

UC Riverside

UC Riverside Electronic Theses and Dissertations

Title

Shaping the Future of Generative AI: Insights into Interpretability, Alignment, and Compositionality of Diffusion Models

Permalink

<https://escholarship.org/uc/item/1z92k5n1>

ISBN

9798263308599

Author

Kong, Xianghao

Publication Date

2025-08-25

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Shaping the Future of Generative AI: Insights into Interpretability, Alignment, and
Compositionality of Diffusion Models

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Xianghao Kong

September 2025

Dissertation Committee:

Dr. Greg Ver Steeg, Chairperson
Dr. Christian Shelton
Dr. Amit Roy Chowdhury
Dr. Yue Dong

The Dissertation of Xianghao Kong is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I am deeply grateful to my advisor, Prof. Greg Ver Steeg, for his continuous guidance, encouragement, and patience throughout my Ph.D. journey. His insightful advice, high standards, and generous support have shaped not only this dissertation but also my approach to research and problem solving. I also thank my dissertation committee members, Prof. Christian Shelton, Prof. Amit Roy Chowdhury, and Prof. Yue Dong, for their valuable feedback, constructive discussions, and for generously dedicating their time to help improve this work.

I am fortunate to have had the opportunity to collaborate with many wonderful researchers during my Ph.D. I thank my mentors and colleagues at Adobe Firefly and Sony AI, especially Vikash Sehwal, Lingjuan Lyu, Sachin Kelkar, Hareesh Ravi, and Ajinkya Kale, whose guidance and collaboration enriched both my research and my professional development. I am also grateful to my labmates at UC Riverside, past and present, for their friendship, stimulating conversations, and support. Special thanks go to Rob Brekelmans and Ollie Liu for their help in sharing ideas, and for the countless hours we spent together pushing forward on research problems.

This journey would not have been possible without the unwavering support of my friends and family. I thank my parents for their unconditional love, encouragement, and sacrifices that made my education possible. To my friends outside academia, thank you for providing balance, laughter, and perspective during challenging times. Finally, I would like to acknowledge Prof. Nanpeng Yu, for granting me the freedom to pursue my research interests and for his support during the early stages of my doctoral studies.

To my family, for their love and patience that made this journey possible.

ABSTRACT OF THE DISSERTATION

Shaping the Future of Generative AI: Insights into Interpretability, Alignment, and
Compositionality of Diffusion Models

by

Xianghao Kong

Doctor of Philosophy, Graduate Program in Computer Science
University of California, Riverside, September 2025
Dr. Greg Ver Steeg, Chairperson

Generative models, including autoregressive models, GANs, VAEs, and diffusion models, have demonstrated remarkable capabilities in synthesizing high-quality content across diverse modalities. Among these, diffusion models have emerged as a particularly powerful framework due to their stability and generative fidelity.

In this work, we present an information-theoretic perspective on diffusion modeling by establishing a precise connection between data distributions and optimal denoising via the pointwise Information–Minimum Mean Square Error (I-MMSE) relation. This connection refines traditional variational training objectives, enabling more accurate density estimation for both continuous and discrete data. Furthermore, by integrating contrastive learning, our method enhances the model’s ability to approximate complex data distributions. We term this framework Information-Theoretic Diffusion (ITD), which not only improves density modeling but also provides a principled interpretation of cross-modal alignment—modeling mutual information between modalities (e.g., text and image) rather than relying on attention

maps or CLIP scores. In addition, ITD enables improved mutual information estimation directly within the training process.

To fully leverage the information in the training data while minimizing computational cost, we introduce a plug-and-play lightweight path-mixer module, enabling highly efficient diffusion model training for under \$2000. We also explore a conditional autoregressive diffusion framework that progressively injects information for flexibly compositionable generation. Overall, this work bridges information theory and diffusion modeling, advancing both the theoretical foundations and practical performance of generative AI.

Contents

1	Introduction	1
2	Enhance Density Estimation of Diffusion Based on Information Theory	6
2.1	Introduction	6
2.2	Information-Theoretic Diffusion	8
2.2.1	Fundamental Pointwise Denoising Relation	8
2.2.2	Diffusion as Thermodynamic Integration	10
2.3	What Your Diffusion Model is Hiding: An Optimal Classifiers	18
2.3.1	Contrastive Diffusion Loss (CDL)	19
2.4	Implementation	20
2.5	Experiments	25
2.6	Related Work	29
2.7	Conclusion	31
3	Bidirectional Synergy Between Mutual Information and Diffusion Models	33
3.1	Introduction	33
3.2	Interpretable Diffusion via Information Decomposition	36
3.2.1	Information-theoretic Perspective on Diffusion Models	36

3.2.2	Mutual Information and Pointwise Estimators	38
3.2.3	Pixel-wise Information Decomposition	40
3.2.4	Numerical Information Estimates	41
3.3	Mutual Information Estimation via Neural Diffusion	42
3.3.1	Model Training	42
3.3.2	Adaptive Importance Sampling	44
3.4	Results	45
3.4.1	Relation Testing with Pointwise Information	45
3.4.2	Pixel-wise Information and Word Localization	48
3.4.3	Selective Image Editing via Prompt Intervention	52
3.4.4	40 MI Estimation Benchmark	54
3.4.5	High MI Estimation	56
3.4.6	MI Consistency Test	56
3.5	Related Work	58
3.6	Conclusion	60
4	Information-Preserved and Efficient Diffusion via Architectural Innovation	62
4.1	Stretching Each Dollar: Diffusion Training from Scratch on a Micro-Budget .	62
4.1.1	Introduction	62
4.1.2	Background: Diffusion-based generative models and diffusion trans- formers	66
4.1.3	Approaches for minimizing the computational cost of training diffusion transformers	69

4.1.4	Alleviating performance bottleneck in patch masking using deferred masking	71
4.1.5	Experimental Setup	73
4.1.6	Effectiveness of Deferred Masking	74
4.1.7	Comparing the Effectiveness of Deferred Masking with Baselines . . .	77
4.1.8	Micro-budget Training at Scale	78
4.1.9	Understanding role of synthetic data in micro-budget training	81
4.1.10	Conclusion	83
4.2	LayerGen: Redefining image generation as scene composition	84
4.2.1	Introduction	84
4.2.2	Related Work	86
4.2.3	Latent Causality Diffusion Models	89
4.2.4	Experiments	91
4.2.5	Datasets	91
4.2.6	Conclusion	99
5	Conclusions	100
	Bibliography	103
A	Appendix for Chapter 2	126
A.1	Derivations	126
A.1.1	Derivation of the Fundamental Pointwise Denoising Relation	126
A.1.2	High SNR KL divergence limit	129

A.1.3	Simplifying Density with an Integral Identity	131
A.1.4	Non-Gaussian Density Representation	131
A.1.5	Gaussian properties	133
A.1.6	MMSE Upper Bounds via Bregman Divergence Interpretation	134
A.1.7	Comparison with Variational Bounds	136
A.1.8	Mixture Distribution Density	144
A.2	Implementation Details	146
A.2.1	Discrete probability estimator bound	146
A.2.2	Relationship Between $\log(\text{SNR})$ and Timesteps	148
A.2.3	Details on Model Training	149
A.2.4	Soft Discretization Function	151
A.2.5	Variance Estimates	153

B Appendix for Chapter 3 157

B.1	Derivations of the negative log-likelihood	157
B.2	Derivations of pointwise information via the orthogonality principle	159
B.3	Additional Results	161
B.3.1	Relationship between image-level MI and CMI	161
B.3.2	Image-level MMSE curves and pixel-level MMSE visualizaiton	162
B.3.3	Examples of word location for object nouns	163
B.3.4	Examples of word location for other seven entities	163
B.3.5	Intervention experiments additional results	163
B.4	Experimental Settings	165

B.4.1	Relation Testing with Pointwise Information	165
B.4.2	Localizing Word Information in Images	167
B.5	COCO-IT Dataset Preparation	169
B.6	Implementation Details of MIND Training	171
B.6.1	Ablation Study on Adaptive Sampling	171
B.6.2	Visual Analysis of the MIND Estimator	172
C	Appendix for Chapter 5	188
C.1	Related Work	188
C.2	Additional Details on Experimental Setup	190
C.3	Background on Layer-wise Scaling and Mixture-of-experts	197
C.4	Additional Results on Effectiveness of Deferred Masking	199
C.4.1	Out-of-the-box performance: Making high masking ratios feasible with deferred masking	200
C.4.2	Ablation study of our training pipeline	200
C.4.3	Validating improvements in diffusion transformer architecture	203
C.4.4	Deferred masking as pretraining + unmasked finetuning.	205
C.5	Additional Results on Micro-budget Training of Large-scale Models	206
C.5.1	Challenge with canonical evaluation metrics in determining effect of synthetic data.	207

List of Figures

1.1	The overall research blueprint.	3
2.1	The integral of the gap between MMSE curves for data from the target distribution versus data from a Gaussian distribution is used in Eq. equation 2.10 to get an exact expression for the entropy, or expected Negative Log Likelihood (NLL), of the data.	14
2.2	(Left) MSE curves for different denoisers, used in estimating Negative Log Likelihood (NLL). (Right) Continuous NLL estimates for diffusion models using variational bounds and Information-Theoretic (IT) bounds (ours). Variance estimates are shown in App. A.2.5. *Uses a discrete estimator made continuous by assuming uniform density in each bin.	27
2.3	(Left) MSE curves for different denoisers, used in estimating Negative Log Likelihood (NLL), shown with or without rounding using a soft discretization function (App. A.2.4). (Right) Discrete NLL estimates for diffusion models using variational bounds and information-theoretic bounds (ours). Variance estimates are shown in App. A.2.5. *Using the denoiser with soft discretization. **Indicates a continuous estimator made discrete via uniform dequantization.	28

2.5	Parallel DDPM sampler generated Dino data. Comparing to Dino sampled from DDPM loss, CDL-loss sampled Dino has better sample quality and density estimate around hard areas.	29
3.1	We start (left) with a real image from the COCO dataset. We do a “prompt intervention” (§3.4.3) to generate a new image. Next we show conditional mutual information, illustrated using our pixel-wise decomposition, and attention maps for the modified word. Top row shows an image where prompt intervention has an effect, while in the bottom row it has little effect. Conditional mutual information reflects the effect of intervention while attention does not.	34
3.2	The mutual information is exactly half the area between MMSE curves for conditional and unconditional denoising. We use denoising diffusion models to approximate the MMSE curves, then numerically integrate to get an estimate of the mutual information.	43
3.3	The schematic of the MIND training process. Noise at a given γ level is added to the data $\mathbf{x}$, and a denoiser is used to recover it, with or without conditioning on y at a 50% probability. Finally, the MI loss, as defined in Eq. 3.9, is computed to backpropagate the gradient.	43
3.4	Examples of localizing different types of words in images. The left half presents noun words, while the right half displays abstract words.	49

3.5	COCO images edited by omitting words from the prompt. Conditional mutual information better reflects the actual changes in the image after intervention. Table shows Pearson correlation between metrics (CMI or attention) versus L2 changes in image after intervention, at the image-level and pixel-level as discussed in §3.4.3.	54
3.6	High MI benchmark: original (column (b)) and transformed variants (columns (a) and (c)).	57
3.7	Consistency Tests over MNIST dataset: (a) evaluation of $\frac{I(A;B_r)}{I(A;B)}$; (b) evaluation of $\frac{I(A;[B_{r+k},B_r])}{I(A;B_{r+k})}$ for $k > 0$; (c) evaluation of $\frac{I([A^1,A^2];[B_r^1,B_r^2])}{I(A^1;B_r^1)}$	57
4.1	Qualitative evaluation of our 512×512 image generation. Our model is trained in 2.6 days on a single $8 \times H100$ machine (about \$1,890 GPU cost) without proprietary or billion-image datasets. In (e)–(g) we generate diverse styles for the prompt “ <i>Image of an astronaut riding a horse in — style</i> ” (Origami, Pixel art, Line art). In (h) we compare training cost and fidelity (FID [85]) for zero-shot COCO [145]. More samples: Fig. C.9–C.22.	63

4.2	Compressing patch sequence to reduce computational cost. As the training cost of diffusion transformers is proportional to sequence size, i.e., number of patches, it is desirable to reduce the sequence size without degrading performance. It can be achieved by a) using larger patches, b) naively masking a fraction of patches at random, or c) using MaskDiT [283] that combines naive masking with an additional autoencoding objective. We find all three approaches lead to significant degradation in image generation performance, especially at high masking ratios. To alleviate this issue, we propose a straightforward deferred masking strategy, where we mask patches after they are processed by a patch-mixer. Our approach is analogous to naive masking in all aspects except the use of the patch-mixer. In comparison to MaskDiT, our approach doesn't require optimizing any surrogate objectives and has nearly identical computational costs.	70
4.3	Ablating individual design choices in deferred masking, spanning from training hyperparameters to patch masking strategies.	75
4.4	Comparing performance of patch masking strategies. Using a lightweight patch-mixer before patch masking in our deferred masking approach significantly improves image generation performance over baseline masking strategies.	77

4.5	Human preference based evaluation. We report the win percentage based on human preference when comparing images across different MicroDiT models. Here, alignment refers to alignment between caption and generated image. We first compare the preference among identical models trained with only 22M real images vs. combined set (37M) of 22M real and 15M synthetic images (subfig a). Next, we scale up synthetic images to 490M and train a model only on these images, which achieves better human preference than our previous model trained on 37M real and synthetic images (subfig b).	82
4.6	This is diagram of Latent Auto-Regression Diffusion model.	84
4.7	The illustration of the pipeline used for layered images captioning.	92
4.8	Examples of prepared layered image datasets.	93
4.9	Results of a three-layer image generated by LayerGen. The first row presents three reference images, each paired with three instruction-based prompts at the bottom. The following three rows illustrate the generated images when conditioned on a decreasing number of reference layers.	95
4.10	The demonstration of layer-level image editing by prompts.	96
4.11	The results of object-level continuous image editing via prompts.	97
4.12	Two examples of generated 3-layer image with the captions from Instruction MULAN validation dataset.	98
A.1	$\log(\text{SNR})$ v.s. timestep t/T	150
A.2	Change of denoising MSE curve during fine-tuning process.	151
A.3	Training and testing NLL during fine-tuning process.	151

A.4	Illustrating the soft discretization function for various values of SNR (γ), for regularly spaced discrete data, $x = -1, -0.75, -0.5, \dots, 1$.	152
B.1	Scatter plot for correlation between MI and CMI.	162
B.2	MMSE curves examples for 10 categories.	163
B.3	(a) An example where the correlation between pixel-level changes and CMI or attention are low (0.25 and -0.21 respectively). (b) An example where the pixel-level correlation is high (0.73 and 0.77 respectively).	164
B.4	The diagram of two different diffusion processes.	170
B.16	Integrand analysis on a Spiral-transformed task (GT MI = 9.90). Comparison between (a) the volatile direct MMSE subtraction method and (b) the stable orthogonal principle. In both plots, the adaptive sampler (red) outperforms the baseline (blue) by focusing on the most informative region.	173
B.5	Examples of pixel-level MMSE visualization.	176
B.6	Examples of pixel-level MMSE visualization.	177
B.7	Examples of localizing noun words in images.	178
B.8	Examples of localizing noun words in images.	179
B.9	Examples of localizing noun words in images.	180
B.10	Examples of localizing abstract words in images.	181
B.11	Examples of localizing abstract words in images.	182
B.12	Examples of word swap interventions	183
B.13	Examples of word swap interventions	184
B.14	Examples of word omission interventions	185

B.15	Examples of word omission interventions	186
C.1	Overall architecture of our diffusion transformer. We prepend the backbone transformer model with a lightweight patch-mixer that operates on all patches in the input image before they are masked. Following contemporary works [16, 58], we process the caption embeddings using an attention layer before using them for conditioning. We use sinusoidal embeddings for timesteps. Our model only denoises unmasked patches, thus the diffusion loss (Eq. 4.3) is calculated only on these patches. We modify the backbone transformer using layer-wise scaling on individual layers and use mixture-of-expert layers in alternate transformer blocks.	197
C.2	Mixture-of-experts (MoE). Expert-choice routing based mixture-of-experts [284]. Each patch is passed to a patch router that determines the top-k patches routed to each expert. Make this figure nicer.	199
C.3	Out-of-the-box performance of deferred masking. Without any hyperparameter optimization, we compare the performance of our deferred masking with a naive masking strategy. We find that deferred masking, i.e., using a patch-mixer before naive masking, tremendously improves image generation performance, particularly at high masking ratios.	199

C.4 Deferred masking in pretraining.	We test the advantage of using deferred masking with finetuning over training a model with no masking for a given computational budget. After pretraining with 75% deferred masking, we increase the number of unmasked finetuning steps and compare its performance with another model trained completely without masking. The shaded region represents steps after the isoflops threshold, where deferred masking pretraining and finetuning have higher computation cost. At the isoflops threshold, the finetuned model achieves better performance than training the diffusion model without any masking.	204
C.6 Assessing image quality with GPT-4o.	We supply the following prompt to the GPT-4o model: <i>Given the prompt ‘{prompt}’, which image do you prefer, Image A or Image B, considering factors like image details, quality, realism, and aesthetics? Respond with ‘A’ or ‘B’ or ‘none’ if neither is preferred.</i> For each comparison, we also shuffle the order of images to remove any ordering bias. We evaluate on two prompt databases: DrawBench [207] and PartiPrompts [273]. The y-axis in the bar plots indicates the percentage of comparisons in which an image from a model is preferred; we also break down results by category.	209
C.7 Comparing generations from our three micro-budget models.	We compare synthesized images from our large-scale models trained with different configurations of the training dataset but with identical computational costs.	211

C.8 Comparing generations from our first two micro-budget models.	
Generations from a model trained only on real data (left) and combined real + synthetic data (right) on GenEval benchmark prompts. Both models use identical random seeds for generation.	212
C.9 Comparison with previous works. Figure adapted from [30].	212
C.10 Comparison with previous works. Figure adapted from [30].	213
C.11 Evaluating on PartiPrompts. Synthesized images by our model using randomly selected prompts from PartiPrompts [273]. Rows correspond to following categories: Abstract, Animals, Artifacts, Arts, Food & Beverage, and Illustrations.	214
C.12 Evaluating on PartiPrompts. Synthesized images by our model using randomly selected prompts from PartiPrompts [273]. Rows correspond to following categories: Indoor scenes, Outdoor scenes, People, Produce & Plants, Vehicles, and World knowledge.	215
C.13 Evaluating on DrawBench. Synthesized images by our model using randomly selected prompts from DrawBench [207].	216
C.14 Evaluating on GenEval. Synthesized images by our model using randomly selected prompts from GenEval [73].	217
C.15 Control and diversity in image styles. Evaluating the ability to generate diverse styles. <i>Prompt:</i> A — dressed in a suit posing for a photo in — style. Natural lake landscape in background, detailed light and shadow, high detail.	218

C.16 Control and diversity in image styles. Evaluating the ability to generate diverse styles. <i>Prompt:</i> A luxurious — with a next-generation modern design illustrated in — style.	219
C.17 Control and diversity in image styles. Evaluating the ability to generate diverse styles. <i>Prompt:</i> A moonlit night over a —, mysteriously rendered in — style.	220
C.18 Control and diversity in image styles. Evaluating the ability to generate diverse styles. <i>Prompt:</i> A lush garden full of —, vividly illustrated in — style.	221
C.19 Evaluating on JourneyDB. Synthesized images by our model using randomly selected prompts from the test set of JourneyDB dataset [73].	222
C.20 Evaluating on JourneyDB. Synthesized images by our model using randomly selected prompts from the test set of JourneyDB dataset [73].	223
C.21 Evaluating on JourneyDB. Synthesized images by our model using randomly selected prompts from the test set of JourneyDB dataset [73].	224
C.22 Evaluating on JourneyDB. Synthesized images by our model using randomly selected prompts from the test set of JourneyDB dataset [73].	225
C.23 Samples from our cifar-captions dataset. Selected images with corresponding real and synthetic captions from the cifar-captions dataset. We created the cifar-captions dataset, imitating the widely used CIFAR-10 dataset [130], to enable small scale experimentation on the text-to-image generative models.	226

List of Tables

2.1	$\mathbb{E}[-\log p(\mathbf{x})]$ on Test Data (bpd)	27
2.2	$\mathbb{E}[-\log P(\mathbf{x})]$ on Test Data (bpd)	28
3.1	Accuracy (%) of Stable Diffusion and its OpenCLIP backbone on the ARO Benchmark. $\star$: conducts additional fine-tuning with compositional-aware hard negatives.	46
3.2	Selected results on VG-R.	47
3.3	Unsupervised Object Segmentation mIoU (%) Results on COCO-IT	51
3.4	Pearson Correlation with Image Change	54
3.5	Low MI estimation over 10 seeds using $N = 10$ k test samples against ground truth (GT) [43]. All methods were trained with 100k samples. Color indicates relative negative (red) and positive bias (blue). Blank entries indicate numerical instabilities. Abbreviations: Mn =Multinormal, St =Student-t, Nm =Normal, Hc =Half-cube, Sp =Spiral.	55

4.1	Quantitative evaluation. a) FID: Zero-shot FID on the COCO2014 validation split lin2014microsoftCOCO. b) Compositional image generation using the GenEval ghosh2024geneval benchmark. On both metrics, we find that our approach achieves competitive performance compared to Stable Diffusion models v1.5 and v2.1, while taking $14.2\times$ less training time than the state-of-the-art low-cost training approach (e.g., PixArt- α). We provide expanded FID and GenEval results in Tables C.8 and C.9 (Appendix C.5), respectively.	80
4.2	Win percentages of our best MicroDiT model, trained only on 490M synthetic images, against four open-source baseline models.	83
A.1	The standard deviation of NLL (bpd) estimates with information-theoretic bounds	153
A.2	Bootstrap variance estimates for continuous estimator	155
A.3	Bootstrap variance estimates for discrete estimator	155
A.4	Bootstrap variance estimates using uniform dequantization	155
B.1	Additional Experiment Details for the ARO Benchmark	165
B.2	Additional Accuracy (%) of Stable Diffusion and OpenCLIP.	166
B.3	Mean Estimator Accuracy and Std. Dev. across Random Seeds	167
B.4	The hyper-parameters used in DDIM and ITD schedulers.	168
B.5	Unsupervised Object Segmentation mIoU (%) Standard Error Analysis on COCO-IT	168
B.6	Fine-grained results in Visual Genome Relation dataset.	175

B.7	MIND network training hyper-parameters. <i>Dim</i> of the task correspond the sum of the two variables dimensions, and <i>d</i> corresponds to the randomization probability.	180
B.8	Sampling Hyperparameters	180
B.9	Ablation study comparing MIND-adaptive ("Adaptive") with adaptive importance sampling against the baseline MIND ("Baseline") with a fixed sampling distribution. This analysis uses the non-orthogonal variants to isolate the impact of the sampling strategy. For each task, the estimate closer to the Ground Truth and the lower standard deviation (Std) are independently bolded.	187
C.1	Comparing the computational cost and storage overhead of CLIP [188, 59] and T5 [192, 240] text encoders. We use the state-of-the-art CLIP model from [59]. We report the compute and storage cost for one million image captions. Even though T5 embeddings achieve better generation quality, especially for text generation, computing them is an order of magnitude slower than CLIP embeddings and even precomputing them for our dataset (37M images) poses high storage overheads (36.4 TB). In this table, we use one H100 GPU to estimate the time to process 1M image captions. We save the embeddings in <code>float16</code> precision.	192
C.2	Training hyperparameters. Hyperparameters across both phases of our large-scale training setup.	192

C.3 Ablating individual components in our training pipeline.	In each subsequent ablation, we use the overall best performing model from previous ablation. In each ablation, we train a DiT-Tiny/2 model for 60K training steps using 75% masking ratio.	202
C.4 Using layer-wise scaling.	Layer-wise scaling of transformer architecture is a better fit for masked training in diffusion transformers. We validate its effectiveness in the canonical naive masking with 75% masking ratio.	203
C.5 IsoFLOPs training.	Both patch masking and model downscaling, i.e., reducing the size of the model, reduce the computational cost in training and can be complementary to each other. However, it is natural to ask how the two paradigms compare to each other in training diffusion transformers. Under identical training setup and wall-clock time, we compare the effectiveness of model downscaling and our deferred masking approaches. We find that except at extremely high masking ratios, deferred masking achieves better performance across at least two performance metrics. Based on this finding, we do not use a masking ratio higher than 75% in our models.	205
C.6 Breakdown of computational cost.	Computational cost of the two-stage training of our large-scale model. Our total computational cost is 3.45×10^{20} FLOPs, amounting to a total cost of \$1,890 and 2.6 training days on an 8×H100 GPU machine. Report CLIP score at CFG 5. For the 4-ch encoder, put a table in the appendix. Move FID values to a separate table in the appendix, as they are confusing here. Instead, add 8×A100 and 8×H100 days.	206

C.7 Why prefer 4-channel image encoders over 16-channel image encoders in micro-budget training?	
We ablate the dimension of latent space by training our MicroDiT models in four and sixteen channel latent spaces, respectively. Even though training in higher dimensional latent space is being adopted across large-scale models [58, 44], we find that it underperforms when training on a micro-budget.	207
C.8 Zero-shot FID on COCO2014 validation split.	
We report total training time in terms of number of days required to train the model on a machine with eight A100 GPUs. We observe a $2.5\times$ reduction in training time when using H100 GPUs. Our micro-budget training takes $14.2\times$ less training time than state-of-the-art low-cost training approach while simultaneously achieving competitive FID compared to some open-source models.	207
C.9 Comparing performance on compositional image generation using GenEval [73] benchmark.	
Table adopted from [58]. Higher performance is better. Must provide some visualization on where our methods performs well. .	208
C.10 Scaling synthetic dataset to half billion images.	
Comparing performance on compositional image generation using GenEval [73] benchmark for MicroDiT models trained with different configuration of training dataset.	208

Chapter 1

Introduction

Generative modeling has progressed through several paradigms, each offering distinct trade-offs. Variational Autoencoders (VAEs) [122, 281] provide stable training and principled likelihood estimation but often produce blurry samples due to restrictive Gaussian assumptions. Generative Adversarial Networks (GANs) [75, 191, 6] can generate sharp and realistic outputs but are prone to instability and mode collapse. Normalizing Flows [201, 53, 120] enable exact likelihood computation through invertible transformations, yet their architectural constraints limit scalability to complex, high-dimensional data.

Against this backdrop, diffusion models have emerged as a powerful alternative, modeling data generation as the reversal of a gradual noising process. They connect to several established perspectives: variational formulations (e.g., DDPMs [89], DDIMs [227], VDMs [121]) cast training as optimizing a variational bound on the data likelihood; score-based diffusion models [233, 252, 232] interpret training as learning the score function (the gradient of the log-density) across noise levels, often formulated via stochastic differential

equations (SDEs); and flow matching [146, 147] reframes training as learning continuous-time vector fields that transport noise to data, unifying diffusion with other generative flows. Among the variants, the Latent Diffusion Model (LDM) [203] has been particularly influential by introducing a compression stage in which learning and generation occur in a lower-dimensional latent space rather than directly in pixel space. This design substantially reduces computational costs while preserving semantic fidelity, making large-scale training and deployment practical. Collectively, these models have achieved remarkable success in producing high-quality outputs across modalities, fueling many of the recent breakthroughs in generative AI [26].

While these viewpoints provide principled formulations, they often focus on likelihood-based or score-based objectives, leaving open the question of how information flow between conditioning variables and outputs can guide or explain the generation process. In this thesis, we introduce an information-theoretic perspective on diffusion modeling, grounded in the Information–Minimum Mean Square Error (I-MMSE) relation. Unlike variational or score-based formulations, our approach directly connects the data distribution to optimal denoising, offering exact likelihood expressions and enabling mutual-information–driven analysis of interpretability, alignment, and compositionality in generative models.

In practice, accurate density estimation is crucial for generative models to support high-stakes domains such as medical imaging [117, 140] and natural language processing [94]. To this end, we propose Information-Theoretic Diffusion (ITD), a novel mathematical framework for diffusion models grounded in classical results from information theory—specifically, the Information–Minimum Mean Square Error (I-MMSE) relations [127]. By generalizing

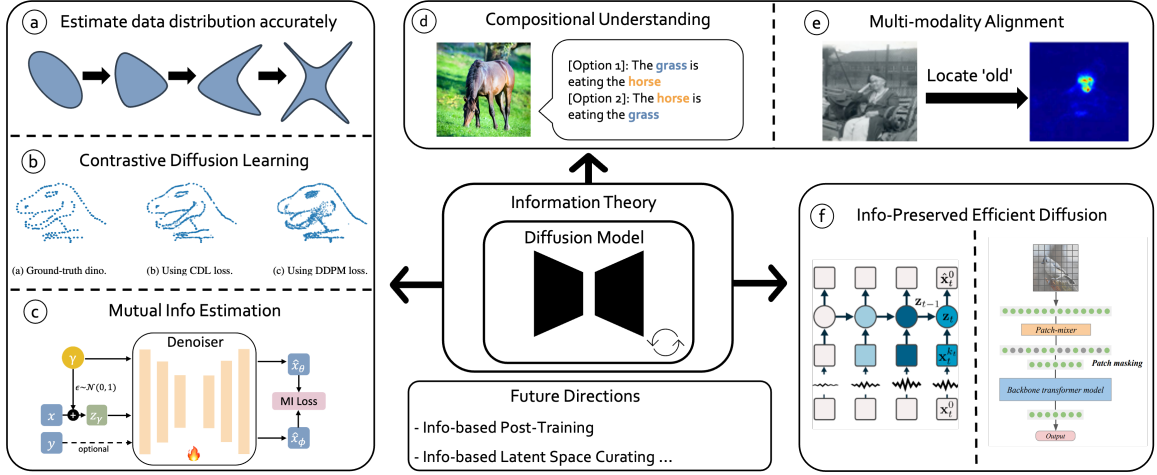


Figure 1.1: The overall research blueprint.

the I-MMSE relations, we establish a precise connection between the underlying data distribution and an optimal denoising regression problem. This formulation refines existing diffusion bounds, enabling more accurate density estimation for both continuous and discrete data (Fig. 1.1(a)). Furthermore, we find that integrating contrastive learning [86] into ITD improves its ability to capture highly skewed data distributions (Fig. 1.1(b)), such as long-tail distributions, albeit at the cost of increased training complexity.

Moreover, this framework offers a novel interpretation of how conditional diffusion models capture relationships between different modalities (e.g., text and image) through information decomposition, rather than relying on traditional attention-score fusion [129], or CLIP score that requires a pretrained model [189]. This information-theoretic view enables a range of new capabilities, including quantitatively assessing the compositional understanding of diffusion models (Fig. 1.1(d)), performing unsupervised object localization in complex scenes (Fig. 1.1(e)), and systematically analyzing the effects of prompt-based selective editing in generated images. By making the generative process more transparent, the framework

enhances interpretability, allowing researchers and practitioners to identify, analyze, and mitigate potential biases—an essential step for the safe and ethical deployment of generative AI in sensitive domains such as healthcare [3], finance [245], and creative content generation [203, 181]. Conversely, this same formulation strengthens mutual information estimation within denoising diffusion training (Fig. 1.1(c)) by defining it as half the gap between conditional and unconditional diffusion MMSE curves, integrated over all Signal-to-Noise Ratios (SNRs) in the noising process. The resulting diffusion-based estimator achieves superior accuracy over variational approaches, particularly in high-mutual information regimes, enabling models to better preserve semantic consistency, reduce information loss during generation, and maintain modality-specific diversity. These improvements directly benefit cross-modality alignment [258], ensuring that generated outputs remain both coherent across modalities and faithful to their respective input signals.

With the advent of scaling laws [110], generative models have grown increasingly large, substantially raising training costs. From an information-preserving perspective, our goal is to reduce the number of patches participating in diffusion while avoiding information loss from the training data. To achieve this, we propose a patch-masking strategy tailored for state-of-the-art Diffusion Transformers (DiT) [214]. Our method employs deferred masking, in which patches are first processed through a patch information mixer before masking (Fig. 1.1(f, right)). This design effectively mitigates performance degradation, outperforms traditional model downscaling in computational efficiency, and enables competitive diffusion model training for under \$2000. Similarly, to maximize the efficiency of information utilization, we integrate recurrent neural network structures with diffusion models, forming a large

diffusion grid [28] (Fig. 1.1(f, left)). This architecture allows intermediate variables to be transferred to the next diffusion iteration without waiting for the previous process to fully complete, thereby accelerating both training and generation while ensuring efficient information flow. Furthermore, this regime supports progressive information injection, enabling flexible composition of different content elements during generation.

Overall, in the thesis we summarize our contributions as follows:

- **Information-Theoretic Diffusion (ITD):** Develop a unified diffusion framework grounded in information theory to improve the model’s density estimation capability. Incorporating contrastive learning further enhances robustness, particularly for skewed or long-tail data distributions.
- **Mutual-Information-Driven Interpretability and Alignment:** Enhance the interpretability of diffusion models and improve cross-modality alignment through mutual information analysis. Conversely, propose a diffusion-based framework that advances mutual information estimation itself, especially in high-mutual information regimes.
- **Information-Preserving Efficient Architectures:** From an information-preserving perspective, introduce a patch mixer that maintains data information even when tokens are dropped, enabling efficient and cost-effective training. Additionally, integrate an autoregressive (AR) structure with diffusion models to support progressive, controllable generation and flexible scene composition while retaining information integrity.

Chapter 2

Enhance Density Estimation of Diffusion Based on Information Theory

2.1 Introduction

Denoising diffusion models [224] incorporating recent improvements [89] now outperform GANs for image generation [50], and also lead to better density models than previously state-of-the-art autoregressive models [121]. The quality and flexibility of image results have led to major new industrial applications for automatically generating diverse and realistic images from open-ended text prompts [194, 206, 204]. Mathematically, diffusion models can be understood in a variety of ways: as classic denoising autoencoders [257] with multiple noise levels and a new architecture [89], as VAEs with a fixed noising encoder [121], as

annealed score matching models [231], as a non-equilibrium process that tractably bridges between a target distribution and a Gaussian [224], or as a stochastic differential equation that does the same [234, 153].

In this paper, we call attention to a connection between diffusion models and a classic result in information theory relating the mutual information to the Minimum Mean Square Error (MMSE) estimator for denoising a Gaussian noise channel [77]. Research on Information and MMSE (often referred to as I-MMSE relations) transformed information theory with new representations of standard measures leading to elegant proofs of fundamental results [256]. This paper uses a generalization of the I-MMSE relation to discover an exact relation between data probability distribution and optimal denoising regression. The information-theoretic formulation of diffusion simplifies and improves existing results. Our main contributions are as follows.

- A new, exact relation between probability density and the global optimum of a mean square error denoising objective of the form, $-\log p(\mathbf{x}) = 1/2 \int_0^\infty \text{mmse}(\mathbf{x}, \gamma) d\gamma + \text{constant terms}$, with useful variations in Eq. equation B.5, equation B.6 and equation 2.11. Because this expression is exact and not a bound, we can convert any statements about density functionals such as entropy into statements about the unconstrained optimum of regression problems easily solved with neural nets.
- The same optimization problem can also be exactly related to *discrete* probability mass, Eq. equation 2.12. In contrast, the variational diffusion bound requires specifying a separate discrete decoder term in the generative model and associated variational bound.

Our unification of discrete and continuous probabilities justifies empirical work applying diffusion to categorical variables.

- In experiments, we show that our approach can take pre-trained discrete diffusion models and re-interpret them as continuous density models with competitive log-likelihoods. Our approach allows us to fine-tune and *ensemble* existing diffusion models to achieve better NLLs.

2.2 Information-Theoretic Diffusion

2.2.1 Fundamental Pointwise Denoising Relation

Let $p(\mathbf{z}|\mathbf{x})$ be a Gaussian noise channel with $\mathbf{z} = \sqrt{\gamma}\mathbf{x} + \epsilon$ and $\epsilon \sim \mathcal{N}(0, \mathbb{I})$, where γ represents the Signal-to-Noise Ratio (SNR) and $p(\mathbf{x})$ is the unknown data distribution. This channel has exploding variance as the SNR increases but we will see that the variance of this channel can be normalized arbitrarily without affecting results. Our convention matches the information theory literature and significantly simplifies proofs.

The seminal result of [77] connects mutual information with MMSE estimators,

$$\frac{d}{d\gamma} I(\mathbf{x}; \mathbf{z}) = 1/2 \text{mmse}(\gamma). \quad (2.1)$$

Here, the MMSE refers to the Minimum Mean Square Error for recovering $\mathbf{x}$ in this noisy channel,

$$\text{mmse}(\gamma) \equiv \min_{\hat{\mathbf{x}}(\mathbf{z}, \gamma)} \mathbb{E}_{p(\mathbf{z}, \mathbf{x})} [\|\mathbf{x} - \hat{\mathbf{x}}(\mathbf{z}, \gamma)\|_2^2]. \quad (2.2)$$

We refer to $\hat{\mathbf{x}}$ as the denoising function. The optimal denoising function $\hat{\mathbf{x}}^*$ corresponds to the conditional expectation, which can be seen using variational calculus or from the fact that the squared error is a Bregman divergence ([11] Prop. 1),

$$\hat{\mathbf{x}}^*(\mathbf{z}, \gamma) \equiv \arg \min_{\hat{\mathbf{x}}(\mathbf{z}, \gamma)} \text{mse}(\gamma) = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|\mathbf{z})}[\mathbf{x}] \quad (2.3)$$

The analytic solution is typically intractable because it requires sampling from the posterior distribution of the noise channel.

Recent work on variational diffusion models writes a lower bound on log likelihood explicitly in terms of MMSE [121], suggesting a potential connection to Eq. equation B.1. However, the precise nature of the connection is not clear because the variational diffusion bound is an inequality while Eq. equation 2.2 is an equality, and the variational bound is formulated pointwise for $\log p(\mathbf{x})$ at a single $\mathbf{x}$, while Eq. equation 2.2 is an expectation.

We now introduce a point-wise generalization of Guo et al’s result that is the foundation for all other new results in this paper.

$$\boxed{\frac{d}{d\gamma} D_{KL}[p(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})] = 1/2 \text{mmse}(\mathbf{x}, \gamma)} \quad (2.4)$$

The marginal is $p(\mathbf{z}) = \int p(\mathbf{z}|\mathbf{x})p(\mathbf{x})d\mathbf{x}$, and the pointwise MMSE is defined as follows,

$$\text{mmse}(\mathbf{x}, \gamma) \equiv \mathbb{E}_{p(\mathbf{z}|\mathbf{x})} [\|\mathbf{x} - \hat{\mathbf{x}}^*(\mathbf{z}, \gamma)\|_2^2]. \quad (2.5)$$

Pointwise MMSE is just the MMSE evaluated at a single point $\mathbf{x}$, and $\mathbb{E}_{p(\mathbf{x})}[\text{mmse}(\mathbf{x}, \gamma)] = \text{mmse}(\gamma)$. Taking the expectation with respect to $\mathbf{x}$ of both sides of Eq. equation B.2 recovers Guo et al’s famous result, Eq. equation B.1. Our proof of Eq. equation B.2 uses special properties of the Gaussian noise channel and repeated application of integration by parts, with a detailed proof given in Appendix A.1.1. This result can also be seen as a special case of Theorem 5 from [178] by replacing their general channel with our channel written in terms of γ and then using the chain rule. In the rest of this paper, we show how this more general version of the I-MMSE relation can be used to reformulate denoising diffusion models.

2.2.2 Diffusion as Thermodynamic Integration

We first use Eq. equation B.2 to derive an expression for log-likelihood that resembles the variational bound. However, using results from the information theory literature [256], we find that the expression can be significantly simplified and certain terms that are typically estimated can be calculated analytically.

Our derivation is inspired by the recent development of thermodynamic variational inference [158, 22] which shows how to construct a path connecting a tractable distribution to some target, such that integrating over this path recovers the log likelihood for the target model. The paths between distributions can be generalized in a number of ways [157, 31]. Typically, though, these path integrals are difficult to estimate because they require expensive sampling from complex intermediate distributions. A distinctive property of diffusion models is that the Gaussian noise channel, which transforms the target distribution into a standard normal distribution, can be easily sampled at any intermediate noise level.

We use the fundamental theorem of calculus to evaluate a function at two points in terms of the integral of its derivative, $\int_{\gamma_0}^{\gamma_1} d\gamma \frac{d}{d\gamma} f(\gamma) = f(\gamma_1) - f(\gamma_0)$. This approach is known as “thermodynamic integration” in the statistical physics literature [173, 72], where evaluation of the endpoints corresponds to a difference in the free energy or log partition function, and the derivatives of these quantities may be more amenable to Monte Carlo simulation. Consider applying the thermodynamic integration trick to the following function, $f(\mathbf{x}, \gamma) \equiv D_{KL}[p(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})]$. As before, we have $p(\mathbf{x})$, the data distribution, and a Gaussian noise channel, $\mathbf{z} = \sqrt{\gamma}\mathbf{x} + \epsilon$ with different signal-to-noise ratios γ ,

$$\begin{aligned} \int_{\gamma_0}^{\gamma_1} d\gamma \frac{d}{d\gamma} f(\mathbf{x}, \gamma) &= D_{KL}[p(\mathbf{z}_{\gamma_1}|\mathbf{x}) \parallel p(\mathbf{z}_{\gamma_1})] - D_{KL}[p(\mathbf{z}_{\gamma_0}|\mathbf{x}) \parallel p(\mathbf{z}_{\gamma_0})] \\ &= D_{KL}[p(\mathbf{z}_{\gamma_1}|\mathbf{x}) \parallel p(\mathbf{z}_{\gamma_1})] - \mathbb{E}_{p(\mathbf{z}_{\gamma_0}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z}_{\gamma_0})] + \log p(\mathbf{x}). \end{aligned}$$

In the second line, we expanded the KL divergence and used Bayes rule. Next, we re-arrange and use Eq. equation B.2 to re-write the integrand

$$-\log p(\mathbf{x}) = \underbrace{D_{KL}[p(\mathbf{z}_{\gamma_1}|\mathbf{x}) \parallel p(\mathbf{z}_{\gamma_1})]}_{\text{Prior loss}} + \underbrace{\mathbb{E}_{p(\mathbf{z}_{\gamma_0}|\mathbf{x})}[-\log p(\mathbf{x}|\mathbf{z}_{\gamma_0})]}_{\text{Reconstruction loss}} - \underbrace{1/2 \int_{\gamma_0}^{\gamma_1} \text{mmse}(\mathbf{x}, \gamma) d\gamma}_{\text{Diffusion loss}}. \quad (2.6)$$

Comparing to a particular variational bound for diffusion models in the continuous time limit (Eq. 15 from [121]), we see this expression looks similar (see App. A.1.7 for more details). However, our derivation so far is exact and we haven’t introduced any variational approximations. Prior and reconstruction loss terms are stochastically estimated in the variational formulation, but we show this is unwise and unnecessary. Consider the limit where

$\gamma_1 \rightarrow 0$. In that case, the prior loss will be zero. The reconstruction term, for continuous densities, becomes infinite in the limit of large γ_0 . For this reason, recent diffusion models only consider reconstruction for discrete random variables, $P(\mathbf{x}|\mathbf{z}_{\gamma_0})$. Estimation of conflicting divergent terms and the need for separate prior and reconstruction objectives can be avoided, as we now show.

