}$ for $g > 0$.

Note that

$$\nabla L(u) = 2 \left((\tilde{c} - \mathbb{E}_t[\phi'_t(\|u\|^2)])u - \tilde{\eta} e^T u e \right) = 0$$

if and only if

$$(\tilde{c} - \mathbb{E}_t[\phi'_t(\|u\|^2)])u = \tilde{\eta} e^T u e.$$

Note that $u \in \{\pm\sqrt{1+\sigma^2}e\}$ satisfies this condition. On the other hand, consider a decomposition $u = \beta e + s$ where $s \perp e$. Set $F(r) = \tilde{c} - \mathbb{E}_t[\phi'_t(r)]$. Then $u = \beta e + s$ is a critical point if and only if

$$\beta(F(\beta^2 + \|s\|^2) - \tilde{\eta})e + F(\beta^2 + \|s\|^2)s = 0.$$

Suppose $\beta = 0$. Then the equation above requires $F(\|s\|^2)s = 0$. If $s \neq 0$, then we need $\|s\| = \sigma$ since $F(\sigma^2) = 0$. If $\beta \neq 0$, then we require $F(\beta^2 + \|s\|^2) = \tilde{\eta} > 0$, so $s = 0$, which means $F(\beta^2) = \tilde{\eta}$. This condition is indeed satisfied by $\beta^2 = 1 + \sigma^2$, i.e., $\beta = \pm\sqrt{1+\sigma^2}$.

Hence the critical points are

$$\left\{ \pm\sqrt{1+\sigma^2}e \right\} \cup \{u : u \perp e, \|u\| = \sigma\}.$$

Note that the Hessian is

$$\nabla^2 L(u) = 4F'(\|u\|^2)uu^T + 2F(\|u\|^2)I_d - 2\tilde{\eta}ee^T.$$

Note that for critical points u such that $u \perp e$ and $\|u\| = \sigma$, we have

$$\nabla^2 L(u)e = 4F'(\|u\|^2)uu^T e + 2F(\|u\|^2)e - 2\tilde{\eta}e = 2(F(\sigma^2) - \tilde{\eta})e = -2\tilde{\eta}e$$

so e is an eigenvector of $\nabla^2 L(u)$ with eigenvalue $-2\tilde{\eta} < 0$, which is strictly negative. Hence e is a descent direction of the objective $L(u)$ so u is a strict saddle point. If $u = \pm\sqrt{1+\sigma^2}e$, then

$$\nabla^2 L(u) = 4(1+\sigma^2)F'(1+\sigma^2)ee^T + 2F(1+\sigma^2)I_d - 2\tilde{\eta}ee^T = (4(1+\sigma^2)F'(1+\sigma^2) - 2\tilde{\eta})ee^T + 2\tilde{\eta}I_d.$$

Note that for directions orthogonal to e , $v \perp e$,

$$\nabla^2 L(u)v = 2\tilde{\eta}v \text{ with } 2\tilde{\eta} > 0.$$

Along e , we have

$$\nabla^2 L(u)e = (2\tilde{\eta} + 4(1+\sigma^2)F'(1+\sigma^2) - 2\tilde{\eta})e = 4(1+\sigma^2)F'(1+\sigma^2)e$$

where $4(1+\sigma^2)F'(1+\sigma^2) > 0$ so $\nabla^2 L(u)$ is in fact positive definite at $u = \pm\sqrt{1+\sigma^2}e$. Hence such points are global minimizers while other points are strict saddles. $\square$

Multiple noise scales It is possible to extend Theorem 1 to the setting in which one has access to a dataset of varying noise levels $\sigma \sim p(\sigma)$ (as in the experiments discussed in Appendix 6.6.3.4). An illustrative example is the case when we have noisy images $y = x + \sigma z$ where σ comes from a finite set of noise levels $\{\sigma_1, \dots, \sigma_K\}$. This can be modeled as $\sigma \sim p(\sigma) = \sum_{k=1}^K \pi_k \delta_{\sigma_k}$ where $\pi_k \geq 0$, $\sum_{k=1}^K \pi_k = 1$ and $\sigma_k > 0$ for each $k \in [K]$. We will minimize the following objective that also considers noise at different scales:

$$\tilde{\mathcal{L}}(\theta) := \mathbb{E}_{\sigma \sim p(\sigma)} \mathcal{L}(\theta) := \mathbb{E}_{\sigma \sim p(\sigma), t \sim \text{Unif}(0,1)} \mathbb{E}_{x_t \sim p_{G_\theta}^{\sigma_t}} \left[\|s_{\sigma, \sigma_t}(x_t) - \nabla \log p_{G_\theta}^{\sigma_t}(x_t)\|_2^2 \right]$$

We prove that, when the noise levels follow a general distribution $\sigma \sim p(\sigma)$, we can also characterize minimizers of the above loss. In particular, the scaling of the covariance now depends on $p(\sigma)$ and (σ_t) . We will show through an example with a finite set of noise levels how the distilled generator can outperform any of the noisy teachers.

Theorem 4. Consider the same setting as Theorem 1. Then for any distribution $p(\sigma)$ supported on $(0, \infty)$, there exists a unique $\lambda^* > 0$ that depends on (σ_t) and $p(\sigma)$ such that the global minimizers of $\tilde{\mathcal{L}}$ over Θ , denoted by $\theta_\sigma^* := (U^*, V_\sigma^*)$, satisfy the following:

$$U^* = EQ \text{ for some orthogonal matrix } Q \text{ and } (V_\sigma^*)^T V_\sigma^* = \lambda^* I_r. \quad (6.11)$$

In particular, λ^* is the unique solution to the equation

$$\mathbb{E}_{p(\sigma), t} [(\sigma^2 + \sigma_t^2 + 1)^{-2}] = \mathbb{E}_t [(\lambda^* + \sigma_t^2)^{-2}].$$

Hence the learned distilled distribution is given by $p_{G_{\theta_\sigma^*}} = \mathcal{N}(0, \lambda^* E E^T)$.

Proof. The proof of this theorem is very similar to the proof of Theorem 1, with a particular modification of Lemma 9 needed to find the precise form of the minimizer. In particular, one can show using similar arguments that the optimal U is given by $U^* = EQ$ for orthogonal Q . To find the form of V , one needs to minimize the following function over feasible V , which only depends on the eigenvalues of $V^T V$:

$$\sum_{i=1}^r F(\lambda_i(V^T V)) := \sum_{i=1}^r \mathbb{E}_{p(\sigma), t} \left[\frac{\lambda_i(V^T V)}{(\sigma^2 + \sigma_t^2 + 1)^2} - \frac{\lambda_i(V^T V)}{\sigma_t^2(\lambda_i(V^T V) + \sigma_t^2)} \right]$$

where $F(u) := \mathbb{E}_{p(\sigma)}[f_\sigma(u)]$ and f_σ is the function defined in Lemma 9. We claim that F is strictly convex and has a unique minimizer λ^* . Note that for any $u > 0$, its derivatives are

$$F'(u) = \mathbb{E}_{p(\sigma), t} \left[\frac{1}{(\sigma^2 + \sigma_t^2 + 1)^2} \right] - \mathbb{E}_t \left[\frac{1}{(\sigma_t^2 + u)^2} \right] \text{ and}$$

$$F''(u) = \mathbb{E}_{p(\sigma), t} \left[\frac{2}{(\sigma_t^2 + u)^3} \right] > 0.$$

Moreover, note that $u \mapsto \psi(u) := \mathbb{E}_{p(\sigma), t} [(\sigma_t^2 + u)^{-2}]$ is strictly decreasing with $\lim_{u \rightarrow \infty} \psi(u) = 0$ and $\lim_{u \rightarrow 0^+} \psi(u) = \mathbb{E}_t[\sigma_t^{-4}] > \mathbb{E}_{p(\sigma), t} [(\sigma^2 + \sigma_t^2 + 1)^{-2}]$. Note the last strict inequality follows by strict monotonicity of ψ . Hence there must exist a unique $\lambda^* \in (0, \infty)$ that minimizes F satisfying $F'(\lambda^*) = 0$, i.e., $\mathbb{E}_{p(\sigma), t} [(\sigma^2 + \sigma_t^2 + 1)^{-2}] = \mathbb{E}_t [(\lambda^* + \sigma_t^2)^{-2}]$. Thus the above function for invertible $V^T V$ is minimized when the gram matrix of V_σ^* has equal eigenvalues $\lambda_i((V_\sigma^*)^T V_\sigma^*) = \lambda^*$ for all $i \in [r]$. Since all of its eigenvalues are the same, by the Spectral Theorem, we must have that $(V_\sigma^*)^T V_\sigma^* = \lambda^* I_r$. $\square$

Example with a finite number of noise levels: Consider the case when $p(\sigma) = \sum_{k=1}^K \pi_k \delta_{\sigma_k}$ for $\sigma_k > 0$ with $\pi_k \geq 0$ and $\sum_{k=1}^K \pi_k = 1$. Then λ^* is the unique solution to

$$\sum_{k=1}^K \pi_k \mathbb{E}_t [(\sigma_k^2 + \sigma_t^2 + 1)^{-2}] = \mathbb{E}_t [(\lambda^* + \sigma_t^2)^{-2}].$$

Since the right-hand side is strictly decreasing with respect to λ^* , one can show that we always have

$$1 + \sigma_{\min}^2 \leq \lambda^* \leq 1 + \sigma_{\max}^2, \quad \sigma_{\min} := \min_k \sigma_k, \quad \sigma_{\max} := \max_k \sigma_k.$$

The precise value of λ^* interestingly depends now on the noise schedule (σ_t) . In particular, when the noise schedule σ_t has much smaller values, the right-hand side increases, requiring λ^* to decrease to satisfy the equation. Likewise, when σ_t focuses on larger noise levels, the right-hand side goes down, requiring a larger λ^* for the equation to be satisfied.

One can also give a mathematical condition on when the distilled distribution is closer in Wasserstein distance to each of the noisy distributions to the ground-truth. In particular, consider the low-rank regime $r \ll d$. Then if λ^* satisfies

$$\lambda^* < \min_{k \in [K]} \left[1 + \sqrt{\left(\sqrt{1 + \sigma_k^2} - 1 \right)^2 + \frac{d-r}{r} \sigma_k^2} \right]^2 \approx 1 + \frac{d}{r} \sigma_{\min}^2$$

then

$$W_2^2(p_{G_{\theta^*}}, p_X) < \min_{k \in [K]} W_2^2(p_{Y, \sigma_k}, p_X),$$

i.e., the distilled generator is closer to the ground-truth distribution than every noisy teacher. Note that if the difference between $\sigma_{\max}$ and $\sigma_{\min}$ is not too large and the data is low-rank enough $d \gg r$, then this can be satisfied. For example, if the largest noise level is less than a multiple of the smallest noise level $\sigma_{\max} < \sqrt{d/r} \cdot \sigma_{\min}$, then this would ensure $\lambda^* \leq 1 + \sigma_{\max}^2 < 1 + \frac{d}{r} \sigma_{\min}^2$. As an example, suppose $d/r = 10$. Then as long as $\sigma_{\max} < 3\sigma_{\min}$, the above condition would be satisfied.

6.6.5.3 General guarantees for distillation

Building on our linear analysis, we derive more general performance guarantees for the distilled generator under capacity and optimization assumptions on the teacher and generator. We first consider general corruptions and then specialize the result to denoising, where we give conditions under which the distilled model improves upon the noisy distribution.

To state our results, we set the notation. Let $\mathcal{F}(p||q) := \mathbb{E}_p[\|\nabla \log p(x) - \nabla \log q(x)\|_2^2]$ denote the Fisher divergence between p and q . For a density p , we let $dp = p(x)dx = p dx$. Let $\chi^2(p||q) = \int (p/q - 1)^2 dq$ denote the chi-square divergence between p and q . For notational simplicity, we let $\lesssim$ denote an inequality up to absolute constants so that $a \lesssim b$ if there exists an absolute constant $C > 0$ such that $a \leq Cb$. We let $y = Ax + \sigma\epsilon$ with $\sigma > 0$ and define $p_Y := \mathcal{T}[p_X] := A_\# p_X * \mathcal{N}(0, \sigma^2 I_m)$. We will use the notation

$$x_t = x + \sigma_t \epsilon, y_t = y + \sigma_t \epsilon'$$

with marginals $p_{X,t}, p_{Y,t}$. For the generator G_θ with clean law p_θ and parameters $\theta \in \Theta$, its measurement law is $p_{\theta,Y} := \mathcal{T}[p_\theta]$ with noise-convolved marginals $p_{\theta,Y,t}$. We consider distilling by minimizing

$$\mathcal{L}_{\text{distill}}(\theta) := \mathbb{E}_{t, \tilde{y}_t \sim p_{\theta,Y,t}} \|f_\phi(\tilde{y}_t, t) - f_\psi(\tilde{y}_t, t)\|_2^2$$

where f_ϕ is the teacher and f_ψ is the fake diffusion model. For each t , let $s_{Y,t}$ and $s_{\theta,Y,t}$ denote the scores of $p_{Y,t}$ and $p_{\theta,Y,t}$, respectively. Throughout, we will assume sufficient regularity of the densities so that all scores $\nabla \log p$ and gradients ∇p are well-defined, including those induced by p_θ for $\theta \in \Theta$. Moreover, put

$$\begin{aligned} \delta_\phi(y, t) &:= f_\phi(y, t) - s_{Y,t}(y) & \delta_\psi(\tilde{y}, t) &:= f_\psi(\tilde{y}, t) - s_{\theta,Y,t}(\tilde{y}) \\ \varepsilon_{\phi,2}^2 &:= \sup_t \mathbb{E}_{p_{Y,t}} \|\delta_\phi(y_t, t)\|_2^2, & \varepsilon_{\phi,4}^2 &:= \sup_t (\mathbb{E}_{p_{Y,t}} \|\delta_\phi(y_t, t)\|_2^4)^{1/2}. \end{aligned}$$

We will assume we have optimized our parameter θ to a point $\hat{\theta}$. Define the local fake-diffusion error at $\hat{\theta}$:

$$\varepsilon_\psi^2(\hat{\theta}) := \sup_t \mathbb{E}_{p_{\hat{\theta},Y,t}} \|\delta_\psi(\tilde{y}_t, t)\|_2^2.$$

Finally, let $\Delta(\hat{\theta})$ denote the local density ratio

$$\Delta(\hat{\theta}) := \sup_t \sup_y w_t(y), \quad w_t(y) := p_{\hat{\theta}, Y, t}(y)/p_{Y, t}(y).$$

We will make the following assumptions on the data and learned parameters:

Assumption 5 (Data distribution). Suppose that the corrupted data distribution $p_{Y, t}$ satisfies a uniform Poincaré-like inequality in the sense that there exists a $\lambda_0 > 0$ such that

$$\chi^2(p||p_{Y, t}) \leq \lambda_0^{-1} \int \|\nabla(p/p_{Y, t})\|^2 dp_{Y, t}, \quad \forall t, p \in \{p_{\theta, Y, t}\}_{\theta \in \Theta}.$$

Assumption 6 (Capacity and local near-optimality). There exists a θ^* such that $p_{\theta^*, Y} = p_Y$ and $\hat{\theta}$ satisfies the following for some $\varepsilon_{\text{opt}} \geq 0$:

$$\mathcal{L}_{\text{distill}}(\hat{\theta}) \leq \mathcal{L}_{\text{distill}}(\theta^*) + \varepsilon_{\text{opt}}.$$

Moreover, the fake diffusion network with parameters ψ satisfies $\mathbb{E}_{t, p_{Y, t}} \|f_\phi(y_t, t) - f_\psi(y_t, t)\|_2^2 \leq \mathbb{E}_{t, p_{Y, t}} \|f_\phi(y_t, t) - s_{Y, t}\|_2^2$. We note that this can be relaxed to an upper bound up to a constant.

We will aim to prove the following:

Theorem 5. *Under Assumptions 5 and 6, we have that the following holds:*

1. (**general bound**) the learned distilled distribution in measurement space satisfies

$$\mathbb{E}_t \mathcal{F}(p_{\hat{\theta}, Y, t} || p_{Y, t}) \lesssim \left(\varepsilon_{\phi, 4}^2 \sqrt{\frac{\Delta(\hat{\theta})}{\lambda_0}} + \sqrt{\varepsilon_{\phi, 4}^4 \frac{\Delta(\hat{\theta})}{\lambda_0} + \varepsilon_{\phi, 2}^2 + \varepsilon_\psi^2(\hat{\theta}) + \varepsilon_{\text{opt}}} \right)^2;$$

2. (**measurement injectivity**) if the measurement operator $\mathcal{T} : p \mapsto A_\# p * \mathcal{N}(0, \sigma^2 I_m)$ satisfies $\mathbb{E}_t \mathcal{F}(p_{\hat{\theta}, t} || p_{X, t}) \leq \hat{\kappa} \cdot \mathbb{E}_t \mathcal{F}(p_{\hat{\theta}, Y, t} || p_{Y, t})$ for some $\hat{\kappa} > 0$, then we have that the learned distilled distribution in image space satisfies

$$\mathbb{E}_t \mathcal{F}(p_{\hat{\theta}, t} || p_{X, t}) \lesssim \hat{\kappa} \left(\varepsilon_{\phi, 4}^2 \sqrt{\frac{\Delta(\hat{\theta})}{\lambda_0}} + \sqrt{4\varepsilon_{\phi, 4}^4 \frac{\Delta(\hat{\theta})}{\lambda_0} + \varepsilon_{\phi, 2}^2 + \varepsilon_\psi^2(\hat{\theta}) + \varepsilon_{\text{opt}}} \right)^2;$$

Discussion. This result guarantees a bound on the Fisher divergence between the distilled generator’s measurement distribution and the true measurement distribution. The key idea is that minimizing the distillation loss encourages the generator to learn an image distribution whose *induced* measurements are close to the true measurements. Lemma 12 more explicitly connects the distillation loss to the reverse Fisher divergence between the measurement distributions. The second key component is the second bound, which shows that if the corruption operator satisfies an injectivity property over the data, then we can transfer this bound to the distilled distribution in image space. Hence distillation has the potential to succeed when 1) the distilled generator learns to create images such that, when corrupted further, look like the measurements and 2) the corruption operator is stable or injective over our distributions. We show in the following Corollary that in the instructive case of denoising, the corruption operator is stable and we can give a condition on when distillation can improve over the noisy distribution.

Corollary 1 (Improvement in denoising). *Under the setting of Theorem 5, when $A = I$, we have that the measurement injectivity condition holds for some $\hat{\kappa} > 0$ that depends on the noise schedule and for $\varepsilon_{\phi,2}, \varepsilon_{\phi,4}, \varepsilon_{\psi}(\hat{\theta}), \varepsilon_{\text{opt}}$ sufficiently small, we have that the distilled distribution improves upon the noisy distribution*

$$\mathbb{E}_t \mathcal{F}(p_{\hat{\theta},t} || p_{X,t}) < \mathbb{E}_t \mathcal{F}(p_{Y,t} || p_{X,t}).$$

To prove these results, we require a number of technical lemmas. The first is a change of measure result that will be useful in transferring expectations.

Lemma 11. *For fixed t , let $e_{\phi}(y, t) := \|\delta_{\phi}(y, t)\|_2^2$. Then we have that*

$$\mathbb{E}_{p_{\theta,Y,t}} e_{\phi}(y, t) \leq \varepsilon_{\phi,2}^2 + \varepsilon_{\phi,4}^2 \sqrt{\chi^2(p_{\theta,Y,t} || p_{Y,t})}.$$

Proof of Lemma 11. Suppose we set $q = p_{Y,t}$, $p = p_{\theta,Y,t}$, and $w = dp/dq$. Recall that for a density q , the induced $L^2(q)$ norm is given by $\|f\|_{L^2(q)}^2 = \int f^2 dq$. Then we have via an

application of Cauchy-Schwarz that

$$\begin{aligned}
|\mathbb{E}_p e_\phi(y, t) - \mathbb{E}_q e_\phi(y, t)| &= \left| \int (w(y) - 1) e_\phi(y, t) dq \right| \\
&\leq \|w - 1\|_{L^2(q)} \|e_\phi\|_{L^2(q)} \\
&= \sqrt{\chi^2(p||q)} \cdot (\mathbb{E}_q e_\phi^2(y, t))^{1/2}.
\end{aligned}$$

The result follows by using the definitions of $\varepsilon_{\phi,2}$ and $\varepsilon_{\phi,4}$. $\square$

The next Lemma is crucial, in that it shows how the distillation loss from SiD [ZZW24] encourages the generator G_θ to produce images whose measurements match the distribution of the true measurements.