Simple and exact probability density via MMSE

We consider applying thermodynamic integration to a slightly different function, and expand the range of integration to $\gamma \in [0, \infty)$. Consider sending samples from either the data distribution $p(\mathbf{x})$ or a standard Gaussian $p_G(\mathbf{x}) = \mathcal{N}(\mathbf{x}; 0, \mathbb{I})$ through our Gaussian noise channel. We denote the MMSE for the channel with Gaussian input as $\text{mmse}_G(\gamma)$, and write its marginal output distribution as $p_G(\mathbf{z}) = \int p(\mathbf{z}|\mathbf{x})p_G(\mathbf{x})d\mathbf{x}$. Finally, we define the function $f(\mathbf{x}, \gamma)$ as ¹

$$f(\mathbf{x}, \gamma) \equiv D_{KL}[p(\mathbf{z}|\mathbf{x}) \parallel p_G(\mathbf{z})] - D_{KL}[p(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})].$$

In the limit of zero SNR, we get $\lim_{\gamma \rightarrow 0} f(\mathbf{x}, \gamma) = 0$. In the high SNR limit we use the following result proved in App. A.1.2,

$$\lim_{\gamma \rightarrow \infty} f(\mathbf{x}, \gamma) = \log \frac{p(\mathbf{x})}{p_G(\mathbf{x})}. \quad (2.7)$$

¹Note that $\mathbb{E}_{p(\mathbf{x})}[f(\mathbf{x}, \gamma)] = \mathbb{E}_{p(\mathbf{x}, \mathbf{z})}[\log p(\mathbf{z}) - \log p_G(\mathbf{z})] = D_{KL}[p(\mathbf{z}) \parallel p_G(\mathbf{z})]$ is the gap in an upper bound $\mathbb{E}_{p(\mathbf{x})}[D_{KL}[p(\mathbf{z}|\mathbf{x}) \parallel p_G(\mathbf{z})]]$ on mutual information $I(\mathbf{x}; \mathbf{z})$, where the upper bound uses the marginal distribution induced by the Gaussian source $p_G(\mathbf{z})$ instead of the data $p(\mathbf{z})$.

Combining this with Eq. equation B.2, we can write the log likelihood *exactly* in terms of the log likelihood of a Gaussian and a one dimensional integral.

$$\begin{aligned} -\log p(\mathbf{x}) &= -\log p_G(\mathbf{x}) - \int_0^\infty d\gamma \frac{d}{d\gamma} f(\mathbf{x}, \gamma) \\ &= -\log p_G(\mathbf{x}) - 1/2 \int_0^\infty d\gamma (\text{mmse}_G(\mathbf{x}, \gamma) - \text{mmse}(\mathbf{x}, \gamma)) \end{aligned} \quad (2.8)$$

This expresses density in terms of a Gaussian density and a correction that measures how much better we can denoise the target distribution than we could using the optimal decoder for Gaussian source data. The density can be further simplified by writing out the Gaussian expressions explicitly and simplifying with an identity given in App. A.1.3.

$$\boxed{-\log p(\mathbf{x}) = d/2 \log(2\pi e) - 1/2 \int_0^\infty d\gamma \left(\frac{d}{1+\gamma} - \text{mmse}(\mathbf{x}, \gamma) \right)} \quad (2.9)$$

This expression shows that density can be written solely in terms of the global optimum of a particular regression problem, the denoising MSE. This is convenient because neural networks excel at unconstrained optimization of MSE loss functions. If we take the expectation of $-\log p(\mathbf{x})$ using Eq. equation B.6, we recover a relatively recently discovered representation of the differential entropy, $h(p)$, in terms of MMSE [256]. This highlights an advantage of our approach, as all density functionals can be rewritten in terms of the solution of an unconstrained regression problem,

$$h(p) \equiv \mathbb{E}_{p(\mathbf{x})}[-\log p(\mathbf{x})] = d/2 \log 2\pi e - 1/2 \int_0^\infty d\gamma \left(\frac{d}{1+\gamma} - \text{mmse}(\gamma) \right). \quad (2.10)$$

Since the first integral in Eq. equation B.6-equation 2.10 does not depend on $\mathbf{x}$, it is tempting to absorb it into a constant. However, the first integral diverges, and the second integral typically divergences as well – only the difference converges. This observation will help improve density estimation, by noticing that only the gap between MMSEs is important and that the gap becomes small at high and low values of SNR.

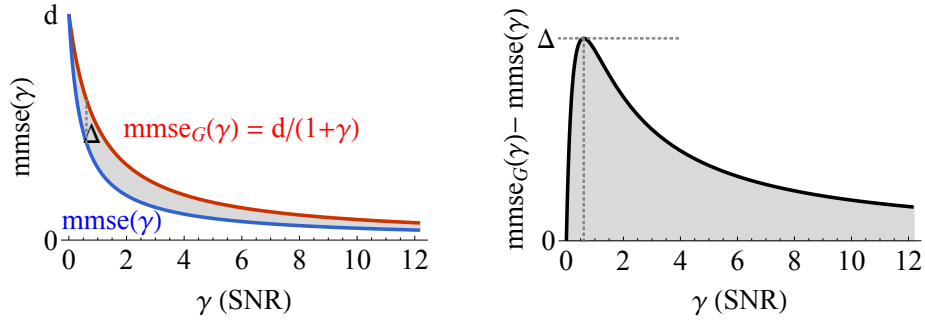


Figure 2.1: The integral of the gap between MMSE curves for data from the target distribution versus data from a Gaussian distribution is used in Eq. equation 2.10 to get an exact expression for the entropy, or expected Negative Log Likelihood (NLL), of the data.

We show example MMSE curves for denoising with Gaussian input versus non-Gaussian inputs in Fig. 2.1. The goal is to integrate the gap between these two curves, one analytic (for Gaussians) and one given by the data. Note that the gap could in principle be positive or negative. However, we know that Gaussians have the maximum possible MMSE, compared to any distribution with the same variance or covariance, when used as an input to a Gaussian noise channel [216]. Therefore, if we always compare to Gaussians with the same variance or covariance as the data, we can guarantee that the gap is positive. In practice, if the gap ever becomes negative, which we show can occur for models appearing in the literature in Sec. 2.5, it means that the discovered estimator is sub-optimal and we can fall back to the optimal Gaussian estimator to improve results in this region.

We also see that for large and small SNR values, the gap becomes small. At low SNR, the noisy data is approximately Gaussian. At high SNR, the noise goes to zero, so a linear (Gaussian) denoiser is sufficient in either case. Recognizing that the important signals come from intermediate SNR values can help save computation over approaches that indiscriminately optimize over the whole curve, as we discuss in Sec. 2.4.

Standardizing data to have unit variance to ensure that the MMSE gap is always positive may be inconvenient, so we derive a variation of Eq. equation B.6 that takes the scale into account. Instead of using $p_G(\mathbf{x}) = \mathcal{N}(\mathbf{x}; 0, \mathbb{I})$, we can take the base measure to be a Gaussian, $p_G(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$, with mean and covariance that match the data. Letting $\lambda_1, \dots, \lambda_d$ be the eigenvalues of the covariance matrix $\boldsymbol{\Sigma}$, we derive a compact expression in App. A.1.4,

$$\mathbb{E}_{p(\mathbf{x})}[-\log p(\mathbf{x})] = \underbrace{1/2 \log \det(2\pi e \boldsymbol{\Sigma})}_{\text{Gaussian entropy}} - \underbrace{1/2 \int_0^\infty d\gamma \left(\sum_{i=1}^d \frac{1}{\gamma + 1/\lambda_i} - \text{mmse}(\gamma) \right)}_{\text{Deviation from Gaussianity} \geq 0}. \quad (2.11)$$

This expression tells us precisely which part of the reconstruction mean square error curve is actually important, namely the part that deviates most from Gaussianity. Note that if the eigenvalues of $\boldsymbol{\Sigma}$ are not feasible to estimate, we can use a diagonal covariance matrix for the base Gaussian and still preserve the desired property that the gap between the data MMSE and Gaussian MMSE is non-negative [216].

Finally, we make several remarks to relate our exact expression for the likelihood in Eq. equation B.6 to existing work in the diffusion literature.

- Since the MSE minimization is performed at each γ , reweighting terms with different γ [121] should not have an effect as long as we achieve the global minima at each γ .
- Several heuristics have been suggested for improving the “noise schedule”, which corresponds to points for evaluating the MMSE integral in our formulation [89, 121, 171]. Our formulation highlights the importance of sampling from SNRs where there is a large gap, and we suggest a simple and effective strategy for this in Sec. 2.4.
- The literature on diffusion models also suggests more complex denoising distributions that also model the covariance [171]. Modeling the covariance of the denoising model is unnecessary in our approach, as our exact expressions never require it.
- Compare Eq. equation B.6 to the exact, continuous density expression in Eq. 39 of [234], which combines neural ODE flows and diffusion models. The form of that expression is:

$$-\log p(\mathbf{x}(0)) = -\log p_G(\mathbf{x}(T)) + \int_0^T \nabla \cdot \mathbf{g}(\mathbf{x}(t), t) dt.$$

The first step to using this expression is to solve a differential equation for the full trajectory, $\mathbf{x}(t)$, that depends on many evaluations of a learned denoising diffusion (or score) model. Then, estimating the integral is highly non-trivial as $\mathbf{g}$ is complex and its divergence is expensive to compute, necessitating some stochastic approximations. See Sec. 3.5 for additional discussion.

Discrete probability estimator Modeling probability distributions over continuous versus discrete random variables typically requires rather different approaches. An unusual feature of I-MMSE relations is that they naturally handle both types of random variables. In this subsection, consider that $\mathbf{x} \in \mathcal{X} \subset \mathcal{Z}^d$, with a domain that is discrete

but numeric (non-numeric discrete variables can be handled via an appropriate mapping function [77]). We will use capital $P(\mathbf{x})$ to denote probability (rather than probability *density*) over this discrete domain. Note that the Gaussian noise channel, $\mathbf{z} = \sqrt{\gamma}\mathbf{x} + \epsilon$, is still continuous with an associated probability density, $p(\mathbf{z}|\mathbf{x})$.

Re-doing the analysis in Eq. equation 2.6 leads to the same expression, but replacing $p(\mathbf{x})$ with $P(\mathbf{x})$. The first two terms in Eq. equation 2.6 go to zero,

$$\lim_{\gamma_0 \rightarrow \infty, \gamma_1 \rightarrow 0} D_{KL}[p(\mathbf{z}_{\gamma_1}|\mathbf{x}) \parallel p(\mathbf{z}_{\gamma_1})] + \mathbb{E}_{p(\mathbf{z}_{\gamma_0}|\mathbf{x})}[-\log P(\mathbf{x}|\mathbf{z}_{\gamma_0})] = 0.$$

This leads to the following form for the discrete probability.

$$\boxed{-\log P(\mathbf{x}) = 1/2 \int_0^\infty \text{mmse}(\mathbf{x}, \gamma) d\gamma} \quad (2.12)$$

At large SNR, $\mathbf{z}$ is very concentrated around a discrete point specified by $\mathbf{x}$. In that case, we should be able to recover the true value from the slightly perturbed value with high probability. Sec. 2.4 derives tail bounds used for approximating this integral numerically. Taking the expectation of this general result recovers an equation for the Shannon entropy from Section VI of [77],

$$H(\mathbf{x}) \equiv - \sum_{\mathbf{x} \in \mathcal{X}} P(\mathbf{x}) \log P(\mathbf{x}) = \mathbb{E}_{\mathbf{x} \sim P(\mathbf{x})}[-\log P(\mathbf{x})] = 1/2 \int_0^\infty \text{mmse}(\gamma) d\gamma. \quad (2.13)$$

Conveniently, our discrete and continuous probability representations are nearly identical – they rely on the same integral and optimization and differ only by constant terms. We

discuss comparisons to the variational bounds in App. A.1.7 and numerical implementation details in Sec. 2.4.

2.3 What Your Diffusion Model is Hiding: An Optimal Classifier

We now introduce our first interesting discovery, which shows that diffusion models implicitly define optimal noise classifiers. Eq. (B.6) expresses the probability density of the data directly in terms of the denoising function. If we apply Eq. (B.6) to the noisy distributions, $p_\zeta(\mathbf{x})$, that bridge the data and a Gaussian, we can see that all mixture densities can be written in terms of the same optimal denoising function, $\hat{\epsilon}(\cdot, \cdot)$. The complete derivation is presented in App. A.1.8.

$$-\log p_\zeta(\mathbf{x}) = c + 1/2 \int_{-\infty}^{\infty} d\alpha \mathbb{E}_{p(\epsilon)} [\|\epsilon - b \cdot \hat{\epsilon}(\mathbf{x}_\alpha, \beta)\|_2^2] \quad (2.14)$$

$$\mathbf{x}_\alpha \equiv \sqrt{\sigma(\alpha)}\mathbf{x} + \sqrt{\sigma(-\alpha)}\epsilon \quad (2.15)$$

$$\beta \equiv \sigma^{-1}(\sigma(\zeta)\sigma(\alpha)), \quad b \equiv \sqrt{\sigma(-\alpha)/\sigma(-\beta)} \quad (2.16)$$

Intuitively, if we find the optimal denoising function for the data distribution, it may be hypothesized that it can denoise an already *noisy* version of the data distribution. Using Eq. B.6, this directly translates into an expression for density of mixture distributions. Differences in log likelihoods lead to cancellation of constants, and these Log Likelihood Ratios (LLR) are related to the optimal classifiers nce as we show below.

To connect LLRs with classification, consider the following generative model. We generate a random binary label $q(y = \pm 1) = 1/2$. Then, conditioned on y , we sample from some distribution $q(\mathbf{x}|y)$. Given samples $(\mathbf{x}, y) \sim q(\mathbf{x}, y) = q(\mathbf{x}|y)q(y)$, the Bayes optimal classifier is:

$$\begin{aligned}
q(y|\mathbf{x}) &= \frac{q(\mathbf{x}|y) q(y)}{q(\mathbf{x})} \\
&= \frac{q(\mathbf{x}|y) q(y)}{q(\mathbf{x}|y=1) q(y=1) + q(\mathbf{x}|y=-1) q(y=-1)} \\
&= \frac{1}{1 + \frac{q(\mathbf{x}|y=-1)}{q(\mathbf{x}|y=1)}} = \frac{1}{1 + \exp(y [\log q(\mathbf{x}|y=-1) - \log q(\mathbf{x}|y=1)])} \\
\log q(y|\mathbf{x}) &= -\log \left(1 + \exp \left(y \log \frac{q(\mathbf{x}|y=-1)}{q(\mathbf{x}|y=1)} \right) \right) \\
&= -\text{softplus} \left(y \log \frac{q(\mathbf{x}|y=-1)}{q(\mathbf{x}|y=1)} \right) \tag{2.17}
\end{aligned}$$

2.3.1 Contrastive Diffusion Loss (CDL)

Next, we build up a new connection between diffusion denoisers and noise classifiers to define a new training objective. We set the distributions $q(\mathbf{x}|y=1)$ and $q(\mathbf{x}|y=-1)$ to be two distributions at different noise levels that we can write in terms of the optimal diffusion denoiser from Eq. 2.14. So we have $q(\mathbf{x}|y=1) \equiv p(\mathbf{x})$, the data distribution, and $q(\mathbf{x}|y=-1) \equiv p_\zeta(\mathbf{x})$, for some noise level, ζ . Then given a sample $(\mathbf{x}, y) \sim q(\mathbf{x}, y)$ the per-sample cross-entropy loss for the noise classifier, Eq. (2.17), is as follows.

$$\mathcal{L}_{CDL} = \mathbb{E}_{q(\mathbf{x}, y)} [\text{softplus}(y(\log p_\zeta(\mathbf{x}) - \log p(\mathbf{x})))] \tag{2.18}$$

We can estimate both densities directly from our denoising model using Eq. (2.14), with the constants canceling out in the process. This loss differs significantly from the standard diffusion loss. Intuitively, to distinguish between a sample from the data distribution, $p(\mathbf{x})$, versus a noisy version of the data distribution, $p_\zeta(\mathbf{x})$, we need to evaluate denoisers on points from both distributions. In standard diffusion training, denoisers at noise level ζ are only trained on samples from $p_\zeta(\mathbf{x})$.

2.4 Implementation

While we start with exact expressions for density and discrete probability in Eqs. equation B.6 and equation 2.12, in practice there will be two sources of error when implementing our approach numerically. First, we must parametrize our estimator as a neural network that may not achieve the global minimum mean square error and, second, we are forced to rely on numerical integration.

MMSE Upper Bounds While the likelihood bounds in Eqs. equation 2.6, equation B.5, and equation B.6 are exact, evaluating each $\text{mmse}(\mathbf{x}, \gamma)$ term requires access to the optimal denoising function or conditional expectation $\hat{\mathbf{x}}^*(\mathbf{z}, \gamma) = \mathbb{E}_{p(\mathbf{x}|\mathbf{z})}[\mathbf{x}]$. Using a suboptimal denoising function $\hat{\mathbf{x}}(\mathbf{z}, \gamma)$ instead of $\hat{\mathbf{x}}^*(\mathbf{z}, \gamma)$ to approximate the MMSE, we obtain an upper bound whose gap can be characterized as

$$\mathbb{E}_{p(\mathbf{x}, \mathbf{z})} [\|\mathbf{x} - \hat{\mathbf{x}}(\mathbf{z}, \gamma)\|_2^2] = \underbrace{\mathbb{E}_{p(\mathbf{x}, \mathbf{z})} [\|\mathbf{x} - \hat{\mathbf{x}}^*(\mathbf{z}, \gamma)\|_2^2]}_{\text{mmse}(\gamma)} + \underbrace{\mathbb{E}_{p(\mathbf{z})} [\|\hat{\mathbf{x}}^*(\mathbf{z}, \gamma) - \hat{\mathbf{x}}(\mathbf{z}, \gamma)\|_2^2]}_{\text{estimation gap}}.$$

In App. A.1.6, we derive this upper bound and its gap using results of [11] for general Bregman divergences. This translates to an upper bound on the Negative Log Likelihood (NLL),

$$\mathbb{E}_{p(\mathbf{x})}[-\log p(\mathbf{x})] \leq 1/2 \log \det(2\pi e \mathbf{\Sigma}) - 1/2 \int_0^\infty d\gamma \left(\sum_{i=1}^d \frac{1}{\gamma + 1/\lambda_i} - \mathbb{E}_{p(\mathbf{x}, \mathbf{z})} [\|\mathbf{x} - \hat{\mathbf{x}}(\mathbf{z}, \gamma)\|_2^2] \right).$$

Restricting Range of Integration

As noted in Fig. 2.1, only the difference $\text{mmse}_G(\gamma) - \text{mmse}(\gamma) \geq 0$ contributes to our expected NLL bound. The non-negativity of this difference is guaranteed by the results of [216], but may not hold in practice due to our suboptimal estimators of the MMSE, $\mathbb{E}_{p(\mathbf{x}, \mathbf{z})} [\|\mathbf{x} - \hat{\mathbf{x}}(\mathbf{z}, \gamma)\|_2^2] \geq \text{mmse}(\gamma)$.

In regions $\gamma \leq \gamma_0$ or $\gamma \geq \gamma_1$ where the difference in mean square error terms appears to be negative, we can simply define $\hat{\mathbf{x}}(\mathbf{z}, \gamma) \equiv \hat{\mathbf{x}}_G^*(\mathbf{z}, \gamma)$, where $\hat{\mathbf{x}}_G^*(\mathbf{z}, \gamma)$ is the optimal, linear decoder for a Gaussian with the same covariance as the data (App. A.1.5). For this choice of decoder, the integrand becomes zero so that we may drop the tails of the integral outside of the appropriate range $\gamma \in [\gamma_0, \gamma_1]$,

$$\mathbb{E}_{p(\mathbf{x})}[-\log p(\mathbf{x})] \leq 1/2 \log \det(2\pi e \mathbf{\Sigma}) - 1/2 \int_{\gamma_0}^{\gamma_1} d\gamma \left(\sum_{i=1}^d \frac{1}{\gamma + 1/\lambda_i} - \mathbb{E}_{p(\mathbf{x}, \mathbf{z})} [\|\mathbf{x} - \hat{\mathbf{x}}(\mathbf{z}, \gamma)\|_2^2] \right).$$

Parametrization

We parametrize $\hat{\mathbf{x}}(\mathbf{z}, \gamma) \equiv (\mathbf{z} - \hat{\epsilon}(\mathbf{z}, \gamma))/\sqrt{\gamma}$, $\forall \gamma \in [\gamma_0, \gamma_1]$.

With this definition, we can re-arrange to see that the error for reconstructing the noise, ϵ , is related to the error for reconstructing the original image, $\hat{\epsilon}(\mathbf{z}, \gamma) - \epsilon = \sqrt{\gamma}(\mathbf{x} - \hat{\mathbf{x}}(\mathbf{z}, \gamma))$. Our neural network then implements $\hat{\epsilon}$, a function that predicts the noise. The inputs to this network are $(\mathbf{z}, \gamma)$. Note that we can equivalently parametrize our network to use the inputs

$(\mathbf{z}/\sqrt{1+\gamma}, \gamma)$. This choice is preferable so that the variance of the noisy image representation is bounded, while the SNR is unchanged. Numerically, it is better to work with log SNR values, so we change variables, $\alpha = \log \gamma$, which leads to the following form

$$\mathbb{E}_{p(\mathbf{x})}[-\log p(\mathbf{x})] \leq 1/2 \log \det(2\pi e \mathbf{\Sigma}) - 1/2 \int_{\alpha_0}^{\alpha_1} d\alpha \left(f_{\Sigma}(\alpha) - \mathbb{E}_{\mathbf{x}, \epsilon} [\|\epsilon - \hat{\epsilon}(\mathbf{z}, \gamma)\|_2^2] \right). \quad (2.19)$$

Here, $f_{\Sigma}(\alpha) \equiv \sum_{i=1}^d \sigma(\alpha + \log \lambda_i)$, using the traditional sigmoid function, $\sigma(t) = 1/(1 + e^{-t})$.

Numerical Integration The last technical issue to solve is how to evaluate the integral in Eq. equation 2.19 numerically. We use importance sampling to write this expression as an expectation over some distribution, $q(\alpha)$, for which we can get unbiased estimates via Monte Carlo sampling. This leads to the final specification of our loss function $\mathbb{E}_{p(\mathbf{x})}[-\log p(\mathbf{x})] \leq \mathcal{L}$, where

$$\mathcal{L} \equiv 1/2 \log \det(2\pi e \mathbf{\Sigma}) - 1/2 \mathbb{E}_{q(\alpha)} \left[1/q(\alpha) \left(f_{\Sigma}(\alpha) - \mathbb{E}_{\mathbf{x}, \epsilon} [\|\epsilon - \hat{\epsilon}(\mathbf{z}, \gamma)\|_2^2] \right) \right]. \quad (2.20)$$

All that remains is to choose $q(\alpha)$. We use our analysis of the Gaussian case to motivate this choice. Note that $f_{\Sigma}(\alpha)$ is a mixture of CDFs for logistic distributions with unit scale and different means. Mixtures of logistics with different means are well approximated by a logistic distribution with a larger scale [42]. So we take $q(\alpha)$ to be a truncated logistic distribution with mean $\mu = \text{Mean}_i(-\log \lambda_i)$ and scale $s = \sqrt{1 + 3/\pi^2 \text{Var}_i(\log \lambda_i)}$, based on moment matching to the mixture of logistics implied by $f_{\Sigma}(\alpha)$. Samples can be generated from a uniform distribution using the quantile function, $\alpha = \mu + s \log t/(1 - t)$, $t \sim \mathcal{U}[t_0, t_1]$. The quantile range is set so that $\alpha \in [\mu - 4s, \mu + 4s]$. The objective is estimated with

Monte Carlo sampling, providing a stochastic, unbiased estimate of our upper bound in Eq. equation 2.19. Numerical integration alternatives include the trapezoid rule [132, 67, 99] or a Riemann sum, due to the monotonicity of $\text{mmse}(\mathbf{x}, \gamma)$ ([121] Fig. 2, [158]).

Comparing between continuous and discrete probability estimators

Recent diffusion models are trained assuming discrete data. We can measure how well they model continuous data by viewing continuous density estimation as the limiting density of discrete points [104]. Treating $p(\mathbf{x})$ as a uniform density in some d -dimensional box or bin of volume Δ^d around each discrete point leads to the relation, $\mathbb{E}[-\log p(\mathbf{x})] = \mathbb{E}[-\log P(\mathbf{x} \in \text{bin})/(\Delta^d)] = \mathbb{E}[-\log P(\mathbf{x} \in \text{bin}) - d \log \Delta]$. In the other direction, when we use a continuous density estimator to model a discrete density, we use uniform dequantization [88].

Direct discrete probability estimate

Finally, we derive a practical upper bound on negative log likelihood for discrete probabilities.

$$\begin{aligned} \mathbb{E}[-\log P(\mathbf{x})] &= 1/2 \int_0^\infty \text{mmse}(\gamma) d\gamma = 1/2 \int_{\gamma_0}^{\gamma_1} \text{mmse}(\gamma) d\gamma + 1/2 \left(\int_0^{\gamma_0} + \int_{\gamma_1}^\infty \right) \text{mmse}(\gamma) d\gamma \\ \mathbb{E}[-\log P(\mathbf{x})] &\leq 1/2 \int_{\gamma_0}^{\gamma_1} \mathbb{E}_{\mathbf{z}, \mathbf{x}} [\|\mathbf{x} - \hat{\mathbf{x}}(\mathbf{z}, \gamma)\|_2^2] d\gamma + c(\gamma_0, \gamma_1) \end{aligned} \quad (2.21)$$

In App. A.2.1, we analytically derive upper bounds for the left and right tail of the integral, expressed using $c(\gamma_0, \gamma_1)$. We also get an upper bound from using a denoiser that is not necessarily globally optimal. The discrete and continuous density estimators differ only by constants, therefore we can use the same importance sampling estimator for the integral, as in Eq. 4.8.

Choice of noise ζ for CDL

Combining Eq. (B.6) and Eq. (2.14) we have

Eq. (3.4), where the constant cancels out.

$$LLR = \log p_{\zeta}(\mathbf{x}) - \log p(\mathbf{x}) = \int_{-\infty}^{\infty} d\alpha \mathbb{E}_{p(\epsilon)}[\|\epsilon - \hat{\epsilon}(\mathbf{z}, \alpha)\|_2^2] - \mathbb{E}_{p(\epsilon)}[\|\epsilon - b\hat{\epsilon}(\mathbf{z}, \beta)\|_2^2] \quad (2.22)$$

$$\text{with: } \mathbf{z} \equiv \sqrt{\sigma(\alpha)}\mathbf{x} + \sqrt{\sigma(-\alpha)}\epsilon$$

Note that the input $\mathbf{x}$ to the LLR term may come from two different distributions, which breaks the standard synchronous denoising pair $(\mathbf{x}_{\alpha}, \alpha)$ into asynchronous. When it's from data distribution $\mathbf{x} \sim p(\mathbf{x})$, $\mathbf{z} = \mathbf{z}_{\alpha}$; and when it's from some noisy data distribution $\mathbf{x} \sim p_{\zeta}(\mathbf{x})$, $\mathbf{z} = \mathbf{z}_{\beta}$.

From Eq. (3.4) we see that $\hat{\epsilon}(\cdot, \cdot)$ is trained on four pairs: $(\mathbf{z}_{\alpha}, \alpha)$, $(\mathbf{z}_{\alpha}, \beta)$, $(\mathbf{z}_{\beta}, \beta)$ and $(\mathbf{z}_{\beta}, \alpha)$, where $\beta \equiv \sigma^{-1}(\sigma(\alpha)\sigma(\zeta)) < \min(\alpha, \zeta)$ (Eq. 2.16). During standard training, only the first two pairs are trained (Eq. 2.3). This means that our CDL objective trains the denoiser to perform correctly even for samples from distributions that are noisier or cleaner than the specified noise level (a pair like $(\mathbf{z}_{\alpha}, \beta)$ or $(\mathbf{z}_{\beta}, \alpha)$). This can be useful for both sequential and parallel sampling settings. During sequential sampling, extra error noise added due to discretization errors can be corrected by the denoiser trained with CDL. As for parallel sampling, CDL helps with evaluations on asymmetric pairs $(\mathbf{z}_{\beta}, \alpha)$ or $(\mathbf{z}_{\alpha}, \beta)$ which we refer to OOD regions for standard diffusion loss.

In practice, diffusion training pipelines are highly tuned on popular datasets like CIFAR10 and ImageNet, so the amplitude of discretization errors during sampling is small, meaning that errors won't nudge points too far away from the true trajectory. Therefore, when evaluating CDL objective, we sample some large-valued ζ s, which corresponds to

classifying only small differences in noise levels. Empirically we find that $\zeta \sim \text{Uniform}[6, 15]$ or $\zeta \sim \text{logistic}[6, 15]$ performed equally good.

2.5 Experiments

Our results establish an exact connection between the data likelihood and the solution to a regression problem, the Gaussian denoising MMSE. If we integrate the MSE curve of a particular denoiser and get a worse bound, it simply means that the denoiser does not achieve the MMSE. Interestingly, variational diffusion models optimize an objective that can be made quite similar with appropriate choices for variational distributions (App. A.1.7). Therefore, we can take previously trained diffusion models and evaluate alternate likelihood bounds using the methods described above. We can attempt to improve these bounds by directly optimizing the denoising MSE. Finally, a straightforward implication of our results is that we would like the minimum MSE denoiser at each SNR value; therefore, if we have different denoisers with lower error at different SNR values, we can combine them to get a density estimator which outperforms any individual one. *Ensembling* diffusion models that “specialize” in different SNR ranges can likely be used to construct new SOTA density models.

For the following experiments we use the CIFAR-10 dataset, scaled to have pixel values in $[-1, 1]$, and consider a pre-trained DDPM model [89] and an Improved DDPM we refer to as IDDPM [171]. We use 4000 diffusion steps for calculating variational bounds. For our IT based methods, the comparable parameter is how many log SNR samples, $\alpha \sim q(\alpha)$, we use for evaluation. For continuous density estimation 100 points were sufficient, while for

the discrete estimator we used 1000 points (as discussed in App. A.2.5). More details on models and training can be found in App. A.2.3.

Continuous Density Estimation with Diffusion

We first explore the results of continuous probability density estimation on test data using variational bounds versus our approach. Recent diffusion based models treat the data as discrete, bounding $\log P(\mathbf{x})$, however, we could also use diffusion models to give continuous density estimates by interpreting the last denoising step as providing a Gaussian distribution over $\mathbf{x}$. For comparison, we can also take the variational bound for discrete data and create a density estimate by making density uniform within each bin. The results are shown in Fig. 2.2. For our estimator, the shaded integral between the Gaussian MMSE and the denoising MSE is used, and this provides noticeably improved density estimates. Note that for our bound calculated using the IDDPM model, we throw away the part of their network that estimates the covariance and still achieve improved results, validating our assertion that variational modeling of covariance is unnecessary.

Continuous density estimators applied to intrinsically discrete pixel data, like CIFAR-10, can in principle lead to very low NLLs if the model learns to put large mass on discrete points. In our case, we are comparing the same model so the comparison would still be meaningful. Furthermore, we can see that the diffusion models are not putting probability mass on discrete points by comparing the MSE curves to denoisers where discretization is purposely introduced in Fig. 2.3.

Discrete Probability

NLL bounds for diffusion models in recent work are for discrete probability estimates. We compare to our estimate based on an exact relation

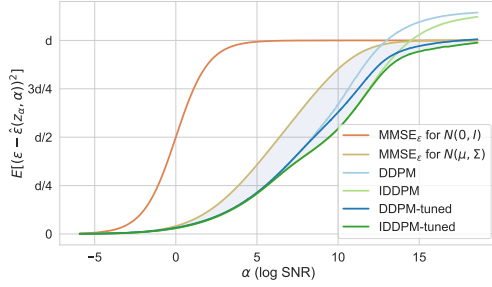


Table 2.1: $\mathbb{E}[-\log p(\mathbf{x})]$ on Test Data (bpd)

Model	Variational Bounds		IT
	Disc.*	Cont.	Cont.
DDPM	-3.62	-3.44	-3.56
IDDPM	-4.05	-3.58	-4.09
DDPM(tune)	-3.55	-3.51	-3.84
IDDPM(tune)	-3.85	-3.55	-4.28
Ensemble	-	-	-4.29

Figure 2.2: (Left) MSE curves for different denoisers, used in estimating Negative Log Likelihood (NLL). (Right) Continuous NLL estimates for diffusion models using variational bounds and Information-Theoretic (IT) bounds (ours). Variance estimates are shown in App. A.2.5. *Uses a discrete estimator made continuous by assuming uniform density in each bin.

between MMSE and discrete likelihood, Eqs. equation 2.12 and equation 2.21, and to our continuous NLL estimator treated as a discrete estimator with uniform dequantization. The diffusion architectures we tried do not naturally concentrate predictions on grid points at high SNR, so we had to explicitly add rounding via a “soft discretization” function (described in App. A.2.4), which leads to much lower error at high SNR, as seen in Fig. 2.3. Further improvements from ensembling the best models at different SNRs are described below.

Fine-tuning

Fig. 2.2 and 2.3 show that we can improve log likelihoods by fine-tuning existing models using our regression objective derived in Eq. 4.8, rather than the variational bound. In particular, we see that the improvements for continuous density estimation come from reducing error at high SNR levels. The inability of existing architectures to exploit discreteness in the data is the reason fine-tuning improves the continuous but not the discrete estimator in Table 2.2. The final discrete estimates are improved by including a soft discretization nonlinearity and using ensembling, as described next. See App. A.2.3 for additional experimental details.

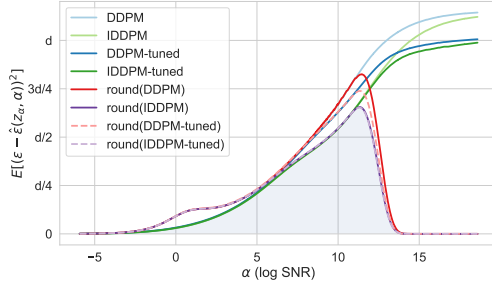


Table 2.2: $\mathbb{E}[-\log P(\mathbf{x})]$ on Test Data (bpd)

Model	Var.	IT	
		Disc.*	Cont.**
DDPM	3.37	3.68	3.51
IDDPM	2.94	3.16	3.17
DDPM(tune)	3.45	3.48	3.41
IDDPM(tune)	3.14	3.15	3.18
Ensemble	-	2.90	3.16

Figure 2.3: (Left) MSE curves for different denoisers, used in estimating Negative Log Likelihood (NLL), shown with or without rounding using a soft discretization function (App. A.2.4). (Right) Discrete NLL estimates for diffusion models using variational bounds and information-theoretic bounds (ours). Variance estimates are shown in App. A.2.5. *Using the denoiser with soft discretization. **Indicates a continuous estimator made discrete via uniform dequantization.

Ensembling

Finally, we propose to ensemble different denoising diffusion models by choosing the denoiser with the lowest MSE at each SNR level in Eq. equation 2.19. Since our estimator depends only on estimating the smallest MSE at each γ , our likelihood estimates will benefit from combining models whose relative performance differs across SNR regions. In Fig. 2.2 and 2.3, we report improved results from ensembling DDPM [89], IDDPM [171], fine-tuned versions, and rounded versions (in the discrete case). For discrete estimators, rounding is useful at high SNR but counterproductive at low SNR. The ensemble MSE (shaded region of Fig. 2.3) gives a better NLL bound than any individual estimator.

CDL Training

We consider the 2D Dino dataset [159], characterized by its highly nonlinear density concentrated on a low-dimensional manifold. For baselines, we employ the standard DDPM loss [90], as all standard diffusion losses similarly minimize a sum of MSE losses between the actual and estimated denoisers. Both CDL-regularized and DDPM-objective-trained models are trained with a MLP where timestep is encoded

by positional encoding. We train it for 2000 epochs to ensure convergence and check the training and validation loss curve to avoid overfitting.

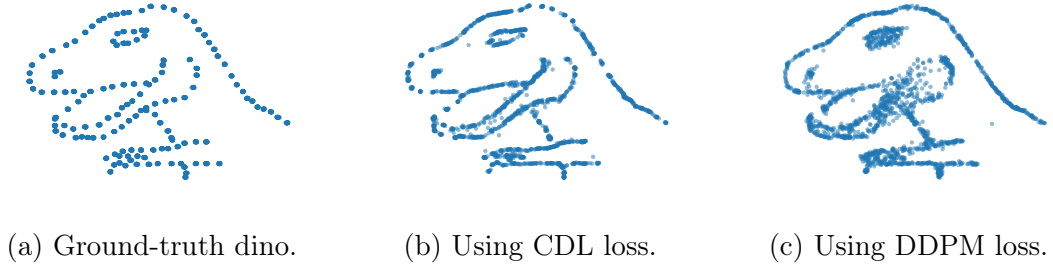


Figure 2.5: Parallel DDPM sampler generated Dino data. Comparing to Dino sampled from DDPM loss, CDL-loss sampled Dino has better sample quality and density estimate around hard areas.

In Fig. 2.5, it is clearly demonstrated that the parallel generated samples from the CDL-regularized model is much better than the model trained only with the standard DDPM loss, especially around the chin, eyes and paws, where the manifolds are close to each other and difficult to distinguish and learn.

2.6 Related Work

The developments in diffusion models that have enabled new industrial applications [194, 206, 204] build on a rich intellectual history of ideas spanning many fields including score matching[100], denoising autoencoders [257], nonequilibrium thermodynamics [224], neural ODEs [32], and stochastic differential equations [234, 96]. The observation that part of the variational diffusion loss could be written as an integral of MMSEs was made by [121]. To the best of our knowledge, the connection between I-MMSE relations in information theory and diffusion models has so far gone unrecognized.

[234] introduced a different way to use diffusion to model probability densities via differential equations. This approach requires solving a differential equation of a dynamic trajectory for each sample. Our estimate can be computed in a single step by evaluating the score model at each SNR in parallel, but the neural probability flow ODE will require thousands of sequential score function evaluations to solve. Concurrent work uses stochastic differential equations along with Girsanov’s theorem, a stochastic version of the change of variables formula, to show that optimizing a regression objective is sufficient to exactly match a stochastic process to some target density [153, 271]. The results are used to improved sampling, but they use standard variational bounds for density estimation. Our work, in contrast, exactly relates regression to density estimation but does not address sampling. The form of their results for discrete distributions suggests a deeper connection between their work and this one.

Our work focused on density modeling, so approaches that forgo density modeling to improve image generation, for instance by doing diffusion in a latent space [204], were not considered. [13] observe that denoising diffusion models using many types of noise can lead to good image sampling, however, our results highlight the special connection between Gaussian denoising and density modeling.

Recent work has studied applicability of diffusion to discrete and categorical variables [7, 76, 34]. Our exact relation between denoising and discrete probability provides theoretical justification for this promising line of research. Finally, this paper gives a theoretical justification for ensembling diffusion models, to achieve the best MSE at different SNR levels. We see from concurrent work that this strategy is already being successfully employed [9].

Learning based on contrastive losses builds on the seminal works [79, 174], with influential applications including SimCLR [33] and CLIP [189]. A few works have considered contrastive learning inspired modifications to diffusion either to enforce multimodal data relationships [134, 285], for style transfer [269], or for guidance during generation [177], but none use the diffusion model as a noise classifier to improve diffusion training as we do. Most similar to our approach are [202, 39] that both train a classifier to distinguish between samples from distributions on a path between a source and target distribution. Our contribution is to relate this approach to diffusion models by noting that standard diffusion models implicitly already implement the required classifiers for distinguishing distributions on the path from the data distribution to a Gaussian distribution.

2.7 Conclusion

Variational methods are powerful, but require many choices in designing variational approximations. More complex choices for the variational distributions could lead to tighter bounds, and a fair amount of work has explored this possibility. Interestingly, though, the most successful refinements of diffusion models have led to objectives that are more similar to the denoising MSE, and the results in this paper finally make it clear that regression is all that is needed for exact probability estimation, for both continuous and discrete variables. We generalized the classic I-MMSE relation from information theory to introduce this simple, *exact* relationship between the data probability distribution and optimal denoisers, $-\log p(\mathbf{x}) = 1/2 \int_0^\infty \text{mmse}(\mathbf{x}, \gamma) d\gamma + \text{constant terms}$. By integrating contrastive learning, our method further enhances the model’s ability to approximate complex data distributions.

This result pares away the unnecessary ingredients in the variational approach, and its simple, evocative form opens the door for many future improvements and generalizations.

Chapter 3

Bidirectional Synergy Between Mutual Information and Diffusion Models

3.1 Introduction

Denoising diffusion models are the state-of-the-art for modeling relationships between complex data like images and text. While diffusion models exhibit impressive generative abilities, we have little insight into precisely what relationships are learned (or neglected). Often, models have limited value without the ability to dissect their contents. For instance, in biology specifying which variables have an effect on health outcomes is critical. As AI advances, more principled ways to probe learned relationships are needed to reveal and correct gaps between human and AI perspectives.

Quantifying the relationships learned in a complex space like text and images is difficult. Information theory offers a black-box method to gauge how much information flows

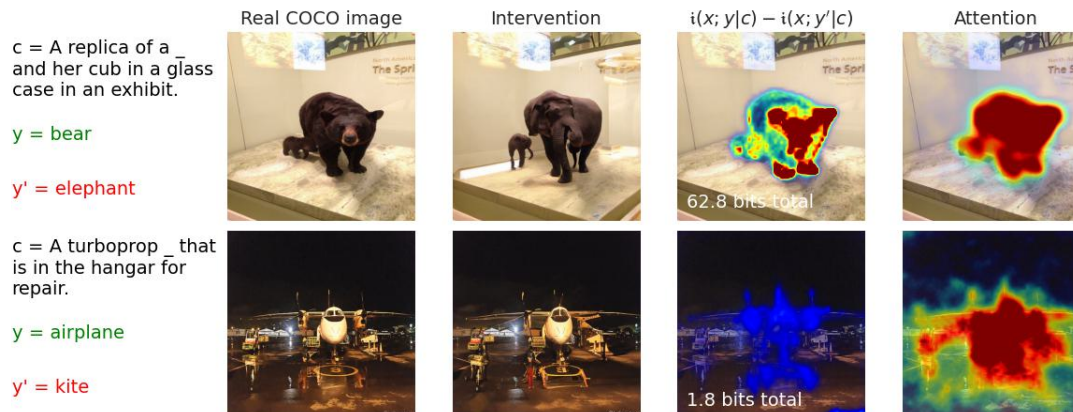


Figure 3.1: We start (left) with a real image from the COCO dataset. We do a “prompt intervention” (§3.4.3) to generate a new image. Next we show conditional mutual information, illustrated using our pixel-wise decomposition, and attention maps for the modified word. Top row shows an image where prompt intervention has an effect, while in the bottom row it has little effect. Conditional mutual information reflects the effect of intervention while attention does not.

from inputs to outputs. This work proceeds from the novel observation that diffusion models naturally admit a simple and versatile information decomposition that allows us to pinpoint information flows in fine detail which allows us to understand and exploit these models in new ways.

For denoising diffusion models, recent work has explored how *attention* can highlight how models depends on different words during generation [244, 151, 282, 248, 260, 278, 155, 82]. Our information-theoretic approach diverges from attention-based methods in three significant ways. First, attention requires not just white-box access but also dictates a particular network design. Our approach abstracts away from architecture details, and may be useful in the increasingly common scenario where we interact with large generative models only through black-box API access [194]. Second, while attention is engineered toward specific tasks such as segmentation [244] and image-text matching [82], our information estimators can adapt to diverse applications. As an illustrative example, in §3.4.1 we automate the evaluation

of compositional understanding for Stable Diffusion [204]. Third, information flow as a dependence measure better captures the effects of interventions. Attention within a neural network does not necessarily imply that the final output depends on the attended input. Our Conditional Mutual Information (CMI) estimator correctly reflects that a word with small CMI will not affect the output (Fig. 3.1).

On the other hand, estimating MI from samples remains a fundamental challenge with broad applications. Early methods can suffer from sample complexity or variance that scales exponentially with the true MI [70, 161, 254], making reliable estimation difficult when MI is unknown. To address this, [225] proposed three self-consistency tests for assessing estimator robustness, and [43] introduced a benchmark of 40 synthetic datasets from diverse distributions to standardize evaluation. Recently, MINDE [66], a score-based diffusion MI estimator, achieved an 87.5% success rate on the benchmark and passed all self-consistency tests. Building on these advances, our information-theoretic diffusion framework provides an even more principled and effective approach to MI estimation, delivering strong empirical performance across extensive experiments.

We summarize our main contributions in this chapter below:

- We show that denoising diffusion models directly provide a natural and tractable way to decompose information in a fine-grained way, distinguishing relevant information at a per-sample (image) and per-variable (pixel) level. The utility of information decomposition is validated on a variety of tasks below.

- We provide a better quantification of compositional understanding capabilities of diffusion models. We find that on the ARO benchmark [274], diffusion models are significantly underestimated due to sub-optimal alignment scores.
- We examine how attention and information in diffusion models localize specific text in images. While neither exactly align with the goal of object segmentation, information measures more effectively localize abstract words like adjectives, adverbs, and verbs.
- How does a prompt intervention modify a generated image? It is often possible to surgically modify real images using prompt intervention techniques, but sometimes these interventions are completely ignored. We show that CMI is more effective at capturing the effects of intervention, due to the ability to take contextual information into account.
- We find that integrating half the gap between the conditional and unconditional diffusion MMSE curves over all noise levels yields the Mutual Information via Neural Diffusion (MIND) estimator, which achieves superior accuracy over variational approaches, particularly in high-mutual information regimes.

3.2 Interpretable Diffusion via Information Decomposition

3.2.1 Information-theoretic Perspective on Diffusion Models

A diffusion model can be seen as a noisy channel that takes samples from the data distribution, $\mathbf{x} \sim p(X = \mathbf{x})$, and progressively adds Gaussian noise, $\mathbf{z} \equiv \sqrt{\sigma(\alpha)}\mathbf{x} + \sqrt{\sigma(-\alpha)}\epsilon$, with $\epsilon \sim \mathcal{N}(0, \mathbb{I})$ (a variance preserving Gaussian channel with log SNR, α , using the standard sigmoid function). By learning to reverse or *denoise* this noisy channel, we can generate samples from the original distribution [224], a result with remarkable applications [194]. The

Gaussian noise channel has been studied in information theory since its inception [217]. A decade before diffusion models appeared in machine learning, [77] demonstrated that the information in this Gaussian noise channel, $I(X; X_\alpha)$, is *exactly* related to the mean square error for optimal signal reconstruction. This result was influential because it demonstrated for the first time that *information-theoretic* quantities could be related to *estimation of optimal denoisers*. In this paper, we are interested in extending this result to other mutual information estimators, and to *pointwise* estimates of mutual information. Our focus is not on learning the reverse or denoising process for generating samples, but instead to *measure relationships* using information theory.

For our results, we require the optimal denoiser, or Minimum Mean Square Error (MMSE) denoiser for predicting ϵ at each noise level, α .

$$\hat{\epsilon}(\mathbf{x}) \equiv \arg \min_{\bar{\epsilon}(\cdot)} \mathbb{E}_{p(\mathbf{x}), p(\epsilon)} [\|\epsilon - \bar{\epsilon}(\mathbf{z})\|_2^2] \quad (3.1)$$

Note that we predict the noise, ϵ , but could equivalently predict $\mathbf{x}$. This optimal denoiser is exactly what diffusion models are trained to estimate. Instead of using denoisers for generation, we will see how to use them for measuring information-theoretic relationships.

For the denoiser in Eq. 3.1, the following expression holds exactly.

$$-\log p(\mathbf{x}) = 1/2 \int \mathbb{E}_{p(\epsilon)} [\|\epsilon - \hat{\epsilon}(\mathbf{z})\|_2^2] d\alpha + \text{const} \quad (3.2)$$

The value of the constant, *const*, will be irrelevant as we proceed to build Mutual Information (MI) estimators and a decomposition from this expression. This expression shows that

solving a denoising *regression* problem (which is easy for neural networks) is *equivalent* to density modeling. No differential equations need to be referenced or solved to make this exact connection, unlike the approaches appearing in [234] and [160]. The derivation of this result in [128] closely mirrors [77]’s original result, and is shown in App. B.1 for completeness. This expression is extremely powerful and versatile for deriving fine-grained information estimators, as we now show.

3.2.2 Mutual Information and Pointwise Estimators

Note that equation 3.2 also holds with arbitrary conditioning. Let $\mathbf{x}, \mathbf{y} \sim p(X = \mathbf{x}, Y = \mathbf{y})$ and $\hat{\epsilon}(\mathbf{z}|\mathbf{y})$ be the optimal denoiser for $p(\mathbf{x}|\mathbf{y})$ as in equation 3.1. Then we can write the conditional density as follows.

$$-\log p(\mathbf{x}|\mathbf{y}) = 1/2 \int \mathbb{E}_{p(\epsilon)} [\|\epsilon - \hat{\epsilon}(\mathbf{z}|\mathbf{y})\|_2^2] d\alpha + \text{const} \quad (3.3)$$

This directly leads to an estimate of the following useful Log Likelihood Ratio (LLR).

$$\log p(\mathbf{x}|\mathbf{y}) - \log p(\mathbf{x}) = 1/2 \int \mathbb{E}_{p(\epsilon)} [\|\epsilon - \hat{\epsilon}(\mathbf{z})\|_2^2 - \|\epsilon - \hat{\epsilon}(\mathbf{z}|\mathbf{y})\|_2^2] d\alpha \quad (3.4)$$

The LLR is the integrated reduction in MMSE from conditioning on auxiliary variable, $\mathbf{y}$.

The mutual information, $I(X; Y)$ can be defined via this LLR:

$$I(X; Y) \equiv \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} [\log p(\mathbf{x}|\mathbf{y}) - \log p(\mathbf{x})].$$

We write MI using information theory notation, where capital X, Y are used to refer to functionals of random variables with the distributions $p(\mathbf{x}, \mathbf{y})$ [41]. MI is an average measure of dependence, but we are often interested in the strength of a relationship for a single point, or *pointwise information*. Pointwise information for a specific $\mathbf{x}, \mathbf{y}$ is sometimes written with lowercase as $\mathbf{i}(\mathbf{x}; \mathbf{y})$ and is defined so that the average recovers MI [63]. [60] referred to what we call “mutual information” as “average mutual information”, and considered what we call pointwise mutual information to be the more fundamental quantity. Pointwise information has been especially influential in NLP [136].

$$I(X; Y) = \mathbb{E}_{p(\mathbf{x}, \mathbf{y})}[\mathbf{i}(\mathbf{x}; \mathbf{y})] \quad \text{Defining property of pointwise information}$$

Pointwise information is not unique and both quantities below satisfy this property.

$$\begin{aligned} \mathbf{i}^s(\mathbf{x}; \mathbf{y}) &\equiv 1/2 \int \mathbb{E}_{p(\epsilon)} [\|\epsilon - \hat{\epsilon}(\mathbf{z})\|_2^2 - \|\epsilon - \hat{\epsilon}(\mathbf{z}|\mathbf{y})\|_2^2] d\alpha \\ \mathbf{i}^o(\mathbf{x}; \mathbf{y}) &\equiv 1/2 \int \mathbb{E}_{p(\epsilon)} [\|\hat{\epsilon}(\mathbf{z}) - \hat{\epsilon}(\mathbf{z}|\mathbf{y})\|_2^2] d\alpha \end{aligned} \tag{3.5}$$

The first **standard definition** comes from using $\log p(\mathbf{x}|\mathbf{y}) - \log p(\mathbf{x})$ written via equation 3.4.

The second more compact definition is derived using the **orthogonality principle** in §B.2. $\mathbf{i}^s$ has higher variance due to the presence of extra ϵ terms, while $\mathbf{i}^o$ has lower variance and is always non-negative. We will explore both estimators, but find the lower variance version that exploits the orthogonality principle is generally more useful (see §B.3.2).

Note that while (average) MI is always non-negative, pointwise MI can be negative as

$\mathbf{i}^s(\mathbf{x}; \mathbf{y}) = \log p(\mathbf{x}|\mathbf{y}) - \log p(\mathbf{x}) < 0$ occurs when the observation of $\mathbf{y}$ makes $\mathbf{x}$ appear *less* likely. We can say negative pointwise information signals a “misinformative” observation [63].

All these expressions can be given conditional variants, where we condition on a random variable, C , defining the context. CMI and its pointwise expression can be related as $I(X; Y|C) = \mathbb{E}_{p(\mathbf{x}, \mathbf{y}, \mathbf{c})}[\mathbf{i}(\mathbf{x}; \mathbf{y}|\mathbf{c})]$. The pointwise versions of equation 3.5 can be obtained by conditioning all the denoisers on C , e.g., $\hat{\epsilon}(\mathbf{z}|\mathbf{y}) \rightarrow \hat{\epsilon}(\mathbf{z}|\mathbf{y}, \mathbf{c})$.

3.2.3 Pixel-wise Information Decomposition

The pointwise information, $\mathbf{i}(\mathbf{x}; \mathbf{y})$, does not tell us which variables, x_j ’s, are informative about which variables, y_k . If $\mathbf{x}$ is an image and $\mathbf{y}$ represents a text prompt, then this would tell us which parts of the image a particular word is informative about. One reason that information decomposition is highly nontrivial is that scenarios can arise where information in variables is synergistic, for example [267]. Our decomposition instead proceeds from the observation that the correspondence between information and MMSE leads to a natural decomposition of information into a sum of terms for each variable. If $\mathbf{x} \in \mathbb{R}^n$, we can write $\mathbf{i}(\mathbf{x}; \mathbf{y}) = \sum_{j=1}^n \mathbf{i}_j(\mathbf{x}; \mathbf{y})$ with:

$$\begin{aligned} \mathbf{i}_j^s(\mathbf{x}; \mathbf{y}) &\equiv 1/2 \int \mathbb{E}_{p(\epsilon)} [(\epsilon - \hat{\epsilon}(\mathbf{x}_\alpha))_j^2 - (\epsilon - \hat{\epsilon}(\mathbf{x}_\alpha|\mathbf{y}))_j^2] d\alpha \\ \mathbf{i}_j^o(\mathbf{x}; \mathbf{y}) &\equiv 1/2 \int \mathbb{E}_{p(\epsilon)} [(\hat{\epsilon}(\mathbf{x}_\alpha) - \hat{\epsilon}(\mathbf{x}_\alpha|\mathbf{y}))_j^2] d\alpha \end{aligned} \tag{3.6}$$

In other words, both variations of pointwise information can be written in terms of squared errors, and we can decompose the squared error into the error for each variable. For

pixel-wise information for images with multiple channels, we sum over the contribution from each channel.

We can easily extend this for conditional information. Let $\mathbf{x}$ represent a particular image, $\mathbf{y} = \{y_*, \mathbf{c}\} = \{\text{“object”}, \text{“a person holding an _”}\}$. Then we can estimate $\mathbf{i}_j(\mathbf{x}; y_* | \mathbf{c})$, which represents the information that word y_* has about variable $\mathbf{x}_j$, conditioned on the context, $\mathbf{c}$. To get estimates using equation 3.6, we just add conditioning on $\mathbf{c}$ on both sides. Denoising images conditioned on arbitrary text is a standard task for diffusion models. An example of the estimator is shown in Fig. 3.1, where the highlighted region represents the pixel-wise value of $\mathbf{i}_j^o(\mathbf{x}; y_* | \mathbf{c})$ for pixel j .

3.2.4 Numerical Information Estimates

All information estimators we have introduced require evaluating a one-dimensional integral over an infinite range of SNRs. To estimate in practice we use importance sampling to evaluate the integral as in [121, 128]. We use a truncated logistic for the importance sampling distribution for α . Empirically, we find that contributions to the integral for both very small and very large values of α are close to zero, so that truncating the distribution has little effect. Unlike MINE [14] or variational estimators [186] the estimators presented here do not depend on optimizing a direct upper or lower bound on MI. Instead, the estimator depends on finding the MMSE of both the conditional and unconditional denoising problems, and then combining them to estimate MI using equation 3.4. However, these two MMSE terms appear with opposite signs. In general, any neural network trained to minimize MSE may not achieve the global minimum for either or both terms, so we cannot guarantee that the estimate is either an upper or lower bound. In practice, neural networks excel at

regression problems, so we expect to achieve reasonable estimates. In all our results we use pretrained diffusion models which have been trained to minimize mean square under Gaussian noise, as required by equation 3.1 (different papers differ in how much weight each α term receives in the objective, but in principle the MMSE for each α is independent, so the weighting shouldn't strongly affect an expressive enough neural network).

3.3 Mutual Information Estimation via Neural Diffusion

From Eq. 3.4, we know the pointwise MI is

$$\log p(\mathbf{x}|y) - \log p(\mathbf{x}) = 1/2 \int_0^\infty d\gamma (\text{mmse}(\mathbf{x}|\gamma) - \text{mmse}(\mathbf{x}|\gamma, y)) \quad (3.7)$$

Therefore, we can train expert denoisers to estimate the MMSE for various distributions of $q(\gamma)$, as different density consistently lead to distinct distributions of the MMSE gap in Figure 3.2. It also means the Mutual Information could be estimated by two Neural Diffusion processes (MIND).

$$\boxed{I(\mathbf{x}; y) = 1/2 \mathbb{E}_{\gamma, x, y} [(\text{mmse}_x(\gamma) - \text{mmse}_{x|y}(\gamma)) / q(\gamma)]} \quad (3.8)$$

3.3.1 Model Training

In practice, we parametrize the denoising function in terms of a neural network. At training time, we have to solve two minimization problems, or actually a continuum of

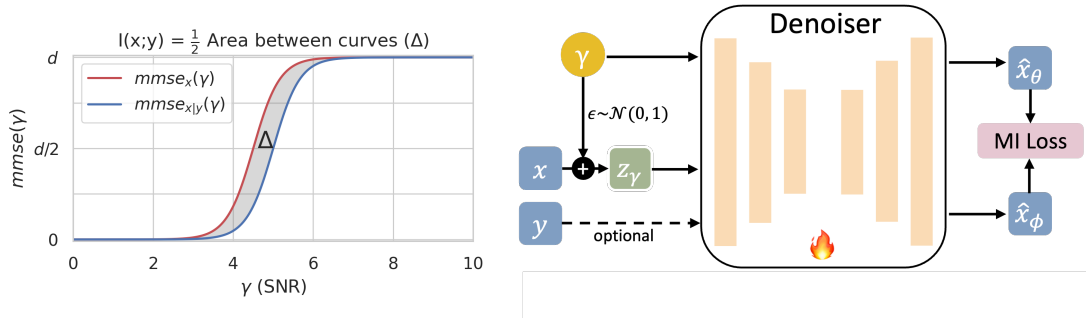


Figure 3.2: The mutual information is exactly half the area between MMSE curves. Figure 3.3: The schematic of the MIND training process. Noise at a given γ level is added to the data $\mathbf{x}$, and a denoiser is used to recover it, with or without conditioning on y at a 50% probability. Finally, the MI loss, as defined numerically integrate to get an estimate in Eq. 3.9, is computed to backpropagate the gradient.

minimization problems for each SNR level.

$$\theta^* = \arg \min_{\theta} \mathbb{E}_{p(\mathbf{z}|\mathbf{x})p(\mathbf{x},y)}[(\mathbf{x} - \hat{\mathbf{x}}_\theta(\mathbf{z}, \gamma, y))^2] \quad (3.9)$$

$$\phi^* = \arg \min_{\phi} \mathbb{E}_{p(\mathbf{z}|\mathbf{x})p(\mathbf{x})}[(\mathbf{x} - \hat{\mathbf{x}}_\phi(\mathbf{z}, \gamma))^2]$$

We parametrize $\hat{\mathbf{x}}_\theta(\mathbf{z}, \gamma, y)$ as a neural network that is trained with a mean square error loss to recover the signal, $\mathbf{x}$, from noise and the auxiliary signal, y , across different SNRs, γ (see Figure 3.3). In principle we should train a separate neural network that recovers data from noise without conditioning on y . Conditional diffusion models have shown compelling results in [169, 194, 206]. During training we simply replace y with a null value, $y = \emptyset$, some fraction of the time so that the same (conditional) network can learn to denoise in the absence of conditioning.

In practice, we can't guarantee that the neural network finds the true global optimum, the MMSEs in Eq. 3.8. Because MMSEs appear with both signs in that expression,

we can’t even guarantee that our network gives an upper or lower bound. While this is unfortunate, conventional wisdom suggests that sufficiently expressive neural networks do converge to global optima [55, 103].

3.3.2 Adaptive Importance Sampling

The core of the experiment is training a denoiser capable of effectively performing denoising across different noise levels, characterized by varying SNRs, to simulate an MMSE curve. The MI is then computed from the gap between the conditional and unconditional MMSE curves (Eq. 3.8). This MMSE-based theoretical approach offers greater flexibility to enhance the accuracy and robustness of MI estimation.

As in [127], we estimate integrals using importance weighted Monte Carlo estimators, with log-SNR values chosen from a logistic distribution with some location, μ and scale, σ . Since the shape of the integration area (Δ area in Figure 3.2) is data-dependent, a fixed sampling distribution can be inefficient. For our adaptive variants, we therefore optimize this distribution for each specific task. To find the data-adapted parameters (μ, σ) , we employ a two-stage procedure. First, we train a preliminary model and analyze its conditional MMSE curve, $\text{MMSE}_{x|y}$. Inspired by the use of error landmarks in density estimation [127], our heuristic identifies the critical transition region of this curve, where the denoiser becomes effective. Specifically, we define the parameters as follows (where d is the data dimensionality):

- The location μ is set to the log-SNR where the MMSE curve crosses the $d/2$ error threshold.

This centers our sampling distribution on the midpoint of the denoiser’s transition from high to low error.

- The scale σ is derived from the log-SNR where the curve crosses the $d/4$ threshold, capturing the steepness of this transition. For example, it can be set as the difference between the log-SNRs at the $d/2$ and $d/4$ crossings.

The final adaptive estimators are then trained with this optimized distribution, targeting the critical SNR range to yield more accurate and efficient MI estimates.

3.4 Results

Section 3.2 establishes a precise connection between optimal denoisers and information. For experiments we consider diffusion models as approximating optimal denoisers which can be used to estimate information. All our experiments are performed with pre-trained latent space diffusion models [204], Stable Diffusion v2.1 from Hugging Face unless otherwise noted. Latent diffusion models use a pre-trained autoencoder to embed images in a lower resolution space before doing typical diffusion model training. We always consider $\mathbf{x}$ as the image in this lower dimensional space, but in displayed images we show the images after decoding, and we use bilinear interpolation when visualizing heat maps at the higher resolution. See §B.4 for experiment details and links to open source code.

3.4.1 Relation Testing with Pointwise Information

First, we consider information decomposition at the “pointwise” or per-image level. We make use of our estimator to compute summary statistics of an image-text pair and quantify qualitative differences across samples. As a novel application scenario, we apply our pointwise estimator to analyze compositional understanding of Stable Diffusion on the

ARO benchmark [274]. Referring readers to [274] for more detailed descriptions, the ARO benchmark is a suite of discriminative tasks wherein a VLM is commissioned to align an image $\mathbf{x}$ with its ground-truth caption $\mathbf{y}$ from a set of perturbations $\mathcal{P} = \{\tilde{\mathbf{y}}_j\}_{j=1}^M$ induced by randomizing constituent orders from $\mathbf{y}$. The compositional understanding capability of a VLM is thus measured by the accuracy:

$$\mathbb{E}_{\mathbf{x}, \mathbf{y}}[1(\mathbf{y} = \arg \max_{\mathbf{y}' \in \{\mathbf{y}\} \cup \mathcal{P}} s(\mathbf{x}, \mathbf{y}'))],$$

where $s : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ is an alignment score. For contrastive VLMs, s is chosen to be the cosine similarity between encoded image-text representations. We choose $s(\mathbf{x}, \mathbf{y}) \equiv \mathbf{i}^o(\mathbf{x}; \mathbf{y})$ as our score function for diffusion models. In contrast, [82] compute s as an aggregate of latent attention maps, while [131] adopt the negative MMSE. In Table 3.1, we report performances of OpenCLIP [101] and Stable Diffusion 2.1 [204] on the ARO benchmark, while controlling model checkpoints with the same text encoder for fair comparison.

Table 3.1: Accuracy (%) of Stable Diffusion and its OpenCLIP backbone on the ARO Benchmark. $\star$: conducts additional fine-tuning with compositional-aware hard negatives.

Method	VG-A	VG-R	COCO	Flickr30k
OpenCLIP [101]	64.6	51.4	32.8	40.5
DiffITM [131]	62.9	50.0	23.5	33.2
HardNeg-DiffITM $\star$ [131]	67.6	52.3	34.4	48.6
Info. (Ours)	72.0	69.1	40.1	49.3

We observe that Stable Diffusion markedly improves in compositional understanding over OpenCLIP. Since the text encoder is frozen, we can attribute this improvement solely to the denoising objective and the visual component. More importantly, our information

estimator significantly outperforms MMSE [131], decidedly proving that previous works underestimate compositional understanding capabilities of diffusion models. Our observation provides favorable evidence for adapting diffusion models for discriminative image-text matching [82]. However, these improvements are smaller compared to contrastive pre-training approaches that make use of composition-aware negative samples [274].

In Table 3.2 we report a subset of fine-grained performances across relation types, highlighting those with over 30% performance improvement in **green** and those with over 30% performance decrease in **magenta**. Interestingly, our highlighted categories correlate well with performance changes incurred by composition-aware negative training, despite using different CLIP backbones (c.f. Table 2 of [274]). Most improvements are attributed to verbs that associate subjects with conceptually distinctive objects (e.g. “*A boy sitting on a chair*”). Our observation suggests that improvements in compositional understanding may stem predominantly from these “low-hanging fruits”, and partially corroborate the hypothesis in [198], which posits that incorrect associations between entities and their visual attributes are attributed to the text encoder’s inability to encode linguistic structure. We refer readers to §B.4.1 for complete fine-grained results, implementation details, and error analysis.

Table 3.2: Selected results on VG-R.

	Info. (↑)	OpenCLIP (↑)
Accuracy (%)	69.1	51.4
beneath	50.0	90.0
covered in	14.3	50.0
feeding	50.0	100.0
grazing on	60.0	30.0
sitting on	78.9	49.7
wearing	84.1	44.9

3.4.2 Pixel-wise Information and Word Localization

Next, we explore “pixel-wise information” that words in a caption contain about specific pixels in an image. According to §3.2.2 and §3.2.3, it naturally leads us to consider two potential approaches for validating this nuanced relationship. The first one entails concentrating solely on the mutual information between the object word and individual pixels. The second approach involves investigating this mutual information when given the remaining context in the caption, i.e., conditional mutual information. As the success of our experiment relies heavily on the alignment between images and text, we carefully filtered two datasets, COCO-IT and COCO-WL from the MSCOCO [144] validation dataset. For specific details about dataset construction, please refer to §B.5. Meanwhile, we also provide an image-level information analysis on these two datasets in §B.3.1, and visualize the diffusion process and its relation to information for 10 cases in §B.3.2.

Visualize mutual information and conditional mutual information

Given an image-text pair $(\mathbf{x}, \mathbf{y})$ where $\mathbf{y} = \{y_*, \mathbf{c}\}$, we compute pixel-level mutual information as $i_j^o(\mathbf{x}; y_*)$, and the conditional one as $i_j^o(\mathbf{x}; y_* | \mathbf{c})$ for pixel j , from Eq. 3.6. Eight cases are displayed in Fig. 3.4, and we put more examples in §B.3.3 and §B.3.4.

In Fig. 3.4, the visualizations of pixel-level MI and CMI are presented using a common color scale. Upon comparison, it becomes evident that CMI has a greater capacity to accentuate objects while simultaneously diminishing the background. This is attributed to the fact that MI solely computes the relationship between noun word y_* and pixels, whereas CMI factors out context-related information from the complete prompt-to-pixel information.

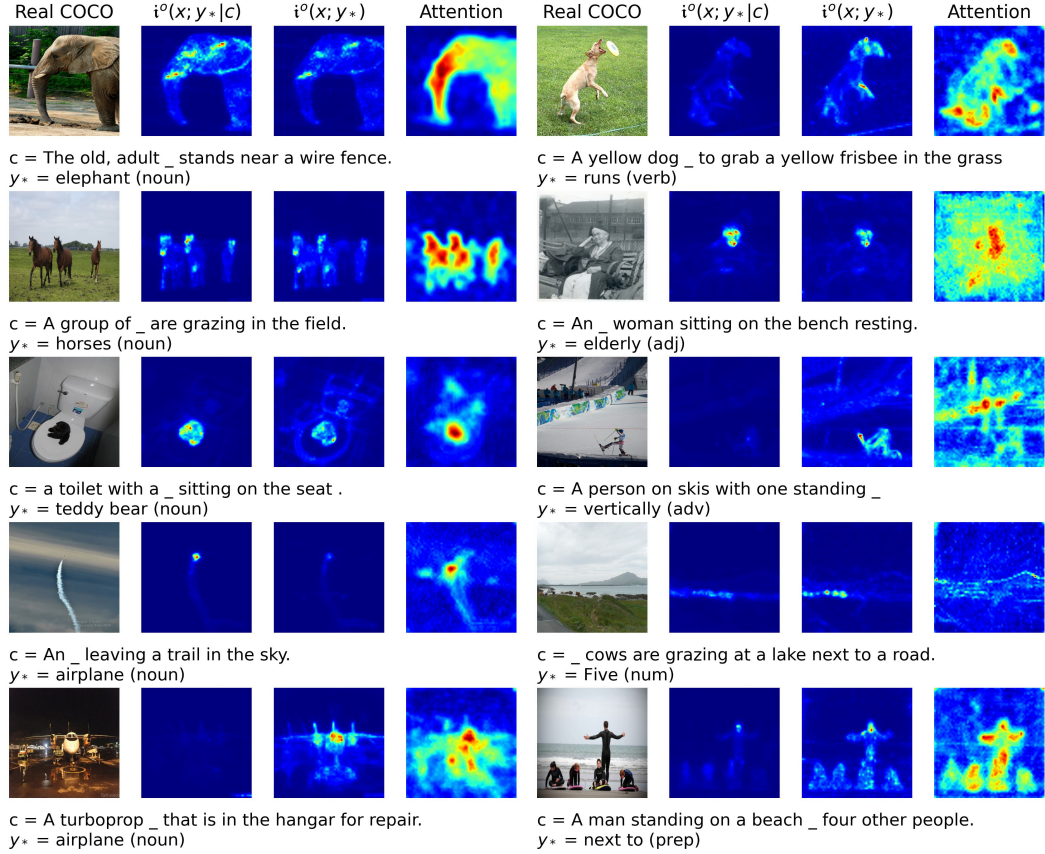


Figure 3.4: Examples of localizing different types of words in images. The left half presents noun words, while the right half displays abstract words.

However, there are occasional instances of the opposite effect, as observed in the example where ‘ $y_* = \text{airplane}$ ’ (left bottom in Fig. 3.4). In this case, it appears that CMI fails to highlight pixels related to ‘airplane’, while MI succeeds. This discrepancy arises from the presence of the word ‘turboprop’ in the context. Therefore the context, c , accurately describes the content in the image and ‘airplane’ adds no additional information. Compared to attention, CMI and MI qualitatively appear to focus more on fine details of objects, like the eyes, ears, and tusks of an elephant, or the face of horses. For object words, we can use a segmentation task to make more quantitative statements in §3.4.2.

We also explore whether pixel-level CMI or MI can provide intriguing insights for other types of entities. Fig. 3.4 (right) presents four different types of entities besides nouns, including verbs, adjectives, adverbs, numbers and prepositions. These words are quite abstract, and even through manual annotation, it can be challenging to precisely determine the corresponding pixels for them. The visualization results indicate that MI gives intuitively plausible results for these abstract items, especially adjectives and adverbs, that highlight relevant finer details of objects more effectively than attention. Interestingly, MI’s ability to locate abstract words within images aligns with the findings presented in §3.4.1 regarding relation testing (Table B.6).

Localizing word information in images

The visualizations provided above offer an intuitive demonstration of how pixel-wise CMI relates parts of an image to parts of a caption. Therefore, our curiosity naturally extends to whether this approach can be applied to word localization within images. Currently, the prevalent evaluation involves employing attention layers within Stable Diffusion models for object segmentation [244, 248, 260]. These methodologies heavily rely on attention layers and meticulously crafted heuristic heatmap generation. It’s worth highlighting that during image generation, the utilization of multi-scale cross-attention layers allows for the rapid computation of the Diffuse Attention Attribution Map (DAAM) [244]. This not only facilitates segmentation but also introduces various intriguing opportunities for word localization analyses. We opted for DAAM as our baseline choice due to its versatile applicability, and the detailed experimental design is documented in the §B.4.2. We use

Table 3.3: Unsupervised Object Segmentation mIoU (%) Results on COCO-IT

Method	50 steps	100 steps	200 steps
Whole Image Mask	14.94	14.94	14.94
Attention [244]	34.52	34.90	35.35
Conditional Mutual Info.	32.31	33.24	33.63
Attention+Information	42.46	42.71	42.84

mean Intersection over Union (mIoU) as the evaluation metric for assessing the performance of pixel-level object segmentation. Table 3.3 illustrates that, in contrast to attention-based methods, pixel-wise CMI proves less effective for object segmentation, with an error analysis appearing in Table B.5. The attention mechanism in DAAM combines high and low resolution image features across the multi-scale attention layers, akin to Feature Pyramid Networks (FPN) [143], facilitating superior feature fusion. CMI tends to focus more on the specific details that are unique to the target object rather than capturing the overall context of it. Although pixel-level CMI did not perform exceptionally well in object segmentation, the results from ‘Attention+Information’ clearly demonstrate that the information-theoretic diffusion process [128] does enhance the capacity of attention layers to capture features.

Discussion: attention versus information versus segmentation Looking at the heatmaps, it is clear that some parts of an object can be more informative than others, like faces or edges. Conversely, contextual parts of an image that are not part of an object can still be informative about the object. For instance, we may identify an airplane by its vapor trail (Fig. 3.4). Hence, we see that neither attention nor information perfectly aligns with the goal of segmentation. One difference between attention and mutual information is that when a pixel pays “attention” to a word in the prompt, it *does not* imply that modifications

to the word would modify the prompt. This is best highlighted with conditional mutual information, where we see that an informative word (“jet”) may contribute little information in a larger context (“turboprop jet”). To highlight this difference, we propose an experiment where we test whether intervening on words changes a generated image. Our hypothesis is that if a word has low CMI, then its omission should not change the result. For attention, on the other hand, this is not necessarily true.

3.4.3 Selective Image Editing via Prompt Intervention

Diffusion models have gained widespread adoption by providing non-technical users with a natural language interface to create diverse, realistic images. Our focus so far has been on how well diffusion models understand structure in *real* images. We can connect real and generated images by studying how well we can *modify* a real image, which is a popular use case for diffusion models discussed in §3.5. We can validate our ability to measure informative relationships by seeing how well the measures align with effects under prompt intervention.

For this experiment, we adopt the perspective that diffusion models are equivalent to continuous, invertible normalizing flows [234]. In this case, the denoising model is interpreted as a score function, and an ODE depending on this score function smoothly maps the data distribution to a Gaussian, or vice versa. The solver we use for this ODE is the 2nd order deterministic solver with 100 steps from [112]. We start with a real image and prompt, then use the (conditional) score model to map the image to a Gaussian latent space. In principle, this mapping is invertible, so if we reverse the dynamics we will recover the original image. We see in §B.3.5 that the original image is almost always recovered with high fidelity.

Next, we consider adding an intervention. While doing the reverse dynamics, we *modify* the (conditional) score by changing the denoiser prompt in some way. We focus in experiments on the effects of omitting a word, or swapping a word for a categorically similar word (“bear” $\rightarrow$ “elephant”, for example). Typically, we find that much of the detail in the original image is preserved, with only the parts that relate to the modified word altered. Interestingly, we also find that in some cases interventions to a word are seemingly ignored (see Fig. 3.1, 3.5).

We want to explore whether attention or information measures predict the effect of interventions. When intervening by omitting a word, we consider the conditional (pointwise) mutual information $i(\mathbf{x}; y | \mathbf{c})$ from equation 3.5, and for word swaps we use a difference of CMIs with y, y' representing the swapped word, as in Fig. 3.1. For attention, we aggregate the attention corresponding to a certain word during generation using the code from [244]. We find that a word can be ignored *even if the attention heatmap highlights an object related to the word*. In other words, attention to a word does not imply that it affects the generated outcome. One reason for this is that a word may provide little information beyond that in the context. For instance, in Fig. 3.5, a woman in a hospital is assumed to be on a bed, so omitting this word has no effect. CMI, on the other hand, correlates well with the effect of intervention, and “bed” in this example has low CMI.

To quantify our observation that conditional mutual information better correlates with the effect of intervention than attention models, we measure the Pearson correlation between a score (CMI or attention heatmap) and the effect of the intervention using L2 distance between images before and after intervention. To get an image-level correlation, we

correlated the aggregate scores per image across examples on COCO100-IT. We also consider the pixel-level correlations between L2 change per pixel and metrics, CMI and attention heatmaps. We average these correlations over all images and report the results in Table 3.4. CMI is a much better predictor of changes at the image-level, which reflects the fact that it directly quantifies what additional information a word contributes after taking context into account. At the per-pixel level, CMI and attention typically correlated very well when changes were localized, but both performed poorly in cases where a small prompt change led to a global change in the image. Results visualizing this effect are shown along with additional experiments with word swap interventions in §B.3.5. Small dependence, as measured by information, correctly implied small effects from intervention. Large dependence, however, can lead to complex, global changes due to the nonlinearity of the generative process.

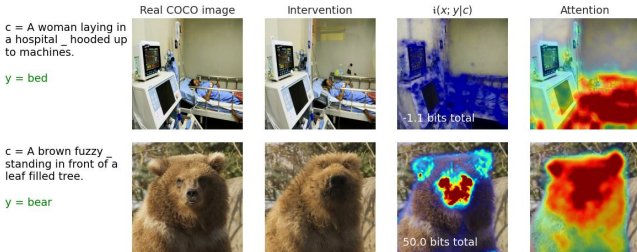


Table 3.4: Pearson Correlation with Image Change

	$i(x; y c)$	Attention
Image-level	0.34 ($\pm .010$)	0.24 ($\pm .011$)
Pixel-level	0.27 ($\pm .002$)	0.21 ($\pm .002$)

Figure 3.5: COCO images edited by omitting words from the prompt. Conditional mutual information better reflects the actual changes in the image after intervention. Table shows Pearson correlation between metrics (CMI or attention) versus L2 changes in image after intervention, at the image-level and pixel-level as discussed in §3.4.3.

3.4.4 40 MI Estimation Benchmark

We evaluate on the benchmark of [43], which combines base distributions (Uniform, Normal with dense or sparse correlation, and long-tailed Student-t) with MI-preserving trans-

formations (Half-Cube, Asinh, Swiss-roll, Spiral). This setup introduces high dimensionality, heavy tails, sparsity, and non-linear geometry. We compare MIND against neural estimators, such as MINE [14], INFONCE [175], NWJ [168], DOE [161] and MINDE [65].

Table 3.5: Low MI estimation over 10 seeds using $N = 10k$ test samples against ground truth (GT) [43]. All methods were trained with 100k samples. Color indicates relative negative (red) and positive bias (blue). Blank entries indicate numerical instabilities. Abbreviations: Mn =Multinormal, St =Student-t, Nm =Normal, Hc =Half-cube, Sp =Spiral.

GT	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.2	0.4	0.2	0.3	0.2	0.4	0.3	0.4	1.7	0.3	0.4		
MINE	0.2	0.4	0.2	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	0.9	0.9	0.9	0.8	0.7	0.8	0.9	0.9	0.9	0.0	0.0	0.1	0.1	0.1	0.2	0.2	0.4	1.7	0.3	0.4
InfoNCE	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	0.9	1.0	1.0	0.8	0.8	0.8	0.9	1.0	1.0	0.2	0.3	0.2	0.3	0.2	0.4	0.3	0.4	1.7	0.3	0.4
D-V	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	0.9	1.0	1.0	0.8	0.8	0.8	0.9	1.0	1.0	0.0	0.0	0.1	0.1	0.2	0.2	0.2	0.4	1.7	0.3	0.4
NWJ	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	0.9	1.0	1.0	0.8	0.8	0.8	0.9	1.0	1.0	0.0	0.0	0.0	0.0	0.1	0.1	0.2	0.4	1.7	0.3	0.4
DoE(Gaussian)	0.2	0.5	0.3	0.6	0.4	0.4	0.4	0.7	1.0	1.0	1.0	0.4	0.7	7.8	1.0	0.6	0.9	1.3		0.4	0.7	1.0	1.0	0.5	0.6	0.6	0.6	0.7	0.8	0.8	6.7	7.9	1.8	2.5	0.6	4.2	1.2	1.6	0.1	0.4
DoE(Logistic)	0.1	0.4	0.2	0.4	0.4	0.4	0.4	0.6	0.9	0.9	1.0	0.3	0.7	7.8	1.0	0.6	0.9	1.3		0.4	0.8	1.1	1.0	0.5	0.6	0.6	0.7	0.8	0.8		2.0	0.5	0.8	0.3	1.5	0.6	1.6	0.1	0.4	
MINDE-J ($\sigma = 1$)	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.1	1.0	1.0	1.0	0.3	0.9	1.2	1.0	0.4	1.0	0.6	1.7	0.4	1.0	1.0	1.0	0.9	0.9	0.9	1.0	0.9	1.0	0.2	0.4	0.2	0.3	0.2	0.5	0.3	0.5	1.6	0.3	0.4
MINDE-J	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.2	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.7	0.4	1.1	1.0	1.0	1.0	0.9	0.9	1.1	1.0	1.0	0.1	0.2	0.2	0.3	0.2	0.5	0.3	0.4	1.7	0.3	0.4
MINDE-C ($\sigma = 1$)	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	0.9	1.0	1.0	0.9	0.9	0.9	0.9	1.0	0.9	0.1	0.3	0.2	0.3	0.2	0.4	0.3	0.3	1.7	0.3	0.4
MINDE-C	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	1.0	1.0	1.0	0.9	0.9	0.9	1.0	1.0	1.0	0.1	0.3	0.2	0.3	0.2	0.4	0.3	0.4	1.7	0.3	0.4
MIND	0.2	0.4	0.3	0.5	0.4	0.4	0.4	0.9	1.1	0.9	1.0	0.2	0.9	1.2	1.0	0.4	1.0	0.6	1.4	0.5	1.0	1.0	1.0	1.1	1.0	1.0	1.2	1.0	1.0	0.2	0.3	0.2	0.3	0.2	0.4	0.2	0.3	1.7	0.3	0.4
MIND-dynamic	0.2	0.4	0.3	0.4	0.4	0.4	0.4	0.9	1.0	0.9	1.0	0.3	1.0	1.2	1.0	0.4	1.0	0.6	1.4	0.4	1.0	1.0	1.1	1.0	1.0	1.1	1.0	1.0	1.0	0.2	0.3	0.2	0.3	0.2	0.5	0.2	0.3	1.7	0.3	0.4
MIND-orthogonal	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	1.0	1.0	1.0	0.9	1.0	1.0	1.0	1.0	1.0	0.1	0.3	0.2	0.3	0.2	0.4	0.3	0.4	1.7	0.3	0.4
MIND-orthogonal-dynamic	0.2	0.4	0.3	0.4	0.4	0.4	0.4	1.0	1.0	1.0	1.0	0.3	1.0	1.3	1.0	0.4	1.0	0.6	1.6	0.4	1.0	1.0	1.0	0.9	1.0	1.0	1.0	1.0	1.0	0.1	0.3	0.2	0.3	0.2	0.4	0.3	0.4	1.7	0.3	0.4
Ashish @ S 1 × 1 (dof=1)	Swiss roll 2 × 1 Uniform 1 × 1 (additive noise=0.1) Uniform 1 × 1 (additive noise=0.75) Wiggly @ Bivariate Nm 1 × 1																																							
Ashish @ S 2 × 2 (dof=1)																																								
Ashish @ S 3 × 3 (dof=2)																																								
Ashish @ S 4 × 4 (dof=3)																																								
Ashish @ S 5 × 5 (dof=4)																																								
Bimodal 1 × 1	Swiss roll 2 × 1 Uniform 1 × 1 (additive noise=0.1) Uniform 1 × 1 (additive noise=0.75) Wiggly @ Bivariate Nm 1 × 1																																							
Bivariate Nm 1 × 1																																								
Hc @ Bivariate Nm 1 × 1																																								
Hc @ Mn 25 × 25 (2-pair)																																								
Hc @ Mn 3 × 3 (2-pair)																																								
Hc @ Mn 3 × 3 (2-pair)	Swiss roll 2 × 1 Uniform 1 × 1 (additive noise=0.1) Uniform 1 × 1 (additive noise=0.75) Wiggly @ Bivariate Nm 1 × 1																																							
Hc @ Mn 5 × 5 (2-pair)																																								
Hc @ Mn 2 × 2 (2-pair)																																								
Mn 2 × 2 (dense)																																								
Mn 25 × 25 (2-pair)																																								
Mn 25 × 25 (dense)	Swiss roll 2 × 1 Uniform 1 × 1 (additive noise=0.1) Uniform 1 × 1 (additive noise=0.75) Wiggly @ Bivariate Nm 1 × 1																																							
Mn 3 × 3 (2-pair)																																								
Mn 3 × 3 (dense)																																								
Mn 5 × 5 (2-pair)																																								
Mn 5 × 5 (dense)																																								
Mn 50 × 50 (dense)	Swiss roll 2 × 1 Uniform 1 × 1 (additive noise=0.1) Uniform 1 × 1 (additive noise=0.75) Wiggly @ Bivariate Nm 1 × 1																																							
Nm CDF @ Bivariate Nm 1 × 1																																								
Nm CDF @ Mn 25 × 25 (2-pair)																																								
Nm CDF @ Mn 3 × 3 (2-pair)																																								
Nm CDF @ Mn 5 × 5 (2-pair)																																								
Nm CDF @ Mn 5 × 5 (2-pair)	Swiss roll 2 × 1 Uniform 1 × 1 (additive noise=0.1) Uniform 1 × 1 (additive noise=0.75) Wiggly @ Bivariate Nm 1 × 1																																							
Sy @ Mn 25 × 25 (2-pair)																																								
Sy @ Mn 3 × 3 (2-pair)																																								
Sy @ Mn 5 × 5 (2-pair)																																								
Sy @ Nm CDF @ Mn 25 × 25 (2-pair)																																								
Sy @ Nm CDF @ Mn 3 × 3 (2-pair)	Swiss roll 2 × 1 Uniform 1 × 1 (additive noise=0.1) Uniform 1 × 1 (additive noise=0.75) Wiggly @ Bivariate Nm 1 × 1																																							
Sy @ Nm CDF @ Mn 5 × 5 (2-pair)																																								
Sy 1 × 1 (dof=1)																																								
Sy 2 × 2 (dof=1)																																								
Sy 2 × 2 (dof=2)																																								
Sy 3 × 3 (dof=2)	Swiss roll 2 × 1 Uniform 1 × 1 (additive noise=0.1) Uniform 1 × 1 (additive noise=0.75) Wiggly @ Bivariate Nm 1 × 1																																							
Sy 3 × 3 (dof=3)																																								
Sy 5 × 5 (dof=2)																																								
Sy 5 × 5 (dof=3)																																								
Swiss roll 2 × 1																																								
Uniform 1 × 1 (additive noise=0.1)	Swiss roll 2 × 1 Uniform 1 × 1 (additive noise=0.1) Uniform 1 × 1 (additive noise=0.75) Wiggly @ Bivariate Nm 1 × 1																																							
Uniform 1 × 1 (additive noise=0.75)																																								
Wiggly @ Bivariate Nm 1 × 1																																								

The results in Table 3.5 establish our method’s state-of-the-art performance. Our **MIND-orthogonal-adaptive** and **MIND-orthogonal** variants succeed on 39/40 and 37/40 tasks respectively, surpassing the MINDE (35/40). This robustness is rooted in our two main contributions: adaptive importance sampling ensures accuracy by focusing the integral on critical SNR regions, while the orthogonal principle guarantees a low-variance integrand for stability. This combination allows our method to excel on complex non-linear datasets where traditional methods fail.