Lemma 12. *For $\hat{\theta}$, define $\Gamma(\hat{\theta}) := \mathbb{E}_t \left[\varepsilon_{\phi,4}^2 \sqrt{\chi^2(p_{\hat{\theta},Y,t}||p_{Y,t})} \right]$. Then we have that the distillation loss satisfies*

$$\frac{1}{2} \mathbb{E}_t \mathcal{F}(p_{\hat{\theta},Y,t}||p_{Y,t}) - 2\varepsilon_\psi^2(\hat{\theta}) - 2\varepsilon_{\phi,2}^2 - 2\Gamma(\hat{\theta}) \leq \mathcal{L}_{\text{distill}}(\hat{\theta})$$

Proof of Lemma 12. We first consider the decomposition

$$f_\phi - f_\psi = (s_{Y,t} - s_{\hat{\theta},Y,t}) + \delta_\phi - \delta_\psi.$$

Using $\|x + y - w\|_2^2 \leq 3(\|x\|_2^2 + \|y\|_2^2 + \|w\|_2^2)$ for $x = s_{Y,t} - s_{\hat{\theta},Y,t}$, $y = \delta_\phi$ and $w = \delta_\psi$ and taking expectations, we have that for fixed t ,

$$\begin{aligned}
&\mathbb{E}_{p_{\hat{\theta},Y,t}} \|f_\phi(\tilde{y}_t, t) - f_\psi(\tilde{y}_t, t)\|_2^2 \\
&\leq 3\mathbb{E}_{p_{\hat{\theta},Y,t}} \|s_{Y,t}(\tilde{y}_t, t) - s_{\hat{\theta},Y,t}(\tilde{y}_t, t)\|_2^2 + 3\mathbb{E}_{p_{\hat{\theta},Y,t}} e_\phi(\tilde{y}_t, t) + 3\mathbb{E}_{p_{\hat{\theta},Y,t}} \|\delta_\psi(\tilde{y}_t, t)\|_2^2 \\
&\leq 3\mathbb{E}_{p_{\hat{\theta},Y,t}} \|s_{Y,t}(\tilde{y}_t, t) - s_{\hat{\theta},Y,t}(\tilde{y}_t, t)\|_2^2 + 3 \left(\varepsilon_{\phi,2}^2 + \varepsilon_{\phi,4}^2 \sqrt{\chi^2(p_{\hat{\theta},Y,t}||p_{Y,t})} \right) + 3\varepsilon_\psi^2(\hat{\theta})
\end{aligned}$$

where the last line follows by Lemma 11 and the definition of $\varepsilon_\psi^2(\hat{\theta})$. Taking an expectation over t yields

$$\mathcal{L}_{\text{distill}}(\hat{\theta}) \leq 3\mathbb{E}_t \mathcal{F}(p_{\hat{\theta},Y,t}||p_{Y,t}) + 3\varepsilon_{\phi,2}^2 + 3\Gamma(\hat{\theta}) + 3\varepsilon_\psi^2(\hat{\theta}).$$

The lower bound holds by using the bound $\|x + y - w\|_2^2 \geq \frac{1}{2}\|x\|_2^2 - 2(\|y\|_2^2 + \|w\|_2^2)$, applying the same bounds, and taking expectations. Note that this bound holds because for any z ,

$$\|x + z\|_2^2 = \|x\|_2^2 + 2\langle x, z \rangle + \|z\|_2^2.$$

Recall Young's inequality: $|\langle x, z \rangle| \leq \frac{\alpha^2}{2}\|x\|_2^2 + \frac{1}{2\alpha^2}\|z\|_2^2$ for $\alpha > 0$. Hence we have the lower bound

$$\|x + z\|_2^2 \geq \|x\|_2^2 - \alpha^2\|x\|_2^2 - \alpha^{-2}\|z\|_2^2 + \|z\|_2^2.$$

Choosing $\alpha^2 = 1/2$, setting $z = y - w$, and using $\|y - w\|_2^2 \leq 2(\|y\|_2^2 + \|w\|_2^2)$ yields the desired inequality. $\square$

An additional ingredient we need is control over the χ^2 distance and relating it to the Fisher divergence. For that, we need the following Lemma.

Lemma 13. *Under Assumption 5, we have that*

$$\Gamma(\hat{\theta}) \leq \varepsilon_{\phi,4}^2 \sqrt{\frac{\Delta(\hat{\theta})}{\lambda_0}} \cdot \left(\mathbb{E}_t \mathcal{F}(p_{\hat{\theta},Y,t} \| p_{Y,t}) \right)^{1/2}.$$

Proof of Lemma 13. We first fix t and set $q = p_{Y,t}$, $p = p_{\hat{\theta},Y,t}$ and $w = p/q$. Then Assumption 5 yields

$$\chi^2(p||q) \leq \lambda_0^{-1} \int \|\nabla w\|_2^2 dq = \lambda_0^{-1} \int w \frac{\|\nabla w\|_2^2}{w} dq \leq \frac{\Delta(\hat{\theta})}{\lambda_0} \int \frac{\|\nabla w\|_2^2}{w} dq = \frac{\Delta(\hat{\theta})}{\lambda_0} \mathcal{F}(p||q)$$

where the first line follows by assumption, the second inequality follows by definition of $\Delta(\hat{\theta})$ and the last equality follows by definition of the Fisher divergence. Indeed, for the last equality, note that if $w = p/q$, then $\nabla w = \nabla(p/q) = \frac{p}{q}(\nabla \log p - \nabla \log q) = w(s_p - s_q)$ where s_p and s_q are the scores of p and q , respectively. This ensures that

$$\int \frac{\|\nabla w\|_2^2}{w} dq = \int w \|s_p - s_q\|^2 dq = \int \|s_p - s_q\|^2 p dx = \mathcal{F}(p||q).$$

Taking square roots and applying an expectation over t along with Jensen's inequality for the concave map $v \mapsto \sqrt{v}$ yields the desired bound. $\square$

Armed with these technical results, we now prove Theorem 5.

Proof of Theorem 5. First, recall that by Lemma 12, we have the lower bound

$$\mathcal{L}_{\text{distill}}(\hat{\theta}) \geq \frac{1}{2} \mathbb{E}_t \mathcal{F}(p_{\hat{\theta}, Y, t} \| p_{Y, t}) - 2\varepsilon_{\psi}^2(\hat{\theta}) - 2\varepsilon_{\phi, 2}^2 - 2\Gamma(\hat{\theta}). \quad (6.12)$$

Moreover, by definition of θ^* , we have that

$$\mathcal{L}_{\text{distill}}(\theta^*) = \mathbb{E}_{t, p_{\theta^*, Y, t}} \|f_{\phi} - f_{\psi}\|_2^2 = \mathbb{E}_{t, p_{Y, t}} \|f_{\phi} - f_{\psi}\|_2^2 \leq \mathbb{E}_{t, p_{Y, t}} \|f_{\phi} - s_{Y, t}\|_2^2 \leq \varepsilon_{\phi, 2}^2$$

where we used $p_{\theta^*, Y} = p_Y$ in the second equality and the assumption on ψ in the second-to-last inequality. Then note that by Assumption 6, we have that

$$\mathcal{L}_{\text{distill}}(\hat{\theta}) \leq \mathcal{L}_{\text{distill}}(\theta^*) + \varepsilon_{\text{opt}} \leq \varepsilon_{\phi, 2}^2 + \varepsilon_{\text{opt}}. \quad (6.13)$$

Combining 6.12 and 6.13 along with Lemma 13 yields the quadratic inequality $\frac{1}{2}X \leq A + B\sqrt{X}$ where we have set $X := \mathbb{E}_t \mathcal{F}(p_{\hat{\theta}, Y, t} \| p_{Y, t})$, $A := 3\varepsilon_{\phi, 2}^2 + \varepsilon_{\text{opt}} + 2\varepsilon_{\psi}^2(\hat{\theta})$ and $B := 2\varepsilon_{\phi, 4}^2 \sqrt{\frac{\Delta(\hat{\theta})}{\lambda_0}}$. Solving this inequality, we have that $X \leq (B + \sqrt{B^2 + 2A})^2$. Substituting the values of X , A , and B yields the desired result. □

Finally, we show the denoising Corollary.

Proof of Corollary 1. The Corollary is a consequence of the fact that when $A = I$, we have that $p_{Y, t} = \mathcal{T}[p_X] * \mathcal{N}(0, \sigma_t^2 I) = p_X * \mathcal{N}(0, \sigma^2 I) * \mathcal{N}(0, \sigma_t^2 I) = p_X * \mathcal{N}(0, (\sigma^2 + \sigma_t^2)I)$. Hence if we define the time-shift map $\tau(t)$ by $\sigma_{\tau(t)}^2 = \sigma_t^2 + \sigma^2$, we have that for any θ and t , $\mathcal{F}(p_{\theta, Y, t} \| p_{Y, t}) = \mathcal{F}(p_{\theta, \tau(t)} \| p_{X, \tau(t)})$. Denote the distribution of $t \sim \rho$ where ρ has support in $\tau([t_{\min}, t_{\max}])$ and set

$$\hat{\kappa} := \sup_{t \in [t_{\min}, t_{\max}]} \frac{\rho(\tau(t))\tau'(t)}{\rho(t)} \in (0, \infty).$$

Then we have that

$$\mathbb{E}_t \mathcal{F}(p_{\hat{\theta}, t} \| p_{X, t}) \leq \hat{\kappa} \cdot \mathbb{E}_t \mathcal{F}(p_{\hat{\theta}, \tau(t)} \| p_{X, \tau(t)}) = \hat{\kappa} \cdot \mathbb{E}_t \mathcal{F}(p_{\hat{\theta}, Y, t} \| p_{Y, t}).$$

Define $\Delta_\sigma := \mathbb{E}_t \mathcal{F}(p_{Y,t} || p_{X,t})$. Then using the measurement injectivity bound in Theorem 5, there exists a universal constant C such that if

$$\hat{\kappa} \cdot \left(\varepsilon_{\phi,4}^2 \sqrt{\frac{\Delta(\hat{\theta})}{\lambda_0}} + \sqrt{\varepsilon_{\phi,4}^4 \frac{\Delta(\hat{\theta})}{\lambda_0} + \varepsilon_{\phi,2}^2 + \varepsilon_\psi^2(\hat{\theta}) + \varepsilon_{\text{opt}}} \right)^2 \leq \Delta_\sigma / C$$

then we have

$$\mathbb{E}_t \mathcal{F}(p_{\hat{\theta},t} || p_{X,t}) < \mathbb{E}_t \mathcal{F}(p_{Y,t} || p_{X,t}).$$

□

Connection to mode-seeking. The previous theory connects to mode-seeking behavior in the following way. In particular, under the setting of our theory, obtaining a small Fisher divergence encourages small reverse KL-divergence. To see this, recall that the KL-divergence and χ^2 -divergence satisfy $D_{\text{KL}}(p||q) \leq \chi^2(p||q)$ when $p \ll q$ (which follows by the inequality $\log t \leq t - 1$ for $t > 0$). Following the proof of Lemma 13, this shows that we have the inequality

$$D_{\text{KL}}(p_{\theta,Y,t} || p_{Y,t}) \leq \chi^2(p_{\theta,Y,t} || p_{Y,t}) \leq \frac{\Delta}{\lambda_0} \mathcal{F}(p_{\theta,Y,t} || p_{Y,t})$$

where $\Delta = \Delta(\theta)$ is the density ratio of $p_{\theta,Y,t}$ and $p_{Y,t}$ and λ_0 is the constant in Assumption 5. Hence if one can minimize the Fisher divergence and the ratio Δ , this will also encourage smaller reverse KL-divergence (mode-seeking behavior). A further theoretical investigation of this connection is an important direction for future work.

6.6.6 Phase I: Variants of Pretraining Details

In Section 6.3.1, we introduced four variants of diffusion model pretraining methods. Here, we provide the training details for each of these variants.

6.6.6.1 Standard Diffusion

Standard Diffusion aims to learn the distribution of $y^{(i)} = \mathcal{A}(x^{(i)}) + \sigma\varepsilon$ directly. To do this, we train the diffusion model on the corrupted dataset $\{y^{(i)}\}_{i=1}^N$. The training objective is given by:

$$\mathcal{L}_{\text{SD}} = \mathbb{E}_{\sigma_t, y, \varepsilon} [\lambda(t) \|f_\phi(y + \sigma_t \varepsilon, t) - y\|_2^2], \quad (6.14)$$

where $\lambda(t)$ is a time-dependent weighting function and σ_t is sampled from a predefined noise schedule. We are using the EDM schedule for $\lambda(t)$ and σ_t same as in [KAA22].

The detailed training procedure is summarized in Algorithm 13.

Algorithm 13 Standard Diffusion Training

```

1: procedure Standard-Diffusion( $\{y^{(i)}\}_{i=1}^N, \sigma, p(\sigma_t), K$ )
2:   for  $k = 1$  to  $K$  do
3:     Sample a batch  $y \sim \{y^{(i)}\}_{i=1}^N$ 
4:     Sample noise level  $\sigma_t \sim p(\sigma_t)$ 
5:     Sample noise  $\varepsilon \sim \mathcal{N}(0, I_d)$ 
6:     Construct noisy input:  $y_t = y + \sigma_t \cdot \varepsilon$ 
7:     Update parameters  $\phi$  via gradient descent on  $\mathcal{L}_{\text{SD}}$  (Eq. 6.14)
8:   end for
9:   return Trained diffusion model  $f_\phi$ 
10: end procedure

```

6.6.6.2 Diffusion for Denoising

Diffusion for denoising aims to learn the distribution of $\mathcal{A}(x^{(i)})$ from noisy observations $y^{(i)} = \mathcal{A}(x^{(i)}) + \sigma\varepsilon$, where σ is a known (constant) noise level. The objective is to mitigate

the impact of additive noise during training. Related denoising strategies have been explored in [DCD25].

The Tweedie adjustment is compatible with more advanced diffusion training techniques (Appendix 6.6.6.3 and Appendix 6.6.6.4), tailored for random inpainting and Fourier-space inpainting, respectively.

Training loss. We use the following objective:

$$\mathcal{L}_D = \mathbb{E}_{\sigma_t, y, \varepsilon} \left[\left\| \frac{\sigma_t^2 - \sigma^2}{\sigma_t^2} f_\phi(y_t, t) + \frac{\sigma^2}{\sigma_t^2} y_t - y \right\|_2^2 \right], \quad (6.15)$$

where σ_t is sampled from the diffusion noise schedule and clipped so that $\sigma_t \geq \sigma$, and $y_t := y + \sqrt{\sigma_t^2 - \sigma^2} \varepsilon$.

This formulation induces differences in both training and inference. Full details are given in Algorithm 14 and Algorithm 15. For sampling, we denote by **Teacher-Full** the standard EDM sampling from $\sigma_{\max}$ down to 0, and by **Teacher-Truncated** the EDM sampling truncated at the data noise level σ . See Algorithm 15 for details.

Algorithm 14 Diffusion for Denoising (Training)

- 1: **procedure** Diffusion-for-Denoising-Training($\{y^{(i)}\}_{i=1}^N, \sigma, p(\sigma_t), K$)
 - 2: **for** $k = 1$ to K **do**
 - 3: Sample a minibatch $y \sim \{y^{(i)}\}_{i=1}^N$, $\sigma_t \sim p(\sigma_t)$, $\varepsilon \sim \mathcal{N}(0, I_d)$
 - 4: $\sigma_t \leftarrow \max\{\sigma, \sigma_t\}$ ▷ Clip noise level
 - 5: $y_t \leftarrow y + \sqrt{\sigma_t^2 - \sigma^2} \varepsilon$
 - 6: Update f_ϕ by descending $\nabla_\phi \mathcal{L}_D$ in Eq. 6.15
 - 7: **end for**
 - 8: **return** trained diffusion model f_ϕ
 - 9: **end procedure**
-

Algorithm 15 Diffusion for Denoising (Sampling)

```
1: procedure Diffusion-for-Denoising-Sampling( $f_\phi, \sigma, \{\sigma_t\}_{t=0}^T$ )
2:   Sample  $x_T \sim \mathcal{N}(0, \sigma_T^2 I_d)$ 
3:   for  $t = T, T-1, \dots, 1$  do
4:      $\hat{x}_0 \leftarrow f_\phi(x_t, t)$ 
5:     if truncation enabled  $\wedge \sigma_{t-1} < \sigma$  then
6:       return  $\hat{x}_0$  ▷ Teacher-Truncated
7:     end if
8:      $x_{t-1} \leftarrow x_t - \frac{\sigma_t - \sigma_{t-1}}{\sigma_t} (x_t - \hat{x}_0)$  ▷ EDM-style update
9:   end for
10:  return  $\hat{x}_0$  ▷ Teacher-Full
11: end procedure
```

6.6.6.3 Diffusion for Random Inpainting Task

Daras et al. [DSD23] introduced a diffusion training objective specifically designed for the random inpainting task. The goal is to learn the underlying clean data distribution p_X from partial observations of the form $y^{(i)} = Mx^{(i)}$, where M is a binary inpainting mask applied to the image. To introduce further stochasticity, a secondary corruption mask $\tilde{M}$ is applied during training.

The training loss for random inpainting is given by:

$$\mathcal{L}_{\text{RI}} = \mathbb{E}_{\sigma_t, y, \varepsilon} \left[\left\| M \left(f_\phi(\tilde{M}, \tilde{M}y_t, t) - y \right) \right\|_2^2 \right], \quad (6.16)$$

where $y_t = y + \sigma_t \varepsilon$, and f_ϕ is conditioned on both the corrupted observation and the masking pattern. All training schedules and hyperparameters follow the same configuration as in the original paper [DSD23].

The full training procedure is outlined in Algorithm 16.

Algorithm 16 Diffusion Training for Random Inpainting

```
1: procedure Diffusion-Inpainting-Training( $\{y^{(i)}\}_{i=1}^N, M, p(\sigma_t), K$ )
2:   for  $k = 1$  to  $K$  do
3:     Sample batch  $y \sim \{y^{(i)}\}_{i=1}^N$ 
4:     Sample noise level  $\sigma_t \sim p(\sigma_t)$ 
5:     Sample noise  $\varepsilon \sim \mathcal{N}(0, I_d)$ 
6:     Sample a further corruption mask  $\tilde{M}$  conditioned on  $M$ 
7:     Compute  $y_t \leftarrow y + \sigma_t \cdot \varepsilon$ 
8:     Update  $f_\phi$  using gradient descent on  $\mathcal{L}_{\text{RI}}$  in Eq. 6.16
9:   end for
10:  return Trained diffusion model  $f_\phi$ 
11: end procedure
```

6.6.6.4 Diffusion Training for Fourier Space Inpainting

[ADL25] introduced a diffusion training objective specifically designed for the multi-coil MRI on Fourier Space. The goal is to learn the underlying clean data distribution p_X from

$$y = \underbrace{\left(\sum_{i=1}^{N_c} S_i^H \mathcal{F}^{-1} M \mathcal{F} S_i \right)}_{\mathcal{A}} x,$$

where S_i denotes the coil sensitivity profile of the i -th coil, $\mathcal{F}$ is the Fourier transform, and M is the masking operator in Fourier space. The further corrupted observation would be

$$\tilde{y} = \underbrace{\left(\sum_{i=1}^{N_c} S_i^H \mathcal{F}^{-1} \tilde{M} \mathcal{F} S_i \right)}_{\tilde{\mathcal{A}}} x.$$

The training loss for multi-coil MRI is given by:

$$\mathcal{L}_{\text{FS}} = \mathbb{E}_{\sigma_t, y, \varepsilon} \left[\left\| \mathcal{A}(f_\phi(\tilde{y}_t, \tilde{M}, t) - y) \right\|_2^2 \right], \quad (6.17)$$

where $y_t = y + \sigma_t \varepsilon$, and f_ϕ is conditioned on both the corrupted observation and the masking pattern.

The full training procedure is outlined in Algorithm 17. All training schedules and hyperparameters follow the same configuration as in the original paper [ADL25].

Algorithm 17 Diffusion Training for Fourier Space Inpainting

```

1: procedure Diffusion-FSInpainting-Training( $\{y^{(i)}\}_{i=1}^N, M, p(\sigma_t), K, S$ )
2:   for  $k = 1$  to  $K$  do
3:     Sample batch  $y \sim \{y^{(i)}\}_{i=1}^N$ 
4:     Sample noise level  $\sigma_t \sim p(\sigma_t)$ 
5:     Sample noise  $\varepsilon \sim \mathcal{N}(0, I_d)$ 
6:     Sample a further corruption mask  $\tilde{M}$  conditioned on  $M$ 
7:     Compute  $y_t \leftarrow y + \sigma_t \cdot \varepsilon$ 
8:     Update  $f_\phi$  using gradient descent on  $\mathcal{L}_{\text{RI}}$  in Eq. 6.17
9:   end for
10:  return Trained diffusion model  $f_\phi$ 
11: end procedure

```

6.6.7 Phase II: Distillation

In Section 6.2.3, we introduced the SiD generator loss (Eq. 6.5). The SiD objective admits additional design choices, as discussed in the original paper [ZZW24]. For completeness, we present the exact generator-loss formulation used in our implementation and defer details such as time-step scheduling and hyperparameter settings to the cited work.

SiD generator loss. Let $x_g = G_\theta(z)$ and $x_t = x_g + \sigma_t \varepsilon$ with $z, \varepsilon \sim \mathcal{N}(0, I_d)$. The loss is

$$\begin{aligned} \mathcal{L}_{\text{SiD}}(w_g) = \mathbb{E}_{z, t, \varepsilon} \Big[& (1 - \alpha) \lambda(t) \|f_\psi(x_t, t) - f_\phi(x_t, t)\|_2^2 \\ & + \lambda(t) (f_\phi(x_t, t) - f_\psi(x_t, t))^\top (f_\psi(x_t, t) - x_g) \Big], \end{aligned} \quad (6.18)$$

where f_ϕ is the teacher (pretrained diffusion model), f_ψ is the fake diffusion model, $\lambda(t)$ is a time-dependent weight, and α balances the two terms. Unless otherwise noted, we use $\alpha = 1.2$, following [ZZW24], which reports strong performance across tasks.

Note that, following common practice, we initialize both the auxiliary diffusion model f_ψ and the generator G_θ from the teacher diffusion model f_ϕ [ZZW24, YGP24, YGZ24], which is crucial for facilitating and stabilizing training. Importantly, all three networks share the same architecture and capacity. Nonetheless, the generator has been shown to outperform the teacher [ZZW24], as it avoids the accumulation error inherent in multi-step sampling.

6.6.8 Implementation Details

Hardware and measurement. Unless otherwise noted, all pretraining and distillation runs use 8×NVIDIA A6000 GPUs. Inference wall-clock time is measured on 4× NVIDIA A6000 GPUs with a batch size of 1024. Images are normalized to $[-1, 1]$ prior to adding Gaussian noise.

Teacher pretraining: denoising task. For the denoising task, we follow the EDM setup [KAA22]. On CIFAR-10, we train for **200 M** image iterations to match the EDM computational budget; on FFHQ, CelebA-HQ, and AFHQ-v2 we train for **100 M** iterations (half of EDM’s budget). We adopt EDM hyperparameters verbatim [KAA22]. All images are corrupted with additive Gaussian noise at the prescribed level.

Distillation: denoising task. For the distillation phase, we train the one-step generator on CIFAR-10 for **100 M** image iterations and on FFHQ, CelebA-HQ, and AFHQ-v2 for **15 M** iterations; this budget suffices to reach competitive FID. Unless stated otherwise, hyperparameters mirror those of SiD [ZZW24]. For CelebA-HQ, we use the same configuration as FFHQ/AFHQ-v2 except for a dropout rate of 0.15.

Teacher-consistency. For experiments in Tab 6.2 involving Teacher-Consistency [DDD24], we use 8 reverse steps and 32 Monte Carlo samples to approximate expectations. The consistency-loss weight is selected from $\{0.1, 1.0, 10.0\}$ as a fixed coefficient to maximize performance.