3.4.5 High MI Estimation

To test the robustness and limits of our estimator, we extend the high-MI study from MINDE [66], pushing their experimental setup from a range of $MI \leq 5$ to the significantly more challenging regime of $MI \in [10, 15]$. Following their protocol, we use a sparse 3×3 Multinormal setup and its two MI-preserving transforms (Half-cube, Spiral). Figure 3.6 reports the mean $\pm$ one standard deviation over 10 seeds.

On the *original* and *Half-cube* cases (Fig. 3.6 a&b), **MIND-adaptive** remains the most accurate as MI grows, while other estimators falter due to distinct sources of error. **MINDE** struggles significantly as its score-matching objective requires approximating the sharp, high-frequency score functions of high-MI distributions, a task where neural networks' spectral bias leads to significant underestimation [193]. In contrast, our orthogonal variants exhibit a different limitation: a systematic conservative bias. This occurs because the orthogonal estimator measures the distance between neural network approximations of the denoisers, and the distance between these "smoothed-out" approximations is inherently smaller than the true distance between the optimal denoisers. This underestimation bias becomes more pronounced as the true MI gap grows larger.

3.4.6 MI Consistency Test

We conduct self-consistency tests inspired by [225] to evaluate the properties of MIND using high-dimensional real-world data, specifically samples from the MNIST dataset (28 $\times$ 28 resolution) [47]. Let A represent an image and B_r denote the image consisting of the top r rows of A . These tests are designed to verify whether the estimators adhere to

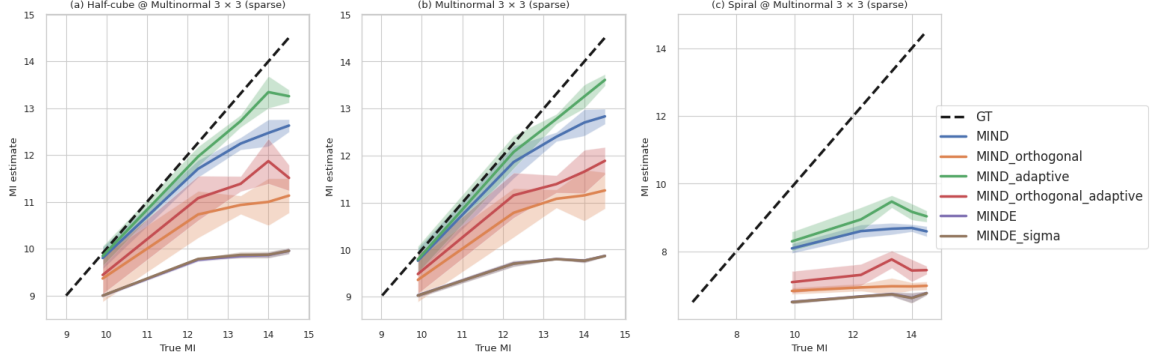


Figure 3.6: High MI benchmark: original (column (b)) and transformed variants (columns (a) and (c)).

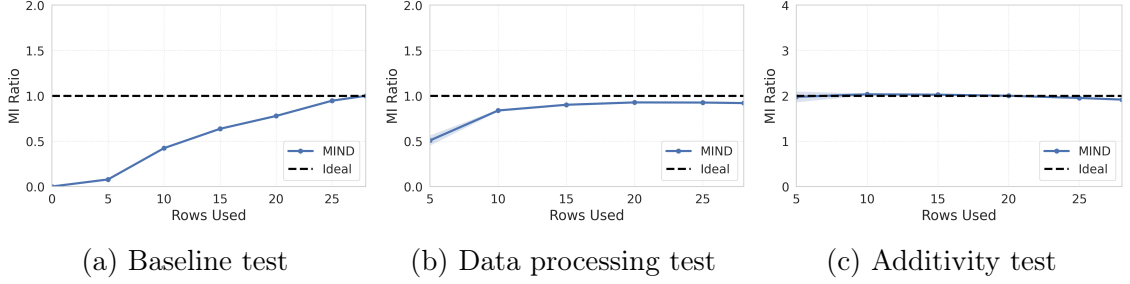


Figure 3.7: Consistency Tests over MNIST dataset: (a) evaluation of $\frac{I(A;B_r)}{I(A;B)}$; (b) evaluation of $\frac{I(A;[B_{r+k},B_r])}{I(A;B_{r+k})}$ for $k > 0$; (c) evaluation of $\frac{I([A^1,A^2];[B_r^1,B_r^2])}{I(A^1;B_r^1)}$.

fundamental properties of MI through three subtasks: Independence Test, Data-Processing Test, and Additivity Test for two independent images sampled from dataset. Ideally, as r increases, $\frac{I(A;B_r)}{I(A;B)}$ should monotonically approach 1, $\frac{I(A;[B_{r+k},B_r])}{I(A;B_{r+k})}$ should consistently equal 1, and $\frac{I([A^1,A^2];[B_r^1,B_r^2])}{I(A^1;B_r^1)}$ should consistently equal 2. To ensure a fair comparison with MINDE [66], we aligned our experimental settings and parameters and tested MIND using five random seeds. The results of the three tests are presented in Figure 3.7. Overall, MIND performed well and successfully passed all tests.

3.5 Related Work

Visual perception via diffusion models: Diffusion models’ success in image generation has piqued interest in their text-image understanding abilities. A variety of pipeline designs are emerging in the realm of visual perception and caption comprehension, offering fresh perspectives and complexities. Notably, ODISE introduced a pipeline that combines diffusion models with discriminative models, achieving excellent results in open-vocabulary segmentation [268]. Similarly, OVDiff uses diffusion models to sample support images for specific textual categories and subsequently extracts foreground and background features [111]. It then employs cosine similarity, often known as the CLIP [190] filter, to segment objects effectively. MaskDiff [133] and DiffSS [243] introduce a new breed of conditional diffusion models tailored for few-shot segmentation. DDP, on the other hand, takes the concept of conditional diffusion models and applies it to dense visual prediction tasks [105]. VDP incorporates diffusion for a range of downstream visual perception tasks such as semantic segmentation and depth estimation [282]. Several methods have begun to explore the utilization of attention layers in diffusion models for understanding text-image relationships in object segmentation [244, 248, 260, 278, 155]. Diffusion models have also been explored as a way to generate synthetic datasets for segmentation [71].

Image editing via diffusion: Editing real images is an area of growing interest both academically and commercially with approaches using natural text, similar images, and latent space modifications to achieve desired effects [169, 156, 151, 9, 115, 165]. [237]’s approach most closely resembles the procedure used in our intervention experiments. Unlike

previous work, we focus not on edit quality but on using edits to validate the learned dependencies as measured by our information estimators.

Interpretable ML: Traditional methods for attribution based on gradient sensitivity rather than attention [239, 154] are seldom used in computer vision due to the well-known phenomenon that (adversarial) perturbations based on gradients are uncorrelated with human perception [241]. Information-theoretic approaches for interpretability based on information decomposition [267] are mostly unexplored because no canonical approach exists [126] and existing approaches are completely intractable for high-dimensional use cases [200], though there are some recent attempts to decompose *redundant information* with neural networks [124, 141]. Our decomposition decomposes an information contribution from each variable, but does not explicitly separate unique and redundant components. “Interpretability” in machine learning is a fuzzy concept which should be treated with caution [148]. We adopted an *operational interpretation* from information theory, which considers $\mathbf{y} \rightarrow \mathbf{x}$ as a noisy channel and asks how many bits of information are communicated, using diffusion models to characterize the channel.

MI Estimation: Neural mutual information (MI) estimation has largely built on the Donsker–Varadhan lower bound [15] and its extensions, including annealed and energy-based methods [21]. Contrastive approaches, such as InfoNCE [254], train a critic to distinguish joint from marginal samples [186, 166, 250, 142], but suffer from the limitation that reliable estimates are bounded by $\log K$ for batch size K [161]. Improvements address this through bias–variance trade-offs [186, 226], multi-positive-sample bounds, and cross-entropy–based estimators for strong dependencies [215]. Other work seeks upper bounds

[37], multivariate extensions [36], and partial information decomposition [125, 199]. The limitations of methods such as MINE [15] and InfoNCE [254] can be traced to the inadequacy of the marginal distribution as a proposal for importance sampling. To address this, [202] proposed bridging the “density chasm” with a telescoping product of importance weights along a path between marginal and joint samples. While similar in spirit to annealed importance sampling and energy-based improvements [21], this method requires heuristics to construct intermediate bridge samples.

3.6 Conclusion

The eye-catching image generation capabilities of diffusion models have overshadowed the equally important and underutilized fact that they also represent the state-of-the-art in density modeling [121]. Using the tight link between diffusion and information, we were able to introduce a novel and tractable information decomposition. This significantly expands the usefulness of neural information estimators [14, 186, 20] by giving an interpretable measure of fine-grained relationships at the level of individual samples and variables. While we focused on vision tasks for ease of presentation and validation, information decomposition can be particularly valuable in biomedical applications like gene expression where we want to identify informative relationships for further study [183]. Another promising application relates to contemporaneous works on *mechanistic interpretability* [261, 80] which seeks to identify “circuits”—a subgraph of a neural network responsible for certain behaviors—by ablating individual network components to observe performance differences. For language models, differences are typically measured as the change in total probability of interested

vocabularies; whereas metric design remains an open question for diffusion models. Our formulation enhances mutual information estimation in denoising diffusion training, with the resulting CMI estimator outperforming variational methods in high-mutual information regimes while preserving semantic consistency and capturing compositional understanding and localizing image edits for stronger cross-modality alignment.

Chapter 4

Information-Preserved and Efficient Diffusion via Architectural Innovation

4.1 Stretching Each Dollar: Diffusion Training from Scratch on a Micro-Budget

4.1.1 Introduction

While modern visual generative models excel at creating photorealistic visual content and have propelled the generation of more than a billion images each year [253], the cost and effort of training these models from scratch remain very high [205, 195, 207]. In text-to-image diffusion models (T2I) [228, 90, 236], some previous works have successfully scaled down the computational cost compared to the 200,000 A100 GPU hours used by Stable Diffusion 2.1 [205, 30, 184, 74]. However, the computational cost of even the state-of-the-art approaches



(a) An elegant squirrel pirate on a ship; A hippo in a ballet performance in a white swan dress; A photo-realistic koala bear playing guitar; A rabbit magician pulling a hat.



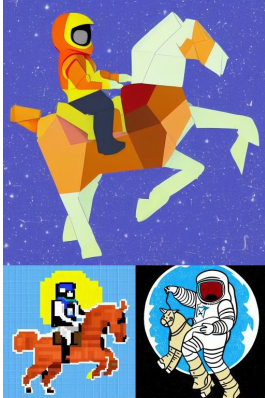
(b) A real life samurai standing in a green bamboo grove; A fairy tale castle in the clouds; A real looking vibrant butterfly on a photorealistic flower; A mystical cave with glowing crystals.



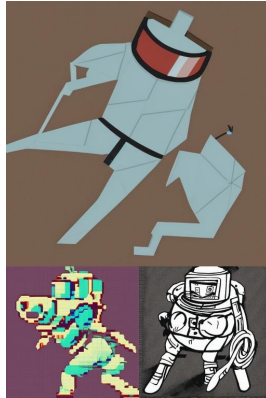
(c) A cute rabbit in a stunning, detailed Chinese coat; A frog on a lotus leaf; Toy elephants made of mahogany wood on a shelf; A logo of a fox in neon light.



(d) Ancient castle on a cliff edge; Serene beach with colorful pebbles; Photorealistic marketplace in a medieval town; Alien landscape with two moons.



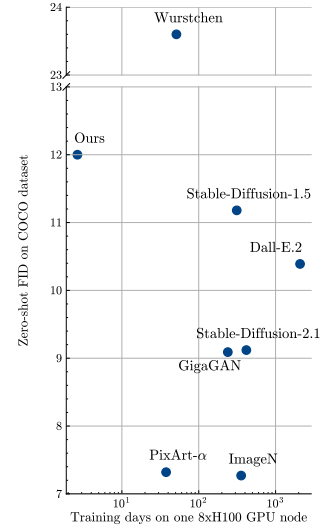
(e) Stable-Diffusion-1.5



(f) Stable-Diffusion-2.1



(g) Ours



(h) FID vs. training time

Figure 4.1: Qualitative evaluation of our 512×512 image generation. Our model is trained in 2.6 days on a single $8 \times H100$ machine (about \$1,890 GPU cost) without proprietary or billion-image datasets. In (e)–(g) we generate diverse styles for the prompt “Image of an astronaut riding a horse in ___ style” (Origami, Pixel art, Line art). In (h) we compare training cost and fidelity (FID [85]) for zero-shot COCO [145]. More samples: Fig. C.9–C.22.

(18,000 A100 GPU hours) remains very high [30], requiring more than a *month* of training time on the leading $8 \times H100$ GPU machine. Furthermore, previous works either leverage larger-scale datasets spanning on the order of a billion images or use proprietary datasets to enhance performance [205, 30, 273, 207]. The high training cost and the dataset requirements

create an inaccessible barrier to participating in the development of large-scale diffusion models. In this work, we aim to address this issue by developing a low-cost and end-to-end pipeline for competitive text-to-image diffusion models that achieve more than an order of magnitude reduction in training cost than the state-of-the-art while not requiring access to billions of training images or proprietary datasets

We consider vision transformer based latent diffusion models for text-to-image generation [54, 182], mainly due to their simplified design and wide adoption across the latest large-scale diffusion models [16, 58, 30]. To reduce the computational cost, we exploit the strong dependence of the transformer’s computational cost on the input sequence size, i.e., number of patches per image. We aim to reduce the effective number of patches processed per image by the transformer during training. This can be easily achieved by randomly masking a fraction of the tokens at the input layer of the transformer. However, existing masking approaches fail to scale beyond a 50% masking ratio without drastically degrading performance [283], particularly because, at high masking ratios, a large fraction of input patches are completely unobserved by the diffusion transformer.

To mitigate the massive performance degradation with masking, we propose a deferred masking strategy where all patches are preprocessed by a lightweight patch-mixer before being transferred to the diffusion transformer. The patch mixer comprises a fraction of the number of parameters in the diffusion transformer. In contrast to naive masking, masking after patch mixing allows non-masked patches to retain semantic information about the whole image and enables reliable training of diffusion transformers at high masking ratios, while incurring no additional computational cost compared to existing state-of-the-art masking [283].

We also demonstrate that under an identical computational budget, our deferred masking strategy achieves better performance than model downscaling, i.e., reducing the size of the model. Finally, we integrate recent advancements in transformer architecture, such as layer-wise scaling [162] and sparse transformers using mixture-of-experts [219, 286, 284], to improve performance in large-scale training.

Our low-cost training pipeline naturally reduces the overhead of ablating experimental design choices at scale. One such ablation we run is examining the impact of the choice of training datasets on the performance of micro-budget training. We demonstrate that when using small-scale real datasets, combining additional synthetic images during training can significantly improve performance. Furthermore, our combined dataset comprises only 37M images, significantly fewer than most existing large-scale models, thus providing strong evidence of high-quality generations from moderate-sized datasets in micro-budget training. We also investigate the generalization benefit from the synthetic dataset by scaling it up to nearly 500M images and demonstrate that training only on these synthetic images suffices to achieve high-quality image generation.

In summary, we make the following key contributions.

- We propose a novel deferred patch masking strategy that preprocesses all patches before masking by appending a lightweight patch-mixer model to the diffusion transformer.
- We conduct extensive ablations of deferred masking and demonstrate that it enables reliable training at high masking ratios and achieves significantly better performance compared to alternatives such as other masking strategies and model downscaling.

- Using a micro-budget of only \$1,890, we train a 1.16 billion parameter sparse diffusion transformer on 37M images and a 75% masking ratio that achieves a 12.7 FID in zero-shot generation on the COCO dataset. The wall-clock time of our training is only 2.6 days on a single 8×H100 GPU machine, 14× lower than the current state-of-the-art approach that would take 37.6 training days (\$28,400 GPU cost).
- We demonstrate that rather than limiting to only real images, combining additional synthetic images in micro-budget training or training only on a sufficiently large amount of synthetic images ($\sim 500\text{M}$) can significantly improve the quality of generated images and compete with existing open-source models in automated and human-preference-based evaluations.

We first provide a background on diffusion models and a review of common approaches, particularly those based on patch masking, to reduce computational costs in training diffusion transformers. Then, we provide a detailed description of our proposed deferred patch masking strategy.

4.1.2 Background: Diffusion-based generative models and diffusion transformers

We denote the data distribution by $\mathcal{D}$, such that $(\mathbf{x}, \mathbf{c}) \sim \mathcal{D}$, where $\mathbf{x}$ and $\mathbf{c}$ represent the image and the corresponding natural language caption, respectively. We assume that $p(\mathbf{x}; \sigma)$ is the image distribution obtained after adding σ^2 -variance Gaussian noise.

Diffusion-based probabilistic models [223, 90, 228, 235] use a forward-reverse process-based approach, where the forward process corrupts the input data and the reverse process tries to reconstruct the original signal by learning to reconstruct the data degradation at

each step. Though multiple variations of diffusion models exists [90, 235, 12], it is common to use time-dependent Gaussian noise ($\sigma(t)$) for model data corruption. Under deterministic sampling, both the forward and reverse processes in diffusion models can be modeled using an ordinary differential equation (ODE) [228, 113].

$$d\mathbf{x} = -\dot{\sigma}(t)\sigma(t)\nabla_{\mathbf{x}} \log p(\mathbf{x}; \sigma(t))dt, \quad (4.1)$$

where $\dot{\sigma}(t)$ represents the time derivative and $\nabla_{\mathbf{x}} \log p(\mathbf{x}; \sigma(t))$ is the score of the underlying density function. Thus the move towards (or away from) higher density regions in the diffusion process is proportional to both the scores of the density function and changes in the noise distribution ($\sigma(t)$) with time. Going beyond the deterministic sampling based on the ODE formulation in Equation 4.1, the sampling process in diffusion models can also be formulated as a stochastic differential equation [235],

$$d\mathbf{x} = \underbrace{-\dot{\sigma}(t)\sigma(t)\nabla_{\mathbf{x}} \log p(\mathbf{x}; \sigma(t))dt}_{\text{Probability flow ODE (Eq. 4.1)}} - \underbrace{\beta(t)\sigma(t)^2\nabla_{\mathbf{x}} \log p(\mathbf{x}; \sigma(t))dt}_{\text{Deterministic noise decay}} + \underbrace{\sqrt{2\beta(t)}\sigma(t)dw_t}_{\text{Noise injection}} \quad (4.2)$$

where dw_t is the standard Wiener process and the parameter $\beta(t)$ controls the rate of injection of additional noise. The generation process in diffusion models starts by sampling an instance from $p(\mathbf{x}, \sigma_{\max})$ and iteratively denoising it using the ODE or SDE-based formulation.

Training. Diffusion model training is based on parameterizing the score of the density function, which doesn't depend on the generally intractable normalization factor, using a denoiser, $\nabla_{\mathbf{x}} \log p(\mathbf{x}; \sigma(t)) \approx (F_{\theta}(\mathbf{x}, \sigma(t)) - \mathbf{x}) / \sigma(t)^2$. The denoising function $F_{\theta}(\mathbf{x}, \sigma(t))$ is generally modeled using a deep neural network. For text-to-image generation, the denoiser is

conditioned on both noise and captions and trained using the loss,

$$\mathcal{L} = \mathbb{E}_{(\mathbf{x}, \mathbf{c}) \sim \mathcal{D}} \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma(t)^2 \mathbf{I})} \|F_{\theta}(\mathbf{x} + \boldsymbol{\epsilon}; \sigma(t), \mathbf{c}) - \mathbf{x}\|_2^2. \quad (4.3)$$

The noise distribution during training is lognormal ($\ln(\sigma) \sim \mathcal{N}(P_{\text{mean}}, P_{\text{std}})$), where the choice of the mean and standard deviation of the noise distribution strongly influences the quality of generated samples [113]. For example, the noise distribution is shifted to the right to sample larger noise values on higher resolution images, as the signal-to-noise ratio increases with resolution [247]. Modern diffusion model training can be easily scaled to web-scale datasets [16, 58, 205, 207].

Classifier-free guidance. To increase alignment between input captions and generated images, classifier-free guidance [91, 114] modifies the image denoiser to output a linear combination of denoised samples in the presence and absence of input captions.

$$\hat{F}_{\theta}(\mathbf{x}; \sigma(t), \mathbf{c}) = F_{\theta}(\mathbf{x}; \sigma(t)) + w \cdot (F_{\theta}(\mathbf{x}; \sigma(t), \mathbf{c}) - F_{\theta}(\mathbf{x}; \sigma(t))), \quad (4.4)$$

where $w \geq 1$ controls the strength of guidance. During training, a fraction of image captions (typically 10%) are randomly dropped to learn unconditional generation.

Latent diffusion models. In contrast to modeling higher dimensional pixel space ($\mathbb{R}^{h \times w \times 3}$) [207, 170], latent diffusion models [205] are trained in a much lower dimensional compressed latent space ($\mathbb{R}^{\frac{h}{n} \times \frac{w}{n} \times c}$), where n is the compression factor and c is the number of channels in the latent space. The image-to-latent encoding and its reverse mapping are performed by a variational autoencoder. Latent diffusion models achieve faster convergence than training in

pixel space and have been heavily adopted in large-scale diffusion models [185, 16, 30, 10]. We consider a transformer based architecture for its superior performance in diffusion models [58, 30, 16]. We provide an additional review of the related work in Appendix C.1.

4.1.3 Approaches for minimizing the computational cost of training diffusion transformers

We describe the common key approaches to reduce the training cost in diffusion transformers in Figure 4.2(a-d). As images are first converted to a sequence of patches in transformers, using a larger patch (Figure 4.2b) size quadratically reduces the number of patches per image. This can significantly degrade performance due to the aggressive compression of larger regions of the image into a single patch. Instead, we opt to use patch masking, where we keep the patch size intact but drop a large fraction of patches at the input layer of the transformer [81], referred to as naive masking (Figure 4.2c). Due to its effectiveness, masking-based training of transformers has been adopted across vision and language domains [48, 81]. We do not significantly decrease the model size or the number of training steps to reduce costs, as web-scale diffusion models require high expressive power and training for a larger number of steps [205, 10].

Masking patches with autoencoding (MaskDiT - Figure 4.2d). MaskDiT aims to improve naive masking by using an auxiliary learning objective. In order to also encourage representation learning from masked patches, [283] adds an auxiliary autoencoding loss that encourages the reconstruction of masked patches. This formulation was motivated by the success of the autoencoding loss in learning self-supervised representations [81]. MaskDiT first appends a lightweight decoder, a small-scale transformer network, to the larger backbone transformer.

Before the decoder, it expands the backbone transformer output by inserting spurious learnable patches in place of masked patches.

Assuming that the binary mask m represents the masked patches, the final training loss is the following,

$$\begin{aligned}\mathcal{L}_{\text{diff}} &= \mathbb{E}_{(\mathbf{x}, \mathbf{c}) \sim \mathcal{D}} \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma(t)^2 \mathbf{I})} \| (\bar{F}_{\theta}((\mathbf{x} + \boldsymbol{\epsilon}) \odot (1 - m); \sigma, \mathbf{c}) - \mathbf{x}) \odot (1 - m) \|_2^2 \\ \mathcal{L}_{\text{mae}} &= \mathbb{E}_{(\mathbf{x}, \mathbf{c}) \sim \mathcal{D}} \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma(t)^2 \mathbf{I})} \| (\bar{F}_{\theta}((\mathbf{x} + \boldsymbol{\epsilon}) \odot (1 - m); \sigma, \mathbf{c}) - (\mathbf{x} + \boldsymbol{\epsilon})) \odot m \|_2^2 \\ \mathcal{L} &= \mathcal{L}_{\text{diff}} + \gamma \mathcal{L}_{\text{mae}}\end{aligned}\quad (4.5)$$

where $\bar{F}_{\theta}$ represents the sequential backbone transformer and decoder module, and γ is a hyperparameter to balance the influence of the masked autoencoding loss with respect to the diffusion training loss.

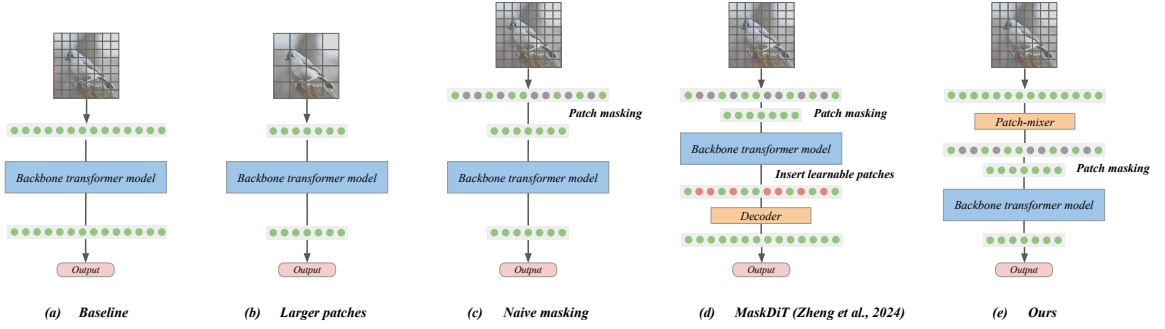


Figure 4.2: **Compressing patch sequence to reduce computational cost.** As the training cost of diffusion transformers is proportional to sequence size, i.e., number of patches, it is desirable to reduce the sequence size without degrading performance. It can be achieved by a) using larger patches, b) naively masking a fraction of patches at random, or c) using MaskDiT [283] that combines naive masking with an additional autoencoding objective. We find all three approaches lead to significant degradation in image generation performance, especially at high masking ratios. To alleviate this issue, we propose a straightforward deferred masking strategy, where we mask patches after they are processed by a patch-mixer. Our approach is analogous to naive masking in all aspects except the use of the patch-mixer. In comparison to MaskDiT, our approach doesn’t require optimizing any surrogate objectives and has nearly identical computational costs.

4.1.4 Alleviating performance bottleneck in patch masking using deferred masking

To drastically reduce the computational cost, patch masking requires dropping a large fraction of input patches before they are fed into the backbone transformer, thus making the information from masked patches unavailable to the transformer. High masking ratios, e.g., 75% masking, significantly degrade the overall performance of the transformer. Even with MaskDiT, we only observe a marginal improvement over naive masking, as this approach also drops the majority of image patches at the input layer itself. Maybe mention that our observation concurs with [283] finding that masking beyond 50% drastically degrades performance.

Deferred masking to retain semantic information of all patches. As high masking ratios remove the majority of valuable learning signals from images, we are motivated to ask whether it is necessary to mask at the input layer. As long as the computational cost is unchanged, it is merely a design choice and not a fundamental constraint. In fact, we uncover a significantly better masking strategy, with nearly identical cost to the existing MaskDiT approach. As the patches are derived from non-overlapping image regions in diffusion transformers [182, 54], each patch embedding does not embed any information about other patches in the image. Thus, we aim to preprocess the patch embeddings before masking, such that non-masked patches can embed information about the whole image. We refer to the preprocessing module as *patch-mixer*.

Training diffusion transformers with a patch-mixer. We consider a patch mixer to be any neural architecture that can fuse the embeddings of individual patches. In transformer

models, this objective is naturally achieved with a combination of attention and feedforward layers. So we use a lightweight transformer comprising only a few layers as our patch mixer. We mask the input sequence tokens after they have been processed by the patch mixer (Figure 4.2e). Assuming a binary mask m for masking, we train our model using the following loss function,

$$\mathcal{L} = \mathbb{E}_{(\mathbf{x}, \mathbf{c}) \sim \mathcal{D}} \mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma(t)^2 \mathbf{I})} \|F_{\theta}(M_{\phi}(\mathbf{x} + \epsilon; \sigma(t), \mathbf{c}) \odot (1 - m); \sigma(t), \mathbf{c}) - \mathbf{x} \odot (1 - m)\|_2^2 \quad (4.6)$$

where M_{ϕ} is the patch-mixer model and F_{θ} is the backbone transformer. Note that compared to MaskDiT, our approach also simplifies the overall design by not requiring an additional loss function and corresponding hyperparameter tuning between two losses during training. We don’t mask any patches during inference.

Unmasked finetuning. As a very high masking ratio can significantly reduce the ability of diffusion models to learn global structure in the images and introduce a train-test distribution shift in sequence size, we consider a small degree of unmasked finetuning after the masked pretraining. The finetuning can also mitigate any undesirable generation artifacts due to the use of patch masking. Thus, it has been crucial in previous work [283] to recover the sharp performance drop from masking, especially when classifier-free guidance is used in sampling. However, we don’t find it completely necessary, as even in masked pretraining, our approach achieves comparable performance to the baseline unmasked pretraining. We only employ it in large-scale training to mitigate any unknown-unknown generation artifacts from a high degree of patch masking.

Improving backbone transformer architecture with mixture-of-experts (MoE) and layer-wise scaling. We also leverage innovations in transformer architecture design to boost

the performance of our model under computational constraints. We use mixture-of-experts layers as they increase the parameters and expressive power of our model without significantly increasing the training cost [219, 61]. We use a simplified MoE layer based on expert-choice routing [284], where each expert determines the tokens routed to it, as it doesn’t require any additional auxiliary loss functions to balance the load across experts [219, 286]. We also consider a layer-wise scaling approach, which has recently been shown to outperform canonical transformers in large-language models [162]. It linearly increases the width, i.e., hidden layer dimension in attention and feedforward layers, of transformer blocks. Thus, deeper layers in the network are assigned more parameters than earlier layers. We argue that since deeper layers in visual models tend to learn more complex features [275], using higher parameters in deeper layers would lead to better performance. We provide background details on both techniques in Appendix C.3. We describe the overall architecture of our diffusion transformer in Figure C.1 in Appendix C.2.

4.1.5 Experimental Setup

We provide key details in our experimental setup below and additional details in Appendix C.2. We use two variants of Diffusion Transformer (DiT) architecture from [182]: DiT-Tiny/2 and DiT-XL/2, both comprising a patch size of 2. When training sparse models, we replace every alternate transformer block with an 8-expert mixture-of-experts block. We train all models using the AdamW optimizer with cosine learning rate decay and high weight decay. We provide an exhaustive list of our training hyperparameters in Table C.2 in Appendix C.2. We refer to our diffusion transformer as `MicroDiT`. We use Fréchet inception distance (FID) [85], both with the original Inception-v3 model [242] and a CLIP model [188],

and CLIP score [84] to measure the performance of the generative models, along with human preference based evaluations. We use a \$30/hour cost estimate for an 8×H100 GPU node¹ (equivalent to \$12/hour for an 8×A100 GPU node) for the economical cost.

Training datasets. We use the following three real image datasets comprising 22M image-text pairs: Conceptual Captions (CC12M) [27], Segment Anything (SA1B) [123], and TextCaps [220]. We add two synthetic image datasets comprising 15M image-text pairs in large-scale training: JourneyDB [238] and DiffusionDB [266]. We also use Diffusion1B [87], a dataset comprising one billion synthetic images from diffusion models, in experiments on dataset scaling. For small-scale ablations, we construct a text-to-image dataset, named cifar-captions, by subsampling images of ten CIFAR-10 classes from the larger COYO-700M [25] dataset. Thus, cifar-captions serves as a drop-in replacement for any image-text dataset and simultaneously enables fast convergence due to being closed-domain. We provide additional details on construction of this dataset in Appendix C.2 and sampled images in Figure C.23.

4.1.6 Effectiveness of Deferred Masking

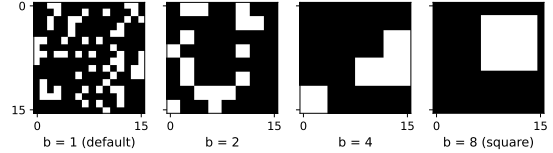
We validate the effectiveness of our deferred masking approach and investigate the impact of individual design choices on image generation performance. We use the DiT-Tiny/2 model and the cifar-captions dataset (256×256 image resolution) for all experiments in this section.

Out-of-the-box performance: Deferred masking enables high masking ratios. A key limitation of patch masking strategies is poor performance when a large fraction of

¹<https://cloud-gpus.com/>

	FID ($\downarrow$)	CLIP-FID ($\downarrow$)	CLIP-Score ($\uparrow$)
Out-of-box	8.82	5.03	26.72
Higher weight decay	8.38	4.90	27.00
Sigma distribution	8.49	4.93	27.47
Larger patch-mixer	7.40	4.57	27.79
Higher learning rate	7.09	4.10	28.24

(a) Summarizing the factors that led to improved performance during our ablation study. We conduct this ablation study at 75% masking ratio. Detailed results are provided in Table C.3 in Appendix C.4.



(b) **Is block masking of patches more effective?** We retain a continuous region of image patches using block masking to test its impact on image generation performance. In this figure, we illustrate the masks for 75% masking ratio with varying block sizes. By default, we mask each patch randomly, i.e., block size (b) = 1.

Figure 4.3: Ablating individual design choices in deferred masking, spanning from training hyperparameters to patch masking strategies.

patches are masked. For example, [283] observed large degradation in MaskDiT performance with a masking ratio higher than 50%. We use commonly used default hyperparameters to simulate the out-of-the-box performance of our approach at high masking ratios. We provide our results in Figure C.3 in Appendix C.4, along with additional details on training hyperparameters. We observe that the deferred masking approach achieves much better performance than the baseline naive masking across all performance metrics. Furthermore, the performance gaps widen with an increase in the masking ratio. For example, with a 75% masking ratio, naive masking degrades the FID score to 16.5, while our approach achieves 5.03, much closer to the FID score of 3.79 with no masking.

Ablation study of our training pipeline. We ablate design choices across masking, patch mixer, and training hyperparameters to improve performance beyond the out-of-the-box performance. We summarize the techniques that improved out-of-the-box performance in Figure 4.3a, with supplementary results in Table C.3 in Appendix C.4. We find that the optimal design choices for the diffusion transformer are *similar* to those for transformer-based

large language models (LLMs). Similar to common LLM training setups [249, 40], we find that SwiGLU activation [218] outperforms GELU [83], and higher weight decay improves performance. We update the noise distribution $(P_{\text{mean}}, P_{\text{std}})$ to $(-0.6, 1.2)$ as it improves the alignment between captions and generated images. By default, we mask each patch at random. We ablate the masking strategy to retain more continuous regions of patches using block masking (Figure 4.3b). We find that increasing the block size leads to a degradation in the performance of our approach; thus, we retain the original strategy of masking *each patch at random*.

Validating improvements in diffusion transformer architecture. We measure the effectiveness of two key modifications in the transformer architecture: layer-wise scaling and mixture-of-experts (MoE) layers. First, we train two DiT-Tiny transformers under identical training costs by varying the choice of layer-wise scaling. We find that the layer-wise scaling approach outperforms the baseline constant-width approach across all three performance metrics (Table C.4 in Appendix C.4). Next, we add expert-choice routing based on MoE layers and observe that it slightly improves the Clip-score from 28.11 to 28.66 but degrades the FID score from 6.92 to 6.98. We believe that the lack of improvement stems from limited training steps (60K), as multiple experts further reduce the training samples observed by each router. We observe a significant improvement in performance with MoE blocks when training on a larger scale for longer training schedules (Section 4.1.8).

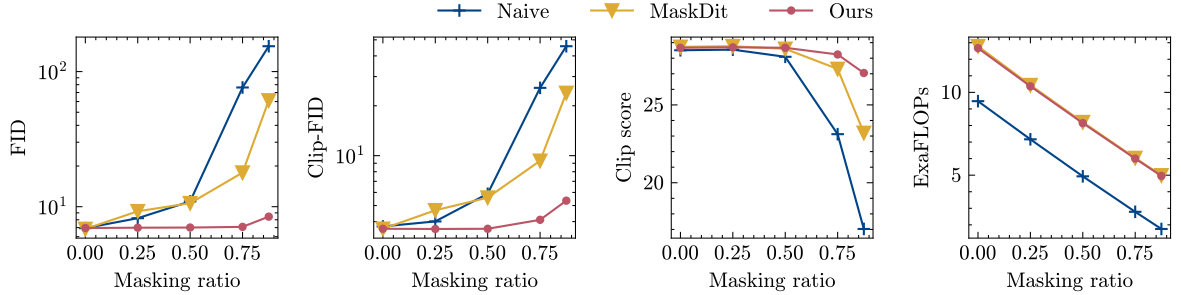


Figure 4.4: **Comparing performance of patch masking strategies.** Using a lightweight patch-mixer before patch masking in our deferred masking approach significantly improves image generation performance over baseline masking strategies.

4.1.7 Comparing the Effectiveness of Deferred Masking with Baselines

In this section, we provide a concrete comparison with multiple baselines. First, we compare with techniques aimed at reducing computational cost by reducing the transformer patch sequence size. Next, we consider reducing network size, i.e., downscaling, while keeping the patch sequence intact. Under isoflops budget, i.e., identical wall-clock training time and FLOPs, we show that our approach achieves better performance than model downscaling.

Comparing different masking strategies. We first compare our approach with the strategy of using larger patches. We increase the patch size from two to four, equivalent to 75% patch masking. However, it underperforms compared to deferred masking and only achieves 9.38, 6.31, and 26.70 FID, Clip-FID, and Clip-score, respectively. In contrast, deferred masking achieves 7.09, 4.10, and 28.24 FID, Clip-FID, and Clip-score, respectively. Next, we compare all three masking strategies (Figure 4.2). We train each model for 60K steps and report total training flops for each approach to indicate its computational cost. We find that naive masking significantly degrades model performance, and using additional proxy loss functions [283] only marginally improves performance. We find that simply deferring

the masking and using a small transformer as a patch-mixer significantly bridges the gap between masked and unmasked training of diffusion transformers. Provide maskdit baseline hyperparameters in appendix.

IsoFLOPs study - Why bother using masking instead of smaller transformers?

Using patch masking is a complementary approach to downscaling transformer size to reduce computation cost in training. We compare the two approaches in Table C.5 (Appendix C.4). At each masking ratio, we reduce the size of the unmasked network to match FLOPs and total wall-clock training time of masked training. We train both networks for 60K training steps. Until a masking ratio of 75%, we find that deferred masking outperforms network down-scaling across at least two out of three metrics. However, at extremely high masking ratios, deferred masking tends to achieve lower performance. This is likely because the information loss from masking is too high at these ratios. For example, for 32×32 resolution latent and a patch size of two, only 32 patches are retained (out of 256 patches) at an 87.5% masking ratio.

4.1.8 Micro-budget Training at Scale

In this section, we validate the effectiveness of our approach in training open-world text-to-image generative models on a micro-budget. Moving beyond the small-scale cifar-captions dataset, we train the large-scale models on a mixture of real and synthetic image datasets. We demonstrate the high-quality image generation capability of our model and investigate the influence of training dataset on the performance of micro-budget training.

Training process. We train a DiT-XL/2 transformer with eight experts in alternate transformer blocks. We train each model for 305K steps, including 250K steps on 256×256

resolution images (Phase-1) and 55K steps on 512×512 resolution images (Phase-2). We provide details on training hyperparameters in Table C.2 in the Appendix C.2. We conduct the training in the following two phases. We refer to any diffusion transformer trained using our micro-budget training as **MicroDiT**.

Computational cost. We provide a breakdown of computational cost, including both training FLOPs and economical cost, across each training phase in Table C.6. Our Phase-1 and Phase-2 training consumes 56% and 44% of the total computational cost, respectively. The total wall-clock training time of our model is 2.6 days on an $8 \times \text{H100}$ GPU cluster (amounting to \$1,890 economical cost), equivalent to 6.6 days on an $8 \times \text{A100}$ GPU cluster.

Comparing zero-shot image generation on COCO dataset (Table C.8). We follow the evaluation methodology in previous work [207, 273, 10] and sample 30K random images and corresponding captions from the COCO 2014 validation set. We generate 30K synthetic images using the captions and measure the distribution similarity between real and synthetic samples using FID (referred to as FID-30K in Table C.8). Our model achieves a competitive 12.66 FID-30K, while trained with $14 \times$ lower computational cost than the state-of-the-art low-cost training method and without any proprietary or humongous dataset to boost performance. Our approach also outperforms Wüsthchen [184], a recent low-cost training approach for diffusion models, while requiring $19 \times$ lower computational cost.

Fine-grained image generation comparison (Table C.9). We use the GenEval [73] framework to evaluate the compositional image generation properties, such as object positions, co-occurrence, count, and color. Similar to most other models, our model achieves near perfect accuracy on single-object generation. Across other tasks, our model performs similar

Table 4.1: **Quantitative evaluation.** a) FID: Zero-shot FID on the COCO2014 validation split lin2014microsoftCOCO. b) Compositional image generation using the GenEval ghosh2024geneval benchmark. On both metrics, we find that our approach achieves competitive performance compared to Stable Diffusion models v1.5 and v2.1, while taking $14.2\times$ less training time than the state-of-the-art low-cost training approach (e.g., PixArt- α). We provide expanded FID and GenEval results in Tables C.8 and C.9 (Appendix C.5), respectively.

Model	Params ($\downarrow$)	Sampling steps ($\downarrow$)	Open source	Training images ($\downarrow$)	8×H100 GPU days ($\downarrow$)	FID-30K ($\downarrow$)	GenEval ($\uparrow$)
DALL-E 2 [195]	6.50B	–	–	650M	2083.2	10.39	0.52
GigaGAN [109]	1.01B	1	–	980M	239.1	9.09	–
Imagen [207]	3.00B	–	–	860M	356.6	7.27	–
Stable Diffusion 2.1 [205]	0.86B	50	✓	3900M	416.5	9.12	0.50
Stable Diffusion 1.5 [205]	0.86B	50	✓	4800M	312.5	11.18	0.43
Würstchen [184]	0.99B	60	✓	1420M	51.2	23.60	–
PixArt- α [30]	0.61B	20	✓	25M ^a	37.6	7.32	0.48
MicroDiT (ours)	1.16B	30	✓	37M	2.6	12.66	0.46

^a Includes 10M proprietary high-quality images.

to early Stable-Diffusion variants, notably outperforming Stable-Diffusion-1.5. On the color attribution task, our model also outperforms Stable-Diffusion-XL-turbo and PixArt- α models.

Ablating design choices on scale. We also ablate the choice of mixture-of-experts (MoE) layers by training another MicroDiT model without them under an identical training setup. We find that using MoE layers improves the zero-shot FID-30K from 13.7 to 12.7 on the COCO dataset. Due to the better performance of the sparse transformers with MoE layers, we use MoE layers by default in large-scale training. We also consider increasing the dimensionality of the latent space, as it provides improvements in image generation abilities for state-of-the-art models [44, 58]. Intriguingly, we find that using a higher dimensional latent space in micro-budget training hurts performance (Table C.7 in Appendix C.5), likely because an increase in latent dimensionality allows for better modeling of data distribution; however, it also simultaneously increases the training budget required to train higher quality models.

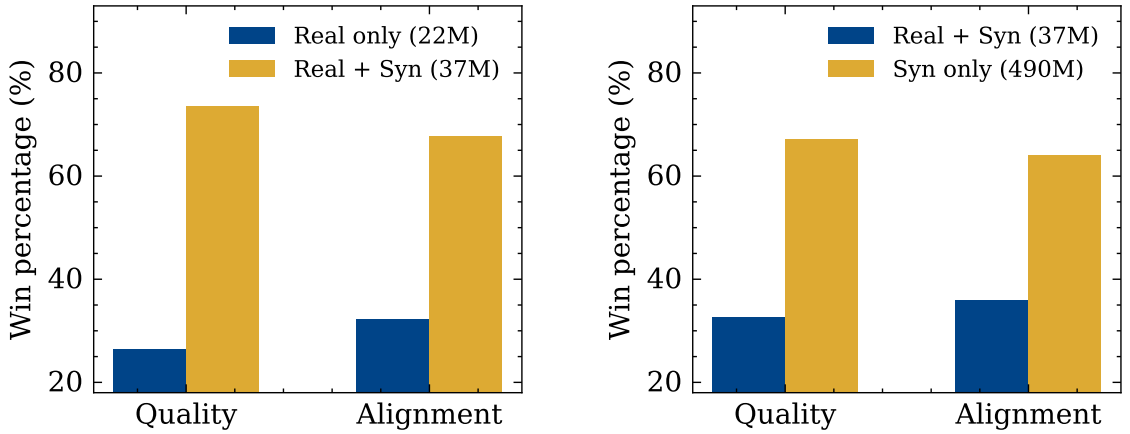
4.1.9 Understanding role of synthetic data in micro-budget training

In addition to quantitative metrics (FID and GenEval scores), we also conduct a human preference-based evaluation to alleviate the shortcomings of these metrics [179, 184]. To achieve the best performance of the large-scale model, we previously combined commonly used real and synthetic datasets, resulting in a total of 37M training images. Note that our synthetic dataset comprises nearly 10M images from early generations of latent diffusion models, often of significantly poor quality.

Training only on real datasets significantly degrades performance. To isolate the effect of synthetic data, we train another MicroDiT model on only real images (22M images) under an identical training configuration and computational budget. Despite the real images comprising nearly 60% of the total samples in the joint real and synthetic dataset, training only on real images achieves significantly poorer image quality compared to the model trained on the joint dataset. Intriguingly, both models lead to nearly identical FID and GenEval scores. However, even with an automated multimodal-based evaluation [176], we clearly observe that images generated from the model trained on the combined dataset are preferred nearly three times more than those generated from the model trained on real-only data. We validate this further with a human preference study (for which we provide details in the supplementary), where images from the former model are preferred more than 65% of the time across both image quality and alignment with prompt criteria (Figure 4.5a).

Determining asymptotic influence of synthetic data at scale. We consider training only on synthetic data, in particular training on massive amount of synthetic images to understand its asymptotic behaviour. Using the identical training cost (\$1,890) as previous

models, we train a MicroDiT model on 490M synthetic images generated from various diffusion models [87]. We find that scaling synthetic data suffice to achieve an even better MicroDiT model than training on the joint real and synthetic dataset of 37M images. It improves the GenEval score from 0.46 to 0.52, while simultaneously being preferred in at least 68% of evaluations by human raters on both image quality and alignment with prompt (Figure 4.5b). We provide a qualitative comparison of samples with other models in Figure C.7.



(a) Model trained only on real data receives lower preference.

(b) Model trained only on massive synthetic data receives higher preference.

Figure 4.5: Human preference based evaluation. We report the win percentage based on human preference when comparing images across different MicroDiT models. Here, alignment refers to alignment between caption and generated image. We first compare the preference among identical models trained with only 22M real images vs. combined set (37M) of 22M real and 15M synthetic images (subfig a). Next, we scale up synthetic images to 490M and train a model only on these images, which achieves better human preference than our previous model trained on 37M real and synthetic images (subfig b).

Human preference evaluation of our final model. We consider the MicroDiT model trained only on nearly 0.5B image-caption pairs, while incurring identical training of \$1,890 as other MicroDiT models, as our final model and compare its performance with other

Table 4.2: Win percentages of our best MicroDiT model, trained only on 490M synthetic images, against four open-source baseline models.

Model	Quality ($\uparrow$)	Alignment ($\uparrow$)
Stable-Diffusion-1.5	47.3	55.8
Stable-Diffusion-2.1	53.8	51.3
Würstchen	47.6	53.3
PixArt- α	48.5	45.2

open-source models using human preferences (Table 4.2) . Except the PixArt- α model, our model is slightly preferred more than the baseline models along at least one of the two evaluation criteria.

4.1.10 Conclusion

In this work, we focus on patch masking strategies to reduce the computational cost of training diffusion transformers. We propose a deferred masking strategy to alleviate the shortcomings of existing masking approaches and demonstrate that it provides significant improvements in performance across all masking ratios. With a deferred masking ratio of 75%, we conduct large-scale training on commonly used real image datasets, combined with additional synthetic images. Despite being trained at an order of magnitude lower cost than the current state-of-the-art, our model achieves competitive zero-shot image generation performance. We hope that our low-cost training mechanism will empower more researchers to participate in the training and development of large-scale diffusion models.

4.2 LayerGen: Redefining image generation as scene composition

4.2.1 Introduction

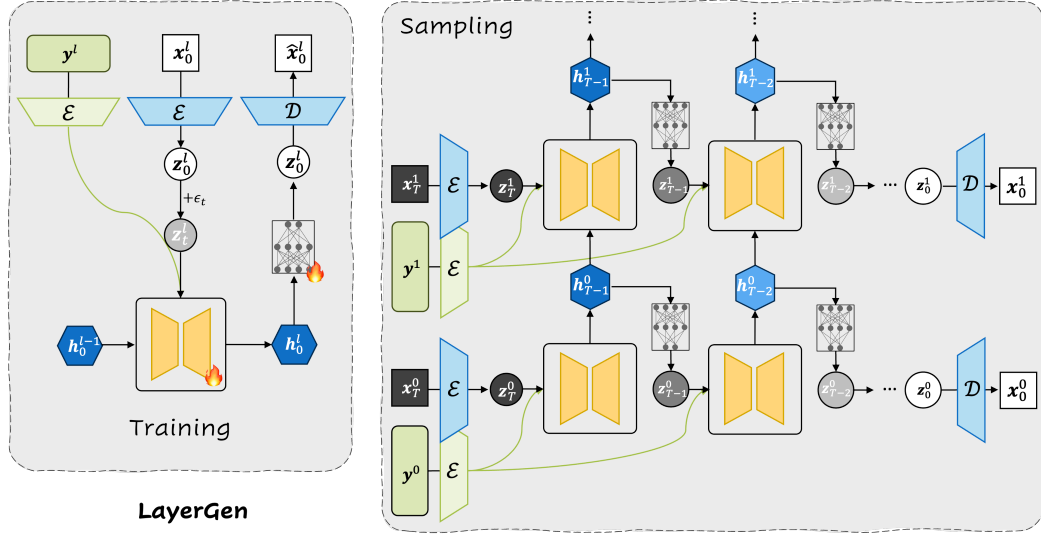


Figure 4.6: This is diagram of Latent Auto-Regression Diffusion model.

Layered image generation and editing have become fundamental tasks in computer vision, enabling applications such as progressive scene composition, object-specific editing, and controllable image synthesis. These tasks require models to not only generate high-quality content at each layer but also ensure spatial and semantic consistency across layers. Maintaining causality between layers is particularly important for enabling flexible editing operations, such as adding, removing, or modifying individual objects while preserving the coherence of the entire scene.

Recent advancements in generative models, particularly diffusion models, have demonstrated remarkable success in high-quality image synthesis. However, most existing

diffusion-based approaches focus on end-to-end image generation, limiting their ability to perform localized modifications. Models such as Stable Diffusion have been adapted for layered image generation by incorporating techniques like LoRA [95] and encoder modifications. For instance, TILD [276] employs a foreground-background mixing strategy to facilitate progressive image generation, significantly reducing the number of trainable parameters. Similarly, Text2Layer [279] modifies the encoder structure of Stable Diffusion to generate two-layered images. However, these approaches lack flexibility, as they are difficult to adjust after training and fail to capture causality between layers. Additionally, LayerDiff [97] attempts multi-layer generation by adapting the Stable Diffusion framework, but its performance remains suboptimal, often suffering from high FID scores.

Despite these advancements, existing methods heavily rely on datasets with precise mask or layout annotations, which, while improving the model’s understanding of individual layer content, restrict their ability to learn causal relationships between layers. Furthermore, some models employ parallel generation processes, such as the LRDiff [187], which heavily depends on masks to control layer layouts but lacks inter-layer coherence. To address these limitations, we propose **LayerGen**, a Diffusion Forcing-based framework that integrates recurrent neural networks to ensure both high-quality layer generation and inter-layer causality. Unlike previous approaches, an autoregressive (AR) diffusion model must finish each diffusion step before generating the next layer, while parallel diffusion suffers from a lack of inter-layer communication, often resulting in poor composition. In contrast, Diffusion Forcing offers a unified training framework that generates multiple layers in parallel yet preserves the causal order among them through a recurrent architecture, leading to more coherent and

comprehensible compositions. Furthermore, its flexible conditioning mechanism helps the model converge faster during training. This work’s contributions are as follows:

- **Latent Diffusion Forcing for Layered Generation:** We propose a novel object-centric generation framework that enables progressive scene composition while preserving cross-layer consistency, resulting in enhanced compositional fidelity.
- **Flexible Image Editing:** Unlike existing models that struggle with image modifications, our approach allows seamless object addition and removal while maintaining spatial coherence with surrounding contents.
- **Object-Level Control via Text Prompts:** Our model enables fine-grained manipulation of object attributes such as color and spatial positioning through textual prompts, without affecting other scene components.
- **Scalability Beyond Training Constraints:** Despite being trained on three-layer images, our model generalizes to arbitrary layer depths during sampling, demonstrating strong adaptability and robustness.

4.2.2 Related Work

Denoising Diffusion Models

Diffusion models [90, 233] are a class of generative models that have achieved remarkable success in image [205] and video [18] generation. These models rely on a process of gradually transforming random noise into structured data through a series of denoising steps and then backwards. In the forward process, noise is progressively added to a data sample $\mathbf{x}_0 \sim q(\mathbf{x})$, creating a sequence of noisy data $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T$ that eventually approaches

a Gaussian distribution. Mathematically, this process is defined as:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}),$$

where $\beta_t \in (0, 1)$ controls the variance of noise at each timestep. In the reverse process, a neural network is trained to model the conditional probabilities

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)),$$

enabling the iterative removal of noise and reconstruction of the original data. The training objective minimizes the evidence lower bound (ELBO), often reparameterized and simplified by $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}} \epsilon_\theta(\mathbf{x}_t, t) \right)$ to predicting the added noise using a mean squared error loss:

$$L(\theta) = \mathbb{E}_{\mathbf{x}_0, \epsilon_t, t} \left[\|\epsilon_t - \hat{\epsilon}_\theta(\mathbf{x}_t, t)\|_2^2 \right].$$

Layer Diffusion

Recently, several works have aimed to address the challenges of generating or editing layered images. [98, 138] proposed a diffusion-based approach for layered image editing, demonstrating that the generated images better align with prompt descriptions compared to end-to-end generation methods. Zhang et al. [276] represents a significant contribution to image generation, leveraging pretrained Stable Diffusion and LoRA techniques to design a foreground-background mixing training strategy. This approach enables a progressively generative pipeline, significantly reducing the number of trainable parameters. Similarly,

Text2Layer [279] modifies the encoder structure of Stable Diffusion, achieving promising results for 2-layer image generation. While LayerDiff [97] attempts to generate more than two layers, it only adjusts the Stable Diffusion framework to accommodate multi-layer inputs, but its performance remains suboptimal, with high FID scores. In contrast to [276], Text2Layer, and LayerDiff, which lack flexibility after training and fail to maintain causality between layers, the LRDiff [187] introduces a novel approach. During the reverse diffusion process, LRDiff applies individual diffusion processes to each layer in the initial steps, merges the denoised results, and then completes the reverse diffusion process. However, because each layer is generated in parallel, the model heavily relies on masks to control the layout of the final image. All these models require datasets with precise mask or layout annotations, which, while improving the model’s understanding of image content, significantly limit the model’s ability to learn the causal relationships between different layers.

Text-Guided Image Editing

Text-guided image editing has gained significant attention with the rise of diffusion models, particularly Stable Diffusion [205]. Early works such as SDEdit [164] introduced a stochastic denoising process to modify images while maintaining structure, but it lacked fine-grained object control. Later, Blended Diffusion [8] improved local editing by constraining diffusion updates within masked regions. To enable interactive instruction-based editing, InstructPix2Pix [23] fine-tuned Stable Diffusion on paired real and edited images, achieving realistic modifications but lacking strong spatial control. Composable Diffusion Models (CDM) [152] addressed multi-object editing by introducing energy-based constraints, ensuring structured scene generation. More recently, InstructEdit [264] improved text-image alignment,

enhancing edit quality without requiring explicit fine-tuning. Despite these advances, existing models struggle with multi-layer scene composition and flexible object manipulation. In contrast, our approach extends Stable Diffusion into a layered generation framework, enabling progressive object addition, removal, and modification while preserving inter-layer consistency.

4.2.3 Latent Causality Diffusion Models

To achieve progressive generation with moderate causality, we propose a layer diffusion model that integrates auto-regressive principles into the diffusion process. This approach naturally builds on the strengths of combining diffusion models and recurrent architectures, as exemplified in Diffusion Forcing [29], which enables high-quality layer-by-layer generation while maintaining modest causality and consistency.

Diffusion Forcing introduces recurrent structures into diffusion models to capture temporal dependencies and ensure inter-layer consistency. Instead of directly operating in the pixel space $\mathbf{x}$, we utilize a latent variable $\mathbf{z}$, which is derived from $\mathbf{x}$ through an encoder-decoder architecture. Specifically, given an input $\mathbf{x}$, the latent variable is obtained as:

$$\mathbf{z} = \text{Encoder}(\mathbf{x}).$$

This latent space representation reduces dimensionality and simplifies the diffusion process, similar to techniques employed in Stable Diffusion [205].

A recurrent neural network (RNN [57]) is employed to iteratively update hidden states in the latent space, defined as:

$$\mathbf{h}_t^l = f(\mathbf{h}_t^{l-1}, \mathbf{z}_t^l), \quad (4.7)$$

where $\mathbf{h}_t^l$ is the l -th layer hidden state at step t , and $f(\cdot)$ represents the recurrent unit, such as an LSTM [93] or GRU [38]. The hidden state $\mathbf{h}_t^l$ provides contextual information for layer-wise generation, influencing the reverse diffusion parameters $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\Sigma}_\theta$ in the latent space.

The training objective minimizes the discrepancy between the true noise ϵ and the predicted noise $\hat{\epsilon}$ across all layers in the latent space:

$$\mathcal{L} = \mathbb{E}_{q,l} \left[\|\epsilon_t^l - \hat{\epsilon}(\mathbf{z}_t^l, \mathbf{h}_{t-1}^l, \mathbf{y}^l, t)\|^2 \right], \quad (4.8)$$

$\mathbf{y}^l$ denotes the conditioning at l -th layer.

The generation process begins with an initialized hidden state $\mathbf{h}_{0:T}^0 = \mathbf{0}$. At each step, the model iteratively generates a latent representation by sampling from the reverse diffusion distribution:

$$\mathbf{h}_{t-1}^l \sim p_\theta(\mathbf{z}_{t-1}^l \mid \mathbf{z}_t^l, \mathbf{h}_{t-1}^l), \quad (4.9)$$

$$\mathbf{z}_{t-1}^l = g(\mathbf{h}_{t-1}^l), \quad (4.10)$$

and subsequently updates the hidden state with Eq. 4.7. Here, $g(\cdot)$ is a small fully-connected neural network used to extract $\mathbf{z}$ from the hidden state $\mathbf{h}$. Finally, the reconstructed image $\mathbf{x}$ is obtained using the decoder:

$$\mathbf{x}_0^l = \text{Decoder}(\mathbf{z}_0^l).$$

This iterative mechanism ensures the generated layers are both causally linked and consistent in the latent space, aligning with the overarching goal of progressive generation while leveraging the benefits of latent representation for improved efficiency and scalability.

4.2.4 Experiments

Leveraging advantages of auto-regressive diffusion models, we hypothesize that this architecture can effectively address the following three key challenges: (1) Can image generation be carried out progressively on an object-by-object basis without disrupting the spatial relationships among existing content? (2) Can layers be freely added or removed without compromising the coherence and causality between layers? (3) Can prompts be used to control newly added objects, including their shape, size, and spatial positioning within the image?

4.2.5 Datasets

To explore the potential of the model, we constructed a multi-layer structured image dataset. Starting with simple scenes, like CLEVR [107] dataset, we used Blender [19] to render random combinations of objects with varying quantities, shapes, sizes, and materials

under different lighting and shadow conditions. Using an open-source VLM, InternVL [35], we added instruction-based descriptions to each layer of the images. These descriptions specify the new object to be added and its relationship to the content of the previous layer (see Fig. 4.7). Finally, we trained LARD model on these multi-layer image-text pairs, Instruction CLEVR.

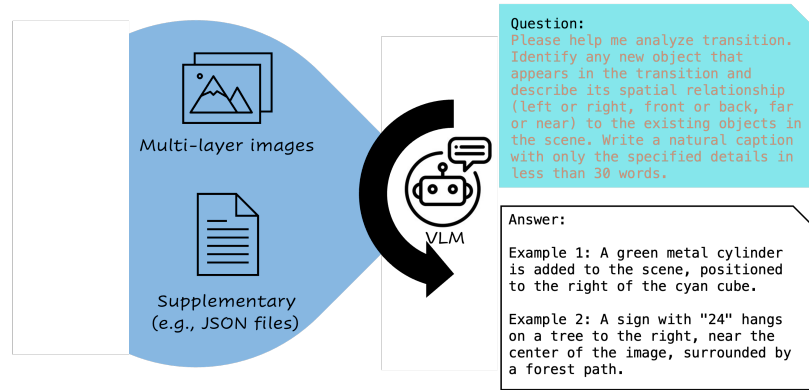


Figure 4.7: The illustration of the pipeline used for layered images captioning.

In addition to the Instruction CLEVR dataset for simple scenes, Tudosiu et al. [251] introduced the MULAN dataset for multi-layer image generation and inpainting. However, the captions in this dataset were generated by BLIP2 [137], which described each layer independently, resulting in a loss of inter-layer relationships in the text. Furthermore, newly added objects were always often overlooked in these captions. To address this, we re-captioned the MULAN dataset using InternVL, ensuring that inter-layer relationships and newly added objects were accurately captured.

Instruction CLEVR consists of approximately 70k sets of layered images, with each image containing 3 to 6 layers. We randomly split the dataset, using 90% for training and 10% for

testing. The prompt for the first layer provides a complete description of the scene, while each subsequent layer is paired with an instruction-based prompt, as illustrated in Fig. 4.8a.

Instruction MULAN comprises two real-world datasets: **MULAN_COCO** and **MULAN_LAION** [251]. Similarly, we split each dataset into 90% for training and 10% for testing, as illustrated in Fig. 4.8b.

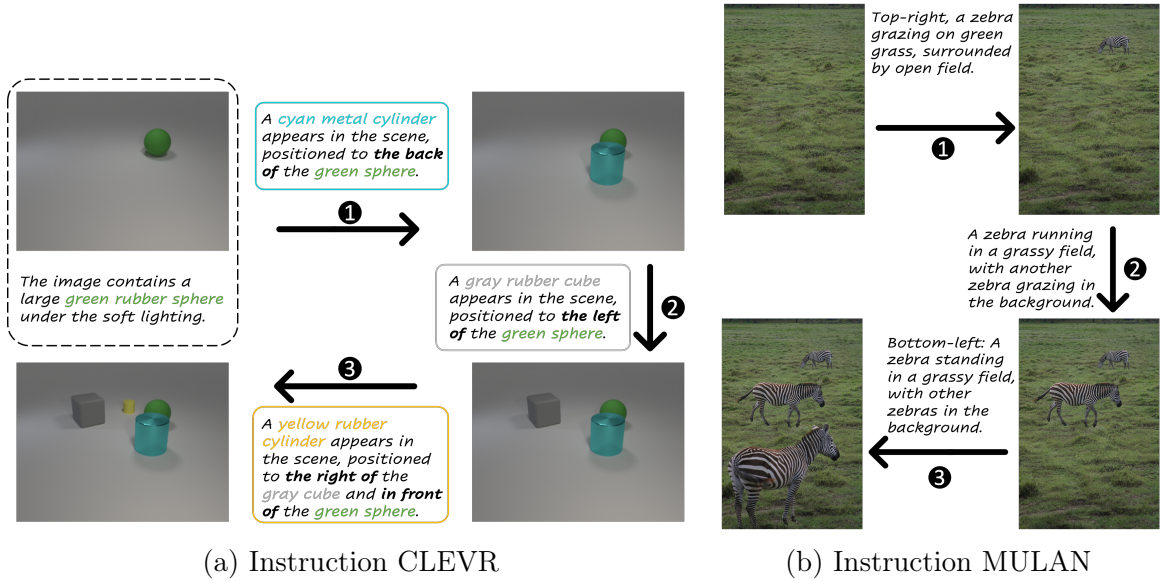


Figure 4.8: Examples of prepared layered image datasets.

Model Training and Sampling

We designed two versions of the model for training. In the pixel space, we adopt the vanilla Diffusion Forcing architecture [29]. However, to fuse image and text information, we introduce cross-attention layers after the self-attention layers. Additionally, the time embedding is integrated into the ResNet using FiLM [56]. In the latent space, the model consists of two latent variables: $\mathbf{z}$, which represents the image encoding, and $\mathbf{h}$, which serves as the transition variable. The whole pipeline is shown in Fig. 4.6. We allocate different

numbers of channels to these variables, assigning more channels to $\mathbf{h}$ than to $\mathbf{z}$ to ensure that the accumulated changes in the transition process are properly captured.

During training, we adopt velocity prediction [209] to avoid singularities, ensuring a smoother decrease of loss in Eq. 4.8. Furthermore, we randomly drop 10% of the prompts to enable classifier-free guidance [92] sampling. In sampling phase, we employ DDIM [228] for efficient and deterministic sampling, and dynamic thresholding [208] to prevent the influence of extreme values on the generated images, thereby improving image quality. In particular, our model is trained using only three-layer images. However, during the sampling phase, it can recurrently generate an arbitrary number of layers.

Layer-Conditioned Generation

To address question (1), which aims to verify whether the combination of diffusion and recurrent neural networks can add a new object at a random position in layer $N + 1$ while preserving the content of the previous N layers, we conducted *Layer-Conditioned Generation (LCG)*. Specifically, given an arbitrary image, the model generates subsequent layers conditioned on the existing ones. The experimental results are shown in Figure 4.9. We observe that whether the reference image is provided as an original image with one or two layers or generated from scratch (given zero layers), the recurrent structure successfully preserves the content of the previous layer while adding a new object that aligns with the caption description. Additionally, it effectively simulates new lighting and shadow effects. Afterwards, the experiments are conducted without any given layers.

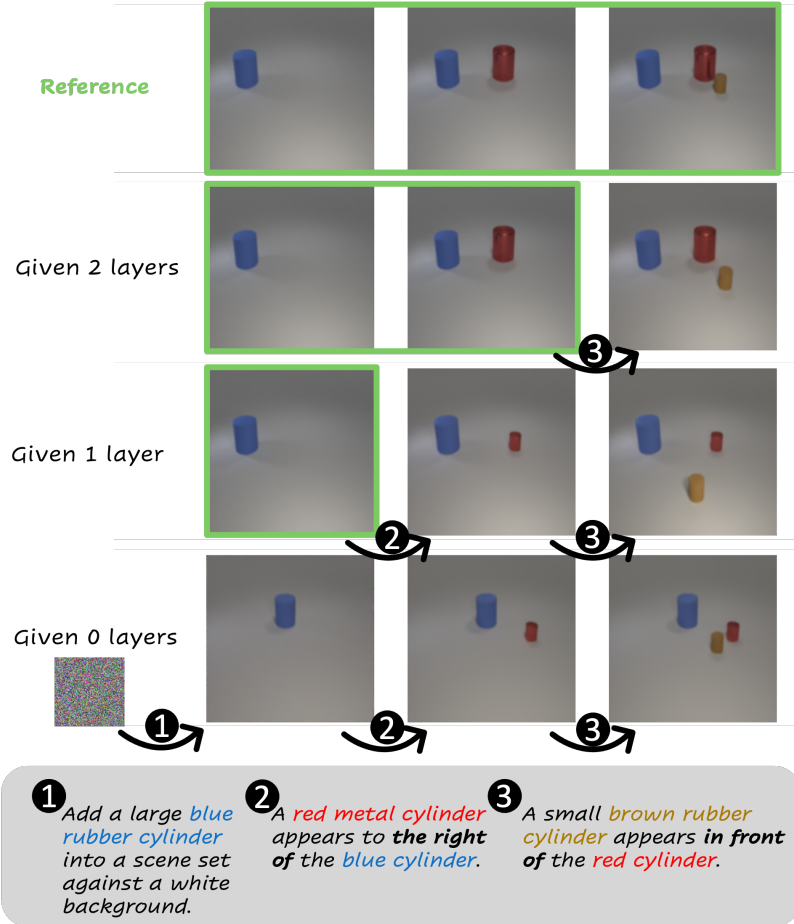


Figure 4.9: Results of a three-layer image generated by LayerGen. The first row presents three reference images, each paired with three instruction-based prompts at the bottom. The following three rows illustrate the generated images when conditioned on a decreasing number of reference layers.

Layer-Level Editing

Further, we investigate questions (2) to examine whether the model supports flexible layer editing. Since our training process only utilizes three-layer images, it is crucial to determine whether the model can generate additional layers during sampling. To test this, we add a fourth layer (see Fig. 4.10) and observe that the generation quality remains high.

However, removing an object is relatively challenging, as it requires reverting to the layer before the object was added and regenerating subsequent layers while preserving other objects. Since our model uses the latent transition variable $\mathbf{h}$ to store intermediate states during training, removing an object does not alter the positions of other objects, even if they belong to later layers. This demonstrates that our model effectively maintains weak causality between layers in the latent space, enabling flexible layer editing.

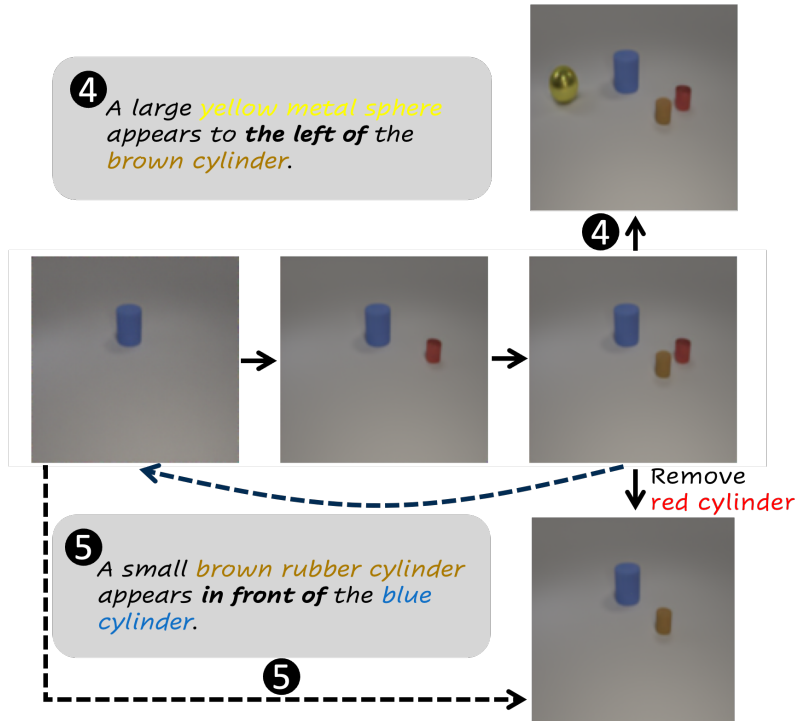


Figure 4.10: The demonstration of layer-level image editing by prompts.

Object-Level Editing

A highly editable model must not only maintain flexible relationships between layers but also allow fine-grained adjustments to individual objects. Therefore, we conduct experiments to validate question (3). Our model enables object-level editing by simply modifying the prompt, allowing changes to object attributes (see Fig. 4.11). Notably, such object-specific edits do not affect other objects in the scene, further demonstrating the model’s strong control over localized modifications.

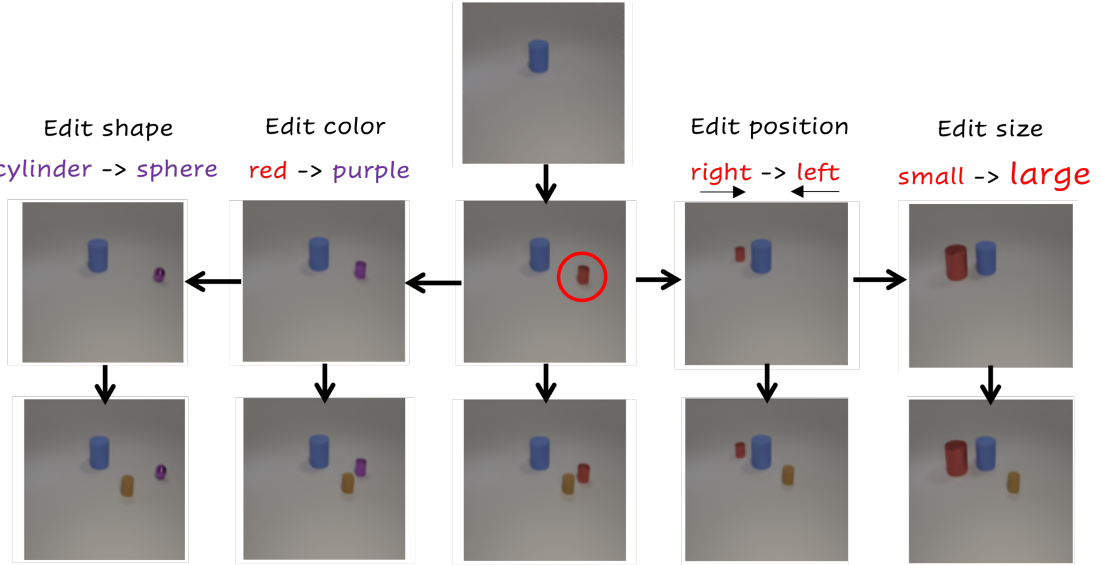


Figure 4.11: The results of object-level **continuous** image editing via prompts.

Realistic Image Generation

We integrate the model architecture shown in Fig. 1 with the pretrained U-Net structure from Stable Diffusion 2.1 [2]. To incorporate the transition variable $\mathbf{h}$, we modify the input and output layers of the U-Net, specifically the `unet.conv_in` and `unet.conv_out` layers. We fine-tune the model on the MULAN dataset (512×512 resolution), and given the

first two layers, we generate the third layer based on the provided prompt. The experimental results are shown in Fig. 4.12. Unfortunately, although our model is able to add new objects at the correct locations without disrupting unrelated pixels, it fails to generate the new objects accurately. On the one hand, we attribute this limitation to the small batch size during training. When using an 8-GPU node, we are only able to achieve a batch size of 4 on the MULAN dataset. On the other hand, the limitation also stems from the architectural mismatch: the pretrained U-Net is designed for input and output of the same size, namely the latent image variable, $\mathbf{z}$, whereas our model, modified based on diffusion forcing, takes both $\mathbf{z}$ and the $\mathbf{h}$ as input, but only outputs $\mathbf{h}$. A lightweight network is then used to decode $\mathbf{z}$ from $\mathbf{h}$. This architecture poses challenges when modeling complex distributions such as realistic images, indicating the need for exploring new network designs.



Figure 4.12: Two examples of generated 3-layer image with the captions from **Instruction MULAN** validation dataset.

4.2.6 Conclusion

In this work, we presented a novel framework for layered image generation based on Latent Diffusion Forcing, enabling structured and interpretable scene composition through autoregressive reasoning. Our model supports object-level control via sub-prompts, allowing flexible editing operations such as addition, removal, and transformation of individual scene elements while maintaining spatial and semantic consistency. Unlike previous approaches that rely on rigid encoder modifications or predefined masks, our method generalizes to arbitrary layer depths and reduces dependency on layout annotations. Despite certain limitations—such as challenges in generating visually realistic new objects and memory constraints during high-resolution training—our approach provides a promising step toward scalable, controllable, and causality-aware image synthesis. Future work will explore more expressive backbone architectures and memory-efficient training strategies to further improve realism and scalability.

Chapter 5

Conclusions

In this thesis, we introduced an information-theoretic perspective on diffusion modeling that reshapes how we understand and extend generative AI. Through Information-Theoretic Diffusion (ITD), we generalized the I-MMSE relation to establish a precise link between data distributions and optimal denoising, yielding exact probability estimation for both continuous and discrete data. This formulation refined density estimation, improved robustness through contrastive learning, and opened new avenues for principled integration of information theory with diffusion. Complementing this, our work on Information Decomposition demonstrated how mutual information and partial information decomposition provide rigorous tools for analyzing interpretability, compositionality, and cross-modal alignment. Together, these contributions move diffusion models beyond being viewed primarily as likelihood optimizers or score matchers, toward being understood as information processors.

The influence of this perspective is evident in subsequent research that has cited and extended these ideas. Diffusion PID [49] built directly on our decomposition framework,

applying partial information decomposition to deepen the interpretability of diffusion dynamics. Yu et al. [272] developed lift scores to improve compositional generalization, echoing our emphasis on mutual-information-driven compositionality. Wang et al. [259] formalized text-to-image alignment using information-theoretic objectives, offering a principled alternative to reinforcement learning-based methods and directly aligning with the motivations of ITD. Finally, ItDPDM [17] extended information-theoretic diffusion principles to discrete Poisson processes, illustrating the adaptability of our ideas to discrete domains. Collectively, these works demonstrate how the information-theoretic framing introduced in this thesis has already influenced research on interpretability, alignment, compositionality, and discrete generative modeling—helping to shape the emerging agenda in diffusion research.

Looking forward, several promising directions emerge from this foundation:

- **Information-Theoretic Post-Training:** Instead of retraining models from scratch, information-theoretic regularizers could be applied post-training to enhance robustness and alignment of large pretrained diffusion models. This would provide a lightweight yet principled means of adapting models to new domains.
- **Curated Latent Spaces for Compositionality and Efficiency:** By shaping latent representations using mutual information and decomposition criteria, future work can ensure that latents retain semantically meaningful factors while discarding redundancy. Such information-curated latent spaces would not only strengthen compositionality and controllability—by enabling disentangling and recombining across modalities—but also improve efficiency by reducing computational overhead while preserving fidelity.

- **Information-Theoretic Evaluation Metrics:** Building on our interpretability framework, new evaluation metrics grounded in mutual information and information decomposition could provide principled alternatives to heuristic scores (e.g., CLIP similarity, FID), yielding more transparent and reliable assessments of alignment and diversity in generative models.

In conclusion, this thesis demonstrates that diffusion models can be fruitfully reinterpreted through the lens of information theory, offering both theoretical advances and practical improvements. The follow-up works that have extended these ideas highlight the growing role of information-theoretic tools in shaping the future of generative AI. As the field continues to evolve, we anticipate that information theory will serve as a unifying foundation—bridging density estimation, interpretability, compositionality, and efficiency to guide the next generation of diffusion models.

Bibliography

- [1] Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.
- [2] Stability AI. Stable diffusion 2.1. <https://huggingface.co/stabilityai/stable-diffusion-2-1>, 2022. Accessed: 2025-04-01.
- [3] Juan Miguel Lopez Alcaraz and Nils Strodthoff. Diffusion-based conditional ecg generation with structured state space models. *Computers in biology and medicine*, 163:107115, 2023.
- [4] Shun-ichi Amari. *Information geometry and its applications*, volume 194. Springer, 2016.
- [5] Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.
- [6] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017.
- [7] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.
- [8] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 18187–18197. IEEE, June 2022.
- [9] 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.
- [10] Yogesh Balaji, Yogesh Balaji, Nah, Seungjun, Seungjun Nah, Huang, Xun, Xun Huang, Vahdat, Arash, Arash Vahdat, Song, Jiaming, Jiaming Song, Kreis, Karsten, Karsten Kreis, Aittala, Miika, Miika Aittala, Aila, Timo, Timo Aila, Laine, Samuli, Samuli

- Laine, Catanzaro, Bryan, Bryan Catanzaro, Karras, Tero, Tero Karras, Liu, Ming-Yu, and Mingyu Li. eDiff-I: Text-to-Image Diffusion Models with an Ensemble of Expert Denoisers. *arXiv preprint arXiv:2211.01324*, 2022.
- [11] Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. *Journal of machine learning research*, 6(10), 2005.
 - [12] Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. *Advances in Neural Information Processing Systems*, 36, 2024.
 - [13] Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie S Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. *arXiv preprint arXiv:2208.09392*, 2022.
 - [14] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *Proceedings of the 35th International Conference on Machine Learning*, pages 531–540, 2018.
 - [15] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International Conference on Machine Learning*, pages 531–540. PMLR, 2018.
 - [16] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science*. <https://cdn.openai.com/papers/dall-e-3.pdf>, 2(3):8, 2023.
 - [17] Sagnik Bhattacharya, Abhiram Gorle, Ahsan Bilal, Connor Ding, Amit Kumar Singh Yadav, and Tsachy Weissman. Itdpdm: Information-theoretic discrete poisson diffusion model, 2025.
 - [18] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.
 - [19] Blender Foundation. Blender - Open Source 3D Creation Suite. <https://www.blender.org/>. Accessed: 2025-01-06.
 - [20] Rob Brekelmans, Sicong Huang, Marzyeh Ghassemi, Greg Ver Steeg, Roger Grosse, and Alireza Makhzani. Improving mutual information estimation with annealed and energy-based bounds. *arXiv preprint arXiv:2303.06992*, 2023.
 - [21] Rob Brekelmans, Sicong Huang, Marzyeh Ghassemi, Greg Ver Steeg, Roger Baker Grosse, and Alireza Makhzani. Improving mutual information estimation with annealed

- and energy-based bounds. In *International Conference on Learning Representations*, 2021.
- [22] Rob Brekelmans, Vaden Masrani, Frank Wood, Greg Ver Steeg, and Aram Galstyan. All in the exponential family: Bregman duality in thermodynamic variational inference. In *International Conference on Machine Learning (ICML)*, 2020.
 - [23] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions, 2023.
 - [24] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
 - [25] Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Sae-hoon Kim. Coyo-700m: Image-text pair dataset. <https://github.com/kakaobrain/coyo-dataset>, 2022.
 - [26] Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S Yu, and Lichao Sun. A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt. *arXiv preprint arXiv:2303.04226*, 2023.
 - [27] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3558–3568, 2021.
 - [28] Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion, 2024.
 - [29] Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *arXiv preprint arXiv:2407.01392*, 2024.
 - [30] Jun Song Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James T. Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt- α : Fast Training of Diffusion Transformer for Photorealistic Text-to-Image Synthesis. *arXiv preprint arXiv:2310.00426*, 2023.
 - [31] Junya Chen, Danni Lu, Zidi Xiu, Ke Bai, Lawrence Carin, and Chenyang Tao. Variational inference with holder bounds. *arXiv preprint arXiv:2111.02947*, 2021.
 - [32] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, pages 6571–6583, 2018.

- [33] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.
- [34] Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022.
- [35] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.
- [36] Pengyu Cheng, Weituo Hao, and Lawrence Carin. Estimating total correlation with mutual information bounds. *NeurIPS Workshop on Deep Learning through Information Geometry*, 2020.
- [37] Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. Club: A contrastive log-ratio upper bound of mutual information. In *International Conference on Machine Learning*, pages 1779–1788. PMLR, 2020.
- [38] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- [39] Kristy Choi, Chenlin Meng, Yang Song, and Stefano Ermon. Density ratio estimation via infinitesimal classification. *arXiv preprint arXiv:2111.11010*, 2021.
- [40] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 10, 2022.
- [41] Thomas M Cover and Joy A Thomas. *Elements of information theory*. Wiley-Interscience, 2006.
- [42] Gavin E Crooks. Logistic approximation to the logistic-normal integral. *Technical Report Lawrence Berkeley National Laboratory*, 2009.
- [43] Paweł Czyż, Frederic Grabowski, Julia E. Vogt, Niko Beerenwinkel, and Alexander Marx. Beyond normal: On the evaluation of mutual information estimators, 2023.
- [44] Xiaoliang Dai, Ji Hou, Chih-Yao Ma, Sam Tsai, Jialiang Wang, Rui Wang, Peizhao Zhang, Simon Vandenhende, Xiaofang Wang, Abhimanyu Dubey, et al. Emu: Enhancing image generation models using photogenic needles in a haystack. *arXiv preprint arXiv:2309.15807*, 2023.