General corruption tasks: pretraining. We again follow the EDM training protocol [KAA22] and pretrain for **100 M** image iterations: (i) For Gaussian deblurring and super-resolution with $\sigma = 0$, we use the Standard Diffusion objective (Eq. 6.14). (ii) For the same tasks with $\sigma = 0.2$, we adopt the Diffusion for denoising loss (Eq. 6.15). (iii) For random inpainting with $\sigma = 0$, we use the publicly available Diffusion for random inpainting checkpoint at <https://github.com/giannisdaras/ambient-diffusion/tree/main>. (iv) For random inpainting with $\sigma = 0.2$, we train with the Diffusion for denoising loss (Eq. 6.15); we avoid the dedicated inpainting loss (Eq. 6.16) in this noisy regime due to instability, consistent with [DSD23]. (v) For Fourier-space random inpainting on MRI, we initialize from the checkpoint at <https://github.com/utcsilab/ambient-diffusion-mri.git>. Unless noted, pretrained diffusion models use EDM hyperparameters [KAA22].

General corruption tasks: distillation. During distillation we train the one-step generator for **50–100 M** image iterations, with early stopping if the validation FID begins to diverge.

Reproducibility. Upon acceptance, we will release code and checkpoints to facilitate reproduction and further research.

CHAPTER 7

Flow Priors for Linear Inverse Problems via Iterative Corrupted Trajectory Matching

7.1 Introduction

Linear inverse problems are ubiquitous across many imaging domains, pervading areas such as astronomy [Rod88, Jan14], medical imaging [RYF19, Sue17], and seismology [Nol08, RFS14]. In these problems the goal is to reconstruct an unknown image $x_* \in \mathbb{R}^n$ from observed measurements $y \in \mathbb{R}^m$ of the form:

$$y = \mathcal{A}(x_*) + \text{noise}, \quad (7.1)$$

where $\mathcal{A} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ with $m \leq n$ is a linear operator that degrades the clean image x_* , and the additive noise is drawn from a known distribution. In this work, we assume the noise follows $\mathcal{N}(0, \sigma_y^2 I)$. Due to the under-constrained nature of such problems, they are typically ill-posed, i.e., there are an infinite number of undesirable images that fit to the observed measurements. Hence, one requires further structural information about the underlying images, which constitutes our prior.

With the advent of large generative models [KW13, GPM20, SSK20, RM15, CZF24, ZYW25, ZYW24], there has been a surge of interest in exploiting generative models as priors to solve inverse problems. Given a pretrained generator to sample from a distribution or grant access to image probabilities, one can solve a variety of inverse problems in a task- or forward model-agnostic fashion, without the need for large-scale supervision

[OJM20]. This has been successfully done for a variety of models, including implicit generators such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) [BJP17, MDH20], invertible generators such as Normalizing Flows [ADL20, WLD21], and more recently Diffusion models [CKM23, RRD24, ZDZ24].

A recent paradigm in generative modeling [SSK20, XLZ16b, KG23, YZH24], based on the concept of flow matching [LGL22a, LCB22], has made significant strides in scaling ODE-based generators to high-resolution images. Flow matching models map a simple base distribution, such as a Gaussian, to a complex, high-dimensional data distribution by defining a flow field that represents the transformation between these distributions. These generative models have demonstrated scalability to high dimensions, forming the backbone of several state-of-the-art generative models [LZM24, EKB24, YLP24]. Moreover, flow matching models follow straighter and more direct probability paths compared to diffusion models, allowing for more efficient and faster sampling [LCB22, LGL22a, EKB24]. Additionally, due to their invertibility, flow matching models provide direct access to image likelihoods through the instantaneous change-of-variables formula [CRB18, GCB19]. Given these advantages and the relatively recent application of these models to inverse problems [PMC23, BPG24], we investigate their use as image priors in this work.

Leveraging knowledge about the corruption process $p(y|x)$ and a natural image prior $p(x)$, the Bayesian approach suggests analyzing the image reconstruction posterior $p(x|y) \propto p(y|x)p(x)$ to solve the inverse problem. A proven and effective method based on this approach is maximum-a-posteriori (MAP) estimation [BL14, HB15], which maximizes the posterior to identify the image most likely to match the observed measurements:

$$\operatorname{argmin}_{x \in \mathbb{R}^n} -\log p(x|y) = \operatorname{argmin}_{x \in \mathbb{R}^n} -\log p(y|x) - \log p(x). \quad (7.2)$$

MAP estimation provides a single, most probable point estimate of the posterior distribution, making it simple and interpretable. This deterministic approach ensures consistency and reproducibility, which are essential in applications requiring reliable outcomes, particularly in compressed sensing tasks such as Computed Tomography (CT) [Buz11] and Magnetic

Resonance Imaging (MRI) [VB13]. While posterior sampling methods can offer diverse reconstructions to quantify uncertainty, they can be prohibitively slow in high-dimensions [BGJ11]. Hence, in this work, we propose to integrate flow priors to solve linear inverse problems by MAP estimation.

A significant challenge in employing flow priors for MAP estimation lies in the slow computation of the image probabilities, as it requires backpropagating through an ODE solver [SDM21, FSR23, FB24]. In this work, we show how one can address this challenge via Iterative Corrupted Trajectory Matching (ICTM), a novel algorithm to approximate the MAP solution in a computationally efficient manner. In particular, we show how one can approximately find an MAP solution by sequentially optimizing a novel simpler, auxiliary objective that approximates the true MAP objective in the limit of infinite function evaluations. For finite evaluations, we demonstrate that this approximation is sufficient to optimize by showcasing strong empirical performance for flow priors across a variety of linear inverse problems. We summarize our **contributions** as follows:

1. We propose ICTM, an algorithm to approximate the MAP solution to a variety of linear inverse problems using a flow prior. This algorithm optimizes an auxiliary objective that partitions the flow model’s trajectory into N “local MAP” objectives, where N is the number of function evaluations (NFEs). By leveraging Tweedie’s formula, we show that we can perform gradient steps to sequentially optimize these objectives.
2. Theoretically, we demonstrate that the auxiliary objective converges to the true MAP objective as the NFEs goes to infinity. We validate the correctness of our algorithm in finding the MAP solution on a denoising problem.
3. We demonstrate the utility of ICTM on a wide variety of linear inverse problems on both natural and scientific image datasets, with problems including denoising, inpainting, super-resolution, deblurring, and compressed sensing. Extensive results show that

ICTM is both computationally efficient and obtains high-quality reconstructions, outperforming other reconstruction algorithms based on flow priors.

7.2 Background

Notation We follow the convention for flow-based models, where Gaussian noise is sampled at timestep 0, and the clean image corresponds to timestep 1. Note that this is the opposite of diffusion models. For $t \in [0, 1]$, we denote $x_t(x_0)$ as the point at time t whose initial condition is x_0 . In this work, we use x and x_1 interchangeably, i.e., $x_1(x_0) = x(x_0)$.

7.2.1 Flow-Based Models

We consider generative models that map samples x_0 from a noise distribution $p(x_0)$, e.g., Gaussian, to samples x_1 of a data distribution $p(x_1)$ using an ordinary differential equation (ODE):

$$dx_t = v_\theta(x_t, t) dt, \tag{7.3}$$

where the velocity field v is a θ -parameterized neural network, e.g., using a UNet [LCB22, LGL22a, RFB15b] or Transformer [EKB24, VSP17] architecture. Generative models based on flow matching [LCB22, LGL22a] can be seen as a simulation-free approach to learning the velocity field. This approach involves pre-determining paths that the ODE should follow by specifying the interpolation curve x_t , rather than relying on the MLE algorithm to implicitly discover them [CRB18]. To construct such a path, which is not necessarily Markovian, one can define a **differentiable** nonlinear interpolation between x_0 and x_1 :

$$x_t = \alpha_t x_1 + \beta_t x_0, \quad x_0 \sim \mathcal{N}(0, I), \tag{7.4}$$

where both α_t and β_t are differentiable functions with respect to t satisfying $\alpha_0 = 0$, $\beta_0 = 1$, and $\alpha_1 = 1$, $\beta_1 = 0$. This ensures that x_t is transported from a standard Gaussian

distribution to the natural image manifold from time 0 to time 1. In contrast, the diffusion process [SSK20, SWM15, HJA20b] induces a non-differentiable trajectory due to the diffusion term in the SDE formulation.

The idea behind flow matching is to utilize the power of deep neural networks to efficiently predict the velocity field at each timestep. To achieve this, we can train the neural network by minimizing an L_2 loss between the sampled velocity and the one predicted by the neural network:

$$\mathcal{L}(\theta) = \mathbb{E}_{t,p(x_1),p(x_0)} \|v_\theta(x_t, t) - (\dot{\alpha}_t x_1 + \dot{\beta}_t x_0)\|^2. \quad (7.5)$$

We denote the optimal (not necessarily unique) solution to $\arg \min_\theta \mathcal{L}(\theta)$ as $\hat{\theta}$. The optimal velocity field $v_{\hat{\theta}}$ can be derived in closed form and is the expected velocity at state x_t :

$$v_{\hat{\theta}}(x_t, t) = \mathbb{E}_{p(x_1),p(x_0)} [\dot{\alpha}_t x_1 + \dot{\beta}_t x_0 \mid x_t]. \quad (7.6)$$

For convenience, in the following text, we use v_θ to refer to the optimal $v_{\hat{\theta}}$. In the rest of the paper, we assume that the flow v_θ and its parameters are pretrained on a dataset of interest and fixed. We are then interested in leveraging its utility as a prior to solve inverse problems.

7.2.2 Probability Computation for Flow Priors

Denote the probability of x_t in Eq. equation 7.3 as $p(x_t)$ dependent on time. Assuming that v_θ is uniformly Lipschitz continuous in x_t and continuous in t , the change in log probability also follows a differential equation [CRB18, GCB19]:

$$\frac{\partial \log p(x_t)}{\partial t} = -\text{tr} \left(\frac{\partial}{\partial x} v_\theta(x_t, t) \right). \quad (7.7)$$

One can additionally obtain the likelihood of the trajectory via integrating Eq. equation 7.7 across time

$$\log p(x_t) = \log p(x_\tau) - \int_\tau^t \text{tr} \left(\frac{\partial}{\partial x} v_\theta(x_s, s) \right) ds, \quad 0 \leq \tau < t \leq 1. \quad (7.8)$$

7.3 Method

In this work, we aim to solve the MAP estimation problem in Eq. equation 7.2 where $p(x)$ is given by a pretrained flow prior. We first discuss in Section 7.3.1 how the MAP problem could, in principle, be solved via a latent-space optimization problem. As we will see, this problem is challenging to solve computationally due to the need to backpropagate through an ODE solver. To overcome this, we show in Section 7.3.2 that the ideal MAP problem can be approximated by a weighted sum of “local MAP” optimization problems, which operates by partitioning the flow’s trajectory to a reconstructed solution. We then introduce our ICTM algorithm to sequentially optimize this auxiliary objective. Finally, in Section 7.3.3, we experimentally validate that our algorithm finds a solution that is faithful to the MAP estimate in a simplified setting where the globally optimal MAP solution is known.

7.3.1 Flow-Based MAP

Given a pretrained flow prior, one can compute the log-likelihood of x generated from an initial noise sample x_0 via Eq. equation 7.8. Hence, to find the MAP estimate, one could equivalently optimize the initial point of the trajectory x_0 and return $x_1(x_0)$ where x_0 is found by solving

$$\min_{x_0 \in \mathbb{R}^n} \underbrace{\frac{1}{2\sigma_y^2} \|y - \mathcal{A}(x_1(x_0))\|^2}_{\text{data likelihood}} + \underbrace{\frac{1}{2} \|x_0\|^2 + \int_0^1 \text{tr} \left(\frac{\partial}{\partial x} v_\theta(x_t, t) \right) dt}_{\text{prior}}, \quad (7.9)$$

where $x_t := x_t(x_0)$ denotes the intermediate state x_t generated from x_0 . Intuitively, this loss encourages finding an initial point x_0 such that the reconstruction $x_1 := x_1(x_0)$ fits the observed measurements, but is also likely to be generated by the flow.

In practice, x_1 and the prior term can be approximated by an ODE solver. The trajectory of $x_t = x_0 + \int_0^t v_\theta(x_t, t) dt$ can be approximated by an ODE sampler, i.e. $\text{ODESolve}(x_0, 0, t, v_\theta)$, where x_0 is the initial point, and the second and third arguments represent the starting time and the ending time, respectively. For example, with an Euler

sampler, we iterate over $x_{t+\Delta t} = x_t + v_\theta(x_t, t)\Delta t$ where $\Delta t = 1/N$ and N is the predetermined NFEs. After acquiring the optimal $\hat{x}_0$ by optimizing the Eq. equation 7.9, we obtain the MAP solution x_1 by using $\text{ODESolve}(\hat{x}_0, 0, 1, v_\theta)$ again.

7.3.2 Flow-Based MAP Approximation

The global flow-based MAP objective Eq. equation 7.9 is tractable for low-dimensional problems. The challenge for high-dimensional problems, however, is that optimizing Eq. equation 7.9 is simulation-based, and thus each update iteration requires full forward and backward propagation through an ODE solver, resulting in issues regarding memory inefficiency and time, making it hard to optimize [CRB18, FB24, FSR23, SDM21].

As a way to address this, we prove a result in Theorem 6 that shows that the MAP objective can be approximated by a weighted sum of N local posterior objectives. These objectives are “local” in the sense that they mainly depend on likelihoods and probabilities of intermediate trajectories x_t and $x_t + v_\theta(x_t, t)\Delta t$ for $t = 0, \Delta t, \dots, N\Delta t$ where $\Delta t := 1/N$. Given an initial noise input x_0 , each local posterior objective depends on a non-Markovian **auxiliary path** $y_t = \alpha_t y + \beta_t \mathcal{A}(x_0)$ by connecting the points between y and $\mathcal{A}x_0$. We prove this result for straight paths $\alpha_t = t$ and $\beta_t = 1 - t$ for simplicity, but other interpolation paths can be used. The proof is in Section 7.7.1.2.

Theorem 6. *For $N \geq 1$, set $\gamma_i := (\frac{1}{2})^{N-i+1}$ and $\Delta t = 1/N$. Suppose $y = \mathcal{A}(x_*) + \epsilon$ where $x_* = x_1(x_0)$ with x_0 being the solution to Eq. equation 7.9, $\epsilon \sim \mathcal{N}(0, \sigma_y^2 I)$, and x_t exactly follows the straight path $x_t = tx + (1 - t)x_0$ for any timestep $t \in [0, 1]$. Suppose the velocity field $v_\theta : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$ satisfies $\sup_{z \in \mathbb{R}^n, s \in [0, 1]} |\text{tr} \frac{\partial}{\partial x} v_\theta(z, s)| \leq C_1$ for some universal constant C_1 . Then, there exists a constant $c(N)^1$ that does not depend on x_0 such that*

$$\lim_{N \rightarrow \infty} \left| \log p(x(x_0)|y) - \sum_{i=1}^N \gamma_i \hat{\mathcal{J}}_i - c(N) \right| = 0,$$

¹This is given by $c(N) := \sum_{i=1}^N \gamma_i c_i - \log p(y)$. Please see the proof of Theorem 6 in Appendix 7.7.1.2.

where $\hat{\mathcal{J}}_i = \log p(x_{(i-1)\Delta t}) - \text{tr} \left(\frac{\partial v_\theta(x_{(i-1)\Delta t}, (i-1)\Delta t)}{\partial x} \right) \Delta t + \log p(y_{i\Delta t} | x_{i\Delta t})$.

This result shows that the true MAP objective evaluated at the optimal solution can be approximated by a weighted sum of objectives that depend locally at a time t for the trajectory $\{x_t : t \in [0, 1]\}$. The intuition regarding $\hat{\mathcal{J}}_i$ arises from the fact that $\hat{\mathcal{J}}_i \approx \mathcal{J}_i$, where $\mathcal{J}_i$ is the local posterior distribution

$$\mathcal{J}_i = \log p(y_{i\Delta t} | x_{i\Delta t}(x_{(i-1)\Delta t})) + \log p(x_{i\Delta t}).$$

Optimizing each of these local posterior distributions in a sequential fashion captures the fact that we would like each intermediate point in our trajectory $x_{i\Delta t}$ to be likely and fit to our measurements, ideally resulting in a final reconstruction x_1 that satisfies this as well. The benefit of $\hat{\mathcal{J}}_i$, as we will show in the sequel, is that it is efficient to optimize.

Discussion of assumptions: We assume that the trajectory $\{x_t\}_t$ exactly follows the predefined interpolation path $\{\alpha_t x + \beta_t x_0\}_t$. In Section 7.7.2 of the appendix, we analyze this assumption and show that we can bound the deviation from the predefined interpolation path to the learned path via a path compliance measure. Moreover, we impose a regularity assumption on the velocity field v_θ , effectively requiring a uniform bound on the spectrum of the Jacobian of v_θ . This can be easily satisfied with neural networks using Lipschitz continuous and differentiable activation functions.

As we see in Theorem 6, one can approximate the true MAP objective via a sum of local objectives of the form

$$\hat{\mathcal{J}}_i := \underbrace{\log p(y_{i\Delta t} | x_{i\Delta t})}_{\text{local data likelihood}} + \underbrace{\log p(x_{(i-1)\Delta t}) - \text{tr} \left(\frac{\partial v_\theta(x_{(i-1)\Delta t}, (i-1)\Delta t)}{\partial x} \right) \Delta t}_{\text{local prior}}. \quad (7.10)$$

At first glance, $\hat{\mathcal{J}}_i$ still appears challenging to optimize, but there are additional insights we can exploit for computation. We discuss each term in $\hat{\mathcal{J}}_i$ below.

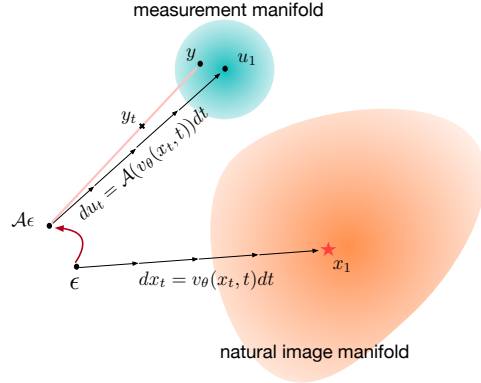


Figure 7.1: **Illustration of the idea of ICTM.** The corrupted trajectory $u_t := \mathcal{A}(x_t)$ follows the corrupted flow ODE $du_t = \mathcal{A}(v_\theta(x_t, t))dt$.

Local data likelihood: The intuition behind ICTM is that we aim to match a **corrupted** trajectory $\{u_t\}_t$ with an auxiliary path $\{y_t\}_t$ specified by an interpolation between our measurements y and $\mathcal{A}(x_0)$ for each timestep t , defined by $y_t := \alpha_t y + \beta_t \mathcal{A}(x_0)$. The corrupted trajectory $u_t := \mathcal{A}(x_t)$ follows the **corrupted flow ODE** $du_t = \mathcal{A}(v_\theta(x_t, t))dt$. To optimize the above “local MAP” objectives, we must understand the distribution of $p(y_t|x_t)$. Generally speaking, this distribution is intractable. However, by assuming exact **compliance** of the trajectory generated by flow to the predefined interpolation path (as done in Theorem 6), we can show that $y_t|x_t \sim \mathcal{N}(u_t, \alpha_t^2 \sigma_y^2)$. This is proven in Lemma 16 in the appendix. While exact compliance of the trajectory may not hold for learned flow matching models, we show empirically that making this assumption leads to strong performance in practice. We further analyze this notion of compliance in Section 7.7.2 of the appendix.

Local prior: The approximation in Eq. equation 7.10 addresses one of the main concerns of MAP in that the intensive integral computation is circumvented with a simpler Riemannian sum. This approximation holds for small time increments Δt : $\int_t^{t+\Delta t} \text{tr} \left(\frac{\partial}{\partial x} v_\theta(x_s, s) \right) ds \approx \text{tr} \left(\frac{\partial}{\partial x} v_\theta(x_t, t) \right) \Delta t$. Note that one can additionally improve the efficiency of this term by employing a Hutchinson-Skilling estimate [Ski89, Hut89] for the trace of the Jacobian matrix.

However, at first glance, it appears we have simply shifted the problem to the computation of the prior at timestep $(i - 1)\Delta t$. Fortunately, it is possible to derive a formula for the gradient of $\log p(x_t)$ for all timesteps $t \in [0, 1]$ using Tweedie’s formula [Efr11]. This allows us to optimize each objective $\hat{\mathcal{J}}_i$ using gradient-based optimizers. The following result gives a precise characterization of $\nabla_{x_t} \log p(x_t)$, proven in Section 7.7.1.1.

Proposition 1. *Let $\lambda_t = \alpha_t/\beta_t$ denote the signal-to-noise ratio. The relationship between the score function $\nabla_{x_t} \log p(x_t)$ and the velocity field $v_\theta(x_t, t)$ is given by:*

$$\nabla_{x_t} \log p(x_t) = \frac{1}{\beta_t^2} \left[\left(\frac{d \log \lambda_t}{dt} \right)^{-1} \left(v_\theta(x_t, t) - \frac{d \log \beta_t}{dt} x_t \right) - x_t \right]. \quad (7.11)$$

In summary, we have derived an efficient approximation to the MAP objective. For our algorithm, we iteratively optimize each term $\hat{\mathcal{J}}_t$ sequentially for each $t = 0, \Delta t, \dots, N\Delta t$, fitting our current iterate x_t to induce an increment $x_{t+\Delta t}$ such that $\mathcal{A}(x_{t+\Delta t})$ fits to our auxiliary corrupted path $y_{t+\Delta t}$ while being likely under our local prior. We call this approach Iterative Corrupted Trajectory Matching (ICTM). Our algorithm is summarized in Algo. 18. In lines 7 and 12, instead of directly optimizing the local data likelihood, we choose λ as a new hyper-parameter to tune. We find a constant λ works well in practice.