- [45] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pages 7480–7512. PMLR, 2023.
- [46] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [47] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.
- [48] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [49] Shaurya Dewan, Rushikesh Zawar, Prakanshul Saxena, Yingshan Chang, Andrew F. Luo, and Yonatan Bisk. Diffusionpid: Interpreting diffusion via partial information decomposition. *ArXiv*, abs/2406.05191, 2024.
- [50] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34, 2021.
- [51] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.
- [52] Ming Ding, Wendi Zheng, Wenyi Hong, and Jie Tang. Cogview2: Faster and better text-to-image generation via hierarchical transformers. *Advances in Neural Information Processing Systems*, 35:16890–16902, 2022.
- [53] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation, 2015.
- [54] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.
- [55] Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International conference on machine learning*, pages 1675–1685. PMLR, 2019.
- [56] Vincent Dumoulin, Ethan Perez, Nathan Schucher, Florian Strub, Harm de Vries, Aaron Courville, and Yoshua Bengio. Feature-wise transformations. *Distill*, 3(7):e11, 2018.
- [57] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

- [58] 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. In *Forty-first International Conference on Machine Learning*, 2024.
- [59] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. Data filtering networks. *arXiv preprint arXiv:2309.17425*, 2023.
- [60] Robert M Fano. *Transmission of Information: A Statistical Theory of Communications*. MIT Press, 1961.
- [61] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.
- [62] William Feller. On the theory of stochastic processes, with particular reference to applications. In *Selected Papers I*, pages 769–798. Springer, 2015.
- [63] Conor Finn and Joseph T Lizier. Pointwise partial information decomposition using the specificity and ambiguity lattices. *Entropy*, 20(4):297, 2018.
- [64] Jean Baptiste Joseph Fourier. *Théorie analytique de la chaleur*, volume 504. Didot Paris, 1822.
- [65] Giulio Franzese, Mustapha Bounoua, and Pietro Michiardi. Minde: Mutual information neural diffusion estimation, 2023.
- [66] Giulio Franzese, Mustapha Bounoua, and Pietro Michiardi. Minde: Mutual information neural diffusion estimation, 2024.
- [67] Nial Friel, Merrilee Hurn, and Jason Wyse. Improving power posterior estimation of statistical evidence. *Statistics and Computing*, 24(5):709–723, 2014.
- [68] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *European Conference on Computer Vision*, pages 89–106. Springer, 2022.
- [69] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Mdtv2: Masked diffusion transformer is a strong image synthesizer. *arXiv preprint arXiv:2303.14389*, 2023.
- [70] Shuyang Gao, Greg Ver Steeg, and Aram Galstyan. Efficient estimation of mutual information for strongly dependent variables. In *Artificial intelligence and statistics*, pages 277–286. PMLR, 2015.
- [71] Yunhao Ge, Jiashu Xu, Brian Nlong Zhao, Neel Joshi, Laurent Itti, and Vibhav Vineet. Beyond generation: Harnessing text to image models for object detection and segmentation. *arXiv preprint arXiv:2309.05956*, 2023.

- [72] Andrew Gelman and Xiao-Li Meng. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical science*, pages 163–185, 1998.
- [73] Dhruva Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36, 2024.
- [74] Aaron Gokaslan, A Feder Cooper, Jasmine Collins, Landan Seguin, Austin Jacobson, Mihir Patel, Jonathan Frankle, Cory Stephenson, and Volodymyr Kuleshov. Common-canvas: Open diffusion models trained on creative-commons images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8250–8260, 2024.
- [75] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.
- [76] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10696–10706, 2022.
- [77] Dongning Guo, Shlomo Shamai, and Sergio Verdú. Mutual information and minimum mean-square error in gaussian channels. *IEEE transactions on information theory*, 51(4):1261–1282, 2005.
- [78] Lanqing Guo, Yingqing He, Haoxin Chen, Menghan Xia, Xiaodong Cun, Yufei Wang, Siyu Huang, Yong Zhang, Xintao Wang, Qifeng Chen, Ying Shan, and Bihan Wen. Make a cheap scaling: A self-cascade diffusion model for higher-resolution adaptation, 2024.
- [79] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.
- [80] Michael Hanna, Ollie Liu, and Alexandre Variengien. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *arXiv preprint arXiv:2305.00586*, 2023.
- [81] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.
- [82] Xuehai He, Weixi Feng, Tsu-Jui Fu, Varun Jampani, Arjun Akula, Pradyumna Narayana, Sugato Basu, William Yang Wang, and Xin Eric Wang. Discriminative diffusion models as few-shot vision and language learners, 2023.

- [83] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.
- [84] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clip-score: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021.
- [85] 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.
- [86] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.
- [87] hlky. Diffusion1b. [<https://huggingface.co/datasets/bigdata-pw/Diffusion1B>] (<https://huggingface.co/datasets/bigdata-pw/Diffusion1B>), 2024.
- [88] Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *International Conference on Machine Learning*, pages 2722–2730. PMLR, 2019.
- [89] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [90] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [91] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- [92] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- [93] S Hochreiter. Long short-term memory. *Neural Computation MIT-Press*, 1997.
- [94] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34:12454–12465, 2021.
- [95] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.
- [96] Chin-Wei Huang, Jae Hyun Lim, and Aaron C Courville. A variational perspective on diffusion-based generative models and score matching. *Advances in Neural Information Processing Systems*, 34:22863–22876, 2021.

- [97] Runhui Huang, Kaixin Cai, Jianhua Han, Xiaodan Liang, Renjing Pei, Guansong Lu, Songcen Xu, Wei Zhang, and Hang Xu. Layerdiff: Exploring text-guided multi-layered composable image synthesis via layer-collaborative diffusion model, 2024.
- [98] Wenjing Huang, Shikui Tu, and Lei Xu. Pfb-diff: Progressive feature blending diffusion for text-driven image editing, 2023.
- [99] Sabine Hug, Michael Schwarzfischer, Jan Hasenauer, Carsten Marr, and Fabian J Theis. An adaptive scheduling scheme for calculating bayes factors with thermodynamic integration using simpson’s rule. *Statistics and Computing*, 26(3):663–677, 2016.
- [100] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.
- [101] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021.
- [102] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021.
- [103] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.
- [104] Edwin T Jaynes. *Probability theory: the logic of science*. Cambridge university press, 2003.
- [105] Yuanfeng Ji, Zhe Chen, Enze Xie, Lanqing Hong, Xihui Liu, Zhaoqiang Liu, Tong Lu, Zhenguo Li, and Ping Luo. Ddp: Diffusion model for dense visual prediction, 2023.
- [106] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.
- [107] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning, 2016.
- [108] Alexia Jolicoeur-Martineau, Ke Li, Rémi Piché-Taillefer, Tal Kachman, and Ioannis Mitliagkas. Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080*, 2021.
- [109] Minguk Kang, Jun-Yan Zhu, Richard Zhang, Jaesik Park, Eli Shechtman, Sylvain Paris, and Taesung Park. Scaling up gans for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10124–10134, 2023.

- [110] 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.
- [111] Laurynas Karazija, Iro Laina, Andrea Vedaldi, and Christian Rupprecht. Diffusion models for zero-shot open-vocabulary segmentation. *arXiv preprint arXiv:2306.09316*, 2023.
- [112] 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.
- [113] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364*, 2022.
- [114] Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. *arXiv preprint arXiv:2406.02507*, 2024.
- [115] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023.
- [116] Steven M Kay. *Fundamentals of statistical signal processing: estimation theory*. Prentice-Hall, Inc., 1993.
- [117] Amirhossein Kazerooni, Ehsan Khodapanah Aghdam, Moein Heidari, Reza Azad, Mohsen Fayyaz, Ilker Hacihaliloglu, and Dorit Merhof. Diffusion models in medical imaging: A comprehensive survey. *Medical Image Analysis*, 88:102846, 2023.
- [118] Saehoon Kim, Sanghun Cho, Chiheon Kim, Doyup Lee, and Woonhyuk Baek. mindall-e on conceptual captions. <https://github.com/kakaobrain/minDALL-E>, 2021.
- [119] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [120] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions, 2018.
- [121] Diederik P Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *arXiv preprint arXiv:2107.00630*, 2021.
- [122] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.
- [123] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

- [124] Michael Kleinman, Alessandro Achille, Stefano Soatto, and Jonathan C Kao. Redundant information neural estimation. *Entropy*, 23(7):922, 2021.
- [125] Michael Kleinman and Jonathon Kao. Redundant information neural estimation. *ICLR Workshop on Neural Compression*, 2021.
- [126] Artemy Kolchinsky. A novel approach to the partial information decomposition. *Entropy*, 24(3):403, 2022.
- [127] Xianghao Kong, Rob Brekelmans, and Greg Ver Steeg. Information-theoretic diffusion. *arXiv preprint arXiv:2302.03792*, 2023.
- [128] Xianghao Kong, Rob Brekelmans, and Greg Ver Steeg. Information-theoretic diffusion. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2022.
- [129] Xianghao Kong, Ollie Liu, Han Li, Dani Yogatama, and Greg Ver Steeg. Interpretable diffusion via information decomposition. *arXiv preprint arXiv:2310.07972*, 2023.
- [130] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- [131] Benno Krojer, Elinor Poole-Dayana, Vikram Voleti, Christopher Pal, and Siva Reddy. Are diffusion models vision-and-language reasoners? In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [132] Nicolas Lartillot and Hervé Philippe. Computing bayes factors using thermodynamic integration. *Systematic biology*, 55(2):195–207, 2006.
- [133] Minh-Quan Le, Tam V. Nguyen, Trung-Nghia Le, Thanh-Toan Do, Minh N. Do, and Minh-Triet Tran. Maskdiff: Modeling mask distribution with diffusion probabilistic model for few-shot instance segmentation, 2023.
- [134] Chaejeong Lee, Jayoung Kim, and Noseong Park. Codi: Co-evolving contrastive diffusion models for mixed-type tabular synthesis. *arXiv preprint arXiv:2304.12654*, 2023.
- [135] Nunzio Alexandro Letizia, Nicola Novello, and Andrea M Tonello. Mutual information estimation via f -divergence and data derangement based learning models.
- [136] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, pages 2177–2185, 2014.
- [137] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023.
- [138] Pengzhi Li, Qinxuan Huang, Yikang Ding, and Zhiheng Li. Layerdiffusion: Layered controlled image editing with diffusion models. In *SIGGRAPH Asia 2023 Technical Communications*, SA '23, New York, NY, USA, 2023. Association for Computing Machinery.

- [139] Xianhang Li, Haoqin Tu, Mude Hui, Zeyu Wang, Bingchen Zhao, Junfei Xiao, Sucheng Ren, Jieru Mei, Qing Liu, Huangjie Zheng, et al. What if we recaption billions of web images with llama-3? *arXiv preprint arXiv:2406.08478*, 2024.
- [140] Yunxiang Li, Xianghao Kong, Jiacheng Xie, Greg Ver Steeg, and You Zhang. Denoising diffusion wavelet models for zero-shot medical image translation. *Knowledge-Based Systems*, page 113800, 2025.
- [141] Paul Pu Liang, Zihao Deng, Martin Ma, James Zou, Louis-Philippe Morency, and Ruslan Salakhutdinov. Factorized contrastive learning: Going beyond multi-view redundancy. *arXiv preprint arXiv:2306.05268*, 2023.
- [142] Ruizhi Liao, Daniel Moyer, Polina Golland, and William M Wells. Demi: Discriminative estimator of mutual information. *arXiv preprint arXiv:2010.01766*, 2020.
- [143] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection, 2017.
- [144] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.
- [145] 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*, pages 740–755. Springer, 2014.
- [146] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023.
- [147] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code, 2024.
- [148] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.
- [149] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023.
- [150] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022.
- [151] Nan Liu, Yilun Du, Shuang Li, Joshua B Tenenbaum, and Antonio Torralba. Unsupervised compositional concepts discovery with text-to-image generative models. *arXiv preprint arXiv:2306.05357*, 2023.
- [152] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B. Tenenbaum. Compositional visual generation with composable diffusion models, 2023.

- [153] Xingchao Liu, Lemeng Wu, Mao Ye, and Qiang Liu. Let us build bridges: Understanding and extending diffusion generative models. *arXiv preprint arXiv:2208.14699*, 2022.
- [154] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017.
- [155] Chaofan Ma, Yuhuan Yang, Chen Ju, Fei Zhang, Jinxiang Liu, Yu Wang, Ya Zhang, and Yanfeng Wang. Diffusionseg: Adapting diffusion towards unsupervised object discovery, 2023.
- [156] Jiafeng Mao, Xueting Wang, and Kiyoharu Aizawa. Guided image synthesis via initial image editing in diffusion model. *arXiv preprint arXiv:2305.03382*, 2023.
- [157] Vaden Masrani, Rob Brekelmans, Thang Bui, Frank Nielsen, Aram Galstyan, Greg Ver Steeg, and Frank Wood. q-paths: Generalizing the geometric annealing path using power means. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2021.
- [158] Vaden Masrani, Tuan Anh Le, and Frank Wood. The thermodynamic variational objective. In *Advances in Neural Information Processing Systems*, pages 11521–11530, 2019.
- [159] Justin Matejka and George Fitzmaurice. Same stats, different graphs: generating datasets with varied appearance and identical statistics through simulated annealing. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pages 1290–1294, 2017.
- [160] David McAllester. On the mathematics of diffusion models. *arXiv preprint arXiv:2301.11108*, 2023.
- [161] David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. In *International Conference on Artificial Intelligence and Statistics*, pages 875–884. PMLR, 2020.
- [162] Sachin Mehta, Mohammad Hossein Sekhavat, Qingqing Cao, Maxwell Horton, Yanzi Jin, Chenfan Sun, Iman Mirzadeh, Mahyar Najibi, Dmitry Belenko, Peter Zatloukal, et al. Openelm: An efficient language model family with open-source training and inference framework. *arXiv preprint arXiv:2404.14619*, 2024.
- [163] Kangfu Mei, Zhengzhong Tu, Mauricio Delbracio, Hossein Talebi, Vishal M Patel, and Peyman Milanfar. Bigger is not always better: Scaling properties of latent diffusion models. *arXiv preprint arXiv:2404.01367*, 2024.
- [164] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations, 2022.
- [165] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023.

- [166] Sudipto Mukherjee, Himanshu Asnani, and Sreeram Kannan. Ccmi: Classifier based conditional mutual information estimation. In *Uncertainty in Artificial Intelligence*, pages 1083–1093. PMLR, 2020.
- [167] Radford M Neal. Annealed importance sampling. *Statistics and computing*, 11(2):125–139, 2001.
- [168] XuanLong Nguyen, Martin J Wainwright, and Michael Jordan. Estimating divergence functionals and the likelihood ratio by penalized convex risk minimization. In *Advances in Neural Information Processing Systems*, 2007.
- [169] 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.
- [170] 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.
- [171] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.
- [172] Frank Nielsen. What is an information projection. *Notices of the AMS*, 65(3):321–324, 2018.
- [173] Yoshihiko Ogata. A monte carlo method for high dimensional integration. *Numerische Mathematik*, 55(2):137–157, 1989.
- [174] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- [175] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *Advances in neural information processing systems*, 2018.
- [176] OpenAI. Hello gpt-4o. <https://openai.com/index/hello-gpt-4o/>, 2024.
- [177] Yidong Ouyang, Liyan Xie, and Guang Cheng. Improving adversarial robustness by contrastive guided diffusion process. *arXiv preprint arXiv:2210.09643*, 2022.
- [178] Daniel P Palomar and Sergio Verdú. Gradient of mutual information in linear vector gaussian channels. *IEEE Transactions on Information Theory*, 52(1):141–154, 2005.
- [179] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11410–11420, 2022.

- [180] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
- [181] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.
- [182] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.
- [183] Shirley Pepke and Greg Ver Steeg. Comprehensive discovery of subsample gene expression components by information explanation: therapeutic implications in cancer. *BMC medical genomics*, 10(1):12, 2017.
- [184] Pablo Pernias, Dominic Rampas, Mats L. Richter, Christopher J. Pal, and Marc Aubreville. Wuerstchen: An Efficient Architecture for Large-Scale Text-to-Image Diffusion Models. *ICLR*, 2024.
- [185] Dustin Podell, Zion English, Karen Lacey, Andreas Blattmann, Tim Dockhorn, Judith Müller, Joe Penna, and Robin Rombach. SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. *ICLR*, 2024.
- [186] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.
- [187] Zipeng Qi, Guoxi Huang, Chenyang Liu, and Fei Ye. Layered rendering diffusion model for controllable zero-shot image synthesis, 2024.
- [188] 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*, pages 8748–8763. PMLR, 2021.
- [189] 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*, pages 8748–8763. PmLR, 2021.
- [190] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.
- [191] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016.

- [192] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.
- [193] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pages 5301–5310. PMLR, 2019.
- [194] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.
- [195] 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.
- [196] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021.
- [197] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506, 2020.
- [198] 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. *arXiv preprint arXiv:2306.08877*, 2023.
- [199] Kyle Reing, Greg Ver Steeg, and Aram Galstyan. Discovering higher-order interactions through neural information decomposition. *Entropy*, 23(1):79, 2021.
- [200] Kyle Reing, Greg Ver Steeg, and Aram Galstyan. Influence decompositions for neural network attribution. In *The 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.
- [201] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows, 2016.
- [202] Benjamin Rhodes, Kai Xu, and Michael U Gutmann. Telescoping density-ratio estimation. *Advances in neural information processing systems*, 33:4905–4916, 2020.
- [203] 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*, pages 10684–10695, 2022.

- [204] 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*, pages 10684–10695, 2022.
- [205] 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*, pages 10684–10695, 2022.
- [206] 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, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.
- [207] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.
- [208] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.
- [209] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.
- [210] Guillaume Sanchez, Honglu Fan, Alexander Spangher, Elad Levi, Pawan Sasanka Ammanamanchi, and Stella Biderman. Stay on topic with classifier-free guidance. *arXiv preprint arXiv:2306.17806*, 2023.
- [211] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023.
- [212] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.
- [213] Vikash Sehwal, Caner Hazirbas, Albert Gordo, Firat Ozgenel, and Cristian Canton. Generating high fidelity data from low-density regions using diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11492–11501, 2022.
- [214] Vikash Sehwal, Xianghao Kong, Jingtao Li, Michael Spranger, and Lingjuan Lyu. Stretching each dollar: Diffusion training from scratch on a micro-budget. *arXiv preprint arXiv:2407.15811*, 2024.

- [215] Yuval Shalev, Amichai Painsky, and Irad Ben-Gal. Neural joint entropy estimation. *arXiv preprint arXiv:2012.11197*, 2020.
- [216] Shlomo Shamai and Sergio Verdu. Worst-case power-constrained noise for binary-input channels. *IEEE Transactions on Information Theory*, 38(5):1494–1511, 1992.
- [217] C.E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 1948.
- [218] Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.
- [219] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarczyk, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.
- [220] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioning with reading comprehension. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 742–758. Springer, 2020.
- [221] Leslie N Smith. Cyclical learning rates for training neural networks. arxiv. *Preprint at https://arxiv.org/abs/1506.01186*, 2015.
- [222] Leslie N Smith. General cyclical training of neural networks. *arXiv preprint arXiv:2202.08835*, 2022.
- [223] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.
- [224] Jascha Sohl-Dickstein, Eric A Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *arXiv preprint arXiv:1503.03585*, 2015.
- [225] Jiaming Song and Stefano Ermon. Understanding the limitations of variational mutual information estimators. In *International Conference on Learning Representations*, 2019.
- [226] Jiaming Song and Stefano Ermon. Multi-label contrastive predictive coding. *Advances in Neural Information Processing Systems*, 33, 2020.
- [227] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- [228] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- [229] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022.

- [230] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.
- [231] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.
- [232] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution, 2020.
- [233] 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.
- [234] 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.
- [235] 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.
- [236] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations*, 2021.
- [237] Xuan Su, Jiaming Song, Chenlin Meng, and Stefano Ermon. Dual diffusion implicit bridges for image-to-image translation. *arXiv preprint arXiv:2203.08382*, 2022.
- [238] Keqiang Sun, Junting Pan, Yuying Ge, Hao Li, Haodong Duan, Xiaoshi Wu, Renrui Zhang, Aojun Zhou, Zipeng Qin, Yi Wang, et al. Journeydb: A benchmark for generative image understanding. *Advances in Neural Information Processing Systems*, 36, 2024.
- [239] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks, 2017.
- [240] Lintang Sutawika, Aran Komatsuzaki, and Colin Raffel. Pile-t5, 2024. Blog post.
- [241] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.
- [242] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.
- [243] Weimin Tan, Siyuan Chen, and Bo Yan. Diffss: Diffusion model for few-shot semantic segmentation, 2023.

- [244] Raphael Tang, Linqing Liu, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Pontus Stenetorp, Jimmy Lin, and Ferhan Ture. What the daam: Interpreting stable diffusion using cross attention, 2022.
- [245] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csd: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34:24804–24816, 2021.
- [246] Mosaic ML Team. streaming. <<https://github.com/mosaicml/streaming/>>, 2022.
- [247] Jiayan Teng, Wendi Zheng, Ming Ding, Wenyi Hong, Jianqiao Wangni, Zhuoyi Yang, and Jie Tang. Relay diffusion: Unifying diffusion process across resolutions for image synthesis. *arXiv preprint arXiv:2309.03350*, 2023.
- [248] Junjiao Tian, Lavisha Aggarwal, Andrea Colaco, Zsolt Kira, and Mar Gonzalez-Franco. Diffuse, attend, and segment: Unsupervised zero-shot segmentation using stable diffusion, 2023.
- [249] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- [250] Yao-Hung Hubert Tsai, Han Zhao, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. Neural methods for point-wise dependency estimation. *Advances in Neural Processing Systems*, 2020.
- [251] Petru-Daniel Tudosiu, Yongxin Yang, Shifeng Zhang, Fei Chen, Steven McDonagh, Gerasimos Lampouras, Ignacio Iacobacci, and Sarah Parisot. Mulan: A multi layer annotated dataset for controllable text-to-image generation, 2024.
- [252] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34, 2021.
- [253] Alina Valyaeva. AI Has Already Created As Many Images As Photographers Have Taken in 150 Years. Statistics for 2023. <https://journal.everyapixel.com/ai-image-statistics>, 2023. [Online; accessed 11-July-2024].
- [254] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- [255] 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.
- [256] Sergio Verdú and Dongning Guo. A simple proof of the entropy-power inequality. *IEEE Transactions on Information Theory*, 52(5):2165–2166, 2006.
- [257] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

- [258] Chao Wang, Giulio Franzese, Alessandro Finamore, Massimo Gallo, and Pietro Michiardi. Information theoretic text-to-image alignment. *arXiv preprint arXiv:2405.20759*, 2024.
- [259] Chao Wang, Giulio Franzese, Alessandro Finamore, Massimo Gallo, and Pietro Michiardi. Information theoretic text-to-image alignment, 2025.
- [260] Jinglong Wang, Xiawei Li, Jing Zhang, Qingyuan Xu, Qin Zhou, Qian Yu, Lu Sheng, and Dong Xu. Diffusion model is secretly a training-free open vocabulary semantic segmenter, 2023.
- [261] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022.
- [262] Liming Wang, David Edwin Carlson, Miguel RD Rodrigues, Robert Calderbank, and Lawrence Carin. A bregman matrix and the gradient of mutual information for vector poisson and gaussian channels. *IEEE Transactions on Information Theory*, 60(5):2611–2629, 2014.
- [263] Liming Wang, Miguel Rodrigues, and Lawrence Carin. Generalized bregman divergence and gradient of mutual information for vector poisson channels. In *2013 IEEE International Symposium on Information Theory*, pages 454–458. IEEE, 2013.
- [264] Qian Wang, Biao Zhang, Michael Birsak, and Peter Wonka. Instructedit: Improving automatic masks for diffusion-based image editing with user instructions, 2023.
- [265] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023.
- [266] Zijie J Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models. *arXiv preprint arXiv:2210.14896*, 2022.
- [267] P.L. Williams and R.D. Beer. Nonnegative decomposition of multivariate information. *arXiv:1004.2515*, 2010.
- [268] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models, 2023.
- [269] Serin Yang, Hyunmin Hwang, and Jong Chul Ye. Zero-shot contrastive loss for text-guided diffusion image style transfer. *arXiv preprint arXiv:2303.08622*, 2023.
- [270] Guy Yariv, Itai Gat, Lior Wolf, Yossi Adi, and Idan Schwartz. Audiotoken: Adaptation of text-conditioned diffusion models for audio-to-image generation. *arXiv preprint arXiv:2305.13050*, 2023.

- [271] Mao Ye, Lemeng Wu, and Qiang Liu. First hitting diffusion models. *arXiv preprint arXiv:2209.01170*, 2022.
- [272] Chenning Yu and Sicun Gao. Improving compositional generation with diffusion models using lift scores, 2025.
- [273] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *Transactions on Machine Learning Research*, 2022.
- [274] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bag-of-words models, and what to do about it? *arXiv preprint arXiv:2210.01936*, 2022.
- [275] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014.
- [276] Lvmin Zhang and Maneesh Agrawala. Transparent image layer diffusion using latent transparency, 2024.
- [277] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.
- [278] Manlin Zhang, Jie Wu, Yuxi Ren, Ming Li, Jie Qin, Xuefeng Xiao, Wei Liu, Rui Wang, Min Zheng, and Andy J. Ma. Diffusionengine: Diffusion model is scalable data engine for object detection, 2023.
- [279] Xinyang Zhang, Wentian Zhao, Xin Lu, and Jeff Chien. Text2layer: Layered image generation using latent diffusion model, 2023.
- [280] Linxi Zhao, Yihe Deng, Weitong Zhang, and Quanquan Gu. Mitigating object hallucination in large vision-language models via classifier-free guidance. *arXiv preprint arXiv:2402.08680*, 2024.
- [281] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders. *CoRR*, abs/1706.02262, 2017.
- [282] Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. Unleashing text-to-image diffusion models for visual perception, 2023.
- [283] Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers. *Transactions on Machine Learning Research*, 2024.

- [284] Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022.
- [285] Ye Zhu, Yu Wu, Kyle Olszewski, Jian Ren, Sergey Tulyakov, and Yan Yan. Discrete contrastive diffusion for cross-modal and conditional generation. *arXiv preprint arXiv:2206.07771*, 2022.
- [286] Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. St-moe: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*, 2022.

Appendix A

Appendix for Chapter 2

A.1 Derivations

A.1.1 Derivation of the Fundamental Pointwise Denoising Relation

We now derive the following pointwise denoising relation.

$$\frac{d}{d\gamma} D_{KL}[p(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})] = 1/2 \text{mmse}(\mathbf{x}, \gamma)$$

Recall the definition of pointwise MMSE, except we will use a shorthand notation for the analytic MMSE denoiser.

$$\begin{aligned} \text{mmse}(\mathbf{x}, \gamma) &\equiv \mathbb{E}_{p(\mathbf{z}|\mathbf{x})}[\|\mathbf{x} - \mathbb{E}[\mathbf{x}|\mathbf{z}]\|_2^2] \\ \mathbb{E}[\mathbf{x}|\mathbf{z}] &\equiv \int d\bar{\mathbf{x}} \, p(\bar{\mathbf{x}}|\mathbf{z}) \, \bar{\mathbf{x}} = \int d\bar{\mathbf{x}} \, p(\mathbf{z}|\bar{\mathbf{x}})p(\bar{\mathbf{x}})/p(\mathbf{z}) \, \bar{\mathbf{x}} \end{aligned}$$

We begin by expanding the left-hand side.

$$\begin{aligned}
\frac{d}{d\gamma} D_{KL}[p(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})] &= \frac{d}{d\gamma} \mathbb{E}_{p(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{z}|\mathbf{x})] - \frac{d}{d\gamma} \mathbb{E}_{p(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{z})] \\
&= -\frac{d}{d\gamma} \mathbb{E}_{p(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{z})] = -\int d\mathbf{z} \frac{d}{d\gamma} [p(\mathbf{z}|\mathbf{x}) \log p(\mathbf{z})] \\
&= -\int d\mathbf{z} \left(\frac{d}{d\gamma} p(\mathbf{z}|\mathbf{x}) [\log p(\mathbf{z})] + p(\mathbf{z}|\mathbf{x}) \frac{d}{d\gamma} \log p(\mathbf{z}) \right)
\end{aligned}$$

The first term in the first line is a constant that does not depend on γ . Then we expand with the product rule. The following is an easily verified identity for our Gaussian noise channel, $p(\mathbf{z}|\mathbf{x})$.

$$\frac{d}{d\gamma} p(\mathbf{z}|\mathbf{x}) = -\mathbf{x}/(2\sqrt{\gamma}) \cdot \nabla_{\mathbf{z}} p(\mathbf{z}|\mathbf{x})$$

We also need the following relation.

$$\begin{aligned}
\frac{d}{d\gamma} \log p(\mathbf{z}) &= 1/p(\mathbf{z}) \frac{d}{d\gamma} p(\mathbf{z}) = 1/p(\mathbf{z}) \int d\bar{\mathbf{x}} \frac{d}{d\gamma} p(\mathbf{z}|\bar{\mathbf{x}}) p(\bar{\mathbf{x}}) \\
&= -1/(2\sqrt{\gamma}) \frac{1}{p(\mathbf{z})} \int d\bar{\mathbf{x}} \bar{\mathbf{x}} \cdot \nabla_{\mathbf{z}} p(\mathbf{z}|\bar{\mathbf{x}}) p(\bar{\mathbf{x}})
\end{aligned}$$

Using these expressions in the derivation above, we get the following.

$$\begin{aligned}
\frac{d}{d\gamma} D_{KL}[p(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})] &= \int d\mathbf{z} \left(\mathbf{x}/(2\sqrt{\gamma}) \cdot \nabla_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}) [\log p(\mathbf{z})] \right. \\
&\quad \left. + 1/(2\sqrt{\gamma}) \frac{p(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})} \int d\bar{\mathbf{x}} \bar{\mathbf{x}} \cdot \nabla_{\mathbf{z}} p(\mathbf{z}|\bar{\mathbf{x}}) p(\bar{\mathbf{x}}) \right)
\end{aligned}$$

Next we use integration by parts on $\mathbf{z}$.

$$\begin{aligned}
\frac{d}{d\gamma} D_{KL}[p(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})] &= - \int d\mathbf{z} \left(1/2 \, p(\mathbf{z}|\mathbf{x}) \, \mathbf{x}/\sqrt{\gamma} \cdot \nabla_{\mathbf{z}} \log p(\mathbf{z}) \right. \\
&\quad \left. + 1/(2\sqrt{\gamma}) \, 1/p(\mathbf{x}) \int d\bar{\mathbf{x}} \, p(\mathbf{z}|\bar{\mathbf{x}}) p(\bar{\mathbf{x}}) \, \bar{\mathbf{x}} \cdot \nabla_{\mathbf{z}} p(\mathbf{x}|\mathbf{z}) \right) \\
&= - \int d\mathbf{z} \left(1/2 \, p(\mathbf{z}|\mathbf{x}) \, \mathbf{x}/\sqrt{\gamma} \cdot \nabla_{\mathbf{z}} \log p(\mathbf{z}) \right. \\
&\quad \left. + 1/(2\sqrt{\gamma}) \, p(\mathbf{z})/p(\mathbf{x}) \mathbb{E}[\mathbf{x}|\mathbf{z}] \cdot \nabla_{\mathbf{z}} p(\mathbf{x}|\mathbf{z}) \right)
\end{aligned}$$

The gradients can be written,

$$\nabla_{\mathbf{z}} p(\mathbf{x}|\mathbf{z}) = p(\mathbf{x}|\mathbf{z}) \sqrt{\gamma} (\mathbf{x} - \mathbb{E}[\mathbf{x}|\mathbf{z}])$$

$$\nabla_{\mathbf{z}} \log p(\mathbf{z}) = \sqrt{\gamma} \mathbb{E}[\mathbf{x}|\mathbf{z}] - \mathbf{z}.$$

$$\begin{aligned}
\frac{d}{d\gamma} D_{KL}[p(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})] &= \int d\mathbf{z} \left(\frac{1}{2} p(\mathbf{z}|\mathbf{x}) \mathbf{x}/\sqrt{\gamma} \cdot (\mathbf{z} - \sqrt{\gamma}\mathbb{E}[\mathbf{x}|\mathbf{z}]) \right. \\
&\quad \left. - 1/(2\sqrt{\gamma}) p(\mathbf{z}|\mathbf{x}) \mathbb{E}[\mathbf{x}|\mathbf{z}] \cdot \sqrt{\gamma}(\mathbf{x} - \mathbb{E}[\mathbf{x}|\mathbf{z}]) \right) \\
&= \frac{1}{2} \int d\mathbf{z} p(\mathbf{z}|\mathbf{x}) \left(-\mathbf{x} \cdot \mathbb{E}[\mathbf{x}|\mathbf{z}] + \mathbf{x} \cdot \mathbf{z}/\sqrt{\gamma} \right. \\
&\quad \left. - \mathbb{E}[\mathbf{x}|\mathbf{z}] \cdot (\mathbf{x} - \mathbb{E}[\mathbf{x}|\mathbf{z}]) \right) \\
&= \frac{1}{2} \int d\mathbf{z} p(\mathbf{z}|\mathbf{x}) \left(-\mathbf{x} \cdot \mathbb{E}[\mathbf{x}|\mathbf{z}] + \mathbf{x} \cdot \mathbf{x} \right. \\
&\quad \left. - \mathbb{E}[\mathbf{x}|\mathbf{z}] \cdot (\mathbf{x} - \mathbb{E}[\mathbf{x}|\mathbf{z}]) \right) \\
&= \frac{1}{2} \int d\mathbf{z} p(\mathbf{z}|\mathbf{x}) \|\mathbf{x} - \mathbb{E}[\mathbf{x}|\mathbf{z}]\|_2^2 \\
&= \frac{1}{2} \text{mmse}(\mathbf{x}, \gamma) \quad \square
\end{aligned}$$

A.1.2 High SNR KL divergence limit

In this appendix, we derive the following result.

$$\lim_{\gamma \rightarrow \infty} D_{KL}[p(\mathbf{z}|\mathbf{x}) \parallel p_G(\mathbf{z})] - D_{KL}[p(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})] = \log \frac{p(\mathbf{x})}{p_G(\mathbf{x})}$$

In this expression, we have two “base distributions”, $p(\mathbf{x}), p_G(\mathbf{x})$, and we consider the marginal distributions after injecting Gaussian noise, $p_G(\mathbf{z}) = \int p(\mathbf{z}|\mathbf{x}) p_G(\mathbf{x}) d\mathbf{x}$, $p(\mathbf{z}) = \int p(\mathbf{z}|\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$. Start by expanding and canceling out terms.

$$\begin{aligned}
f(\mathbf{x}, \gamma) &\equiv D_{KL}[p(\mathbf{z}|\mathbf{x}) \parallel p_G(\mathbf{z})] - D_{KL}[p(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})] && \text{(Cancel log } p(\mathbf{z}|\mathbf{x}) \text{ terms)} \\
&= \mathbb{E}_{p(\mathbf{z}|\mathbf{x})} [\log ((p(\mathbf{z})\gamma^{d/2})/(p_G(\mathbf{z})\gamma^{d/2}))] && \text{(Multiply by 1)} \\
&= \mathbb{E}_{p(\epsilon)} [\log ((p(\mathbf{z} = \sqrt{\gamma}\mathbf{x} + \epsilon)\gamma^{d/2})/(p_G(\mathbf{z} = \sqrt{\gamma}\mathbf{x} + \epsilon)\gamma^{d/2}))] && \text{(Reparametrize)}
\end{aligned}$$

Set $Z = (2\pi)^{d/2}$ for $\mathbf{x} \in \mathbb{R}^d$ and re-arrange.

$$\begin{aligned} p(\mathbf{z})\gamma^{d/2} &= \int d\bar{\mathbf{x}} \, p(\bar{\mathbf{x}}) 1/Z \, e^{-1/2(\mathbf{z}-\sqrt{\gamma}\bar{\mathbf{x}})^2} \gamma^{d/2} \\ &= \int d\bar{\mathbf{x}} \, p(\bar{\mathbf{x}}) 1/Z \, e^{-1/2(\mathbf{z}/\sqrt{\gamma}-\bar{\mathbf{x}})^2} \gamma^{d/2} \end{aligned}$$

For large γ , we recognize a limit representation of the Dirac delta function in blue.

$$\begin{aligned} p(\mathbf{z} = \sqrt{\gamma}\mathbf{x} + \epsilon)\gamma^{d/2} &= \int d\bar{\mathbf{x}} \, p(\bar{\mathbf{x}}) 1/Z \, e^{-1/2(\mathbf{x}+\epsilon/\sqrt{\gamma}-\bar{\mathbf{x}})^2} \gamma^{d/2} \\ \lim_{\gamma \rightarrow \infty} p(\mathbf{z} = \sqrt{\gamma}\mathbf{x} + \epsilon)\gamma^{d/2} &= \int d\bar{\mathbf{x}} \, p(\bar{\mathbf{x}}) \delta(\mathbf{x} - \bar{\mathbf{x}}) \\ &= p(\mathbf{x}) \end{aligned}$$

Using this result leads to the desired result.

$$\begin{aligned} \lim_{\gamma \rightarrow \infty} f(\mathbf{x}, \gamma) &= \lim_{\gamma \rightarrow \infty} \mathbb{E}_{p(\epsilon)} [\log p(\mathbf{z} = \sqrt{\gamma}\mathbf{x} + \epsilon)\gamma^{d/2} - \log p(\mathbf{z} = \sqrt{\gamma}\mathbf{x} + \epsilon)\gamma^{d/2}] \\ &= \log p(\mathbf{x})/p_G(\mathbf{x}) \quad \square \end{aligned}$$

This informal proof using delta functions could be made more rigorous in a measure-theoretic setting. Once again, we have derived a point-wise generalization of Eq. 177 from [77], which can be recovered by taking the expectation of our result.

A.1.3 Simplifying Density with an Integral Identity

Our goal is to simplify the expression of the density coming from Eq. equation B.5.

$$\begin{aligned}
-\log p(\mathbf{x}) &= -\log p_G(\mathbf{x}) - 1/2 \int_0^\infty d\gamma (\text{mmse}_G(\mathbf{x}, \gamma) - \text{mmse}(\mathbf{x}, \gamma)) \\
&= d/2 \log(2\pi) + x^2/2 - 1/2 \int_0^\infty d\gamma \left(\frac{x^2 + \gamma d}{(1 + \gamma)^2} - \text{mmse}(\mathbf{x}, \gamma) \right) \\
&= d/2 \log(2\pi e) - 1/2 \int_0^\infty d\gamma \left(\frac{d}{1 + \gamma} - \text{mmse}(\mathbf{x}, \gamma) \right)
\end{aligned}$$

In the second line we write out the expressions. Note the the pointwise MMSE for a standard normal input is derived in more detail in App. A.1.5. In the third line we make use of the following integral identity.

$$\int_0^\infty \left(\frac{x^2 + \gamma d}{(1 + \gamma)^2} - \frac{d}{1 + \gamma} \right) d\gamma = x^2 - d$$

This identity can be verified via elementary manipulations (multiply second term in integrand by $(1 + \gamma)/(1 + \gamma)$), but we state it explicitly because of its counter-intuitive form.

A.1.4 Non-Gaussian Density Representation

Instead of using $p_G(\mathbf{x}) = \mathcal{N}(\mathbf{x}; 0, \mathbb{I})$, we can take the base measure to be a Gaussian, $p_G(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$, with mean and covariance that match the data. Let $\lambda_1, \dots, \lambda_d$ be the eigenvalues of the covariance matrix. We could start with the derivation in App. A.1.3 and try to simplify in a similar way with a more complex integral identity. However, a more straightforward derivation proceeds as follows.

Start with the simple density estimator from Eq. equation B.6 assuming a standard normal as the base measure, but replacing the factor of d with a sum over d terms.

$$-\log p(\mathbf{x}) = d/2 \log(2\pi e) - 1/2 \int_0^\infty d\gamma \left(\sum_{i=1}^d \frac{1}{1+\gamma} - \text{mmse}(\mathbf{x}, \gamma) \right)$$

Now, note the following integral identity:

$$\int_0^\infty \left(\frac{1}{\gamma + 1/\lambda_i} - \frac{1}{\gamma + 1} \right) d\gamma = \log \lambda_i.$$

Using this to replace $1/(\gamma + 1)$ terms in the previous expression gives the following.

$$-\log p(\mathbf{x}) = d/2 \log(2\pi e) + 1/2 \sum_{i=1}^d \log \lambda_i - 1/2 \int_0^\infty d\gamma \left(\sum_{i=1}^d \frac{1}{\gamma + 1/\lambda_i} - \text{mmse}(\mathbf{x}, \gamma) \right)$$

We recognize the sum of the log eigenvalues as equivalent to the log determinant. The constants can also be pulled inside the determinant.

$$-\log p(\mathbf{x}) = \underbrace{1/2 \log \det(2\pi e \mathbf{\Sigma})}_{\text{Gaussian entropy}} - \underbrace{1/2 \int_0^\infty d\gamma \left(\sum_{i=1}^d \frac{1}{\gamma + 1/\lambda_i} - \text{mmse}(\mathbf{x}, \gamma) \right)}_{\text{Deviation from Gaussianity} \geq 0}$$

Note that the deviation from Gaussianity need only be non-negative in expectation.

When we change variables of integration to $\alpha = \log \gamma$ we get the following.

$$\int_0^\infty d\gamma \sum_{i=1}^d \frac{1}{\gamma + 1/\lambda_i} = \int_{-\infty}^\infty d\alpha \sum_{i=1}^d \sigma(\alpha + \log \lambda_i) \quad (\text{A.1})$$

Here the traditional sigmoid function is used $\sigma(t) = 1/(1 + e^{-t})$.

A.1.5 Gaussian properties

Consider $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$, for some covariance matrix $\boldsymbol{\Sigma}$. Let $\boldsymbol{\Sigma} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$ be the SVD, with eigenvalues $\Lambda_{ii} = \lambda_i$. For this distribution we would like to derive the ground truth decoder and MMSE, both for testing purposes and to use as a fallback estimator in regions where our discovered decoder is clearly sub-optimal.