7.3.3 Toy Example Validation

We experimentally validate that the reconstruction found via ICTM is close to the optimal MAP solution in a simplified denoising problem where the MAP solution can be obtained in closed-form. Specifically, we fit a Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ using 1,000 samples from the FFHQ dataset. Consider a denoising problem $y = x + \epsilon$ where $x \sim \mathcal{N}(\mu, \Sigma)$ and $\epsilon \sim \mathcal{N}(0, \sigma_y^2 I)$. In this case, the analytical solution to the MAP estimation problem (Eq. equation 7.2) is $x_* = (\Sigma^{-1} + \sigma_y^{-2} I)^{-1} (\Sigma^{-1} \mu + \sigma_y^{-2} y)$. We set $\sigma_y = 0.1$. Then, we train a flow-based model on 10,000 samples from the true Gaussian distribution and showcase the deviation of our reconstruction found via ICTM to the closed-form MAP solution x_* in Fig.

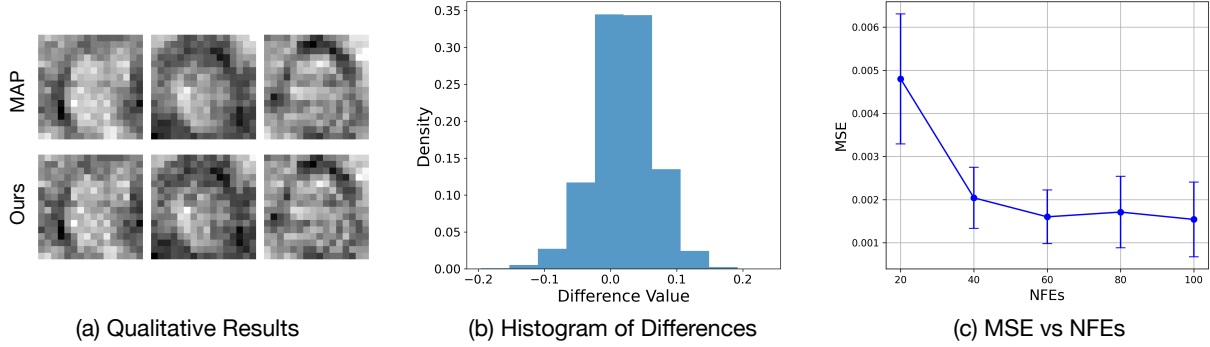


Figure 7.2: Results of a toy example modeling 1,000 FFHQ faces as a Gaussian distribution. Subfigure (a) shows the qualitative results of our method; Subfigure (b) presents the histogram of the differences between ours and the true MAP; Subfigure (c) displays the MSE values as the NFes varies.

7.2. We see that ICTM can obtain a faithful estimate of the MAP solution across many samples.

7.4 Experiments

In our experimental setting, we use optimal transport interpolation coefficients, i.e. $\alpha_t = t$ and $\beta_t = 1 - t$. We test our algorithm on both natural and medical imaging datasets. For natural images, we utilize the pretrained checkpoint from the official Rectified Flow repository² and evaluate our approach on the CelebA-HQ dataset [LLW15b, KAL17b]. We address four common linear inverse problems: super-resolution, inpainting with a random mask, Gaussian deblurring, and inpainting with a box mask. For the medical application, we train a flow-based model from scratch on the Human Connectome Project (HCP) dataset [VSB13] and test our algorithm specifically for compressed sensing at different compression rates. Our algorithm focuses on the reconstruction faithfulness of generated images, therefore

²<https://github.com/gnabitab/RectifiedFlow>

Table 7.1: Quantitative comparison results in terms of PSNR and SSIM on the CelebA-HQ dataset. Our algorithm surpasses all other baselines across all tasks. The best values are highlighted in blue and the second-best are underlined.

	Super-Resolution		Inpainting(random)		Gaussian Deblurring		Inpainting(box)	
Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
OT-ODE	27.46	0.775	28.57	0.838	26.28	0.727	19.80	0.795
DPS-ODE	27.85	0.791	29.57	<u>0.872</u>	25.97	0.704	23.59	0.758
RED-Diff	27.20	0.760	25.13	0.711	27.23	0.765	17.50	0.651
Π GDM	28.33	<u>0.803</u>	<u>29.98</u>	0.858	24.30	0.583	<u>24.10</u>	<u>0.853</u>
Ours (w/o prior)	26.06	0.724	29.01	0.835	25.13	0.676	22.42	0.803
Ours	<u>27.91</u>	0.805	30.65	0.894	<u>26.54</u>	<u>0.760</u>	24.34	0.866

employing PSNR and SSIM [WBS04] as evaluation metrics.

Baselines We compare our method with five baselines. 1) OT-ODE [PMC23]. To our knowledge, this is the only baseline that applies flow-based models to inverse problems. They incorporate a prior gradient correction at each sampling step based on conditional Optimal Transport (OT) paths. For a fair comparison, we follow their implementation of Algorithm 1, providing detailed ablations on initialization time t' in Appendix 7.7.4.3. 2) DPS-ODE. Inspired by DPS [CKM23], we replace the velocity field with a conditional one, i.e., $v(x_t|y) = v(x_t) + \zeta_t \nabla_{x_t} \log p(y|\hat{x}_1(x_t))$, where ζ_t is a hyperparameter to tune. Following the hyperparameter instruction in DPS, we provide detailed ablations on ζ_t in Appendix 7.7.4.3. 3) Ours without local prior. To examine the local prior term’s effectiveness in our optimization algorithm, we drop the local prior term as defined in Eq. equation 7.10 in our algorithm. In the experiments with natural images, in addition to the flow-based baselines, we have included two representative diffusion-based baselines: 4) RED-Diff [MSK24], a variational Bayes-based method; and 5) Π GDM [SVM23], an advanced MCMC-based method.

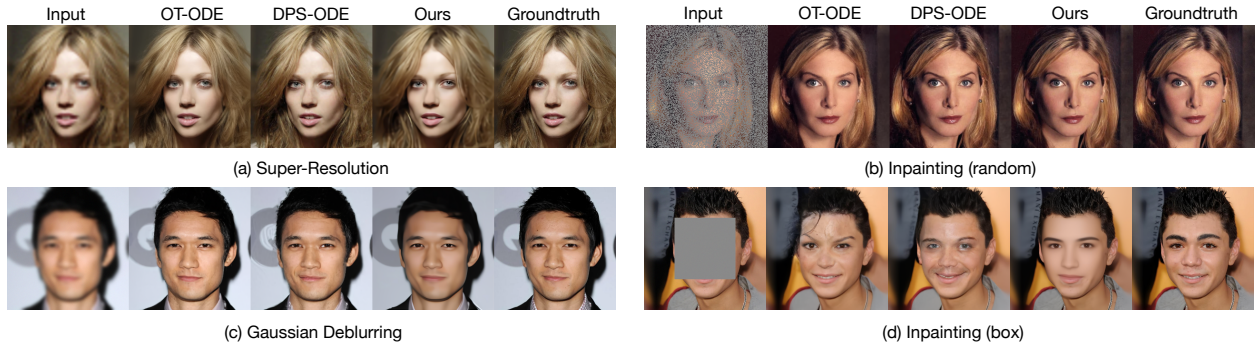


Figure 7.3: Qualitative comparison results on the CelebA-HQ dataset. The reconstructions generated by our method align more faithfully with the ground truth and exhibit a higher degree of refinement.

We also note one concurrent work, D-Flow [BPG24], which formulates the MAP as a constrained optimization problem in their Eq. 9. As documented in their Sec. 3.4, it takes 5-10 minutes to recover each image. This is because each of its optimization step requires backpropagation through an ODE solver to compute the full log-likelihood. In contrast, our method is significantly faster (approximately 1.6 minutes per image) due to our principled local MAP approximation, as demonstrated in Appendix 7.7.3.

7.4.1 Natural Images

Experimental setup We evaluate our algorithm using 100 images from the CelebA-HQ validation set with a resolution of 256×256 , normalizing all images to the $[0, 1]$ range for quantitative analysis. All experiments incorporate Gaussian measurement noise with $\sigma_y = 0.01$. We address the following linear inverse problems: (1) $4\times$ super-resolution using bicubic down-sampling, (2) inpainting with a random mask covering 70% of missing values, (3) Gaussian deblurring with a 61×61 kernel and a standard deviation of 3.0, and (4) box inpainting with a centered 128×128 mask.

We present the quantitative and qualitative results of all the methods in Tab. 7.1 and

Table 7.2: Results of compressed sensing with varying compression rate ν on the HCP T2w dataset. Note that compressed sensing is more challenging due to the complexity of the forward operator, as evidenced by the poor performance of OT-ODE, which assumes a Gaussian distribution of measurement y given x_t . The best values are highlighted in blue.

Method	$\nu = 1/2$		$\nu = 1/4$		$\nu = 1/10$	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Wavelet Prior	18.02 ± 1.38	0.495 ± 0.02	11.99 ± 1.34	0.230 ± 0.02	7.37 ± 1.85	0.090 ± 0.02
TV Prior	25.36 ± 2.79	0.657 ± 0.04	18.70 ± 2.36	0.496 ± 0.03	14.38 ± 3.04	0.309 ± 0.04
OT-ODE	18.71 ± 1.02	0.422 ± 0.17	18.16 ± 1.06	0.271 ± 0.07	12.21 ± 1.43	0.096 ± 0.04
DPS-ODE	31.06 ± 3.91	0.765 ± 0.08	25.01 ± 1.87	0.608 ± 0.08	22.06 ± 1.66	0.479 ± 0.09
Ours	32.72 ± 1.53	0.878 ± 0.05	27.03 ± 1.77	0.733 ± 0.04	24.03 ± 1.23	0.503 ± 0.04

Fig. 7.3, respectively. In Tab. 7.1, our method surpasses all other baselines across all tasks. For more challenging tasks such as Gaussian deblurring and box inpainting, our method significantly outperforms others in terms of SSIM. Based on the MAP framework, as shown in Fig. 7.3, our method prefers more faithful and artifact-free reconstructions, whereas others trade off for perceptual quality. We note that there is an unavoidable tradeoff between perceptual quality and restoration faithfulness [BM18]. Overall, our method presents a higher degree of refinement. The comparison between ours and ours (w/o prior) indicates the effectiveness of the local prior term in enhancing the accuracy of the reconstructions, as evidenced by the increases in both PSNR and SSIM.

7.4.2 Medical application

HCP T2w dataset We utilize images from the publicly available Human Connectome Project (HCP) [VSB13] T2-weighted (T2w) images dataset for the task of compressed sensing, which contains brain images from 47 patients. The HCP dataset includes cross-sectional

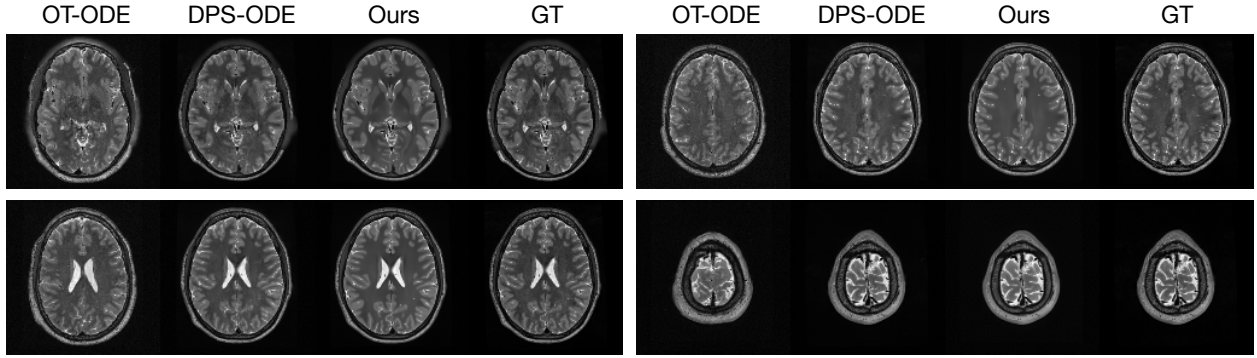


Figure 7.4: Qualitative comparison results on compressed sensing. Our method produces more faithful reconstructions with fewer artifacts, ensuring higher accuracy and clarity in the details.

images of the brain taken at different levels and angles.

Compressed sensing We train a flow-based model from scratch on 10,000 randomly sampled images, utilizing the *ncsnpp* architecture [SSK20] with minor adaptations for grayscale images. We employ compression rates $\nu \in \{1/2, 1/4, 1/10\}$, meaning $m = \nu n$. The measurement operator is given by a subsampled Fourier matrix, whose sign patterns are randomly selected. We evaluate our reconstruction algorithm’s performance on 200 randomly sampled test images.

We present the quantitative and qualitative results of compressed sensing in Tab. 7.2 and Fig. 7.4, respectively. In addition to flow-based methods, we include results for two classical recovery algorithms, Wavelet [Don06, LDS08] and TV [IOF92] priors. As shown in Tab. 7.2, our method outperforms the classical recovery algorithms and other flow-based baselines across varying compression rates ν , demonstrating our method’s capability to handle challenging scenarios and the advantages of utilizing modern generative models as priors. In Fig. 7.4, our method produces reconstructions that are more faithful to the original images, with fewer artifacts, leading to higher accuracy and clearer details.

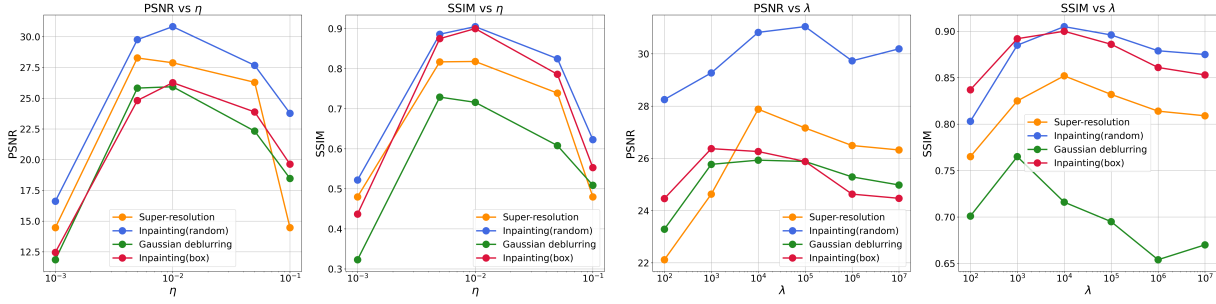


Figure 7.5: Ablation results of step size η and guidance weight λ . The choice of hyperparameters for our algorithm is fairly consistent across all tasks. We choose $\eta = 10^{-2}$ for all experiments on CelebA-HQ. For λ , we choose $\lambda = 10^3$ for Gaussian deblurring and $\lambda = 10^4$ for the other tasks.

7.4.3 Ablation studies

We use the Adam optimizer [Kin14] for our optimization steps due to its effectiveness in neural network computations. For all tasks, we utilize $N = 100$ steps.

Step size η and Guidance weight λ The use of the Adam optimizer ensures that the choice of hyperparameters, particularly the step size η and the guidance weight λ , remains consistent across various tasks, as illustrated in Fig. 7.5. Specifically, a step size of $\eta = 10^{-2}$ is optimal for Inpainting (random), Inpainting (box), and Super-resolution in terms of SSIM. For PSNR, Gaussian deblurring also achieves optimal performance at $\eta = 10^{-2}$. Consequently, we employ $\eta = 10^{-2}$ for all tasks. Based on the results shown in the right two subfigures of Fig. 7.5, we select $\lambda = 10^3$ for Gaussian deblurring and $\lambda = 10^4$ for the other tasks. This consistency extends to the compressed sensing experiments, where we set $\lambda = 10^3$ and $\eta = 10^{-2}$ for all experiments involving medical images.

Iteration number K We present ablation results of the iteration number K on different tasks in Fig. 7.6. We focus on the behavior of K in super-resolution and compressed sensing,

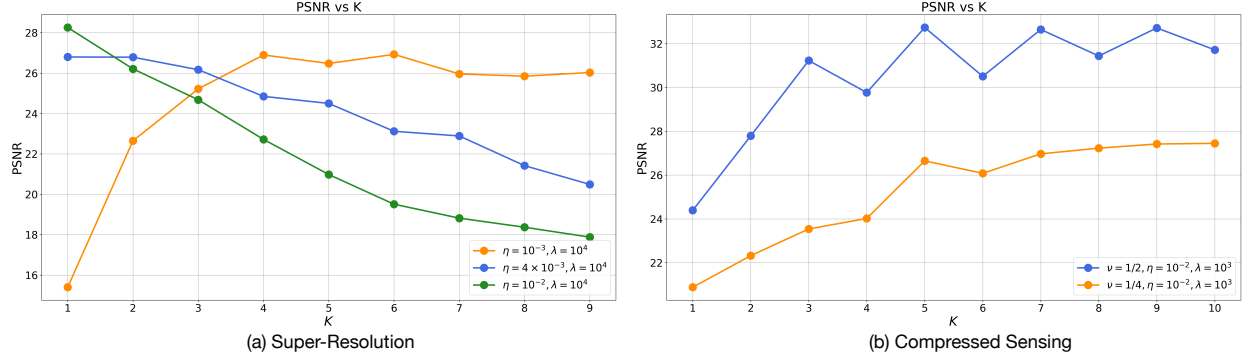


Figure 7.6: Ablation results of iteration number K on different tasks. For super-resolution and the other three tasks, $K = 1$ is sufficient to achieve the best performance with the optimal step size η and guidance weight λ . However, for compressed sensing, it is necessary to increase K to obtain the best performance. We hypothesize that this is due to the increased complexity of the compressed sensing operator, which requires more iteration steps to ensure the correct optimization direction.

as it performs similarly to super-resolution in the other three tasks. With the optimal choice of η and λ in super-resolution, i.e., $\eta = 10^{-2}$ and $\lambda = 10^3$, $K = 1$ provides superior performance on CelebA-HQ. A decreased step size, e.g., $\eta = 10^{-3}$, can help performance as K increases, but it fails to exceed the performance achieved with the optimal parameters at $K = 1$. However, for compressed sensing, it is necessary to increase K to achieve the best performance. Consequently, we set $K = 10$ for all compressed sensing experiments. We hypothesize that the complexity of the compressed sensing operator directly determines the number of iterations required for optimal performance.

7.5 Conclusion

In this work, we have introduced a novel iterative algorithm to incorporate flow priors to solve linear inverse problems. By addressing the computational challenges associated with the slow log-likelihood calculations inherent in flow matching models, our approach lever-

ages the decomposition of the MAP objective into multiple "local MAP" objectives. This decomposition, combined with the application of Tweedie’s formula, enables effective sequential optimization through gradient steps. Our method has been rigorously validated on both natural and scientific images across various linear inverse problems, including super-resolution, deblurring, inpainting, and compressed sensing. The empirical results indicate that our algorithm consistently outperforms existing techniques based on flow matching, highlighting its potential as a powerful tool for high-resolution image synthesis and related downstream tasks.

7.6 Limitations and Future Work

While our algorithm has demonstrated promising results, there are certain limitations that suggest avenues for future research. First, our theoretical framework, built on optimal transport interpolation paths, is currently limited and cannot be applied to solve the general interpolation between Gaussian and data distributions. Additionally, in order to broaden the applicability of flow priors for inverse problems, it is important to generalize our approach to handle nonlinear forward models. Moreover, the algorithm currently lacks the capability to quantify the uncertainty of the generated images, an aspect crucial for many scientific applications. It would be interesting to consider approaches to post-process our solutions to understand the uncertainty inherent in our reconstruction. These limitations highlight important directions for future work to enhance the robustness and applicability of our method.

7.7 Appendix

Algorithm 18 Iterative Corrupted Trajectory Matching (ICTM) with Euler Sampler

Input: measurement y , matrix $\mathcal{A}$, pretrained flow-based model θ , NFEs N , interpolation coefficients $\{\alpha_t\}_t$ and $\{\beta_t\}_t$, step size η , guidance weight λ , and iteration number K

Output: recovered clean image x_1

```

1: Initialize  $\epsilon \sim \mathcal{N}(0, I)$ ,  $x_0 \leftarrow \epsilon$ ,  $t \leftarrow 0$ ,  $\Delta t \leftarrow 1/N$ 
2: Generate an auxiliary path  $y_s = \alpha_s y + \beta_s(\mathcal{A}x_0)$  for  $s \in (0, 1)$ 
3: while  $t < 1$  do
4:    $x_{t+\Delta t} \leftarrow x_t + v_\theta(x_t, t)\Delta t$ 
5:   if  $t = 0$  then
6:     for  $k = 1, \dots, K$  do
7:        $x_t \leftarrow x_t - \eta \nabla_{x_t} \left[ \lambda \|\mathcal{A}(x_{t+\Delta t}(x_t)) - y_{t+\Delta t}\|^2 + \frac{1}{2} \|x_t\|^2 + \text{tr} \left( \frac{\partial v_\theta(x_t, t)}{\partial x} \right) \Delta t \right]$ 
8:     end for
9:   else
10:    for  $k = 1, \dots, K$  do
11:      # use Eq. equation 7.11 to obtain the gradient of  $\log p(x_t)$ 
12:       $x_t \leftarrow x_t - \eta \nabla_{x_t} \left[ \lambda \|\mathcal{A}(x_{t+\Delta t}(x_t)) - y_{t+\Delta t}\|^2 - \log p(x_t) + \text{tr} \left( \frac{\partial v_\theta(x_t, t)}{\partial x} \right) \Delta t \right]$ 
13:    end for
14:  end if
15:   $x_{t+\Delta t} \leftarrow x_t + v_\theta(x_t, t)\Delta t$ 
16:   $t \leftarrow t + \Delta t$ 
17: end while
18: return  $x_1$ 

```

Appendix

7.7.1 Proof

Before we dive into the proof, we provide the following three lemmas.

Lemma 14. *Consider a vector-valued function $f : [0, 1] \rightarrow \mathbb{R}^n$. Then for any $t \in [0, 1]$, we have that*

$$\left\| \int_0^t f(s) ds \right\|^2 \leq \int_0^t \|f(s)\|^2 ds.$$

Proof. For each $s \in [0, 1]$, let $f_i(s) \in \mathbb{R}$ denote the i -th component of $f(s)$. Recall Jensen's inequality: for any convex function $g : \mathbb{R} \rightarrow \mathbb{R}$ and integrable function $h : [0, 1] \rightarrow \mathbb{R}$, we have

$$g\left(\int_a^b h(t) dt\right) \leq \int_a^b g(h(t)) dt.$$

Using convexity of the function $t \mapsto t^2$ and applying Jensen's inequality, we see that

$$\begin{aligned} \left\| \int_0^t f(s) ds \right\|^2 &= \sum_{i=1}^n \left(\int_0^t f_i(s) ds \right)^2 \\ &\leq \sum_{i=1}^n \int_0^t f_i(s)^2 ds \\ &= \int_0^t \sum_{i=1}^n f_i(s)^2 ds \\ &= \int_0^t \|f(s)\|^2 ds. \end{aligned}$$

□

Lemma 15 (Tweedie's Formula [Efr11]). *If $\mu \sim g(\cdot)$, $z|\mu \sim \mathcal{N}(\alpha\mu, \sigma^2 I)$, and therefore $z \sim f(\cdot)$, we have*

$$\mathbb{E}[\mu|z] = \frac{1}{\alpha} [z + \sigma^2 \nabla_z \log f(z)]. \quad (7.12)$$

Lemma 16. Suppose $y = \mathcal{A}(x_*) + \epsilon$ where $x_* = x_1(x_0)$ with x_0 being the solution to Eq. equation 7.9, $\mathcal{A} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is linear, $\epsilon \sim \mathcal{N}(0, \sigma_y^2 I)$, and x_t exactly follows the path $x_t = \alpha_t x + \beta_t x_0$ for any time $t \in [0, 1]$. Then we have

$$p(y_t|x_t) = \mathcal{N}(\mathcal{A}x_t, \alpha_t^2 \sigma_y^2 I), \quad (7.13)$$

and hence

$$\log p(y|x(x_0)) = \log p(y_t|x_t) + \frac{m}{2} \log(\alpha_t^2), \forall t. \quad (7.14)$$

Proof. Recall that the generated auxiliary path $y_t = \alpha_t y + \beta_t \mathcal{A}x_0$. By assumption, we have $\mathcal{A}(x_t) = \mathcal{A}(\alpha_t x + \beta_t x_0) = \alpha_t \mathcal{A}(x(x_0)) + \beta_t \mathcal{A}x_0$. By subtracting these two equations, we have

$$y_t - \mathcal{A}(x_t) = \alpha_t(y - \mathcal{A}(x(x_0))). \quad (7.15)$$

As $y|x(x_0) \sim \mathcal{N}(\mathcal{A}x, \sigma_y^2 I)$, we have $y_t|x_t \sim \mathcal{N}(\mathcal{A}x_t, \alpha_t^2 \sigma_y^2 I)$. The proof for Eq. equation 7.13 is done. Next, we examine the log probability as follows:

$$\log p(y_t|x_t) = -\frac{\|y_t - \mathcal{A}x_t\|^2}{2\alpha_t^2 \sigma_y^2} - \frac{m}{2} \log(2\pi \alpha_t^2 \sigma_y^2) \quad (7.16)$$

$$= -\frac{\|\alpha_t(y - \mathcal{A}(x(x_0)))\|^2}{2\alpha_t^2 \sigma_y^2} - \frac{m}{2} \log(2\pi \alpha_t^2 \sigma_y^2) \quad (7.17)$$

$$= -\frac{\|y - \mathcal{A}(x(x_0))\|^2}{2\sigma_y^2} - \frac{m}{2} \log(2\pi \sigma_y^2) - \frac{m}{2} \log(\alpha_t^2) \quad (7.18)$$

$$:= \log p(y|x(x_0)) - \frac{m}{2} \log(\alpha_t^2). \quad (7.19)$$

□

7.7.1.1 Proof of Proposition 1

Trained by the objective defined in Eq. equation 7.5, the optimal velocity field would be

$$v_\theta(x_t, t) = \mathbb{E}(\dot{\alpha}_t x_1 + \dot{\beta}_t x_0 | x_t) \quad (7.20)$$

$$= \mathbb{E}(\dot{\alpha}_t x_1 + \dot{\beta}_t \frac{x_t - \alpha_t x}{\beta_t} | x_t) \quad \# \text{ Given } x_t, x_0 = \frac{x_t - \alpha_t x}{\beta_t} \quad (7.21)$$

$$= (\dot{\alpha}_t - \dot{\beta}_t \frac{\alpha_t}{\beta_t}) \mathbb{E}(x_1 | x_t) + \frac{\dot{\beta}_t}{\beta_t} x_t \quad (7.22)$$

$$= (\dot{\alpha}_t - \dot{\beta}_t \frac{\alpha_t}{\beta_t}) [\frac{1}{\alpha_t} (x_t + \beta_t^2 \nabla_{x_t} \log p(x_t))] + \frac{\dot{\beta}_t}{\beta_t} x_t. \quad \# \text{ Lemma 15 (Tweedie's Formula)} \quad (7.23)$$

By defining the signal-to-noise ratio as $\lambda_t = \alpha_t / \beta_t$ and rearranging the equation above, we get exactly Eq. equation 7.11 which we display again below:

$$\nabla_{x_t} \log p(x_t) = \frac{1}{\beta_t^2} \left[\left(\frac{d \log \lambda_t}{dt} \right)^{-1} \left(v_\theta(x_t, t) - \frac{d \log \beta_t}{dt} x_t \right) - x_t \right]. \quad (7.24)$$

When $\alpha_t = t$, $\beta_t = 1 - t$, the equation above becomes

$$\nabla_{x_t} \log p(x_t) = \frac{1}{1-t} (-x_t + t v_\theta(x_t, t)). \quad (7.25)$$

7.7.1.2 Proof of Theorem 6

Before we dive into the proof, we first point out $\lim_{\Delta t \rightarrow 0} \sum_{i=1}^N \gamma_i = 1$. Define the timestep $t = (i-1)\Delta t$. Conversely, $i = 1 + t/\Delta t$ is a function of t . In this sense, we define the i -th step Riemannian discretization of the integral $-\int_0^1 \text{tr} \left(\frac{\partial v_\theta(x_t, t)}{\partial x} \right) dt$ as $\Delta p_i = -\text{tr} \left(\frac{\partial v_\theta(x_t, t)}{\partial x} \right) \Delta t$.

We first decompose the global MAP objective as follows:

$$\log p(x(x_0)|y) = \log p(x_0) - \int_0^1 \text{tr} \left(\frac{\partial v_\theta(x_t, t)}{\partial x} \right) dt + \log p(y|x(x_0)) - \log p(y) \quad (7.26)$$

$$= \lim_{\Delta t \rightarrow 0} \sum_{i=1}^N \gamma_i \log p(x_0) + \lim_{\Delta t \rightarrow 0} \sum_{i=1}^N \Delta p_i \quad (7.27)$$

$$+ \lim_{\Delta t \rightarrow 0} \sum_{i=1}^N \gamma_i [\log p(y_{i\Delta t}|x_{i\Delta t}) + c_i] - \log p(y), \quad (7.28)$$

where the decomposition of the second term utilizes the property of the discretization of Riemann integral, and that of the third term utilizes the result in Lemma 16 and thus $c_i = \frac{m}{2} \log(\alpha_{i\Delta t}^2)$. By the property of limits, i.e. $\lim_{\Delta t \rightarrow 0} (\sum_{i=1}^N \gamma_i) (\sum_{i=1}^N \Delta p_i) = \lim_{\Delta t \rightarrow 0} (\sum_{i=1}^N \gamma_i) \lim_{\Delta t \rightarrow 0} (\sum_{i=1}^N \Delta p_i) = \lim_{\Delta t \rightarrow 0} \sum_{i=1}^N \Delta p_i$, we can further decompose the second term in Eq. equation 7.28 into $\lim_{\Delta t \rightarrow 0} (\sum_{i=1}^N \gamma_i) (\sum_{i=1}^N \Delta p_i)$.

By extracting the limit out in Eq. equation 7.28, the equation becomes

$$\begin{aligned} & \lim_{\Delta t \rightarrow 0} \left\{ \gamma_1 [\log p(x_0) + \Delta p_1 + \log p(y_{\Delta t}|x_{\Delta t}) + c_1] \right. \\ & \quad + \gamma_2 [\log p(x_0) + \Delta p_1 + \Delta p_2 + \log p(y_{2\Delta t}|x_{2\Delta t}) + c_2] \\ & \quad + \dots \\ & \quad + \gamma_N [\log p(x_0) + \Delta p_1 + \Delta p_2 + \dots + \Delta p_N + \log p(y_{N\Delta t}|x_{N\Delta t}) + c_N] \\ & \quad \left. + [\gamma_1 \Delta p_2 + (\gamma_1 + \gamma_2) \Delta p_3 + \dots + (\gamma_1 + \gamma_2 + \dots + \gamma_{N-1}) \Delta p_N] - \log p(y) \right\} \quad (7.29) \end{aligned}$$

$$:= \lim_{\Delta t \rightarrow 0} \left[\sum_{i=1}^N \gamma_i \tilde{\mathcal{J}}_i + \sum_{j=2}^N \left(\sum_{i=1}^{j-1} \gamma_i \right) \Delta p_j + \sum_{i=1}^N \gamma_i c_i - \log p(y) \right], \quad (7.30)$$

where $\tilde{\mathcal{J}}_i := \log p(x_0) + \sum_{j=1}^i \Delta p_j + \log p(y_{i\Delta t}|x_{i\Delta t})$. We further define the $c(N) = \sum_{i=1}^N \gamma_i c_i - \log p(y)$.

Recall that $\hat{\mathcal{J}}_i = \log p(x_{(i-1)\Delta t}) - \text{tr} \left(\frac{\partial v_\theta(x_{(i-1)\Delta t}, (i-1)\Delta t)}{\partial x} \right) \Delta t + \log p(y_{i\Delta t}|x_{i\Delta t})$. By triangle

inequality, we have

$$\left| \log p(x(x_0)|y) - \sum_{i=1}^N \gamma_i \hat{\mathcal{J}}_i - c(N) \right| \quad (7.31)$$

$$\leq \left| \log p(x(x_0)|y) - \sum_{i=1}^N \gamma_i \tilde{\mathcal{J}}_i - c(N) \right| + \left| \sum_{i=1}^N \gamma_i \hat{\mathcal{J}}_i - \sum_{i=1}^N \gamma_i \tilde{\mathcal{J}}_i \right|. \quad (7.32)$$

Taking the limit on both sides, we have

$$\lim_{\Delta t \rightarrow 0} \left| \log p(x(x_0)|y) - \sum_{i=1}^N \gamma_i \hat{\mathcal{J}}_i - c(N) \right| \quad (7.33)$$

$$\leq \lim_{\Delta t \rightarrow 0} \left| \log p(x(x_0)|y) - \sum_{i=1}^N \gamma_i \tilde{\mathcal{J}}_i - c(N) \right| + \lim_{\Delta t \rightarrow 0} \left| \sum_{i=1}^N \gamma_i \hat{\mathcal{J}}_i - \sum_{i=1}^N \gamma_i \tilde{\mathcal{J}}_i \right|. \quad (7.34)$$

In the following, we analyze the two terms on the right-hand side one by one.

For the first term: as $|\cdot| : \mathbb{R} \rightarrow \mathbb{R}$ is a continuous function, the first term on the right-hand side is equal to

$$\left| \log p(x(x_0)|y) - \lim_{\Delta t \rightarrow 0} \sum_{i=1}^N \gamma_i \tilde{\mathcal{J}}_i - c(N) \right| \quad (7.35)$$

$$= \left| \lim_{\Delta t \rightarrow 0} \sum_{j=2}^N \left(\sum_{i=1}^{j-1} \gamma_i \right) \Delta p_j \right| \quad (7.36)$$

$$= \left| \lim_{\Delta t \rightarrow 0} \sum_{j=2}^N \left(\frac{1}{2^{N-j+1}} - \frac{1}{2^N} \right) \Delta p_j \right| \quad (7.37)$$

$$\leq \left| \lim_{\Delta t \rightarrow 0} \sum_{j=2}^N \left(\frac{1}{2^{N-j+1}} \right) \Delta p_j \right| + \left| \lim_{\Delta t \rightarrow 0} \sum_{j=2}^N \left(\frac{1}{2^N} \right) \Delta p_j \right|, \quad (7.38)$$

where the first equation is derived by subtracting the first term in Eq. equation 7.30 from Eq. equation 7.26. As the velocity field $v_\theta : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$ satisfies $\sup_{z \in \mathbb{R}^n, s \in [0,1]} |\text{tr} \frac{\partial}{\partial x} v_\theta(z, s)| \leq C_1$ for some universal constant C_1 , we have $|\Delta p_j| \leq C_1 \Delta t$. The first term in equation 7.38 would be

$$\left| \sum_{j=2}^N \left(\frac{1}{2^{N-j+1}} \right) \Delta p_j \right| \leq C_1 \Delta t \sum_{j=2}^N \left(\frac{1}{2^{N-j+1}} \right) \leq C_1 \Delta t = O(\Delta t). \quad (7.39)$$

Similarly, the second term in equation 7.38 would be

$$\left| \sum_{j=2}^n \left(\frac{1}{2^n} \right) \Delta p_j \right| \leq \sum_{j=2}^N \left(\frac{1}{2^N} \right) C_1 \Delta t = C_1 \left(\frac{N-1}{2^N} \right) \Delta t = O(\Delta t). \quad (7.40)$$

Combining the results in Eq. equation 7.39 and Eq. equation 7.40, we get

$$\left| \log p(x(x_0)|y) - \lim_{\Delta t \rightarrow 0} \sum_{i=1}^N \gamma_i \tilde{\mathcal{J}}_i - c(N) \right| = 0. \quad (7.41)$$

For the second term: Intuitively, the error between the integral and the Riemannian discretization goes to 0 as Δt tends to 0. Rigorously,

$$\lim_{\Delta t \rightarrow 0} \left| \sum_{i=1}^N \gamma_i \hat{\mathcal{J}}_i - \sum_{i=1}^N \gamma_i \tilde{\mathcal{J}}_i \right| = \lim_{\Delta t \rightarrow 0} \left| \sum_{i=1}^N \gamma_i (\hat{\mathcal{J}}_i - \tilde{\mathcal{J}}_i) \right| \quad (7.42)$$

$$= \lim_{\Delta t \rightarrow 0} \left| \sum_{i=1}^N \gamma_i \left(\int_0^{t-\Delta t} \text{tr} \left(\frac{\partial v_\theta(x_s, s)}{\partial x} \right) ds - \sum_{j=1}^{i-1} \Delta p_j \right) \right| \quad (7.43)$$

$$\leq \lim_{\Delta t \rightarrow 0} \sum_{i=1}^N \gamma_i \left| \int_0^{t-\Delta t} \text{tr} \left(\frac{\partial v_\theta(x_s, s)}{\partial x} \right) ds - \sum_{j=1}^{i-1} \Delta p_j \right| = 0. \quad (7.44)$$

Combining the results of the first term and the second term, we get the proof of theorem 1 done.

7.7.2 Compliance of Trajectory

To quantify our deviation from the assumption of having x_t exactly follow the interpolation path $\alpha_t x + \beta_t x_0$, we define the following: given a differentiable process $\{z_t\}$ and an interpolation path specified by $\alpha := \{\alpha_t\}$ and $\beta := \{\beta_t\}$, we define the trajectory's **compliance** $S_{\alpha, \beta}(\{z_t\})$ to the interpolation path as

$$S_{\alpha, \beta}(\{z_t\}) := \int_0^1 \mathbb{E}_{p(z_0), p(z_1)} \left[\|\dot{z}_t - (\dot{\alpha}_t z_1 + \dot{\beta}_t z_0)\|^2 \right] dt. \quad (7.45)$$

This generalizes the definition of straightness in [LGL22a] to general interpolation paths. We recover their definition by setting $\alpha_t = t$ and $\beta_t = 1 - t$. In certain cases, we have exact

compliance with the predefined interpolation path. For example, when $\{z_t\}$ is generated by v_θ and $\alpha_t = t$ and $\beta_t = 1 - t$, note that $S_{\alpha,\beta}(\{z_t\}) = 0$ is equivalent to $v_\theta(z_t, t) = c$ where c is a constant, almost everywhere. This ensures that $z_1 = z_0 + c$. In this case, when generating the trajectory through an ODE solver with starting point x_0 and endpoint x_t , we have $x_t = \alpha_t x + \beta_t x_0, \forall t$. When $S_{\alpha,\beta}(\{z_t\})$ is not equal to 0, we show in Proposition 2 that we can bound the deviation of our trajectory from the interpolation path using this compliance measure. When specifying our result to Rectified Flow, we can obtain an additional bound showing that when using L -Rectified Flow, the deviation of the learned trajectory from the straight trajectory is bounded by $O(1/L)$.

Proposition 2. *Consider a differentiable interpolation path specified by $\alpha := \{\alpha_t\}$ and $\beta := \{\beta_t\}$. Then the expected distance between the learned trajectory $z_t = z_0 + \int_0^t v_\theta(z_s, s)ds$ and the predefined trajectory $\hat{z}_t = z_0 + \int_0^t (\dot{\alpha}_s z_1 + \dot{\beta}_s z_0)ds$ can be bounded as*

$$\mathbb{E}_{p(z_0), p(z_1)} [\|\hat{z}_t - z_t\|^2] \leq S_{\alpha,\beta}(\{z_t\}). \quad (7.46)$$

If the differentiable process $\{z_t\}$ is specified by L -Rectified Flow and $\alpha_t = t$ and $\beta_t = 1 - t$ for all $t \in [0, 1]$, then we additionally have

$$\mathbb{E}_{p(z_0), p(z_1)} [\|\hat{z}_t - z_t\|^2] \leq O\left(\frac{1}{L}\right). \quad (7.47)$$

Proof. At time t , we are interested in the distance between a real trajectory $z_t = z_0 + \int_0^t v_\theta(z_s, s)ds$ and a preferred trajectory $\hat{z}_t = z_0 + \int_0^t (\dot{\alpha}_s z_1 - \dot{\beta}_s z_0)ds$. Using the result in Lemma 14, the distance can be bounded by

$$\|\hat{z}_t - z_t\|^2 = \left\| \int_0^t [v_\theta(z_s, s) - (\dot{\alpha}_s z_1 - \dot{\beta}_s z_0)]ds \right\|^2 \quad (7.48)$$

$$\leq \int_0^t \|v_\theta(z_s, s) - (\dot{\alpha}_s z_1 - \dot{\beta}_s z_0)\|^2 ds. \quad (7.49)$$

Therefore,

$$\mathbb{E}_{p(z_0), p(z_1)} \|\hat{z}_t - z_t\|^2 \leq \mathbb{E}_{p(z_0), p(z_1)} \left[\int_0^t \|v_\theta(z_s, s) - (\dot{\alpha}_s z_1 - \dot{\beta}_s z_0)\|^2 ds \right] \quad (7.50)$$

$$= \int_0^t \mathbb{E}_{p(z_0), p(z_1)} \|v_\theta(z_s, s) - (\dot{\alpha}_s z_1 - \dot{\beta}_s z_0)\|^2 ds \quad (7.51)$$

$$\leq \int_0^1 \mathbb{E}_{p(z_0), p(z_1)} \|v_\theta(z_s, s) - (\dot{\alpha}_s z_1 - \dot{\beta}_s z_0)\|^2 ds \quad (7.52)$$

$$:= S_{\alpha, \beta}(\{z\}). \quad (7.53)$$

If $\{z_t, t \in [0, 1]\}$ is a learned L -rectified flow, i.e. $\alpha_t = t$ and $\beta_t = 1 - t$ in this case, where L is the times of rectifying the flow, by Theorem 3.7 in [LGL22a], we have $S_{\alpha, \beta}(\{z\}) = O(1/L)$ and thus

$$\mathbb{E}_{p(z_0), p(z_1)} \|\hat{z}_t - z_t\|^2 = O(1/L). \quad (7.54)$$

□

Empirically, [LZM24, LGL22a] found $L = 2$ generates nearly straight trajectories for high-quality one-step generation. Hence, while this result gives us a simple upper bound, in practice the trajectories may comply more faithfully with the predefined interpolation path than this result suggests.

7.7.3 Computational Efficiency

In Tab. 7.3, we present the computational efficiency comparison results. Note that OT-ODE is the slowest as it requires taking the inverse of a matrix $r_t^2 \mathcal{A} \mathcal{A}^T + \sigma_y^2 I$ each update time. Our method requires taking the gradient over an estimated trace of the Jacobian matrix, which slows the computation.

7.7.4 Implementation Details

Experiments were conducted on a Linux-based system with CUDA 12.2 equipped with 4 Nvidia R9000 GPUs, each of them has 48GB of memory.

Table 7.3: **Computational time comparison.** We compare the time required to recover 100 images for the super-resolution task on a single GPU.

	DPS-ODE	OT-ODE	Ours (w/o prior)	Ours
Time(h)	0.36	4.10	0.83	2.72

Operators For all the experiments on the CelebA-HQ dataset, we use the operators from [CKM23]. For all the experiments on compressed sensing, we use the operator *Compressed-SensingOperator* defined in the official repository of [FBS24]³,

Evaluation Metrics are implemented with different Python packages. PSNR is calculated using basic PyTorch operations, and SSIM is computed using the *pytorch_msssim* package.

7.7.4.1 Toy example

The workflow begins with using 1,000 FFHQ images at a resolution of 1024×1024. These images are then downsampled to 16×16 using bicubic resizing. A Gaussian Mixture model is applied to fit the downsampled images, resulting in mean and covariance parameters. The mean values are transformed from the original range of [0,1] to [-1,1]. Subsequently, 10,000 samples are generated from this distribution to facilitate training a score-based model resembling the architecture of CIFAR10 DDPM++. The training process involves 10,000 iterations, each with a batch size of 64, and utilizes the Adam optimizer [Kin14] with a learning rate of 2e-4 and a warmup phase lasting 100 steps. Notably, convergence is achieved within approximately 200 steps. Lastly, the estimated log-likelihood computation for a batch size of 128 takes around 4 minutes and 30 seconds. We show uncured samples generated from the trained models in Fig. 7.7.

³<https://github.com/Sulam-Group/learned-proximal-networks/tree/main>



Figure 7.7: Generated samples from the flow trained on 10,000 Gaussian samples.

7.7.4.2 Medical Application

In this setting, $\sigma_y = 0.001$. We use the *ncsnpp* architecture, training from scratch on 10k images for 100k iterations with a batch size of 50. We set the learning rate to 1×10^{-2} . Sudden convergence appeared during our training process. We use 2000 warmup steps. Uncured generated images are presented in Fig. 7.8.