We have already mentioned in Eq. equation 2.3 that the ideal decoder is:

$$\hat{\mathbf{x}}^*(\mathbf{z}, \gamma) \equiv \arg \text{mmse}(\gamma) = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|\mathbf{z})}[\mathbf{x}].$$

For $p(\mathbf{x})$ a Gaussian distribution, we can simply look up this conditional mean in a textbook to find that the MMSE estimator is:

$$\begin{aligned} \hat{\mathbf{x}}_G^*(\mathbf{z}, \gamma) &= \boldsymbol{\mu} + \sqrt{\gamma}(\gamma\mathbb{I} + \boldsymbol{\Sigma}^{-1})^{-1}(\mathbf{z} - \sqrt{\gamma}\boldsymbol{\mu}) \\ &= \boldsymbol{\mu} + \sqrt{\gamma}\mathbf{U}(\gamma + \boldsymbol{\Lambda}^{-1})^{-1}\mathbf{U}^T(\mathbf{z} - \sqrt{\gamma}\boldsymbol{\mu}) \end{aligned}$$

Taking the expectation, $\mathbb{E}_{\mathbf{x}, \mathbf{z}}[(\mathbf{x} - \hat{\mathbf{x}}_G^*(\mathbf{z}, \gamma))^2]$, gives the MMSE with some manipulation:

$$\text{mmse}_G(\gamma) = \text{Tr}((\boldsymbol{\Sigma}^{-1} + \gamma\mathbb{I})^{-1}) = \sum_i \frac{1}{1/\lambda_i + \gamma}.$$

Note that we used the cyclic property of the trace to write the final expression in terms of the eigenvalues only. The case for the standard normal distribution [77] follows from setting the eigenvalues to 1.

The negative log likelihood is:

$$-\log p(\mathbf{x}) = 1/2 (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) + 1/2 \log \det(2\pi \boldsymbol{\Sigma})$$

It can be verified using the integral identities in App. A.1.4 and App. A.1.3 that using the Gaussian MMSE in Eq. equation B.6 recovers this equation.

A.1.6 MMSE Upper Bounds via Bregman Divergence Interpretation

In this section, we derive the upper bound in the objective using a suboptimal denoising function $\hat{\mathbf{x}}(\mathbf{z}, \gamma)$. The Bregman divergence generated from the strictly convex function $\phi(\mathbf{x}) = \frac{1}{2} \langle \mathbf{x}, \mathbf{x} \rangle = \frac{1}{2} \|\mathbf{x}\|_2^2$ is

$$d_\phi(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \langle \mathbf{x}, \mathbf{x} \rangle - \frac{1}{2} \langle \mathbf{y}, \mathbf{y} \rangle - \langle \mathbf{x} - \mathbf{y}, \nabla \phi(\mathbf{y}) \rangle = \frac{1}{2} \langle \mathbf{x}, \mathbf{x} \rangle - \frac{1}{2} \langle \mathbf{y}, \mathbf{y} \rangle - \langle \mathbf{x} - \mathbf{y}, \mathbf{y} \rangle = \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|_2^2.$$

Note that the Legendre dual of ϕ , or $\psi(\mathbf{y}) = \sup_{\mathbf{x}} \langle \mathbf{x}, \mathbf{y} \rangle - \frac{1}{2} \phi(\mathbf{x}) = \frac{1}{2} \|\mathbf{y}\|_2^2$, matches $\phi(\mathbf{x})$. Although, in general, we only know that $d_\phi(\mathbf{x}, \mathbf{y}) = d_\psi(\mathbf{y}, \mathbf{x})$, in this case the Bregman divergence is symmetric with $d_\phi(\mathbf{x}, \mathbf{y}) = d_\psi(\mathbf{x}, \mathbf{y})$.

For the Gaussian noise channel $\mathbf{z} = \sqrt{\gamma} \mathbf{x} + \epsilon$ with source $\mathbf{x} \sim p(\mathbf{x})$, we treat the Bregman divergence as a distortion loss for reconstructing $\mathbf{x}$ from observed samples $\mathbf{z}$ using the denoising function $\hat{\mathbf{x}}(\mathbf{z}, \gamma)$. In contrast to the developments in the main text, here we consider the mmse as a function of each noisy sample $\mathbf{z} \sim p(\mathbf{z})$ instead of input sample $\mathbf{x} \sim p(\mathbf{x})$.

Performing minimization in the second argument, [11] show that the arithmetic mean over inputs minimizes the expected Bregman divergence (regardless of the convex

generator ϕ)

$$\hat{\mathbf{x}}^*(\mathbf{z}, \gamma) = \mathbb{E}_{p(\mathbf{x}|\mathbf{z})}[\mathbf{x}] = \arg \min_{\hat{\mathbf{x}}(\mathbf{z}, \gamma)} \mathbb{E}_{p(\mathbf{x}|\mathbf{z})} [d_\phi(\mathbf{x}, \hat{\mathbf{x}}(\mathbf{z}, \gamma))] \quad (\text{A.2})$$

At this minimizing argument, the expected divergence is the MMSE and corresponds to the conditional variance (see [11] Ex. 5)

$$\frac{1}{2} \text{mmse}(\mathbf{z}, \gamma) = \min_{\hat{\mathbf{x}}(\mathbf{z}, \gamma)} \mathbb{E}_{p(\mathbf{x}|\mathbf{z})} [d_\phi(\mathbf{x}, \hat{\mathbf{x}}(\mathbf{z}, \gamma))] = \frac{1}{2} \mathbb{E}_{p(\mathbf{x}|\mathbf{z})} [\|\mathbf{x} - \mathbb{E}_{p(\mathbf{x}|\mathbf{z})}[\mathbf{x}]\|_2^2] = \frac{1}{2} \text{Var}_{p(\mathbf{x}|\mathbf{z})}[\mathbf{x}]$$

To prove that the mean provides the minimizing argument in Eq. equation A.2, [11] Prop. 1 consider the gap in the expected divergence for a suboptimal representative $\hat{\mathbf{x}}(\mathbf{z}, \gamma)$. This can be shown to yield another Bregman divergence, which in our case becomes

$$\begin{aligned} \mathbb{E}_{p(\mathbf{x}|\mathbf{z})} [d_\phi(\mathbf{x}, \hat{\mathbf{x}}(\mathbf{z}, \gamma))] - \mathbb{E}_{p(\mathbf{x}|\mathbf{z})} [d_\phi(\mathbf{x}, \hat{\mathbf{x}}^*(\mathbf{z}, \gamma))] &= d_\phi(\hat{\mathbf{x}}^*(\mathbf{z}, \gamma), \hat{\mathbf{x}}(\mathbf{z}, \gamma)) \\ &= \frac{1}{2} \|\hat{\mathbf{x}}^*(\mathbf{z}, \gamma) - \hat{\mathbf{x}}(\mathbf{z}, \gamma)\|_2^2 \geq 0. \end{aligned}$$

This allows us to derive the gap in the MMSE upper bounds in Sec. 2.4, which arise from using suboptimal neural network denoisers $\hat{\mathbf{x}}(\mathbf{z}, \gamma)$ instead of the true conditional expectation.

$$\mathbb{E}_{p(\mathbf{z}, \mathbf{x})} [\|\mathbf{x} - \hat{\mathbf{x}}(\mathbf{z}, \gamma)\|_2^2] = \underbrace{\mathbb{E}_{p(\mathbf{z}, \mathbf{x})} [\|\mathbf{x} - \hat{\mathbf{x}}^*(\mathbf{z}, \gamma)\|_2^2]}_{\text{mmse}(\gamma)} + \underbrace{\mathbb{E}_{p(\mathbf{z}, \mathbf{x})} [\|\hat{\mathbf{x}}^*(\mathbf{z}, \gamma) - \hat{\mathbf{x}}(\mathbf{z}, \gamma)\|_2^2]}_{\text{estimation gap}}. \quad (\text{A.3})$$

Finally, [11] Sec. 4 proves a bijection between regular Bregman divergences and exponential families, so that minimizing a Bregman divergence loss corresponds to maximum likelihood estimation within a corresponding exponential family. See their Ex. 9 demonstrating the case of the mean-only, spherical Gaussian family. Notably, in this case, the natural parameters θ and expectation parameters η are equivalent. For decoding in our

Gaussian noise channel using the squared error as a (Bregman) loss, we obtain a probabilistic interpretation of the MMSE optimization in Eq. equation A.2 ([11])

$$\min_{\eta=\hat{\mathbf{x}}(\mathbf{z},\gamma)} \mathbb{E}_{p(\mathbf{x}|\mathbf{z})} [d_\phi(\mathbf{x},\eta)] = \max_{\theta=\hat{\mathbf{x}}(\mathbf{z},\gamma)} \mathbb{E}_{p(\mathbf{x}|\mathbf{z})} [\log \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}(\mathbf{z},\gamma), \sigma^2 \mathbb{I})] \quad (\text{A.4})$$

where the Normal family is the appropriate exponential family $p(\mathbf{x};\theta)$ and the optimum is $\hat{\mathbf{x}}^*(\mathbf{z},\gamma) = \eta^* = \theta^* = \mathbb{E}_{p(\mathbf{x}|\mathbf{z})}[\mathbf{x}]$. Finally, we can view the equality in Eq. equation A.3 as an expression of the Pythagorean relation or m -projection ([4] Sec. 1.6, 2.8, [172]) in information geometry. In particular, $\theta^* = \hat{\mathbf{x}}^*(\mathbf{z},\gamma)$ is the projection of $p(\mathbf{x}|\mathbf{z})$ onto the submanifold of fixed-variance, diagonal Gaussian distributions, and for a suboptimal $\theta = \hat{\mathbf{x}}(\mathbf{z},\gamma)$ we have

$$\begin{aligned} D_{KL}[p(\mathbf{x}|\mathbf{z})\|\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}(\mathbf{z},\gamma), \sigma^2 \mathbb{I})] &= D_{KL}[p(\mathbf{x}|\mathbf{z})\|\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}^*(\mathbf{z},\gamma), \sigma^2 \mathbb{I})] \\ &\quad + D_{KL}[\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}^*(\mathbf{z},\gamma), \sigma^2 \mathbb{I})\|\mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}(\mathbf{z},\gamma), \sigma^2 \mathbb{I})]. \end{aligned}$$

In future work, it would be interesting to consider using the I-MMSE relations for general Bregman divergences as in [263, 262].

A.1.7 Comparison with Variational Bounds

In this section, we aim to relate the mmse terms in our bound to terms in the standard variational lower bound for diffusion models.

Forward and Reverse Processes We first review the standard forward and reverse processes used to define denoising diffusion models [224, 89, 121]. Using the notation of [121] in terms of SNR values $\gamma_t = \alpha_t^2/\sigma_t^2$, we set $\sigma_t^2 = 1$ for simplicity and without loss of

generality, since the eventual objective will depend only on γ_t .

$$q(\mathbf{z}_{\gamma_{0:T}}|\mathbf{x}) = q(\mathbf{z}_{\gamma_0}|\mathbf{x}) \prod_{t=1}^T q(\mathbf{z}_{\gamma_t}|\mathbf{z}_{\gamma_{t-1}}) \quad (\text{A.5})$$

where $q(\mathbf{z}_{\gamma_t}|\mathbf{z}_{\gamma_{t-1}}) := \mathcal{N}\left(\mathbf{z}_{\gamma_t}; \sqrt{\frac{\gamma_t}{\gamma_{t-1}}} \mathbf{z}_{\gamma_{t-1}}, \frac{\gamma_{t-1} - \gamma_t}{\gamma_{t-1}} \mathbb{I}\right)$

The Markovian time-reversal $q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t})$ of the forward process is only Gaussian in the limit of $T \rightarrow \infty$ [5, 62], in which case we interpret both processes as stochastic differential equations [234]. However, the conditional $q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}, \mathbf{x})$ is always Gaussian. Using Bayes rule,

$$q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}, \mathbf{x}) = \frac{q(\mathbf{z}_{\gamma_t}|\mathbf{z}_{\gamma_{t-1}})q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{x})}{q(\mathbf{z}_{\gamma_t}|\mathbf{x})} = \mathcal{N}\left(\mathbf{z}_{\gamma_{t-1}}; \sqrt{\frac{\gamma_t}{\gamma_{t-1}}} \mathbf{z}_{\gamma_t} + \frac{\gamma_{t-1} - \gamma_t}{\sqrt{\gamma_{t-1}}} \mathbf{x}, \frac{\gamma_{t-1} - \gamma_t}{\gamma_{t-1}} \mathbb{I}\right) \quad (\text{A.6})$$

since each the forward process is Markovian $q(\mathbf{z}_{\gamma_t}|\mathbf{z}_{\gamma_{t-1}}, \mathbf{x}) = q(\mathbf{z}_{\gamma_t}|\mathbf{z}_{\gamma_{t-1}})$ and each $q(\mathbf{z}_{\gamma_t}|\mathbf{x})$ is Gaussian by construction of the noise channel

$$q(\mathbf{z}_{\gamma_t}|\mathbf{x}) = \sqrt{\gamma_t} \mathbf{x} + \epsilon. \quad (\text{A.7})$$

See [121] App. A2 for example derivations of the Gaussian parameters.

Finally, consider defining a generative model using a variational reverse process. [224] (Sec. 2.2) choose to parameterize each conditional $p(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t})$ as a Gaussian, which is

inspired by the fact that $q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t})$ is Gaussian in the limit as $T \rightarrow \infty$,

$$p(\mathbf{z}_{\gamma_{0:T}}, \mathbf{x}) = p(\mathbf{z}_{\gamma_T}) \prod_{t=1}^T p(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}) \cdot p(\mathbf{x}|\mathbf{z}_{\gamma_0}) \quad (\text{A.8})$$

$$\begin{aligned} \text{where } p(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}) &:= q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}, \hat{\mathbf{x}}(\mathbf{z}_{\gamma_t}, \gamma_t)) \\ &= \mathcal{N}\left(\mathbf{z}_{\gamma_{t-1}}; \sqrt{\frac{\gamma_t}{\gamma_{t-1}}} \mathbf{z}_{\gamma_t} + \frac{\gamma_{t-1} - \gamma_t}{\sqrt{\gamma_{t-1}}} \hat{\mathbf{x}}(\mathbf{z}_{\gamma_t}, \gamma_t), \frac{\gamma_{t-1} - \gamma_t}{\gamma_{t-1}} \mathbb{I}\right). \end{aligned}$$

Note that we have written $p(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t})$ in terms of a denoising function $\hat{\mathbf{x}}(\mathbf{z}_{\gamma_t}, \gamma_t)$. While other forms are possible (e.g. [121] Eq. 28), this expression will be useful to draw connections with the optimal denoising function $\hat{\mathbf{x}}^*(\mathbf{z}, \gamma)$ found in the mmse expression.

Discrete Variational Lower Bound We can now express the discrete variational lower bound using extended state space importance sampling [167, 224]

$$\begin{aligned} \log p(\mathbf{x}) &\geq \mathbb{E}_{q(\mathbf{z}_{\gamma_{0:T}}|\mathbf{x})} \left[\log \frac{p(\mathbf{z}_{\gamma_{0:T}}, \mathbf{x})}{q(\mathbf{z}_{\gamma_{0:T}}|\mathbf{x})} \right] = \mathbb{E}_{q(\mathbf{z}_{\gamma_{0:T}}|\mathbf{x})} \left[\log \frac{p(\mathbf{z}_{\gamma_T}) \prod_{t=1}^T p(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}) \cdot p(\mathbf{x}|\mathbf{z}_{\gamma_0})}{q(\mathbf{z}_{\gamma_0}|\mathbf{x}) \prod_{t=1}^T q(\mathbf{z}_{\gamma_t}|\mathbf{z}_{\gamma_{t-1}})} \right] \\ &= \mathbb{E}_{q(\mathbf{z}_{\gamma_{0:T}}|\mathbf{x})} \left[\log \frac{p(\mathbf{z}_{\gamma_T}) \prod_{t=1}^T p(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}) \cdot p(\mathbf{x}|\mathbf{z}_{\gamma_0})}{q(\mathbf{z}_{\gamma_T}|\mathbf{x}) \prod_{t=1}^T q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}, \mathbf{x})} \right] \\ &= \mathbb{E}_{q(\mathbf{z}_{\gamma_0}|\mathbf{x})} [\log p(\mathbf{x}|\mathbf{z}_{\gamma_0})] - D_{KL}[q(\mathbf{z}_{\gamma_T}|\mathbf{x})||p(\mathbf{z}_{\gamma_T})] + \sum_{t=1}^T D_{KL}[q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}, \mathbf{x})||p(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t})] \end{aligned}$$

Compared to the notation in the main text, note that the forward process $q(\mathbf{z}_{\gamma_0}|\mathbf{x})$ and $q(\mathbf{z}_{\gamma_T}|\mathbf{x})$ represent the Gaussian noise channels instead of $p(\mathbf{z}_{\gamma}|\mathbf{x})$. Instead of the true denoising posterior $p(\mathbf{x}|\mathbf{z}_{\gamma_0})$ is a variational distribution in the above expression. Our goal is now to relate the KL divergence terms, particularly the true time-reversal process $q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t})$ or its Gaussian approximation $p(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t})$, to the MMSE.

Relation to Incremental Noise Channel Proof of I-MMSE Relation

The proof of the I-MMSE relation in [77] relies on the construction of an *incremental channel* which successively adds Gaussian noise to the data. We recognize this channel as being identical to the conditional $q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}, \mathbf{x})$ in the limit as $\delta = \gamma_{t-1} - \gamma_t \rightarrow 0$, and restate the proof using the notation above to highlight connections with terms in the variational lower bound.

Recall that the following distributions are Gaussian: each noise channel $q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{x})$ and $q(\mathbf{z}_{\gamma_t}|\mathbf{x})$ (Eq. equation A.7), the forward conditional $q(\mathbf{z}_{\gamma_t}|\mathbf{z}_{\gamma_{t-1}})$ (Eq. equation A.5), and the reverse data-conditional $q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}, \mathbf{x})$ (Eq. equation A.6). Using subscripts to distinguish different standard Gaussian variables, e.g. $\epsilon_{\gamma_t} \sim \mathcal{N}(0, \mathbb{I})$, we have the following relationships

$$\mathbf{z}_{\gamma_{t-1}} = \sqrt{\gamma_{t-1}} \mathbf{x} + \epsilon_{\gamma_{t-1}} \quad (\text{using Eq. equation A.7}) \quad (\text{A.9})$$

$$= \sqrt{\frac{\gamma_t}{\gamma_{t-1}}} \mathbf{z}_{\gamma_t} + \frac{\gamma_{t-1} - \gamma_t}{\sqrt{\gamma_{t-1}}} \mathbf{x} + \sqrt{\frac{\gamma_{t-1} - \gamma_t}{\gamma_{t-1}}} \epsilon \quad (\text{using Eq. equation A.6}) \quad (\text{A.10})$$

$$\mathbf{z}_{\gamma_t} = \sqrt{\gamma_t} \mathbf{x} + \epsilon_{\gamma_t} \quad (\text{using Eq. equation A.7})$$

$$= \frac{\sqrt{\gamma_t}}{\sqrt{\gamma_{t-1}}} \mathbf{z}_{\gamma_{t-1}} + \sqrt{\frac{\gamma_{t-1} - \gamma_t}{\gamma_{t-1}}} \epsilon_\delta \quad (\text{using Eq. equation A.5}). \quad (\text{A.11})$$

Up to change in notation, Eq. equation A.9 and equation A.11 match the construction of the incremental noise channel in Eq. (30)-(31) of [77], while Eq. equation A.10 matches their Eq. (37).

Proof of I-MMSE Relation using Incremental Noise Channel Our goal is to take the limit as $T \rightarrow \infty$ or $\delta = \gamma_{t-1} - \gamma_t \rightarrow 0$, in order to recover the MMSE relation

$\frac{d}{d\gamma_t} I(\mathbf{x}; \mathbf{z}_{\gamma_t}) = 1/2 \text{mmse}(\mathbf{x}, \gamma_t)$. The MMSE relation for the derivative is equivalent to $I(\mathbf{x}; \mathbf{z}_{\gamma_t}) - I(\mathbf{x}; \mathbf{z}_{\gamma_{t-1}}) = \delta/2 \text{mmse}(\mathbf{x}, \gamma_t) + o(\delta)$ for small δ . Thus, we focus on the difference $I(\mathbf{x}; \mathbf{z}_{\gamma_{t-1}}) - I(\mathbf{x}; \mathbf{z}_{\gamma_t})$.

Using the chain rule for mutual information and the Markov property $\mathbf{z}_{\gamma_t} \perp \mathbf{x} \mid \mathbf{z}_{\gamma_{t-1}}$ (since $\mathbf{x} \rightarrow \mathbf{z}_{\gamma_{t-1}} \rightarrow \mathbf{z}_{\gamma_t}$), we have

$$\begin{aligned} I(\mathbf{x}; \{\mathbf{z}_{\gamma_{t-1}}, \mathbf{z}_{\gamma_t}\}) &= I(\mathbf{x}; \mathbf{z}_{\gamma_{t-1}}) + \cancel{I(\mathbf{x}; \mathbf{z}_{\gamma_t} \mid \mathbf{z}_{\gamma_{t-1}})} = I(\mathbf{x}; \mathbf{z}_{\gamma_t}) + I(\mathbf{x}; \mathbf{z}_{\gamma_{t-1}} \mid \mathbf{z}_{\gamma_t}) \\ \implies I(\mathbf{x}; \mathbf{z}_{\gamma_{t-1}}) - I(\mathbf{x}; \mathbf{z}_{\gamma_t}) &= I(\mathbf{x}; \mathbf{z}_{\gamma_{t-1}} \mid \mathbf{z}_{\gamma_t}) \end{aligned} \quad (\text{A.12})$$

which allows us to restrict attention to $I(\mathbf{x}; \mathbf{z}_{\gamma_{t-1}} \mid \mathbf{z}_{\gamma_t})$. Using the forward or target noise process, note that this mutual information compares the ratio of the time-reversed data-conditional $q(\mathbf{z}_{\gamma_{t-1}} \mid \mathbf{z}_{\gamma_t}, \mathbf{x})$ and the time-reversed Markov process $q(\mathbf{z}_{\gamma_{t-1}} \mid \mathbf{z}_{\gamma_t})$,

$$\begin{aligned} I(\mathbf{x}; \mathbf{z}_{\gamma_{t-1}} \mid \mathbf{z}_{\gamma_t}) &= \mathbb{E}_{q(\mathbf{z}_{\gamma_{t-1}}, \mathbf{x} \mid \mathbf{z}_{\gamma_t})} \left[\log \frac{q(\mathbf{z}_{\gamma_{t-1}}, \mathbf{x} \mid \mathbf{z}_{\gamma_t})}{q(\mathbf{z}_{\gamma_{t-1}} \mid \mathbf{z}_{\gamma_t}) q(\mathbf{x} \mid \mathbf{z}_{\gamma_t})} \right] \\ &= \mathbb{E}_{q(\mathbf{z}_{\gamma_{t-1}}, \mathbf{x} \mid \mathbf{z}_{\gamma_t})} \left[\log \frac{q(\mathbf{z}_{\gamma_{t-1}} \mid \mathbf{z}_{\gamma_t}, \mathbf{x})}{q(\mathbf{z}_{\gamma_{t-1}} \mid \mathbf{z}_{\gamma_t})} \right]. \end{aligned} \quad (\text{A.13})$$

Recalling Eq. equation A.10,

$$\mathbf{z}_{\gamma_{t-1}} = \sqrt{\frac{\gamma_t}{\gamma_{t-1}}} \mathbf{z}_{\gamma_t} + \frac{\gamma_{t-1} - \gamma_t}{\sqrt{\gamma_{t-1}}} \mathbf{x} + \sqrt{\frac{\gamma_{t-1} - \gamma_t}{\gamma_{t-1}}} \epsilon, \quad (\text{A.14})$$

we can see that $I(\mathbf{x}; \mathbf{z}_{\gamma_{t-1}} \mid \mathbf{z}_{\gamma_t})$ is the mutual information over a Gaussian noise channel $q(\mathbf{z}_{\gamma_{t-1}} \mid \mathbf{z}_{\gamma_t}, \mathbf{x})$, where $\mathbf{z}_{\gamma_t}$ is given and the input is drawn from the conditional distribution

$q(\mathbf{x}|\mathbf{z}_{\gamma_t})$. The marginal output distribution $q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t})$, which is analogous to $p(\mathbf{z}_{\gamma_{t-1}})$ for the unconditional channel marginal in Eq. equation B.2, is not Gaussian in general.

Noting that the mutual information $I(\mathbf{x}; \mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}) = I(\mathbf{x}; c\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t})$ is invariant to rescaling of $\mathbf{z}_{\gamma_{t-1}}$ by c , we consider

$$\begin{aligned}\sqrt{\frac{\gamma_{t-1}}{\gamma_{t-1} - \gamma_t}} \mathbf{z}_{\gamma_{t-1}} &= \sqrt{\frac{\gamma_t}{\gamma_{t-1}}} \sqrt{\frac{\gamma_{t-1}}{\gamma_{t-1} - \gamma_t}} \mathbf{z}_{\gamma_t} + \frac{\gamma_{t-1} - \gamma_t}{\sqrt{\gamma_{t-1}}} \sqrt{\frac{\gamma_{t-1}}{\gamma_{t-1} - \gamma_t}} \mathbf{x} + \epsilon \\ &= \sqrt{\frac{\gamma_t}{\gamma_{t-1} - \gamma_t}} \mathbf{z}_{\gamma_t} + \sqrt{\gamma_{t-1} - \gamma_t} \mathbf{x} + \epsilon\end{aligned}\tag{A.15}$$

in the limit as the SNR, $\delta = \gamma_{t-1} - \gamma_t$, approaches 0.

Lemma 1 ([77] Lemma 1). *For a Gaussian noise channel*

$$\mathbf{z} = \sqrt{\delta} \mathbf{x} + \epsilon\tag{A.16}$$

where $\mathbf{x} \sim p(\mathbf{x})$, $\mathbb{E}[\mathbf{x}^2] < \infty$, and $\epsilon \sim \mathcal{N}(0, \mathbb{I})$, the input-output mutual information in the limit as $\delta \rightarrow 0$ is given by

$$I(\mathbf{z}; \mathbf{x}) = \frac{\delta}{2} \mathbb{E}_{p(\mathbf{x})} \left[(\mathbf{x} - \mathbb{E}_{p(\mathbf{x})}[\mathbf{x}])^2 \right] + o(\delta) = \frac{\delta}{2} \text{Var}(\mathbf{x}) + o(\delta)\tag{A.17}$$

In particular, the mutual information is independent of shape of the channel input distribution $p(\mathbf{x})$.

Proof. The proof proceeds by constructing an upper bound on the channel mutual information $D_{KL}[p(\mathbf{z}|\mathbf{x})||g(\mathbf{z})] = I(\mathbf{z}; \mathbf{x}) + D_{KL}[p(\mathbf{z})||g(\mathbf{z})]$ where $g(\mathbf{z}) := \mathcal{N}(\mathbf{z}; \mathbb{E}_{p(\mathbf{x})}[\sqrt{\delta}\mathbf{x}], (\delta \text{Var}[\mathbf{x}] + 1)\mathbb{I})$ is a Gaussian distribution with the same mean and variance as $p(\mathbf{z}) = \int p(\mathbf{x})p(\mathbf{z}|\mathbf{x})d\mathbf{x}$.

As $\delta \rightarrow 0$, it can be shown that $D_{KL}[p(\mathbf{z})||g(\mathbf{z})] = o(\delta)$. For each $\mathbf{x}$, the divergence between Gaussians $\mathbb{E}_{p(\mathbf{x})}[D_{KL}[p(\mathbf{z}|\mathbf{x})||g(\mathbf{z})]]$ is tractable, and the terms involving the mean cancel in expectation. The $1/2 \log \frac{\text{Var}_{g(\mathbf{z})}[\mathbf{z}]}{\text{Var}_{p(\mathbf{z}|\mathbf{x})}[\mathbf{z}]}$ term in the KL divergence contributes the variance term in Eq. equation A.17, where $\text{Var}_{p(\mathbf{z}|\mathbf{x})}[\mathbf{z}] = 1$ and $\text{Var}_{g(\mathbf{z})}[\mathbf{z}] = \delta \text{Var}[\mathbf{x}] + 1$ is chosen to be the same as the variance of the output marginal $p(\mathbf{z})$. See [77] App. II for detailed proof. $\square$

Applying Lemma 1 for the Gaussian channel in Eq. equation A.14, where the input distribution is $q(\mathbf{x}|\mathbf{z}_{\gamma_t})$ and the mutual information is $I(\mathbf{x}; \mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}) = I(\mathbf{x}; c\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t})$ (see Eq. equation A.15), yields the desired MMSE relation. Summarizing the reasoning above, we have

$$I(\mathbf{x}; \mathbf{z}_{\gamma_t}) - I(\mathbf{x}; \mathbf{z}_{\gamma_{t-1}}) \stackrel{\text{equation A.12}}{=} I(\mathbf{x}; \mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}) \stackrel{\text{equation A.17}}{=} \frac{\delta}{2} \text{Var}_{q(\mathbf{x}|\mathbf{z}_{\gamma_t})}[\mathbf{x}] + o(\delta) \quad (\text{A.18})$$

$$\implies \frac{d}{d\gamma} I(\mathbf{x}; \mathbf{z}_{\gamma_t}) = \frac{1}{2} \mathbb{E}_{q(\mathbf{x}, \mathbf{z}_{\gamma_t})} \left[\left(\mathbf{x} - \mathbb{E}_{q(\mathbf{x}|\mathbf{z}_{\gamma_t})}[\mathbf{x}] \right)^2 \right] := \frac{1}{2} \text{mmse}(\gamma_t), \quad (\text{A.19})$$

which proves the MMSE relation in Eq. equation 2.2.

Relation to Variational Lower Bound Compare our expression in Eq. equation 2.6

$$\mathcal{L}_{\infty}^{\text{diff}}(\mathbf{x}) := \underbrace{\mathbb{E}_{p(\mathbf{z}_{\gamma_0}|\mathbf{x})}[-\log p(\mathbf{x}|\mathbf{z}_{\gamma_0})]}_{\text{Reconstruction loss}} - \frac{1}{2} \underbrace{\int_{\gamma_0}^{\gamma_1} \text{mmse}(\mathbf{x}, \gamma) d\gamma}_{\text{Diffusion loss}}.$$

to Eq. (11)-(12) in [121]

$$\mathcal{L}_T^{\text{diff}}(\mathbf{x}) := - \underbrace{\sum_{t=1}^T \mathbb{E}_{q(\mathbf{z}_{\gamma_t}|\mathbf{x})} D_{KL}[q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}, \mathbf{x}) \| p(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t})]}_{\text{Diffusion loss}}. \quad (\text{A.20})$$

The conditional Gaussian parameterization of $p(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t})$ is often justified by the fact that $q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}, \mathbf{x})$ is Gaussian [89, 121].

However, from the information theoretic perspective, our goal should be to estimate the conditional mutual information $I(\mathbf{x}; \mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t})$ in Eq. equation A.13. Let $p_G^*(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}) = \mathcal{N}(\mathbf{z}_{\gamma_{t-1}}; \mu(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}), \sigma^2(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}))$ be the maximum likelihood Gaussian approximation to the channel output marginal $q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}) = \int q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}, \mathbf{x}) q(\mathbf{x}|\mathbf{z}_{\gamma_t}) d\mathbf{x}$ (i.e. with the same mean and variance). Rewriting the mutual information in terms of an upper bound using this Gaussian marginal,

$$\begin{aligned} I(\mathbf{x}; \mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}) &= \mathbb{E}_{q(\mathbf{x}|\mathbf{z}_{\gamma_t})} [D_{KL}[q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}, \mathbf{x}) \| q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t})]] \\ &= \mathbb{E}_{q(\mathbf{x}|\mathbf{z}_{\gamma_t})} [D_{KL}[q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}, \mathbf{x}) \| p_G^*(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t})]] - D_{KL}[q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}) \| p_G^*(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t})]. \end{aligned} \quad (\text{A.21})$$

In the continuous time limit as $\delta \rightarrow 0$ or $T \rightarrow \infty$, we have that $D_{KL}[q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}) \| p_G^*(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t})] = o(\delta)$ (as in the proof of Lemma 1, where the marginal divergence $D_{KL}[p(\mathbf{z}) \| g(\mathbf{z})] = o(\delta)$).

Thus, we have

$$\begin{aligned} I(\mathbf{x}; \mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}) &= \mathbb{E}_{q(\mathbf{x}|\mathbf{z}_{\gamma_t})} [D_{KL}[q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}, \mathbf{x}) \| p_G^*(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t})]] - \underbrace{D_{KL}[q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}) \| p_G^*(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t})]}_{o(\delta) \text{ as } \delta \rightarrow 0} \\ &\xrightarrow{\delta \rightarrow 0} \mathbb{E}_{q(\mathbf{x}|\mathbf{z}_{\gamma_t})} [D_{KL}[q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}, \mathbf{x}) \| p_G^*(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t})]] \end{aligned} \quad (\text{A.22})$$

To summarize, in the continuous time limit, the proof of the I-MMSE relation shows that estimating the reverse process $q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t})$ only requires a Gaussian variational family. The optimal Gaussian approximation $p_G^*(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t})$ requires the variance of $q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t})$, which depends on the variance of the input $q(\mathbf{x}|\mathbf{z}_{\gamma_t})$ to the noise channel $q(\mathbf{z}_{\gamma_{t-1}}|\mathbf{z}_{\gamma_t}, \mathbf{x})$ (as in the proof of Lemma 1). Evaluating this variance involves (learning) the conditional expectation or optimal denoiser $\hat{\mathbf{x}}^*(\mathbf{z}_{\gamma_t}, \gamma_t) = \mathbb{E}_{q(\mathbf{x}|\mathbf{z}_{\gamma_t})}[\mathbf{x}]$ at each SNR, which matches Eq. equation 2.3 and leads to the optimization in Sec. 2.4.

A.1.8 Mixture Distribution Density

In this section, we derive the expression that shows that the density of a continuum of Gaussian mixture distributions can be written in terms of the optimal denoiser, $\hat{\epsilon}$, for the data distribution.

$$\begin{aligned}
-\log p_\alpha(\mathbf{x}) &= c + 1/2 \int_{-\infty}^{\infty} d\bar{\alpha} \mathbb{E}_{p(\epsilon)} [\|\epsilon - \sqrt{\frac{\sigma(-\bar{\alpha})}{\sigma(-\beta)}} \hat{\epsilon}(\sqrt{\sigma(\bar{\alpha})}\mathbf{x} + \sqrt{\sigma(-\bar{\alpha})}\epsilon, \beta)\|_2^2] \\
\beta &\equiv \sigma^{-1}(\sigma(\alpha)\sigma(\bar{\alpha}))
\end{aligned} \tag{A.23}$$

As in the previous section, we will adopt information theory notation. If we define the optimal denoiser for the input distribution, $p_\alpha(\mathbf{x})$, with a subscript as $\hat{\epsilon}_\alpha(\cdot, \cdot)$, we can write the density analogously to Eq. B.6.

$$-\log p_\alpha(\mathbf{x}) = c + 1/2 \int_{-\infty}^{\infty} d\bar{\alpha} \mathbb{E}_{p(\epsilon)} [\|\epsilon - \hat{\epsilon}_\alpha(\sqrt{\sigma(\bar{\alpha})}\mathbf{x} + \sqrt{\sigma(-\bar{\alpha})}\epsilon, \bar{\alpha})\|_2^2] \tag{A.24}$$

Note that we now have to keep track of two log SNR values. One indicates how much noise is added to the new “data” distribution, the other is how much noise we add and then try to remove with our denoiser. The goal is to relate $\hat{\epsilon}_\alpha$ to $\hat{\epsilon}$. We can formally write down the optimal solution using the relation in Eq. 2.3.

$$\hat{\epsilon}_\alpha(\mathbf{x}, \bar{\alpha}) = \mathbb{E}_{\epsilon \sim p(\mathcal{E}|Z=\mathbf{x})}[\epsilon]$$

Now, however, the noise channel is defined differently. The channel mixes the signal, $\bar{\mathbf{x}} \sim p_\alpha(\bar{X})$, with Gaussian noise, $\bar{\epsilon} \sim \mathcal{N}(0, \mathbb{I})$, as $Z \equiv \sqrt{\sigma(\bar{\alpha})}\bar{X} + \sqrt{\sigma(-\bar{\alpha})}\bar{\mathcal{E}}$. And the noisy variable, $\bar{X} = \sqrt{\sigma(\alpha)}X + \sqrt{\sigma(-\alpha)}\mathcal{E}$, where we must be careful to distinguish the two independent sources of Gaussian noise.

We start by expanding definitions.

$$\begin{aligned} \hat{\epsilon}_\alpha(\mathbf{z}, \bar{\alpha}) &= \frac{1}{p(Z=\mathbf{z})} \int d\bar{\epsilon} \ p(\mathcal{E}=\bar{\epsilon}, Z=\mathbf{z}) \ \bar{\epsilon} \\ &= \frac{1}{p(Z=\mathbf{z})} \int d\epsilon d\bar{\epsilon} d\bar{\mathbf{x}} d\mathbf{x} \ \delta(\mathbf{z} - \sqrt{\sigma(\bar{\alpha})}\bar{\mathbf{x}} - \sqrt{\sigma(-\bar{\alpha})}\bar{\epsilon}) \delta(\bar{\mathbf{x}} - \sqrt{\sigma(\alpha)}\mathbf{x} - \sqrt{\sigma(-\alpha)}\epsilon) \ \bar{\epsilon} \\ &= \frac{1}{p(Z=\mathbf{z})} \int d\epsilon d\bar{\epsilon} d\mathbf{x} \ \delta(\mathbf{z} - \sqrt{\sigma(\bar{\alpha})}(\sqrt{\sigma(\alpha)}\mathbf{x} + \sqrt{\sigma(-\alpha)}\epsilon) - \sqrt{\sigma(-\bar{\alpha})}\bar{\epsilon}) \ \bar{\epsilon} \end{aligned}$$

Now do a change of variables, a 2-d rotation with:

$$\bar{\epsilon}' = a\bar{\epsilon} + b\epsilon, \epsilon' = -b\bar{\epsilon} + a\epsilon,$$

$$a = \sqrt{\sigma(\bar{\alpha})\sigma(\alpha)/(1 - \sigma(\alpha)\sigma(\bar{\alpha}))}, b = \sqrt{\sigma(-\bar{\alpha})/(1 - \sigma(\alpha)\sigma(\bar{\alpha}))}.$$

This change of variables leads to the following.

$$\begin{aligned}
\hat{\epsilon}_\alpha(\mathbf{z}, \bar{\alpha}) &= \frac{1}{p(Z = \mathbf{z})} \int d\epsilon' d\bar{\epsilon}' d\mathbf{x} \delta(\mathbf{z} - \sqrt{\sigma(\alpha)\sigma(\bar{\alpha})}\mathbf{x} - \sqrt{1 - \sigma(\alpha)\sigma(\bar{\alpha})}\bar{\epsilon}') (b\bar{\epsilon}' + a\epsilon') \\
&= b \frac{1}{p(Z = \mathbf{z})} \int d\epsilon' d\bar{\epsilon}' d\mathbf{x} \delta(\mathbf{z} - \sqrt{\sigma(\alpha)\sigma(\bar{\alpha})}\mathbf{x} - \sqrt{1 - \sigma(\alpha)\sigma(\bar{\alpha})}\bar{\epsilon}') \bar{\epsilon}' \\
&= b \frac{1}{p(Z = \mathbf{z})} \int d\epsilon' d\bar{\epsilon}' d\mathbf{x} \delta(\mathbf{z} - \sqrt{\sigma(\beta)}\mathbf{x} - \sqrt{1 - \sigma(\beta)}\bar{\epsilon}') \bar{\epsilon}' \\
\hat{\epsilon}_\alpha(\mathbf{z}, \bar{\alpha}) &= b\hat{\epsilon}(\mathbf{z}, \beta), \quad \beta \equiv \sigma^{-1}(\sigma(\bar{\alpha})\sigma(\alpha)), b = \sqrt{\sigma(-\bar{\alpha})/(1 - \sigma(\alpha)\sigma(\bar{\alpha}))}
\end{aligned}$$

Note in the second line that the expectation of ϵ' is zero, and we move the constant for the other term, b , outside the integral. In the third line, we define β which represents the log SNR of the two consecutive noisy channels with $\alpha, \bar{\alpha}$. Then we recognize the resulting integral as Eq. 2.3, the optimal denoiser for recovering samples from the original (non-noisy) data distribution in Gaussian noise.

A.2 Implementation Details

A.2.1 Discrete probability estimator bound

We derive some results that are useful for estimating upper bounds on discrete negative log likelihood.

$$\begin{aligned}
\mathbb{E}[-\log P(\mathbf{x})] &= 1/2 \int_0^\infty \text{mmse}(\gamma) d\gamma \\
&= 1/2 \int_{\gamma_0}^{\gamma_1} \text{mmse}(\gamma) d\gamma + 1/2 \left(\int_0^{\gamma_0} + \int_{\gamma_1}^\infty \right) \text{mmse}(\gamma) d\gamma \\
&\leq 1/2 \int_{\gamma_0}^{\gamma_1} \mathbb{E}_{\mathbf{z}, \mathbf{x}} [\|\mathbf{x} - \hat{\mathbf{x}}(\mathbf{z}, \gamma)\|_2^2] d\gamma + c(\gamma_0, \gamma_1)
\end{aligned}$$

First, consider the left tail integral, $I_L = 1/2 \int_0^{\gamma_0} \text{mmse}(\gamma) d\gamma$. When the SNR is low, the distribution is approximately Gaussian, and we can use the Gaussian MMSE to get an upper bound. The Gaussian MMSE should be for a Gaussian with either the same variance or same covariance matrix as the data, to ensure that it is an upper bound on the true MMSE. The eigenvalues of the covariance matrix are denoted with λ_i . The MMSE results for Gaussians are discussed and derived in App. A.1.5.

$$\begin{aligned}
I_L &= 1/2 \int_0^{\gamma_0} \text{mmse}(\gamma) d\gamma \\
&\leq 1/2 \int_0^{\gamma_0} \text{mmse}_G(\gamma) d\gamma \\
&= 1/2 \int_0^{\gamma_0} \sum_i 1/(\gamma + 1/\lambda_i) d\gamma \\
&= 1/2 \sum_i \log(1 + \gamma_0 \lambda_i)
\end{aligned}$$

Next we consider the right tail integral, $I_R = 1/2 \int_{\gamma_1}^{\infty} \text{mmse}(\gamma) d\gamma$. In this regime, the noise is extremely small. Because the data is discrete, we can get a good estimate by simply rounding to the nearest discrete value. Let the distance between discrete values be Δ .

$$\begin{aligned}
I_R &= 1/2 \int_{\gamma_1}^{\infty} \text{mmse}(\gamma) d\gamma \\
&\leq 1/2 \int_{\gamma_1}^{\infty} \mathbb{E}_{\mathbf{x}, \epsilon} [\|\mathbf{x} - \hat{\mathbf{x}}(\sqrt{\gamma}\mathbf{x} + \epsilon, \gamma)\|_2^2] d\gamma \\
&= 1/2 \sum_{i=1}^d \int_{\gamma_1}^{\infty} \mathbb{E}_{\mathbf{x}, \epsilon} [\|x_i - \hat{x}_i(\sqrt{\gamma}x_i + \epsilon_i, \gamma)\|_2^2] d\gamma
\end{aligned}$$

We separate the analysis into a contribution from each vector component. Next, the possible errors per components will be multiples of Δ , whose appearance depends only on the size of

the noise.

$$\begin{aligned}
I_R &\leq 1/2 \sum_{i=1}^d \int_{\gamma_1}^{\infty} \sum_{j=1}^{j_{\max}} (\Delta j)^2 2P((j-1/2)\Delta \leq \epsilon_i/\sqrt{\gamma} \leq (j+1/2)\Delta) d\gamma \\
&\leq d/2 \int_{\gamma_1}^{\infty} \sum_{j=1}^{j_{\max}} (\Delta j)^2 2P((j-1/2)\Delta \leq \epsilon_i/\sqrt{\gamma}) d\gamma \\
&\leq d \int_{\gamma_1}^{\infty} \sum_{j=1}^{j_{\max}} (\Delta j)^2 e^{-(j-1/2)^2 \Delta^2 \gamma/2} d\gamma \quad (\text{Use Gaussian Chernoff bound}) \\
&= d \int_{\gamma_1}^{\infty} \sum_{j=1}^{j_{\max}} \frac{(\Delta j)^2}{(j-1/2)^2 \Delta^2/2} e^{-(j-1/2)^2 \Delta^2 \gamma/2} d\gamma \\
&= 2d \sum_{j=1}^{j_{\max}} \frac{j^2}{(j-1/2)^2} e^{-(j-1/2)^2 \Delta^2 \gamma_1/2} \quad (\text{Do integral}) \\
&\leq 4d \sum_{j=1}^{j_{\max}} e^{-(j-1/2)^2 \Delta^2 \gamma_1/2} \quad (\text{Bound } j \text{ term})
\end{aligned}$$

To summarize, the tail bound constants for our upper bound on discrete likelihood are as follows.

$$\mathbb{E}[-\log P(\mathbf{x})] \leq 1/2 \int_{\gamma_0}^{\gamma_1} \mathbb{E}_{\mathbf{z}, \mathbf{x}}[\|\mathbf{x} - \hat{\mathbf{x}}(\mathbf{z}, \gamma)\|_2^2] d\gamma + c(\gamma_0, \gamma_1)$$

$$c(\gamma_0, \gamma_1) \equiv 1/2 \sum_{i=1}^d \log(1 + \gamma_0 \lambda_i) + 4d \sum_{j=1}^{j_{\max}} e^{-(j-1/2)^2 \Delta^2 \gamma_1/2}$$

For practical integration ranges, these terms are nearly zero.