7.7.4.3 Implementation of Baselines

OT-ODE As OT-ODE [PMC23] has not released their code and pretrained checkpoints. We reproduce their method with the same architecture as in [LGL22a]. We follow their setting and find initialization time t' has a great impact on the performance. We use the *y* *init* method in their paper. Specifically, the starting point is

$$x_{t'} = t'y + (1 - t')\epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \quad (7.55)$$

where t' is the init time. Note that in the super-resolution task we upscale y with bicubic

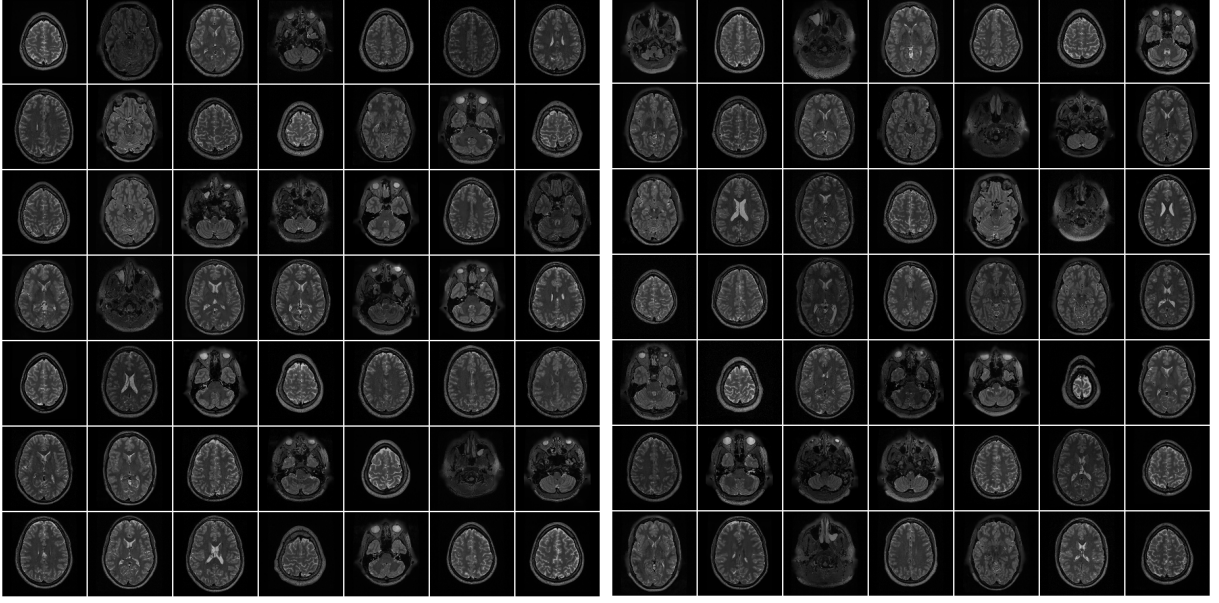


Figure 7.8: Generated samples from the flow trained on 10,000 HCP T2w images.

first. We follow the guidance in the paper and show the ablation results in Fig. 7.9 and Fig. 7.10.

DPS-ODE We use the following formula to update for each step in the flow:

$$v(x_t, y) = v(x_t) + \zeta_t \left(-\nabla_{x_t} \|y - \mathcal{A}\hat{x}_1\|^2 \right),$$

where ζ_t is the step size to tune. We refer to DPS for the method to choose ζ_t . We set $\zeta_t = \frac{\eta}{2\|y - \mathcal{A}\hat{x}_1(x_t)\|}$. We demonstrate the ablation of η for this baseline in Fig. 7.11 and Fig. 7.12. Note that there is a significant divergence in PSNR and SSIM for the task of inpainting (box). As we observe that artifacts are likely to appear when $\eta \geq 100$, we choose the optimal $\eta = 75$ for the best tradeoff.

RED-Diff and IIGDM We use the official repository⁴ from Nvidia to reproduce the results of RED-Diff and IIGDM with the pretrained CelebAHQ checkpoint using the archi-

⁴<https://github.com/NVlabs/RED-diff>

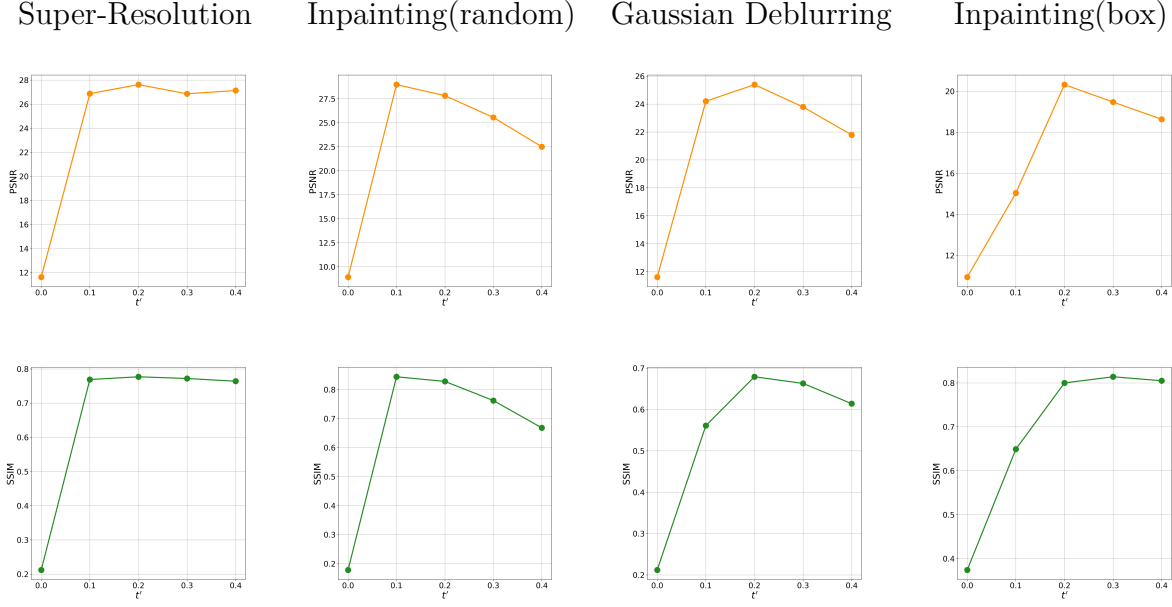


Figure 7.9: Hyperparameter t' selection results for OT-ODE on the CelebA-HQ dataset. We select $t' = 0.2, 0.1, 0.2, 0.2$ for super-resolution, inpainting(random), Gaussian deblurring, and inpainting(box), respectively.

tecture of the guided diffusion repository⁵ from OpenAI.

For RED-Diff, the optimization objective is $\min_{\mu} \|y - \mathcal{A}(\mu)\|^2 + \lambda (sg(\epsilon_{\theta}(x_t, t) - \epsilon))^T \mu$. Following the implementation of the original paper, we use Adam optimizer with 1,000 steps for all tasks. We choose learning rate $lr = 0.25, \lambda = 0.25$ for super-resolution, inpainting(random) and inpainting(box) and $lr = 0.5, \lambda = 0.25$ for deblurring as recommended by the paper.

For IIGDM, we follow the original paper and use 100 diffusion steps. Specifically, we use $\eta = 1.0$ which corresponds to the VE-SDE. Adaptive weights $r_t^2 = \frac{\sigma_{1-t}^2}{1+\sigma_t^2}$ are used if there is an improvement on metrics.

⁵<https://github.com/openai/guided-diffusion>

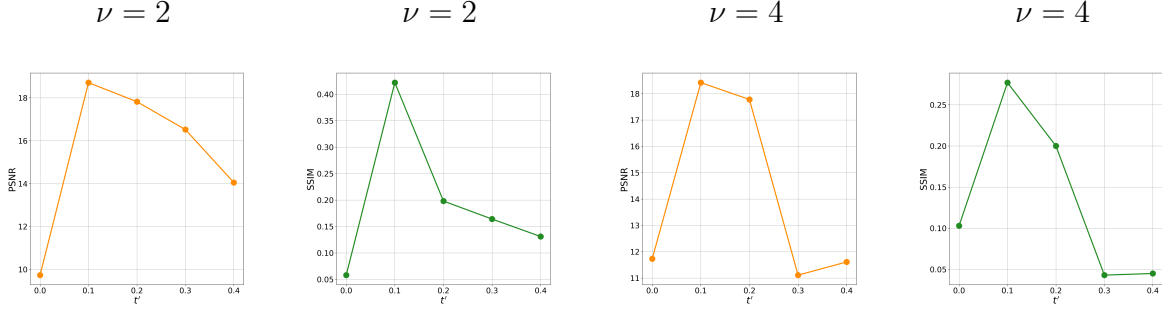


Figure 7.10: Hyperparameter t' selection results for OT-ODE on the HCP T2w dataset. We select $t' = 0.1$ for all the experiments.

Wavelet and TV priors We use the pytorch package DeepInverse⁶ to implement Wavelet and TV priors. For both priors, we use the default Proximal Gradient Descent (PGD) algorithm and perform a grid search for regularization weight λ in the set $\{10^0, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$ and gradient stepsize η in $\{10^1, 10^0, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$. The maximum number of iteration is 3k, 5k, and 10k for compression rate $\nu = 1/2, 1/4$, and $1/10$, respectively. The stopping criterion is the residual norm $\frac{\|x_{t-1} - x_t\|}{\|x_{t-1}\|} \leq 1 \times 10^{-5}$ and the initialization of the algorithm is the backprojected reconstruction, i.e., the pseudoinverse of $\mathcal{A}$ applied to the measurement y .

For the TV prior, the objective we aim to minimize is $\min_x \frac{1}{2} \|\mathcal{A}x - y\|_2^2 + \lambda \|x\|_{TV}$. We find that the optimal combination of hyperparameters is $\lambda = 0.01, \eta = 0.1$ for all the values of ν .

For the Wavelet prior, the objective we want to minimize is $\min_x \frac{1}{2} \|\mathcal{A}x - y\|_2^2 + \lambda \|\Psi x\|_1$. We use the default level of the wavelet transform and select the “db8” Wavelet. The optimal combination of hyperparameters is $\lambda = 0.1, \eta = 0.1$ for all the values of ν .

⁶<https://deepinv.github.io/deepinv/>

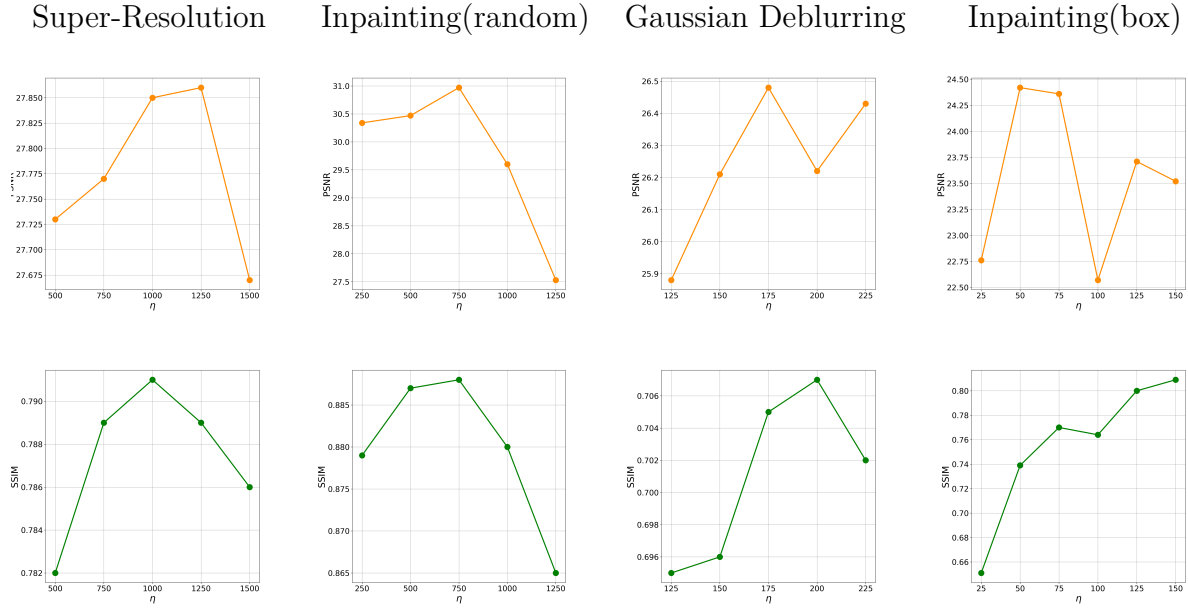


Figure 7.11: Hyperparameter η selection results for DPS-ODE. We select $\eta = 1000, 750, 200, 75$ for super-resolution, inpainting(random), Gaussian deblurring, and inpainting(box), respectively.

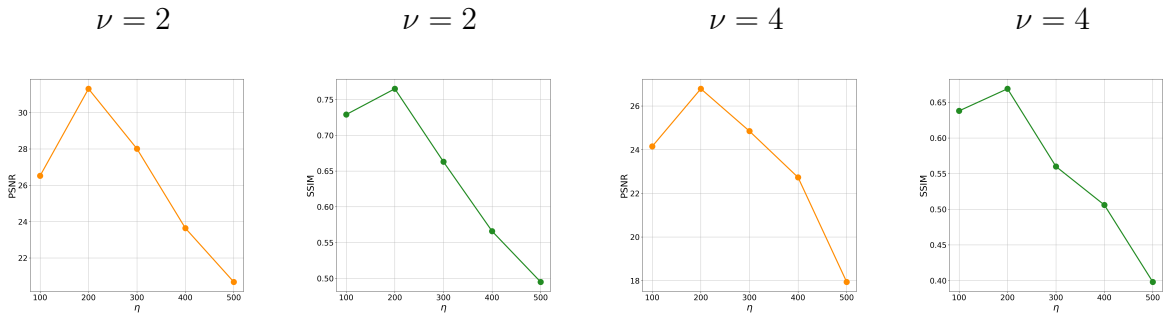


Figure 7.12: Hyperparameter η selection results for DPS-ODE on the HCP T2w dataset. We select $\eta = 200$ for all the experiments.

CHAPTER 8

Conclusion and Future Directions

This dissertation develops a closed-loop view of multimodal generative AI under real-world constraints. The evaluation chapters replace monolithic scores with object-centric, multi-axis measurements and show how those measurements can supervise sequential image editing. The alignment chapters preserve task structure in optimization, whether that structure is the ordinal geometry of numeric judgments or the object–attribute composition of text prompts. The generation chapters extend diffusion and flow models to settings in which only corrupted observations or indirect measurements are available.

Across the six works, a common lesson is that the learning signal should match the structure of the deployment problem. Whole-image similarity is insufficient when only one object should change. Binary reward is insufficient when prediction error is graded. Standard clean-data generation assumptions are insufficient when measurements are noisy or incomplete. Making these structures explicit yields both better objectives and more interpretable failure analysis.

8.1 Future Directions

A first direction is to make evaluation itself adaptive. EdiVal-Agent relies on a modular collection of detectors, vision-language models, feature extractors, and preference models. Future systems should estimate evaluator uncertainty, route difficult cases to specialized tools, and distinguish model failure from evaluator failure.

A second direction is to connect regression-aware alignment with multimodal evaluation. Fine-grained editing scores are naturally ordinal and correlated across turns. A unified learner could optimize their calibrated values rather than thresholding them, while explicitly assigning credit across a long interaction.

A third direction is to combine corrupted-data learning with downstream control. Restoration score distillation and flow priors show complementary ways to use imperfect observations: one learns a fast generator, while the other performs test-time inference with a pretrained prior. Joint training could amortize inverse-problem inference while retaining measurement consistency and calibrated uncertainty.

References

- [AAA19] Kazunori Akiyama, Antxon Alberdi, Walter Alef, Keiichi Asada, Rebecca Azulay, Anne-Kathrin Baczko, David Ball, Mislav Baloković, John Barrett, Dan Bintley, et al. “First M87 event horizon telescope results. IV. Imaging the central super-massive black hole.” *The Astrophysical Journal Letters*, **875**(1):L4, 2019.
- [AAA23a] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. “Gpt-4 technical report.” *arXiv preprint arXiv:2303.08774*, 2023.
- [AAA23b] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. “Gpt-4 technical report.” *arXiv preprint arXiv:2303.08774*, 2023.
- [AAK23] Asad Aali, Marius Arvinte, Sidharth Kumar, and Jonathan I Tamir. “Solving inverse problems with score-based generative priors learned from noisy data.” In *2023 57th Asilomar Conference on Signals, Systems, and Computers*, pp. 837–843. IEEE, 2023.
- [ABC21] Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. “A general language assistant as a laboratory for alignment.” *arXiv preprint arXiv:2112.00861*, 2021.
- [ACG24] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. “Back to Basics: Revisiting REINFORCE-Style Optimization for Learning from Human Feedback in LLMs.”

- In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, Bangkok, Thailand, August 2024. Association for Computational Linguistics.
- [ADL20] Muhammad Asim, Max Daniels, Oscar Leong, Ali Ahmed, and Paul Hand. “Invertible generative models for inverse problems: mitigating representation error and dataset bias.” *Proceedings of the 37th International Conference on Machine Learning*, 2020.
- [ADL24] Asad Aali, Giannis Daras, Brett Levac, Sidharth Kumar, Alexandros G Dimakis, and Jonathan I Tamir. “Ambient diffusion posterior sampling: Solving inverse problems with diffusion models trained on corrupted data.” *arXiv preprint arXiv:2403.08728*, 2024.
- [ADL25] Asad Aali, Giannis Daras, Brett Levac, Sidharth Kumar, Alex Dimakis, and Jon Tamir. “Ambient Diffusion Posterior Sampling: Solving Inverse Problems with Diffusion Models Trained on Corrupted Data.” In *The Thirteenth International Conference on Learning Representations*, 2025.
- [AKJ23] Aishwarya Agarwal, Srikrishna Karanam, KJ Joseph, Apoorv Saxena, Koustava Goswami, and Balaji Vasan Srinivasan. “A-star: Test-time attention segregation and retention for text-to-image synthesis.” In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2283–2293, 2023.
- [BCL25] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibor Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. “Qwen2. 5-vl technical report.” *arXiv preprint arXiv:2502.13923*, 2025.
- [BGJ11] Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of markov chain monte carlo*. CRC press, 2011.

- [BHE23] Tim Brooks, Aleksander Holynski, and Alexei A Efros. “Instructpix2pix: Learning to follow image editing instructions.” In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18392–18402, 2023.
- [BJD23] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. “Training diffusion models with reinforcement learning.” *arXiv preprint arXiv:2305.13301*, 2023.
- [BJP17] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros Dimakis. “Compressed sensing using generative models.” *International Conference on Machine Learning*, 2017.
- [BKL22] Matteo Bottai, Taeho Kim, Benjamin Lieberman, George Luta, and Edsel Peña. “On optimal correlation-based prediction.” *The American Statistician*, **76**(4):313–321, 2022.
- [BL14] Martin Burger and Felix Lucka. “Maximum a posteriori estimates in linear inverse problems with log-concave priors are proper Bayes estimators.” *Inverse Problems*, **30**(11):114004, 2014.
- [BM18] Yochai Blau and Tomer Michaeli. “The perception-distortion tradeoff.” In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6228–6237, 2018.
- [BMR20] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford,

- Ilya Sutskever, and Dario Amodei. “Language Models are Few-Shot Learners.”, 2020.
- [BN06] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.
- [BNH22] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. “ediffi: Text-to-image diffusion models with an ensemble of expert denoisers.” *arXiv preprint arXiv:2211.01324*, 2022.
- [BPG24] Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. “D-Flow: Differentiating through Flows for Controlled Generation.” In *Forty-first International Conference on Machine Learning*, 2024.
- [BR19] Joshua Batson and Loic Royer. “Noise2self: Blind denoising by self-supervision.” In *International conference on machine learning*, pp. 524–533. PMLR, 2019.
- [Buz11] Thorsten M Buzug. “Computed tomography.” In *Springer handbook of medical technology*, pp. 311–342. Springer, 2011.
- [BWC24] Weimin Bai, Yifei Wang, Wenzheng Chen, and He Sun. “An Expectation-Maximization Algorithm for Training Clean Diffusion Models from Corrupted Observations.” In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [BWX24] Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. “Hallucination of multimodal large language models: A survey.” *arXiv preprint arXiv:2404.18930*, 2024.
- [CAV23] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. “Attend-

- and-excite: Attention-based semantic guidance for text-to-image diffusion models.” *ACM Transactions on Graphics (TOG)*, **42**(4):1–10, 2023.
- [CHZ23] Minshuo Chen, Kaixuan Huang, Tuo Zhao, and Mengdi Wang. “Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data.” In *International Conference on Machine Learning*, pp. 4672–4712. PMLR, 2023.
- [CKM22] Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. “Diffusion posterior sampling for general noisy inverse problems.” *arXiv preprint arXiv:2209.14687*, 2022.
- [CKM23] Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. “Diffusion Posterior Sampling for General Noisy Inverse Problems.” In *The Eleventh International Conference on Learning Representations, ICLR 2023*. The International Conference on Learning Representations, 2023.
- [CKS24] Sitan Chen, Vasilis Kontonis, and Kulin Shah. “Learning general gaussian mixtures with efficient score matching.” *arXiv preprint arXiv:2404.18893*, 2024.
- [CKV23] Hugo Cui, Florent Krzakala, Eric Vanden-Eijnden, and Lenka Zdeborová. “Analysis of learning a flow-based generative model from limited sample complexity.” *arXiv preprint arXiv:2310.03575*, 2023.
- [CL23] Cheng-Han Chiang and Hung-Yi Lee. “Can Large Language Models Be an Alternative to Human Evaluations?” In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15607–15631, 2023.
- [CLL23] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Sto-

- ica, and Eric P. Xing. “Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.”, March 2023.
- [CLL25] Cheng-Han Chiang, Hung-yi Lee, and Michal Lukasik. “TRACT: Regression-Aware Fine-tuning Meets Chain-of-Thought Reasoning for LLM-as-a-Judge.” In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2934–2952, Vienna, Austria, July 2025. Association for Computational Linguistics.
- [CRB18] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. “Neural Ordinary Differential Equations.” In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.
- [CTM21] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. “Emerging Properties in Self-Supervised Vision Transformers.” In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.
- [CU22] Colin Conwell and Tomer Ullman. “Testing relational understanding in text-guided image generation.” *arXiv preprint arXiv:2208.00005*, 2022.
- [CWZ24] Tianyu Chen, Zhendong Wang, and Mingyuan Zhou. “Enhancing and Accelerating Diffusion-Based Inverse Problem Solving through Measurements Optimization.” *arXiv preprint arXiv:2412.03941*, 2024.
- [CXK24] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. “Spatialvlm: Endowing vision-language models with spa-

- tial reasoning capabilities.” In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14455–14465, 2024.
- [CYF24] An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. “Spatialrgpt: Grounded spatial reasoning in vision-language models.” *Advances in Neural Information Processing Systems*, **37**:135062–135093, 2024.
- [CZF24] Yingshan Chang, Yasi Zhang, Zhiyuan Fang, Yingnian Wu, Yonatan Bisk, and Feng Gao. “Skews in the Phenomenon Space Hinder Generalization in Text-to-Image Generation.” *arXiv preprint arXiv:2403.16394*, 2024.
- [CZF25] Yingshan Chang, Yasi Zhang, Zhiyuan Fang, Ying Nian Wu, Yonatan Bisk, and Feng Gao. “Skews in the Phenomenon Space Hinder Generalization in Text-to-Image Generation.” In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pp. 422–439, Cham, 2025. Springer Nature Switzerland.
- [CZZ25] Tianyu Chen, Yasi Zhang, Zhi Zhang, Peiyu Yu, Shu Wang, Zhendong Wang, Kevin Lin, Xiaofei Wang, Zhengyuan Yang, Linjie Li, et al. “EdiVal-Agent: An Object-Centric Framework for Automated, Scalable, Fine-Grained Evaluation of Multi-Turn Editing.” *arXiv preprint arXiv:2509.13399*, 2025.
- [CZZ26] Tianyu Chen, Yasi Zhang, Zhi Zhang, Peiyu Yu, Shu Wang, Zhendong Wang, Kevin Lin, Xiaofei Wang, Zhengyuan Yang, Linjie Li, Chung-Ching Lin, Jianwen Xie, Oscar Leong, Lijuan Wang, Ying Nian Wu, and Mingyuan Zhou. “EdiVal-Agent: An Object-Centric Framework for Automated, Fine-Grained Evaluation of Multi-Turn Editing.” In *The Fourteenth International Conference on Learning Representations*, 2026.
- [DCD25] Giannis Daras, Yeshwanth Cherapanamjeri, and Constantinos Costis Daskalakis.

- “How Much is a Noisy Image Worth? Data Scaling Laws for Ambient Diffusion.” In *The Thirteenth International Conference on Learning Representations*, 2025.
- [DDD24] Giannis Daras, Alex Dimakis, and Constantinos Costis Daskalakis. “Consistent Diffusion Meets Tweedie: Training Exact Ambient Diffusion Models with Noisy Data.” In *Forty-first International Conference on Machine Learning*, 2024.
- [Dee25] Google Deepmind. “Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities.”, 2025.
- [DFK07] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. “Image denoising by sparse 3-D transform-domain collaborative filtering.” *IEEE Transactions on image processing*, **16**(8):2080–2095, 2007.
- [DJ94] David L Donoho and Iain M Johnstone. “Ideal spatial adaptation by wavelet shrinkage.” *biometrika*, **81**(3):425–455, 1994.
- [DN21] Prafulla Dhariwal and Alexander Nichol. “Diffusion models beat GANs on image synthesis.” *Advances in neural information processing systems*, **34**:8780–8794, 2021.
- [Don06] David L Donoho. “Compressed sensing.” *IEEE Transactions on information theory*, **52**(4):1289–1306, 2006.
- [DSD23] Giannis Daras, Kulin Shah, Yuval Dagan, Aravind Gollakota, Alex Dimakis, and Adam Klivans. “Ambient diffusion: Learning clean distributions from corrupted data.” *Advances in Neural Information Processing Systems*, **36**:288–313, 2023.
- [DSR22] Arjun D Desai, Andrew M Schmidt, Elka B Rubin, Christopher M Sandino, Marianne S Black, Valentina Mazzoli, Kathryn J Stevens, Robert Boutin, Christopher Ré, Garry E Gold, et al. “Skm-tea: A dataset for accelerated mri reconstruc-

- tion with dense image labels for quantitative clinical evaluation.” *arXiv preprint arXiv:2203.06823*, 2022.
- [DZL25] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. “Emerging properties in unified multimodal pretraining.” *arXiv preprint arXiv:2505.14683*, 2025.
- [Efr11] Bradley Efron. “Tweedie’s formula and selection bias.” *Journal of the American Statistical Association*, **106**(496):1602–1614, 2011.
- [EKB24] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. “Scaling rectified flow transformers for high-resolution image synthesis.” *arXiv preprint arXiv:2403.03206*, 2024.
- [ERO21] Patrick Esser, Robin Rombach, and Bjorn Ommer. “Taming transformers for high-resolution image synthesis.” In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.
- [FB24] Berthy Feng and Katherine Bouman. “Variational Bayesian Imaging with an Efficient Surrogate Score-based Prior.” *Transactions on Machine Learning Research*, 2024.
- [FBS24] Zhenghan Fang, Sam Buchanan, and Jeremias Sulam. “What’s in a Prior? Learned Proximal Networks for Inverse Problems.” In *International Conference on Learning Representations*, 2024.
- [FHF23] Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Reddy Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. “Training-Free Structured Diffusion Guidance for Compositional Text-to-Image Synthesis.” In *The Eleventh International Conference on Learning Representations*, 2023.

- [FR13] Simon Foucart and Holger Rauhut. “An invitation to compressive sensing.” In *A mathematical introduction to compressive sensing*, pp. 1–39. Springer, 2013.
- [FSR23] Berthy T Feng, Jamie Smith, Michael Rubinstein, Huiwen Chang, Katherine L Bouman, and William T Freeman. “Score-Based Diffusion Models as Principled Priors for Inverse Imaging.” In *International Conference on Computer Vision (ICCV)*. IEEE, 2023.
- [FWD23] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. “Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models.” *Advances in Neural Information Processing Systems*, **36**:79858–79885, 2023.
- [GCB19] Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. “FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models.” In *International Conference on Learning Representations*, 2019.
- [Gem25] Google Gemini2. “Experiment with Gemini 2.0 Flash Native Image Generation.” <https://developers.googleblog.com/en/experiment-with-gemini-20-flash-native-image-generation/>, 2025. Accessed: 2025-06-22.
- [GGG25] Yu Gao, Lixue Gong, Qiushan Guo, Xiaoxia Hou, Zhichao Lai, Fanshi Li, Liang Li, Xiaochen Lian, Chao Liao, Liyang Liu, et al. “Seedream 3.0 technical report.” *arXiv preprint arXiv:2504.11346*, 2025.
- [GJS24] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. “A survey on llm-as-a-judge.” *arXiv preprint arXiv:2411.15594*, 2024.

- [Goo23] Google. “Bard: An Experimental AI Chatbot.” <https://blog.google/technology/ai/bard-google-ai-search-updates/>, 2023.
- [GPM20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. “Generative adversarial networks.” *Communications of the ACM*, **63**(11):139–144, 2020.
- [GPM22] Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. “Stylegan-nada: Clip-guided domain adaptation of image generators.” *ACM Transactions on Graphics (TOG)*, **41**(4):1–13, 2022.
- [GYZ25] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. “Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.” *arXiv preprint arXiv:2501.12948*, 2025.
- [HB15] Tapio Helin and Martin Burger. “Maximum a posteriori probability estimates in infinite-dimensional Bayesian inverse problems.” *Inverse Problems*, **31**(8):085009, 2015.
- [HHS23] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. “simple diffusion: End-to-end diffusion for high resolution images.” *arXiv preprint arXiv:2301.11093*, 2023.
- [Hig02] Nicholas J Higham. *Accuracy and stability of numerical algorithms*. SIAM, 2002.
- [HJA20a] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising diffusion probabilistic models.” *Advances in neural information processing systems*, **33**:6840–6851, 2020.
- [HJA20b] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising diffusion probabilistic models.” *Advances in neural information processing systems*, **33**:6840–6851, 2020.

- [HLP23] Benjamin Hoover, Yuchen Liang, Bao Pham, Rameswar Panda, Hendrik Strobelt, Duen Horng Chau, Mohammed J Zaki, and Dmitry Krotov. “Energy transformer.” *arXiv preprint arXiv:2302.07253*, 2023.
- [HM17] Matthew Honnibal and Ines Montani. “spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.” *To appear*, **7**(1):411–420, 2017.
- [HMT23a] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. “Prompt-to-Prompt Image Editing with Cross-Attention Control.” In *ICLR*, 2023.
- [HMT23b] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-or. “Prompt-to-Prompt Image Editing with Cross-Attention Control.” In *The Eleventh International Conference on Learning Representations*, 2023.
- [Hop82] John J Hopfield. “Neural networks and physical systems with emergent collective computational abilities.” *Proceedings of the national academy of sciences*, **79**(8):2554–2558, 1982.
- [HRU17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. “GANs trained by a two time-scale update rule converge to a local nash equilibrium.” *Advances in neural information processing systems*, **30**, 2017.
- [HS22a] Jonathan Ho and Tim Salimans. “Classifier-free diffusion guidance.” *arXiv preprint arXiv:2207.12598*, 2022.
- [HS22b] Jonathan Ho and Tim Salimans. “Classifier-free diffusion guidance.” *arXiv preprint arXiv:2207.12598*, 2022.

- [HSC22] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. “Cascaded diffusion models for high fidelity image generation.” *Journal of Machine Learning Research*, **23**(47):1–33, 2022.
- [HSW22] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. “Lora: Low-rank adaptation of large language models.” *Iclr*, **1**(2):3, 2022.
- [Hut89] Michael F Hutchinson. “A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines.” *Communications in Statistics-Simulation and Computation*, **18**(3):1059–1076, 1989.
- [HWD25] Johannes Hertrich, Hok Shing Wong, Alexander Denker, Stanislas Ducotterd, Zhenghan Fang, Markus Haltmeier, Željko Kereta, Erich Kobler, Oscar Leong, Mohammad Sadegh Salehi, et al. “Learning regularization functionals for inverse problems: A comparative study.” *arXiv preprint arXiv:2510.01755*, 2025.
- [HYZ24] Mude Hui, Siwei Yang, Bingchen Zhao, Yichun Shi, Heng Wang, Peng Wang, Yuyin Zhou, and Cihang Xie. “Hq-edit: A high-quality dataset for instruction-based image editing.” *arXiv preprint arXiv:2404.09990*, 2024.
- [IOF92] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. “Nonlinear total variation based noise removal algorithms.” *Physica D: Nonlinear Phenomena*, **60**:259–268, 1992.
- [JAD21] Ajil Jalal, Marius Arvinte, Giannis Daras, Eric Price, Alexandros G Dimakis, and Jon Tamir. “Robust compressed sensing mri with deep generative priors.” *Advances in Neural Information Processing Systems*, **34**:14938–14954, 2021.
- [Jan14] Peter A Jansson. *Deconvolution of images and spectra*. Courier Corporation, 2014.

- [JSM23] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. “Mistral 7B.”, 2023.
- [KAA22] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. “Elucidating the design space of diffusion-based generative models.” *Advances in neural information processing systems*, **35**:26565–26577, 2022.
- [KAL17a] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. “Progressive growing of GANs for improved quality, stability, and variation.” *arXiv preprint arXiv:1710.10196*, 2017.
- [KAL17b] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. “Progressive growing of gans for improved quality, stability, and variation.” *arXiv preprint arXiv:1710.10196*, 2017.
- [Ken38] M. G. Kendall. “A NEW MEASURE OF RANK CORRELATION.” *Biometrika*, **30**(1-2):81–93, 06 1938.
- [KG23] Diederik Kingma and Ruiqi Gao. “Understanding Diffusion Objectives as the ELBO with Simple Data Augmentation.” In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pp. 65484–65516. Curran Associates, Inc., 2023.
- [KHW19] Wouter Kool, Herke van Hoof, and Max Welling. “Buy 4 REINFORCE Samples, Get a Baseline for Free!”, 2019.
- [Kin14] Diederik P Kingma. “Adam: A method for stochastic optimization.” *arXiv preprint arXiv:1412.6980*, 2014.

- [KJW23] Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhua Chen. “Viescore: Towards explainable metrics for conditional image synthesis evaluation.” *arXiv preprint arXiv:2312.14867*, 2023.
- [KMH20] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. “Scaling laws for neural language models.” *arXiv preprint arXiv:2001.08361*, 2020.
- [KRA20] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Mallocci, Alexander Kolesnikov, et al. “The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale.” *International journal of computer vision*, **128**(7):1956–1981, 2020.
- [KSC23] Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, et al. “Prometheus: Inducing fine-grained evaluation capability in language models.” In *The Twelfth International Conference on Learning Representations*, 2023.
- [KSL24] Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. “Prometheus 2: An Open Source Language Model Specialized in Evaluating Other Language Models.” In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 4334–4353, Miami, Florida, USA, November 2024. Association for Computational Linguistics.
- [KW13] Diederik P Kingma and Max Welling. “Auto-encoding variational bayes.” *arXiv preprint arXiv:1312.6114*, 2013.
- [KZS20] Florian Knoll, Jure Zbontar, Anuroop Sriram, Matthew J Muckley, Mary Bruno,

- Aaron Defazio, Marc Parente, Krzysztof J Geras, Joe Katsnelson, Hersh Chandarana, et al. “fastMRI: A publicly available raw k-space and DICOM dataset of knee images for accelerated MR image reconstruction using machine learning.” *Radiology: Artificial Intelligence*, **2**(1):e190007, 2020.
- [Lab25] Black Forest Labs. “FLUX.2: Frontier Visual Intelligence.” <https://bfl.ai/blog/flux-2>, 2025.
- [LBB25] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. “FLUX.1 Kontext: Flow Matching for In-Context Image Generation and Editing in Latent Space.”, 2025.
- [LCB22] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. “Flow matching for generative modeling.” *arXiv preprint arXiv:2210.02747*, 2022.
- [LCH25] Junzhe Li, Yutao Cui, Tao Huang, Yinping Ma, Chun Fan, Miles Yang, and Zhao Zhong. “MixGRPO: Unlocking Flow-based GRPO Efficiency with Mixed ODE-SDE.”, 2025.
- [LDP07] Michael Lustig, David Donoho, and John M Pauly. “Sparse MRI: The application of compressed sensing for rapid MR imaging.” *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, **58**(6):1182–1195, 2007.
- [LDS08] Michael Lustig, David L. Donoho, Juan M. Santos, and John M. Pauly. “Compressed Sensing MRI.” *IEEE Signal Processing Magazine*, **25**(2):72–82, 2008.

- [LGB24] Yvette Y Lin, Angela F Gao, and Katherine L Bouman. “Imaging An Evolving Black Hole By Leveraging Shared Structure.” In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2475–2479. IEEE, 2024.
- [LGL22a] Xingchao Liu, Chengyue Gong, and Qiang Liu. “Flow straight and fast: Learning to generate and transfer data with rectified flow.” *arXiv preprint arXiv:2209.03003*, 2022.
- [LGL22b] Xingchao Liu, Chengyue Gong, and Qiang Liu. “Flow straight and fast: Learning to generate and transfer data with rectified flow.” *arXiv preprint arXiv:2209.03003*, 2022.
- [LHL24] Youwei Liang, Junfeng He, Gang Li, Peizhao Li, Arseniy Klimovskiy, Nicholas Carolan, Jiao Sun, Jordi Pont-Tuset, Sarah Young, Feng Yang, Junjie Ke, Krishnamurthy Dj Dvijotham, Katie Collins, Yiwen Luo, Yang Li, Kai J Kohlhoff, Deepak Ramachandran, and Vidhya Navalpakkam. “Rich Human Feedback for Text-to-Image Generation.” In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.
- [LHX25] Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. “Step1x-edit: A practical framework for general image editing.” *arXiv preprint arXiv:2504.17761*, 2025.
- [LHZ23] Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. “Diff-Instruct: A universal approach for transferring knowledge from pre-trained diffusion models.” *Advances in Neural Information Processing Systems*, **36**:76525–76546, 2023.
- [LHZ24] Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua

- Zhang. “Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models.” *Advances in Neural Information Processing Systems*, **36**, 2024.
- [LIX23] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. “G-Eval: NLG Evaluation using Gpt-4 with Better Human Alignment.” In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2023.
- [LLD22] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. “Compositional visual generation with composable diffusion models.” In *European Conference on Computer Vision*, pp. 423–439. Springer, 2022.
- [LLL25] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di ZHANG, and Wanli Ouyang. “Flow-GRPO: Training Flow Matching Models via Online RL.” In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.
- [LLW15a] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. “Deep learning face attributes in the wild.” In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.
- [LLW15b] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. “Deep learning face attributes in the wild.” In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.
- [LLX22] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. “Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation.” In *International Conference on Machine Learning*, pp. 12888–12900. PMLR, 2022.

- [LLZ25] Zongjian Li, Zheyuan Liu, Qihui Zhang, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Yang Ye, Wangbo Yu, Yuwei Niu, and Li Yuan. “Uniworld-V2: Reinforce Image Editing with Diffusion Negative-aware Finetuning and MLLM Implicit Feedback.” *arXiv preprint arXiv:2510.16888*, 2025.
- [LMB14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. “Microsoft coco: Common objects in context.” In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.
- [LMN25] Michal Lukasik, Zhao Meng, Harikrishna Narasimhan, Yin-Wen Chang, Aditya Krishna Menon, Felix Yu, and Sanjiv Kumar. “Better autoregressive regression with LLMs via regression-aware fine-tuning.” In *The Thirteenth International Conference on Learning Representations*, 2025.
- [LNM24] Michal Lukasik, Harikrishna Narasimhan, Aditya Krishna Menon, Felix Yu, and Sanjiv Kumar. “Regression Aware Inference with LLMs.” In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 13667–13678, Miami, Florida, USA, November 2024. Association for Computational Linguistics.
- [LSY24] Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Pengfei Liu, et al. “Generative judge for evaluating alignment.” In *International Conference on Learning Representations*, volume 2024, pp. 27547–27574, 2024.
- [LWW26] Xin Luo, Jiahao Wang, Chenyuan Wu, Shitao Xiao, Xiyan Jiang, Defu Lian, Jiajun Zhang, Dong Liu, and Zheng Liu. “EditScore: Unlocking Online RL for Image Editing via High-Fidelity Reward Modeling.” In *The Fourteenth International Conference on Learning Representations*, 2026.

- [LWY25] Haoye Lu, Qifan Wu, and Yaoliang Yu. “Stochastic Forward-Backward Deconvolution: Training Diffusion Models with Finite Noisy Datasets.” *arXiv preprint arXiv:2502.05446*, 2025.
- [LZM24] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. “Instaflow: One step is enough for high-quality diffusion-based text-to-image generation.” In *International Conference on Learning Representations*, 2024.
- [LZR24a] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. “Grounding dino: Marrying dino with grounded pre-training for open-set object detection.” In *European Conference on Computer Vision*, pp. 38–55. Springer, 2024.
- [LZR24b] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. “Grounding dino: Marrying dino with grounded pre-training for open-set object detection.” In *European conference on computer vision*, pp. 38–55. Springer, 2024.
- [LZZ22] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. “Grounded language-image pre-training.” In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10965–10975, 2022.
- [MDH20] Sachit Menon, Alex Damian, McCourt Hu, Nikhil Ravi, and Cynthia Rudin. “PULSE: Self-Supervised Photo Upsampling via Latent Space Exploration of Generative Models.” In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [MGS22] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa

- Dehghani, Zhuoran Shen, et al. “Simple open-vocabulary object detection.” In *European conference on computer vision*, pp. 728–755. Springer, 2022.
- [Mir75] Leon Mirsky. “A trace inequality of John von Neumann.” *Monatshefte für mathematik*, **79**(4):303–306, 1975.
- [MJY24] Yiwei Ma, Jiayi Ji, Ke Ye, Weihuang Lin, Zhibin Wang, Yonghan Zheng, Qiang Zhou, Xiaoshuai Sun, and Rongrong Ji. “I2ebench: A comprehensive benchmark for instruction-based image editing.” *Advances in Neural Information Processing Systems*, **37**:41494–41516, 2024.
- [MPR87] ROBERTJ McEliece, Edwardc Posner, EUGENER Rodemich, and SANTOSHS Venkatesh. “The capacity of the Hopfield associative memory.” *IEEE transactions on Information Theory*, **33**(4):461–482, 1987.
- [MSC13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. “Distributed representations of words and phrases and their compositionality.” *Advances in neural information processing systems*, **26**, 2013.
- [MSK24] Morteza Mardani, Jiaming Song, Jan Kautz, and Arash Vahdat. “A Variational Perspective on Solving Inverse Problems with Diffusion Models.” In *The Twelfth International Conference on Learning Representations*, 2024.
- [MWS25] Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. “HPSv3: Towards Wide-Spectrum Human Preference Score.”, 2025.
- [NDB17] Khanh Nguyen, Hal Daumé III, and Jordan Boyd-Graber. “Reinforcement Learning for Bandit Neural Machine Translation with Simulated Human Feedback.” In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1464–1474, 2017.

- [NDR21] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. “Glide: Towards photorealistic image generation and editing with text-guided diffusion models.” *arXiv preprint arXiv:2112.10741*, 2021.
- [NLS23] Mang Ning, Mingxiao Li, Jianlin Su, Albert Ali Salah, and Itir Onal Ertugrul. “Elucidating the exposure bias in diffusion models.” *arXiv preprint arXiv:2308.15321*, 2023.
- [Nol08] Guust Nolet. “A breviary of seismic tomography.” *A breviary of seismic tomography*, 2008.
- [OJM20] Gregory Ongie, Ajil Jalal, Christopher A. Metzler, Richard G. Baraniuk, Alexandros G. Dimakis, and Rebecca Willett. “Deep Learning Techniques for Inverse Problems in Imaging.” *IEEE Journal on Selected Areas in Information Theory*, **1**(1):39–56, 2020.
- [Ope23] OpenAI. “GPT-3.5-Turbo API Documentation.” <https://platform.openai.com/docs/models/gpt-3.5-turbo>, 2023. Version: gpt-3.5-turbo-0613.
- [Ope25] OpenAI. “Introducing 4o Image Generation.” <https://openai.com/index/introducing-4o-image-generation/>, 2025. Accessed: 2025-06-22.
- [OWJ22] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. “Training language models to follow instructions with human feedback.” *Advances in neural information processing systems*, **35**:27730–27744, 2022.
- [Pea95] Karl Pearson. “VII. Note on regression and inheritance in the case of two parents.” *proceedings of the royal society of London*, **58**(347-352):240–242, 1895.

- [PEL24] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. “SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis.” In *The Twelfth International Conference on Learning Representations*, 2024.
- [PJB22] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. “Dreamfusion: Text-to-3d using 2d diffusion.” *arXiv preprint arXiv:2209.14988*, 2022.
- [PKK23] Geon Yeong Park, Jeongsol Kim, Beomsu Kim, Sang Wan Lee, and Jong Chul Ye. “Energy-Based Cross Attention for Bayesian Context Update in Text-to-Image Diffusion Models.” *arXiv preprint arXiv:2306.09869*, 2023.
- [PMC23] Ashwini Pokle, Matthew J Muckley, Ricky TQ Chen, and Brian Karrer. “Training-free linear image inversion via flows.” *arXiv preprint arXiv:2310.04432*, 2023.
- [PX23] William Peebles and Saining Xie. “Scalable diffusion models with transformers.” In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.
- [QBS25] Yusu Qian, Eli Bocek-Rivele, Liangchen Song, Jialing Tong, Yinfei Yang, Jiasen Lu, Wenze Hu, and Zhe Gan. “Pico-Banana-400K: A Large-Scale Dataset for Text-Guided Image Editing.”, 2025.
- [QGF24] Muhammad Fetrat Qharabagh, Mohammadreza Ghofrani, and Kimon Fountoulakis. “Lvllm-count: Enhancing the counting ability of large vision-language models.” *arXiv preprint arXiv:2412.00686*, 2024.
- [QLF25] Yusu Qian, Jiasen Lu, Tsu-Jui Fu, Xinze Wang, Chen Chen, Yinfei Yang, Wenze Hu, and Zhe Gan. “GIE-Bench: Towards Grounded Evaluation for Text-Guided Image Editing.” *arXiv preprint arXiv:2505.11493*, 2025.

- [RBL22a] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. “High-resolution image synthesis with latent diffusion models.” In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.
- [RBL22b] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. “High-Resolution Image Synthesis With Latent Diffusion Models.” In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.
- [RBL22c] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. “High-resolution image synthesis with latent diffusion models.” In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.
- [RDN22a] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. “Hierarchical text-conditional image generation with clip latents.” *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.
- [RDN22b] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. “Hierarchical text-conditional image generation with clip latents.” *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.
- [RFB15a] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation.” In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.
- [RFB15b] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation.” In *Medical image computing and*

- computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.
- [RFS14] Nicholas Rawlinson, Andreas Fichtner, Malcolm Sambridge, and Mallory K Young. “Seismic tomography and the assessment of uncertainty.” *Advances in geophysics*, **55**:1–76, 2014.
- [RHG23] Royi Rassin, Eran Hirsch, Daniel Glickman, Shauli Ravfogel, Yoav Goldberg, and Gal Chechik. “Linguistic Binding in Diffusion Models: Enhancing Attribute Correspondence through Attention Map Alignment.”, 2023.
- [RJL24] Tianhe Ren, Qing Jiang, Shilong Liu, Zhaoyang Zeng, Wenlong Liu, Han Gao, Hongjie Huang, Zhengyu Ma, Xiaoke Jiang, Yihao Chen, Yuda Xiong, Hao Zhang, Feng Li, Peijun Tang, Kent Yu, and Lei Zhang. “Grounding DINO 1.5: Advance the ”Edge” of Open-Set Object Detection.”, 2024.
- [RKA21] Albert W Reed, Hyojin Kim, Rushil Anirudh, K Aditya Mohan, Kyle Champley, Jingu Kang, and Suren Jayasuriya. “Dynamic ct reconstruction from limited views with implicit neural representations and parametric motion fields.” In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2258–2268, 2021.
- [RKH21a] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. “Learning transferable visual models from natural language supervision.” In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.
- [RKH21b] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark,

- et al. “Learning transferable visual models from natural language supervision.” In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.
- [RM15] Danilo Rezende and Shakir Mohamed. “Variational inference with normalizing flows.” In *International conference on machine learning*, pp. 1530–1538. PMLR, 2015.
- [Rod88] François Roddier. “Interferometric imaging in optical astronomy.” *Physics Reports*, **170**(2):97–166, 1988.
- [RRD24] Litu Rout, Negin Raoof, Giannis Daras, Constantine Caramanis, Alex Dimakis, and Sanjay Shakkottai. “Solving linear inverse problems provably via posterior sampling with latent diffusion models.” *Advances in Neural Information Processing Systems*, **36**, 2024.
- [RRG22] Royi Rassin, Shauli Ravfogel, and Yoav Goldberg. “DALLE-2 is seeing double: Flaws in word-to-concept mapping in Text2Image models.” *arXiv preprint arXiv:2210.10606*, 2022.
- [RRR20] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. “Zero: Memory optimizations toward training trillion parameter models.” In *SC20: international conference for high performance computing, networking, storage and analysis*, pp. 1–16. IEEE, 2020.
- [RSL20] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, et al. “Hopfield networks is all you need.” *arXiv preprint arXiv:2008.02217*, 2020.
- [RYF19] Saiprasad Ravishankar, Jong Chul Ye, and Jeffrey A Fessler. “Image reconstruction: From sparsity to data-adaptive methods and machine learning.” *Proceedings of the IEEE*, **108**(1):86–109, 2019.

- [SB99] Richard S Sutton, Andrew G Barto, et al. “Reinforcement learning.” *Journal of Cognitive Neuroscience*, **11**(1):126–134, 1999.
- [Sch19] Florian Schmidt. “Generalization in generation: A closer look at exposure bias.” *arXiv preprint arXiv:1910.00292*, 2019.
- [SCS22] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. “Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding.”, 2022.
- [SDM21] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. “Maximum likelihood training of score-based diffusion models.” *Advances in neural information processing systems*, **34**:1415–1428, 2021.
- [SE19] Yang Song and Stefano Ermon. “Generative modeling by estimating gradients of the data distribution.” *Advances in neural information processing systems*, **32**, 2019.
- [SHG22] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. “Flava: A foundational language and vision alignment model.” In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15638–15650, 2022.
- [SHW15] John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. “Gradient estimation using stochastic computation graphs.” *Advances in neural information processing systems*, **28**, 2015.
- [Ski89] John Skilling. “The eigenvalues of mega-dimensional matrices.” *Maximum Entropy and Bayesian Methods: Cambridge, England, 1988*, pp. 455–466, 1989.

- [SME21] Jiaming Song, Chenlin Meng, and Stefano Ermon. “Denoising Diffusion Implicit Models.” In *International Conference on Learning Representations*, 2021.
- [Spe61] Charles Spearman. “The proof and measurement of association between two things.” 1961.
- [SPS24] Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. “Emu edit: Precise image editing via recognition and generation tasks.” In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8871–8879, 2024.
- [SSK20] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. “Score-based generative modeling through stochastic differential equations.” *arXiv preprint arXiv:2011.13456*, 2020.
- [SSK21] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. “Score-Based Generative Modeling through Stochastic Differential Equations.” In *International Conference on Learning Representations*, 2021.
- [Sue17] Paul Suetens. *Fundamentals of medical imaging*. Cambridge university press, 2017.
- [SVM23] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. “Pseudoinverse-guided diffusion models for inverse problems.” In *International Conference on Learning Representations*, 2023.
- [SVS25] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. “Dinov3.” *arXiv preprint arXiv:2508.10104*, 2025.

- [SWD17] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. “Proximal policy optimization algorithms.” *arXiv preprint arXiv:1707.06347*, 2017.
- [SWM15] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. “Deep unsupervised learning using nonequilibrium thermodynamics.” In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.
- [SWZ24] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. “Deepseekmath: Pushing the limits of mathematical reasoning in open language models.” *arXiv preprint arXiv:2402.03300*, 2024.
- [SZY24] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. “HybridFlow: A Flexible and Efficient RLHF Framework.” *arXiv preprint arXiv: 2409.19256*, 2024.
- [TDT23] Radhika Tibrewala, Tarun Dutt, Angela Tong, Luke Ginocchio, Mahesh B Keerthivasan, Steven H Baete, Sumit Chopra, Yvonne W Lui, Daniel K Sodickson, Hersh Chandarana, et al. “FastMRI prostate: A publicly available, biparametric MRI dataset to advance machine learning for prostate cancer imaging.” *arXiv preprint arXiv:2304.09254*, 2023.
- [TGZ23] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. “Stanford Alpaca: An Instruction-following LLaMA model.” https://github.com/tatsu-lab/stanford_alpaca, 2023.
- [TMS23] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhos-

- ale, et al. “Llama 2: Open foundation and fine-tuned chat models.” *arXiv preprint arXiv:2307.09288*, 2023.
- [TWM25] Yunhao Tang, Sid Wang, Lovish Madaan, and Remi Munos. “Beyond Verifiable Rewards: Scaling Reinforcement Learning in Language Models to Unverifiable Data.” In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.
- [VB13] Marinus T Vlaardingerbroek and Jacques A Boer. *Magnetic resonance imaging: theory and practice*. Springer Science & Business Media, 2013.
- [Vin11] Pascal Vincent. “A connection between score matching and denoising autoencoders.” *Neural computation*, **23**(7):1661–1674, 2011.
- [VNT25] An Vo, Khai-Nguyen Nguyen, Mohammad Reza Taesiri, Vy Tuong Dang, Anh Totti Nguyen, and Daeyoung Kim. “Vision Language Models are Biased.” *arXiv preprint arXiv:2505.23941*, 2025.
- [VSB13] David C Van Essen, Stephen M Smith, Deanna M Barch, Timothy EJ Behrens, Essa Yacoub, Kamil Ugurbil, Wu-Minn HCP Consortium, et al. “The WU-Minn human connectome project: an overview.” *Neuroimage*, **80**:62–79, 2013.
- [VSP17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. “Attention is all you need.” *Advances in neural information processing systems*, **30**, 2017.
- [WBS04] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. “Image quality assessment: from error visibility to structural similarity.” *IEEE Transactions on Image Processing*, **13**(4):600–612, 2004.
- [Wil92] Ronald J Williams. “Simple statistical gradient-following algorithms for connectionist reinforcement learning.” *Machine learning*, **8**(3):229–256, 1992.

- [WJK26] Keming Wu, Sicong Jiang, Max Ku, Ping Nie, Minghao Liu, and Wenhua Chen. “EditReward: A Human-Aligned Reward Model for Instruction-Guided Image Editing.” In *The Fourteenth International Conference on Learning Representations*, 2026.
- [WLD21] Jay Whang, Erik M. Lindgren, and Alexandros G. Dimakis. “Composing Normalizing Flows for Inverse Problems.” *Proceedings of the 38th International Conference on Machine Learning*, 2021.
- [WLW24] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. “Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation.” *Advances in Neural Information Processing Systems*, **36**, 2024.
- [WLZ25] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. “Qwen-Image Technical Report.”, 2025.
- [WV24] Binxu Wang and John J Vastola. “The unreasonable effectiveness of gaussian score approximation for diffusion models and its applications.” *arXiv preprint arXiv:2412.09726*, 2024.
- [WWY25] Chenxi Whitehouse, Tianlu Wang, Ping Yu, Xian Li, Jason Weston, Ilia Kulikov, and Swarnadeep Saha. “J1: Incentivizing thinking in llm-as-a-judge via reinforcement learning.” *arXiv preprint arXiv:2505.10320*, 2025.
- [WZY25] Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang,

- Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, Ze Liu, Ziyi Xia, Chaofan Li, Haoge Deng, Jiahao Wang, Kun Luo, Bo Zhang, Defu Lian, Xinlong Wang, Zhongyuan Wang, Tiejun Huang, and Zheng Liu. “OmniGen2: Exploration to Advanced Multimodal Generation.” *arXiv preprint arXiv:2506.18871*, 2025.
- [WZZ24] Peng Wang, Huijie Zhang, Zekai Zhang, Siyi Chen, Yi Ma, and Qing Qu. “Diffusion models learn low-dimensional distributions via subspace clustering.” *arXiv preprint arXiv:2409.02426*, 2024.
- [XLW23] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. “ImageReward: learning and evaluating human preferences for text-to-image generation.” In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 15903–15935, 2023.
- [XLZ16a] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. “A theory of generative convnet.” In *International Conference on Machine Learning*, pp. 2635–2644. PMLR, 2016.
- [XLZ16b] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. “A Theory of Generative ConvNet.” In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 2635–2644, New York, New York, USA, 20–22 Jun 2016. PMLR.
- [XNV25] Yilun Xu, Weili Nie, and Arash Vahdat. “One-step Diffusion Models with f -Divergence Distribution Matching.” *arXiv preprint arXiv:2502.15681*, 2025.
- [XSF25] Yixuan Even Xu, Yash Savani, Fei Fang, and J Zico Kolter. “Not all rollouts are useful: Down-sampling rollouts in llm reinforcement learning.” *arXiv preprint arXiv:2504.13818*, 2025.

- [XSZ23] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. “Wizardlm: Empowering large language models to follow complex instructions.” *arXiv preprint arXiv:2304.12244*, 2023.
- [XWG25] Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. “DanceGRPO: Unleashing GRPO on Visual Generation.” *arXiv preprint arXiv:2505.07818*, 2025.
- [XWZ25] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. “Omnigen: Unified image generation.” In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 13294–13304, 2025.
- [XXK24] Sirui Xie, Zhisheng Xiao, Diederik P Kingma, Tingbo Hou, Ying Nian Wu, Kevin Patrick Murphy, Tim Salimans, Ben Poole, and Ruiqi Gao. “EM Distillation for One-step Diffusion Models.” In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [YBK22] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. “When and Why Vision-Language Models Behave like Bags-Of-Words, and What to Do About It?” In *The Eleventh International Conference on Learning Representations*, 2022.
- [YCY25] Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. “Anyedit: Mastering unified high-quality image editing for any idea.” In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 26125–26135, 2025.
- [YGP24] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. “Improved distribution matching distillation

- for fast image synthesis.” *Advances in neural information processing systems*, **37**:47455–47487, 2024.
- [YGZ24] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. “One-step diffusion with distribution matching distillation.” In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6613–6623, 2024.
- [YHL25] Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. “Imgedit: A unified image editing dataset and benchmark.” *arXiv preprint arXiv:2505.20275*, 2025.
- [YHZ25] Siwei Yang, Mude Hui, Bingchen Zhao, Yuyin Zhou, Nataniel Ruiz, and Cihang Xie. “Complex-Edit: CoT-Like Instruction Generation for Complexity-Controllable Image Editing Benchmark.” *arXiv preprint arXiv:2504.13143*, 2025.
- [YLP24] Hanshu Yan, Xingchao Liu, Jiachun Pan, Jun Hao Liew, Qiang Liu, and Jiashi Feng. “PeRFlow: Piecewise Rectified Flow as Universal Plug-and-Play Accelerator.” *arXiv preprint arXiv:2405.07510*, 2024.
- [YLY25] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. “Qwen3 technical report.” *arXiv preprint arXiv:2505.09388*, 2025.
- [YXK22] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. “Scaling Autoregressive Models for Content-Rich Text-to-Image Generation.”, 2022.
- [YXM21] Peiyu Yu, Sirui Xie, Xiaojian Ma, Yixin Zhu, Ying Nian Wu, and Song-Chun Zhu.

- “Unsupervised foreground extraction via deep region competition.” *Advances in Neural Information Processing Systems*, **34**:14264–14279, 2021.
- [YXM22] Peiyu Yu, Sirui Xie, Xiaojian Ma, Baoxiong Jia, Bo Pang, Ruiqi Gao, Yixin Zhu, Song-Chun Zhu, and Ying Nian Wu. “Latent diffusion energy-based model for interpretable text modeling.” *arXiv preprint arXiv:2206.05895*, 2022.
- [YYZ24] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. “Qwen2.5 Technical Report.” *arXiv preprint arXiv:2412.15115*, 2024.
- [YZH24] Peiyu Yu, Dinghuai Zhang, Hengzhi He, Xiaojian Ma, Ruiyao Miao, Yifan Lu, Yasi Zhang, Deqian Kong, Ruiqi Gao, Jianwen Xie, et al. “Latent energy-based odyssey: Black-box optimization via expanded exploration in the energy-based latent space.” *arXiv preprint arXiv:2405.16730*, 2024.
- [YZX24] Peiyu Yu, Yaxuan Zhu, Sirui Xie, Xiaojian Shawn Ma, Ruiqi Gao, Song-Chun Zhu, and Ying Nian Wu. “Learning Energy-Based Prior Model with Diffusion-Amortized MCMC.” *Advances in Neural Information Processing Systems*, **36**, 2024.
- [YZZ25] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. “Dapo: An open-source llm reinforcement learning system at scale.” *arXiv preprint arXiv:2503.14476*, 2025.

- [ZCS23a] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. “Judging llm-as-a-judge with mt-bench and chatbot arena.” *Advances in neural information processing systems*, **36**:46595–46623, 2023.
- [ZCS23b] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. “Judging llm-as-a-judge with mt-bench and chatbot arena.” *Advances in neural information processing systems*, **36**:46595–46623, 2023.
- [ZCY26] Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. “DiffusionNFT: Online Diffusion Reinforcement with Forward Process.” In *The Fourteenth International Conference on Learning Representations*, 2026.
- [ZCZ25] Hongkai Zheng, Wenda Chu, Bingliang Zhang, Zihui Wu, Austin Wang, Berthy T Feng, Caifeng Zou, Yu Sun, Nikola Kovachki, Zachary E Ross, et al. “Inversebench: Benchmarking plug-and-play diffusion priors for inverse problems in physical sciences.” *arXiv preprint arXiv:2503.11043*, 2025.
- [ZDZ24] Yaxuan Zhu, Zehao Dou, Haoxin Zheng, Yasi Zhang, Ying Nian Wu, and Ruiqi Gao. “Think Twice Before You Act: Improving Inverse Problem Solving With MCMC.” *arXiv preprint arXiv:2409.08551*, 2024.
- [ZGV23] Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. “Pytorch fsdp: experiences on scaling fully sharded data parallel.” *arXiv preprint arXiv:2304.11277*, 2023.
- [ZHL25] Zheyuan Zhang, Fengyuan Hu, Jayjun Lee, Freda Shi, Parisa Kordjamshidi, Joyce Chai, and Ziqiao Ma. “Do Vision-Language Models Represent Space and

- How? Evaluating Spatial Frame of Reference under Ambiguities.” In *The Thirteenth International Conference on Learning Representations*, 2025.
- [ZKS18] Jure Zbontar, Florian Knoll, Anuroop Sriram, Tullie Murrell, Zhengnan Huang, Matthew J Muckley, Aaron Defazio, Ruben Stern, Patricia Johnson, Mary Bruno, et al. “fastMRI: An open dataset and benchmarks for accelerated MRI.” *arXiv preprint arXiv:1811.08839*, 2018.
- [ZL25] Yasi Zhang and Oscar Leong. “Learning Difference-of-Convex Regularizers for Inverse Problems: A Flexible Framework with Theoretical Guarantees.” *arXiv preprint arXiv:2502.00240*, 2025.
- [ZMC23] Kai Zhang, Lingbo Mo, Wenhui Chen, Huan Sun, and Yu Su. “Magicbrush: A manually annotated dataset for instruction-guided image editing.” *Advances in Neural Information Processing Systems*, **36**:31428–31449, 2023.
- [ZMC24] Haozhe Zhao, Xiaojian Shawn Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. “Ultraedit: Instruction-based fine-grained image editing at scale.” *Advances in Neural Information Processing Systems*, **37**:3058–3093, 2024.
- [ZWW25] Lianghai Zhu, Xinggang Wang, and Xinlong Wang. “JudgeLM: Fine-tuned Large Language Models are Scalable Judges.” In *The Thirteenth International Conference on Learning Representations*, 2025.
- [ZWY24] Huangjie Zheng, Zhendong Wang, Jianbo Yuan, Guanghan Ning, Pengcheng He, Quanzeng You, Hongxia Yang, and Mingyuan Zhou. “Learning Stackable and Skippable LEGO Bricks for Efficient, Reconfigurable, and Variable-Resolution Diffusion Modeling.” In *The Twelfth International Conference on Learning Representations*, 2024.

- [ZYW24] Yasi Zhang, Peiyu Yu, and Ying Nian Wu. “Object-Conditioned Energy-Based Model for Attention Map Alignment in Text-to-Image Diffusion Models.” In *Synthetic Data for Computer Vision Workshop @ CVPR 2024*, 2024.
- [ZYW25] Yasi Zhang, Peiyu Yu, and Ying Nian Wu. “Object-Conditioned Energy-Based Attention Map Alignment in Text-to-Image Diffusion Models.” In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pp. 55–71, Cham, 2025. Springer Nature Switzerland.
- [ZYZ24] Yasi Zhang, Peiyu Yu, Yaxuan Zhu, Yingshan Chang, Feng Gao, Ying Nian Wu, and Oscar Leong. “Flow Priors for Linear Inverse Problems via Iterative Corrupted Trajectory Matching.” In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [ZZC17] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. “Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising.” *IEEE transactions on image processing*, **26**(7):3142–3155, 2017.
- [ZZC24] Kaijie Zhu, Qinlin Zhao, Hao Chen, Jindong Wang, and Xing Xie. “Promptbench: A unified library for evaluation of large language models.” *Journal of Machine Learning Research*, **25**(254):1–22, 2024.
- [ZZL21] Yan Zeng, Xinsong Zhang, and Hang Li. “Multi-grained vision language pre-training: Aligning texts with visual concepts.” *arXiv preprint arXiv:2111.08276*, 2021.
- [ZZW24] Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. “Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation.” In *Forty-first International Conference on Machine Learning*, 2024.

- [ZZZ18] Kai Zhang, Wangmeng Zuo, and Lei Zhang. “FFDNet: Toward a fast and flexible solution for CNN-based image denoising.” *IEEE Transactions on Image Processing*, **27**(9):4608–4622, 2018.