A.2.2 Relationship Between $\log(\text{SNR})$ and Timesteps

To use pre-trained models in the literature with our estimator, we need to translate “ t ”, a parameter representing time in a Markov chain that progressively adds noise to data,

to a SNR. Referring to Eq. (9) and Eq. (17) in [171], it is easy to build up the mapping from $\alpha = \log(\text{SNR})$ to timestep t in IDDPM. The following derivation shows the relationship:

$$\begin{aligned} \text{sigmoid}(\alpha) &= \cos\left(\frac{\frac{t}{T} + s}{1 + s} \cdot \frac{\pi}{2}\right)^2 / \cos\left(\frac{s}{1 + s} \cdot \frac{\pi}{2}\right)^2 \\ \implies \cos\left(\frac{\frac{t}{T} + s}{1 + s} \cdot \frac{\pi}{2}\right) &= \cos\left(\frac{s}{1 + s} \cdot \frac{\pi}{2}\right) \sqrt{\text{sigmoid}(\alpha)} \\ \implies t &= T \left(\arccos\left(\cos\left(\frac{s}{1 + s} \cdot \frac{\pi}{2}\right) \sqrt{\text{sigmoid}(\alpha)}\right) \frac{2(1 + s)}{\pi} - s \right) \end{aligned}$$

For DDPM, the mapping between α and t is a little bit tricky because it's discrete. Firstly, we construct a one-to-one mapping between the variance β_t and t . In [89], the β_t is scheduled in a linear way. Denote $\eta_t = 1 - \beta_t$ and then we have a mapping between $\bar{\eta}_t = \prod_{s=1}^t \eta_s$ and t . We match the closest value $\bar{\eta}_t$ with $\text{sigmoid}(\alpha)$, and then find the corresponding t of α . Specifically, the mapping between $\log(\text{SNR})$ and t/T (scaled from 0 to 1) is shown in Fig. A.1. Our schedule is generated from CIFAR-10 dataset, which gives more diffusion steps for high $\log(\text{SNR})$. Moreover, the schedule adapts along with the dataset by computing scale and mean from it.

A.2.3 Details on Model Training

In our fine-tuning experiments, we adopted a IDDPM model (<https://github.com/openai/improved-diffusion>) provided by [171], pre-trained on the unconditional CIFAR-10 dataset with the L_{vib} objective and cosine schedule. We also consider a pre-trained DDPM model from Hugging Face (<https://huggingface.co/google/ddpm-cifar10-32>),

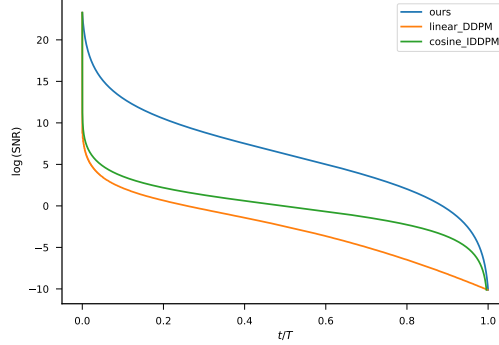


Figure A.1: $\log(\text{SNR})$ v.s. timestep t/T

which is provided by [89]. We process CIFAR-10 32×32 dataset in the same way as that in <https://github.com/openai/improved-diffusion/tree/main/datasets>. Since both models are trained with timesteps, we have to convert our $\log(\text{SNR})$ values to timesteps before passing them into models. The dataset is scaled to $[-1, 1]$ for each pixel.

For fine-tuning, we train each model for 10 epochs with the same ‘learning rate / batch size’ ratio in the [89] and [171], e.g., ‘ $10^{-4}/64$ ’ for DDPM, and ‘ $2.5 \times 10^{-5}/32$ ’ for IDDPM. During training, we reduce the learning rate by a factor of 10 and keep the same batch size after 5 epochs where the training loss starts to be flat. The optimizer for both models is Adam. In testing, we clip denoised images to range $[-1, 1]$, but not during training. The fine-tuning results are displayed in Fig. A.2 and Fig. A.3. It shows that the fine-tuning improves the NLL value by pushing the MSE curve down when the $\log(\text{SNR})$ is high. The NLL results are continuous NLL comparable to Table 2.1.

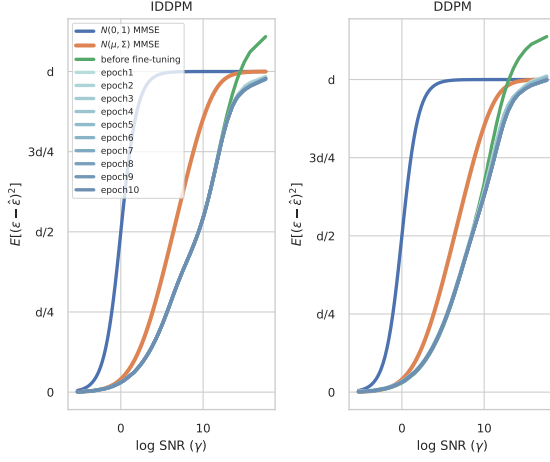


Figure A.2: Change of denoising MSE curve during fine-tuning process.

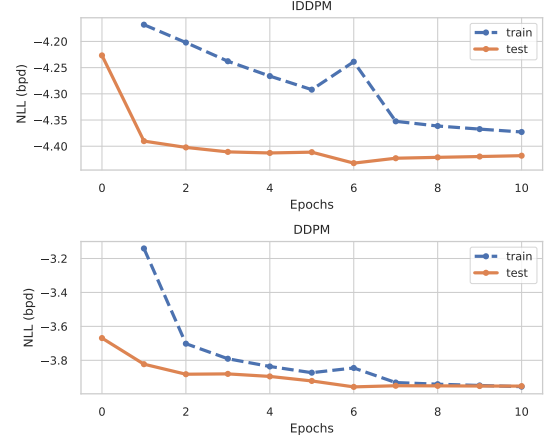


Figure A.3: Training and testing NLL during fine-tuning process.

A.2.4 Soft Discretization Function

We found in our experiments that existing denoising architectures were not able to learn the discreteness of the underlying data for the CIFAR dataset, which takes values $g_i = -1 + \Delta i$, with $\Delta = 2/255, i = 0, \dots, 255$. At high values of SNR, the noisy input, z , is very close to the true value of x , so zero prediction error can be achieved with high probability just by predicting the nearest discrete value. Therefore, we needed to add a discretization function to the denoiser value at high SNR to improve results. At first we tried a simple function that just rounded to the nearest discrete value. This function was overly aggressive in rounding near borderline values, which sometimes caused an increase in the mean square error. Therefore, we devised a “soft discretization” function. Besides improving the MSE, a soft discretization is also more amenable to being used as a differentiable nonlinearity in a trainable neural network, compared to the hard discretization function.

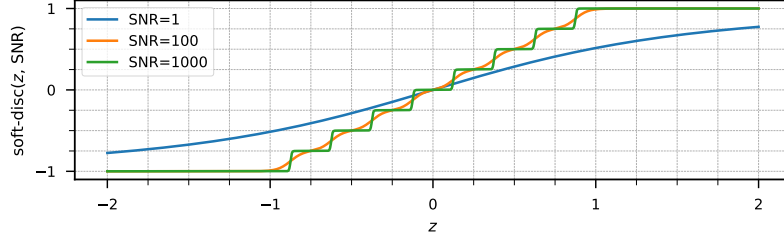


Figure A.4: Illustrating the soft discretization function for various values of SNR (γ), for regularly spaced discrete data, $x = -1, -0.75, -0.5, \dots, 1$.

Consider the noisy channel for scalar random variables, $z = \sqrt{\gamma/(1+\gamma)}x + \sqrt{1/(1+\gamma)}\epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$ and $x \sim P(x)$. Let $P(x)$ be a distribution over N discrete grid points, $g_1, \dots, g_N$. Then we define the soft-discretization function as follows.

$$q(i) = \text{softmax}_i(-0.5(z - g_i)^2(1 + \gamma))$$

$$\text{soft-disc}(z; \gamma) = \sum_i g_i q(i)$$

The softmax can be interpreted as a distribution over nearby grid points, where the closest values are most likely. Then we simply take the expected grid point value as the output. The SNR plays the role of a temperature parameter that leads to a more strongly discretizing function at high SNR, and a more linear function at low SNR, as shown in Fig. A.4. The form of this nonlinearity was inspired by looking at the optimal case where x is uniform, $P(x = g_i) = 1/N$.

In Fig. 2.3, we see that this function is not optimal, leading to higher MSEs at low SNR values. However, it performs well at high SNR, so when we ensemble different denoisers we get the best results in Table 2.2. A more effective strategy would be to replace $(1 + \gamma)$ in the soft discretization function with some learnable function, $f(\gamma)$, then train the

neural network accordingly. In this work, we wanted to see that we could improve density estimates over variational bounds by fine-tuning existing architectures with a new objective. Therefore, we used a fixed discretization function instead of a learnable one to avoid adding new parameters or complexity to the model.

A.2.5 Variance Estimates

We now study two different types of variance estimates for our discrete and continuous log likelihood estimators. We use stochastic estimators for our upper bounds on the negative log likelihood in both cases. We would like to study the variance of these estimators to be sure that the estimates in our results do not look artificially low simply due to noise in the estimators.

Table A.1: The standard deviation of NLL (bpd) estimates with information-theoretic bounds

	Continuous	Discrete	Continuous + Dequantization
DDPM	0.00035	0.00112	0.00033
IDDPM	0.00051	0.00088	0.00046
DDPM(tune)	0.00049	0.00100	0.00061
IDDPM(tune)	0.00063	0.00087	0.00059

Our estimators for the discrete and continuous case are simple Monte Carlo estimates, rewritten in the following expressions with irrelevant constants discarded.

$$\mathcal{L}_{cont} \propto \mathbb{E}_{\alpha, \mathbf{x}, \epsilon} \left[\frac{1}{2} \frac{1}{q(\alpha)} (\|\epsilon - \hat{\epsilon}(\mathbf{z}, \gamma)\|_2^2 - f_{\Sigma}(\alpha)) \right]$$

$$\mathcal{L}_{disc} \propto \mathbb{E}_{\alpha, \mathbf{x}, \epsilon} \left[\frac{1}{2} \frac{1}{q(\alpha)} \|\epsilon - \hat{\epsilon}(\mathbf{z}, \gamma)\|_2^2 \right]$$

The random variables are $\epsilon \sim \mathcal{N}(0, \mathbb{I})$, $\mathbf{x} \sim p(\mathbf{x})$, $\alpha \sim q(\alpha)$, where $q(\alpha)$ is the importance sampling distribution described in Sec. 2.4. Empirical estimates are taken by drawing samples from these distributions and computing an empirical mean. This produces an unbiased estimate that converges to the true expectation as we include more samples. Due to the central limit theorem, we know that the distribution of this estimator will converge to a Gaussian whose mean is the true expectation and whose variance is equal to the variance of samples divided by the total number of samples, N . We used this result to estimate the standard deviation of the estimators in Sec. 2.5, where we used $n = 100$ samples of $\alpha \sim q(\alpha)$ each with 10000 points from $\mathbf{x} \sim p(\mathbf{x})$, $\epsilon \sim N(0, 1)$, so that $N = 10^6$. For the discrete case, we used $n = 1000$ samples from $q(\alpha)$ per point in results in Sec. 2.5, but we report the result here using $n = 100$ samples so that numbers are directly comparable.

The results are shown in Table A.1, and we see that variance estimates are small compared to the values reported in Fig. 2.2. Note that the continuous density estimators have smaller variance than the discrete estimator. This makes sense because the importance sampling distribution that we used was chosen to match the integrand for the continuous density estimator. The more closely the importance sampling distribution matches the target, the lower the variance. The higher variance for the discrete estimator is due to the mismatch between the curves in Fig. 2.3 and the logistic distribution, and we observe a similar phenomenon using the bootstrap estimators studied next. Therefore, we used more samples from $q(\alpha)$ when using the discrete probability estimator.

This analysis has an important caveat. These variance estimates reflect the case where all samples are drawn IID. However, for the purposes of plotting and ensembling, it is

more convenient to have errors on a common set of $\log(\text{SNR})$ values, so that we can estimate average MSE's across samples per $\log(\text{SNR})$ ($\mathbb{E}_{p(\mathbf{x})}[\text{mmse}(\mathbf{x}, \gamma)] = \text{mmse}(\gamma)$). Because we use the same set of $\log(\text{SNR})$ samples (drawn IID from $q(\alpha)$) across *all* samples, these samples are not fully IID. Hence, the standard error estimates in Table A.1, which assume fully IID samples, are imperfect. This prompted us to include an alternate analysis based on bootstrap sampling.

Table A.2: Bootstrap variance estimates for continuous estimator

	$n = 1000$	Bootstrap samples with $n = 100$
DDPM	-3.59	-3.56 ± 0.04
IDDPM	-4.14	-4.09 ± 0.10
DDPM(tune)	-3.87	-3.84 ± 0.07
IDDPM(tune)	-4.31	-4.28 ± 0.11
Ensemble	-4.31	-4.29 ± 0.11

Table A.3: Bootstrap variance estimates for discrete estimator

	$n = 1000$	Boot. $n = 100$
DDPM	3.68	3.56 ± 0.32
IDDPM	3.16	3.05 ± 0.24
DDPM(tune)	3.48	3.36 ± 0.28
IDDPM(tune)	3.15	3.04 ± 0.24
Ensemble	2.90	2.79 ± 0.27

Table A.4: Bootstrap variance estimates using uniform dequantization

	$n = 1000$	Boot. $n = 100$
DDPM	3.47	3.51 ± 0.04
IDDPM	3.12	3.17 ± 0.07
DDPM(tune)	3.37	3.41 ± 0.05
IDDPM(tune)	3.13	3.18 ± 0.07
Ensemble	3.11	3.16 ± 0.07

Bootstrap sampling analysis To get a sense for the variance of test time estimates with IID random $\log \text{SNR}$ values using our data consisting of a fixed set of random $\log(\text{SNR})$ values in common across samples, we did a bootstrap sampling analysis. For this analysis, we first constructed estimates using the full fixed set of $n = 1000$ samples of $\alpha \sim q(\alpha)$ across all samples. Then, we took 10 random subsets of $n = 100$ $\log(\text{SNR})$ values and again estimated

NLL values. We show the mean and standard deviation across the bootstrap samples in each case. The results for different estimators are shown in Tables A.2, A.3, A.4.

The use of non-IID $\log(\text{SNR})$ s does lead to higher variance. In principle, this could be avoided, even in the case when we are ensembling. To do so, we could use a validation set to determine which denoising model to use in each SNR range. Then, at test time, we could use fully IID samples for each $\log(\text{SNR})$ value.

Note that the continuous estimators (including uniform dequantization, which uses a continuous estimator to give a discrete probability estimate) have far lower variance. This is why we used fewer $\log(\text{SNR})$ samples for continuous versus discrete estimators in the main results.

Finally, note that the average bootstrap NLL using $n = 100$ in the discrete case, for example 2.79 for the ensembling (Tab. A.3), is actually lower than the best result reported in the main experiment section, $\text{NLL} = 2.90$ using $n = 1000$. We chose to report the larger but more reliable result using $n = 1000$ since the variance of the discrete estimator using $n = 100$ is large.

Diffusion models are computationally expensive and each point requires many calls per sample at test time to estimate log likelihood. Our variance analysis shows that our continuous density estimator (which can be used for discrete estimation also using uniform dequantization) can significantly reduce variance, leading to reliable estimates with far fewer model evaluations.

Appendix B

Appendix for Chapter 3

B.1 Derivations of the negative log-likelihood

For completeness, we provide a derivation for the negative log-likelihood 3.2 from [128]. We first introduce a seminal result from [77],

$$\frac{d}{d\gamma} I(\mathbf{x}; \mathbf{z}) = 1/2 \text{mmse}(\gamma). \quad (\text{B.1})$$

This relationship admits a point-wise generalization,

$$\frac{d}{d\gamma} D_{KL}[p(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})] = 1/2 \text{mmse}(\mathbf{x}, \gamma), \quad (\text{B.2})$$

The marginal is $p(\mathbf{z}) = \int p(\mathbf{z}|\mathbf{x})p(\mathbf{x})d\mathbf{x}$, and the pointwise MMSE is defined as follows,

$$\text{mmse}(\mathbf{x}, \gamma) \equiv \mathbb{E}_{p(\mathbf{z}|\mathbf{x})} [\|\mathbf{x} - \hat{\mathbf{x}}^*(\mathbf{z}, \gamma)\|_2^2]. \quad (\text{B.3})$$

To obtain the desired result, we apply the thermodynamic integration trick introduced in [121], by first defining the point-wise gap function $f(\mathbf{x}, \gamma)$ as

$$f(\mathbf{x}, \gamma) \equiv D_{KL}[p(\mathbf{z}|\mathbf{x}) \parallel p_G(\mathbf{z})] - D_{KL}[p(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})].$$

We denote $p_G(\mathbf{z}) = \int p(\mathbf{z}|\mathbf{x})p_G(\mathbf{x})d\mathbf{x}$ as the marginal output distribution of the MMSE for the channel with Gaussian input as $\text{mmse}_G(\gamma)$. In the limit of zero SNR, we get $\lim_{\gamma \rightarrow 0} f(\mathbf{x}, \gamma) = 0$. In the high SNR limit, [128] prove that

$$\lim_{\gamma \rightarrow \infty} f(\mathbf{x}, \gamma) = \log \frac{p(\mathbf{x})}{p_G(\mathbf{x})}. \quad (\text{B.4})$$

Combining this with equation B.2, we can write the log likelihood *exactly* in terms of the log likelihood of a Gaussian and a one dimensional integral.

$$\begin{aligned} -\log p(\mathbf{x}) &= -\log p_G(\mathbf{x}) - \int_0^\infty d\gamma \frac{d}{d\gamma} f(\mathbf{x}, \gamma) \\ &= -\log p_G(\mathbf{x}) - 1/2 \int_0^\infty d\gamma (\text{mmse}_G(\mathbf{x}, \gamma) - \text{mmse}(\mathbf{x}, \gamma)) \end{aligned} \quad (\text{B.5})$$

This expresses density in terms of a Gaussian density and a correction that measures how much better we can denoise the target distribution than we could using the optimal decoder for Gaussian source data. The density can be further simplified by writing out the Gaussian expressions explicitly and simplifying with an identity given in,

$$-\log p(\mathbf{x}) = d/2 \log(2\pi e) - 1/2 \int_0^\infty d\gamma \left(\frac{d}{1+\gamma} - \text{mmse}(\mathbf{x}, \gamma) \right). \quad (\text{B.6})$$

Observe that the first term in the integrand does not depend on $\mathbf{x}$, which allows us to derive the desired result Eq. 3.2. We refer readers to [128] for more detailed derivations.

B.2 Derivations of pointwise information via the orthogonality principle

Our goal is to show that the following expression,

$$\mathbf{i}^o(\mathbf{x}; \mathbf{y}) \equiv 1/2 \int \mathbb{E}_{p(\epsilon)} [\|\hat{\epsilon}(\mathbf{z}) - \hat{\epsilon}(\mathbf{z}|\mathbf{y})\|_2^2] d\alpha,$$

is a pointwise information estimator, i.e., that it satisfies the identity,

$$I(X; Y) = \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} [\mathbf{i}^o(\mathbf{x}; \mathbf{y})].$$

To show this fact, we first recall the definition of our optimal denoiser and optimal conditional denoiser.

$$\begin{aligned} \hat{\epsilon}(\mathbf{x}) &\equiv \arg \min_{\bar{\epsilon}(\cdot)} \mathbb{E}_{p(\mathbf{x}), p(\epsilon)} [\|\epsilon - \bar{\epsilon}(\mathbf{z})\|_2^2] \\ \hat{\epsilon}(\mathbf{x}|\mathbf{y}) &\equiv \arg \min_{\bar{\epsilon}(\cdot)} \mathbb{E}_{p(\mathbf{x}|\mathbf{y}), p(\epsilon)} [\|\epsilon - \bar{\epsilon}(\mathbf{z}|\mathbf{y})\|_2^2] \end{aligned}$$

For this optimal denoiser, the following expression holds exactly.

$$\begin{aligned} -\log p(\mathbf{x}) &= 1/2 \int \mathbb{E}_{p(\epsilon)} [\|\epsilon - \hat{\epsilon}(\mathbf{z})\|_2^2] d\alpha + \text{const} \\ -\log p(\mathbf{x}|\mathbf{y}) &= 1/2 \int \mathbb{E}_{p(\epsilon)} [\|\epsilon - \hat{\epsilon}(\mathbf{z}|\mathbf{y})\|_2^2] d\alpha + \text{const} \end{aligned} \tag{B.7}$$

Therefore, we have that,

$$\begin{aligned} I(X; Y) &= \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} [\log p(\mathbf{x}|\mathbf{y}) - \log p(\mathbf{x})] \\ &= \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} \left[\frac{1}{2} \int \mathbb{E}_{p(\epsilon)} [\|\epsilon - \hat{\epsilon}(\mathbf{z})\|_2^2 - \|\epsilon - \hat{\epsilon}(\mathbf{z}|\mathbf{y})\|_2^2] d\alpha \right] \end{aligned}$$

Rearranging we have the following.

$$\begin{aligned} I(X; Y) &= \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} \left[\overbrace{\frac{1}{2} \int \mathbb{E}_{p(\epsilon)} [\|\hat{\epsilon}(\mathbf{z}) - \hat{\epsilon}(\mathbf{z}|\mathbf{y})\|_2^2] d\alpha}^{\text{i}^\circ(\mathbf{x}; \mathbf{y})} \right] \\ &\quad + 2\mathbb{E}_{p(\mathbf{y})} \left[\frac{1}{2} \int \underbrace{\mathbb{E}_{p(\mathbf{x}|\mathbf{y}), p(\epsilon)} [(\hat{\epsilon}(\mathbf{z}) - \hat{\epsilon}(\mathbf{z}|\mathbf{y})) \cdot (\hat{\epsilon}(\mathbf{z}|\mathbf{y}) - \epsilon)]}_{\equiv \mathcal{O}} d\alpha \right] \end{aligned}$$

What remains is to show that the term in red is zero, $\mathcal{O} = 0$, and therefore the whole second term is equal to zero. This fact follows from the orthogonality principle [116], which states the slightly more general result that,

$$\forall \mathbf{f}, \quad \mathbb{E}_{p(\mathbf{x}|\mathbf{y})p(\epsilon)} [\mathbf{f}(\mathbf{z}, \mathbf{y}) \cdot (\hat{\epsilon}(\mathbf{z}|\mathbf{y}) - \epsilon)] = 0.$$

Note that this is stated in a slightly different way, as we have used $\mathbf{z} \equiv \mathbf{z}(\mathbf{x}, \epsilon)$ to write the noisy channel that our MMSE estimator is attempting to use to recover ϵ . The term $(\hat{\epsilon}(\mathbf{z}|\mathbf{y}) - \epsilon)$ is recognized as the error of the MMSE estimator. This error must be orthogonal to any estimator, $\mathbf{f}$. If it isn't, then we can use it to build an estimator with lower MSE than $\hat{\epsilon}(\mathbf{z}|\mathbf{y})$, contradicting our assumption that $\hat{\epsilon}(\mathbf{z}|\mathbf{y})$ is the MMSE estimator. A similar result to the orthogonality principle can be shown in a more general way using Bregman divergences [11].

Therefore, we finally have the desired result that $I(X; Y) = \mathbb{E}_{p(\mathbf{x}, \mathbf{y})}[\mathbf{i}^o(\mathbf{x}; \mathbf{y})]$. Note that this pointwise estimator has a slightly different interpretation from the standard one, $\mathbf{i}^s$, as it is not equal to a log-likelihood ratio pointwise, though it is still in expectation. On the other hand, it has several nice properties. It is non-negative, which is convenient for visualizing heatmaps. It is clear that if mutual information is zero, then the optimal denoiser should learn to ignore $\mathbf{y}$, so $\hat{e}(\mathbf{z}|\mathbf{y}) = \hat{e}(\mathbf{z})$ and our information estimate is then zero.

B.3 Additional Results

B.3.1 Relationship between image-level MI and CMI

On both the COCO100-IT and COCO-WL datasets, we conducted further calculations for image-level MI and CMI, presenting the results in scatterplots in Fig. B.1. These quantitative findings align with our pixel-level visual analysis (§3.4.2). MI and CMI exhibit strong consistency for noun words, with a high Pearson correlation coefficient of 0.89. In most cases, MI values remain higher than CMI, primarily due to MI containing more information from the background context. However, for abstract words, the Pearson coefficient drops to 0.17, and notably, MI is consistently larger than CMI (with most cases being nearly zero), indicating MI’s superior capability in capturing information involving abstract words compared to CMI. This signals a high degree of redundancy between abstract words and context [267].

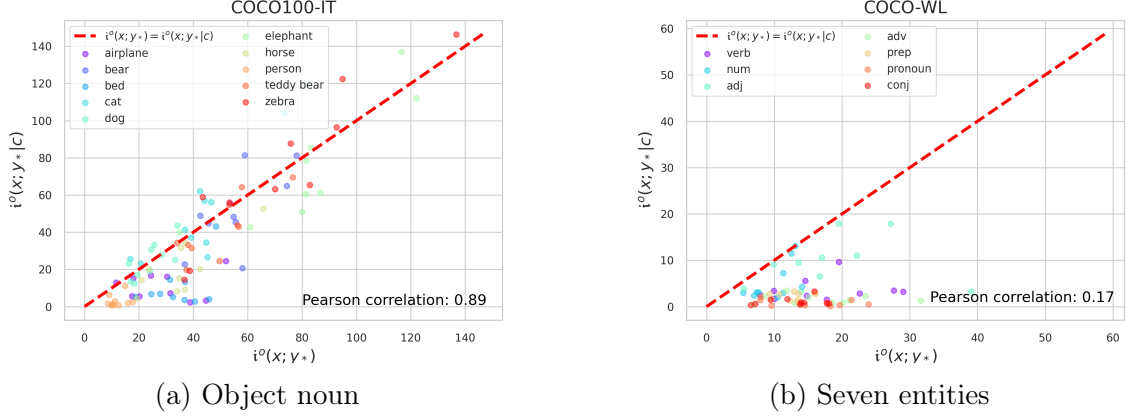


Figure B.1: Scatter plot for correlation between MI and CMI.

B.3.2 Image-level MMSE curves and pixel-level MMSE visualizaiton

We analyze the image-level (Fig. B.2) and pixel-level (Fig. B.5 and B.6) MMSE for 10 cases in COCO100-IT. To fully harness the capabilities of ITD [128], we configured the diffusion steps to be 200. We conducted 50 samples under the same α , and the MMSE results are derived from the average denoising of these samples. However, for the purpose of pixel-level visualization, we selected only 20 steps.

From Fig. B.2, it becomes evident that as α varies, the orthogonal approximation exhibits greater stability with fewer zigzag patterns compared to the standard version. Furthermore, the orthogonal method enhances the consistency between MMSE and conditional MMSE, leading to synchronized peaks and similar distributions. The diffusion process reveals that the optimal performance for locating object-related pixels in the image coincides with the appearance of peaks in Fig B.2. When the α is too high, the highlighted pixels gradually become sparse, while excessively low α values lead to chaos in MMSE.

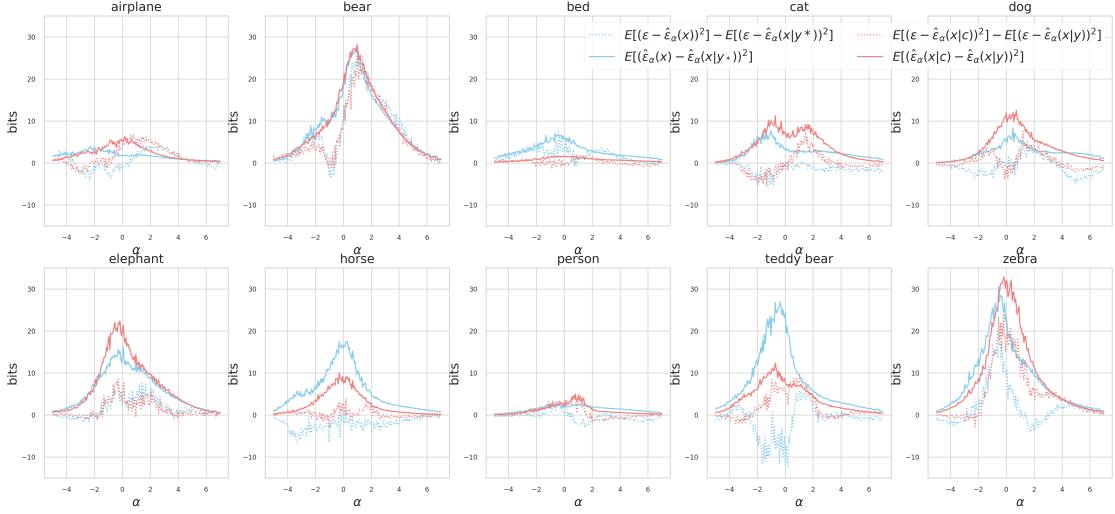


Figure B.2: MMSE curves examples for 10 categories.

B.3.3 Examples of word location for object nouns

We put more pixel-level MI and CMI visualization examples from COCO100-IT, see Fig. B.7 & B.8 & B.9.

B.3.4 Examples of word location for other seven entities

We put more word location visualization examples for seven entities from COCO-WL, see Fig. B.11 & B.10.

B.3.5 Intervention experiments additional results

We observe that correlations between heatmaps (from conditional mutual information or from attention) often correlate strongly with changes in the image after intervention. However, this is not always the case. We show a negative and positive example in Fig. B.3. We see in some images that one word change globally changes the image, leading to poor

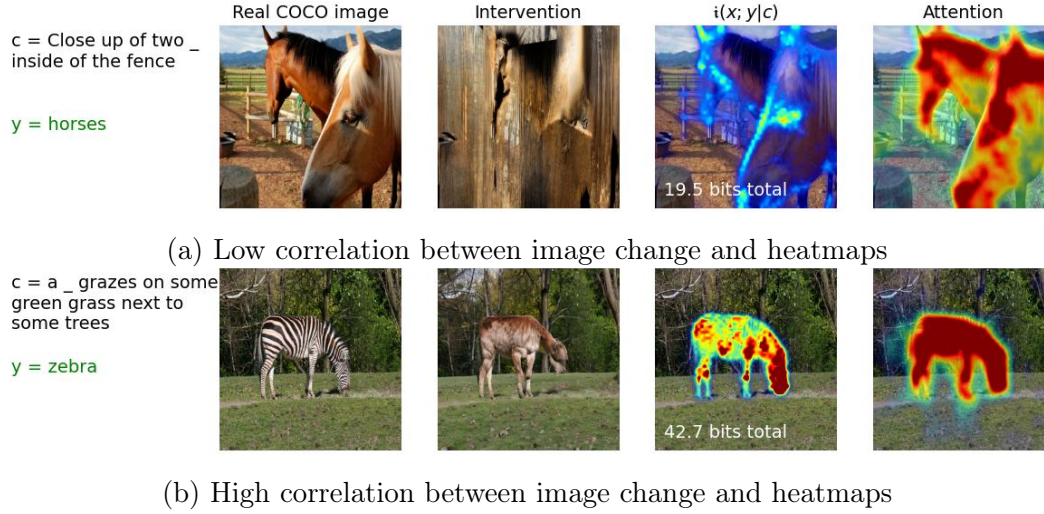


Figure B.3: (a) An example where the correlation between pixel-level changes and CMI or attention are low (0.25 and -0.21 respectively). (b) An example where the pixel-level correlation is high (0.73 and 0.77 respectively).

correlation. This result, however, does not contradict our original hypothesis, which is that small CMI implies that omitting a word will have no effect. We generally observe this to be true. However, when the CMI is large, the effect may or may not be correctly localized. The reason that the effect is not always correctly localized is that generation is an iterative procedure: a small change in the first step can lead to global changes in the image.

Fig. B.12 and B.13 visualize additional examples where we swap a word in a caption with a categorically similar word. For the COCO-IT dataset described in §B.5, we explored the following word swaps: dog $\leftrightarrow$ cat, zebra $\leftrightarrow$ horse, bed $\leftrightarrow$ table, bear $\leftrightarrow$ elephant, airplane $\leftrightarrow$ kite, person $\leftrightarrow$ clown, plus plural versions. In these plots, and also Fig. 3.1 and B.3, the pixel values represent i^o and are shown with a colormap where the maximum value corresponds to 0.15 bits/pixel. However, the “total information” shown in white text uses the unbiased estimate i^s , and hence can sometimes be negative. Attention color maps are normalized as was done by [244].

B.4 Experimental Settings

We provide code for reproducing our experiments at <https://github.com/kxh001/Info-Decomp>.

B.4.1 Relation Testing with Pointwise Information

We refer readers to Table B.1 for additional implementation details for evaluating the ARO benchmark. All datasets are prepared following the official implementation of [274] available at <https://github.com/mertyg/vision-language-models-are-bows.git>. All experiments are run on NVIDIA RTX A6000 GPUs.

Table B.1: Additional Experiment Details for the ARO Benchmark

	VG-A	VG-R	COCO	Flickr30k
Perturbation size	1	1	4	4
Dataset size	28,748	23,937	25,010	5,000
Inference batch size	10	10	5	5
SNR sample size	100	100	100	100

We evaluate the OpenCLIP checkpoint `laion/CLIP-ViT-H-14-laion2B-s32B-b79K`. This checkpoint consists of a 330M BERT-style encoder trained on the LAION-2B Dataset (available at: <https://huggingface.co/datasets/laion/laion2B-en>). Its text encoder is consistent with the one deployed by Stable Diffusion version 2.1 to ensure fair comparison. We use a batch size of 80 for all OpenCLIP evaluations.

Table B.2 reports a more detailed set of experiments conducted on the ARO benchmark. First, we empirically study the effect of SNR importance sampling distributions on the consistency of our estimator, by reporting performances on uniform and logistic

distributions. Sampling from the latter marginally but consistently out-performs its uniform counterpart, which supports the theoretical findings of [128]. We report disagreements across relations in Table B.6, and observe disagreements to be consistently low between two distributions.

In the third-row of Table B.2, we report the performance of OpenCLIP wherein the last layer of its text encoder is removed. This setup is consistent with Stable Diffusion’s usage of text encoder, but we observe the results to be similar for *VG-Relation* and *VG-Attribution* (1 perturbation), and significantly worse for *COCO-Order* and *Flickr30k-Order* (4 perturbations). All other entries are identical to Table 3.1 for reference.

In Table B.6, we report fine-grained performance of Stable Diffusion and OpenCLIP systems across each relation type. In column 3, we report normalized prediction disagreement between uniform and logistic sampling, and observe the predictions to be generally consistent.

Finally, we assess the consistency of our estimator across random seeds. For each dataset, we select the first 1000 samples, evaluate our estimator across 3 random seeds, and provide OpenCLIP baseline on the same subsets as for reference. Numerical results are provided in Table B.3. The information estimates are relatively consistent, and establish statistically significant performance gap compared to OpenCLIP.

Table B.2: Additional Accuracy (%) of Stable Diffusion and OpenCLIP.

Method	VG-A	VG-R	COCO	Flickr30k
Baseline (Random Guess)	50.0	50.0	20.0	20.0
OpenCLIP [101]	64.6	51.4	32.8	40.5
OpenCLIP (all-but-last)	65.6	50.9	22.4	28.8
Info. (Ours, Uniform)	71.2	68.5	39.3	48.7
Info. (Ours, Logistic)	72.0	69.1	40.1	49.3

Table B.3: Mean Estimator Accuracy and Std. Dev. across Random Seeds

	VG-A	VG-R	COCO	Flickr30k
	Accuracy $\pm$ Standard Error (%)			
Info. (Unif.)	71.7 $\pm$ 0.59	72.3 $\pm$ 2.87	36.3 $\pm$ 0.87	50.5 $\pm$ 0.63
Info. (Log.)	72.7 $\pm$ 1.32	73.4 $\pm$ 1.13	36.8 $\pm$ 0.50	51.2 $\pm$ 0.73

B.4.2 Localizing Word Information in Images

In our word localization experiments, we utilized a pre-trained Stable Diffusion v2-1-base model card available at <https://huggingface.co/stabilityai/stable-diffusion-2-1-base> Huggingface. Input images were resized to 512×512 and then normalized to the $[0, 1]$ pixel value range to ensure compatibility with the pre-trained model.

The DAAM [244] is essentially an extension integrated into Stable Diffusion models, designed to generate attention-based heatmaps concurrently with the image generation process. To leverage DAAM, it is imperative to pair it with a diffusion scheduler. In our experiments, we draw inspiration from [151] and employ a DDIM [229] scheduler as a baseline. While our ITD model [128] is capable of independently generating MI and CMI heatmaps using the principles outlined in §3.2, we also had the option to enhance attention heatmaps by integrating DAAM with the information-theoretic diffusion process. Hence, we established three sets of comparative experiments: DAAM+DDIM (Attention), ITD (CMI), and DAAM+ITD (Attention+Info.) respectively. We opt not to use classifier-free guidance since it primarily aids in image generation and introduces additional undesired content onto images in the denoising process. We utilize the “alphas_cumprod” in the scheduler of the Stable Diffusion model to compute the α range spans from -5 to 7. For specific t -to- α transformation calculations, you could refer to Appendix B.2 in [128]. Thus, the parameters

of the corresponding logistic distribution are $[\text{loc}, \text{scale}, \text{clip}] = [1, 2, 3]$. Unfortunately, DAAM only supports a batch size of 1. For DAAM+DDIM and DAAM+ITD, we utilize the following input: $(\mathbf{x}, \mathbf{y}, y^*)$. In the case of ITD, the input configuration is $(\mathbf{x}, \mathbf{y}, \mathbf{c})$. This distinction arises from the fact that DAAM generates heatmaps based on the cross-attention map, enabling the direct calculation of the score for an individual object word on each pixel. On the other hand, ITD relies on conditional mutual information, $i_j^o(\mathbf{x}; y_* | \mathbf{c})$.

Table B.4: The hyper-parameters used in DDIM and ITD schedulers.

random seed	batch size (DDIM/ITD)	logistic distribution (ITD)	guidance (DDIM)
42	1/10	$[\text{loc}, \text{scale}, \text{clip}] = [1, 2, 3]$	1

All experiments were conducted using Nvidia RTX 6000 GPU cards. The hyper-parameters used in these experiments are summarized in Table B.4, with variations in the number of diffusion steps set at 50, 100, and 200. Once the heatmap of the image is computed, we initially rescale it to the $[0, 1]$ range and subsequently apply a uniform hard threshold on them for segmentation. After experimenting with hard thresholds vary in $[0, 1]$, we identify the optimal threshold that yields the highest mIoU value, then record the mIoU in Table 3.3. Unless explicitly stated, all visualization for MI, CMI, and attention heatmaps are based on 100 diffusion steps.

Table B.5: Unsupervised Object Segmentation mIoU (%) Standard Error Analysis on COCO-IT

Method	1 step	50 steps	100 steps	200 steps
Whole Image Mask	14.94 ± 0.0022	14.94 ± 0.0022	14.94 ± 0.0022	14.94 ± 0.0022
Attention	37.89 ± 0.0030	34.52 ± 0.0030	34.90 ± 0.0030	35.35 ± 0.0030
CMI	21.73 ± 0.0023	32.31 ± 0.0026	33.24 ± 0.0026	33.63 ± 0.0026
Attention+Info.	37.96 ± 0.0030	42.46 ± 0.0032	42.71 ± 0.0032	42.84 ± 0.0032

We calculated the standard error of the IoU values for object segmentation experiments conducted on COCO-IT, and the results are documented in Table B.5. This indicates that the number of diffusion steps does not significantly affect the variation in IoU values. Notice that Table B.5 includes an additional column for the 1-step experiment results. The 1-step DAAM-DDIM diffusion process can be regarded as denoising images with imperceptible noise, which is surprisingly effective compared to the multi-step results. However, computing MI and CMI only at a single step, or α , is not directly comparable. The steps in that case are interpreted as elements in a sum approximating an integral, and we don't expect a one step sum to be a good estimate. Additionally, as per the analysis in §B.3.2, peaks are required for an accurate match between relevant pixels and object words, which cannot be predicted in advance. Nonetheless, the results still demonstrate that the information-theoretic diffusion process enhances attention with respect to object segmentation. Additionally, it's noteworthy to mention that the generation process for MI or CMI from ITD differs from the generation of DAAM. DAAM requires continuous noise addition and denoising iterations to compute, while ITD first samples a series of α , and then each α can undergo independent noise addition and denoising computations. Finally, MI or CMI is calculated by one integration, which facilitates parallel computing, see Fig. B.4.

B.5 COCO-IT Dataset Preparation

While the MSCOCO [144] dataset boasts ample image-text pairs, not every object present in the images is mentioned in the captions, even if these objects have been labeled and annotated. In our experiments, we would like to test (1) the mutual information between

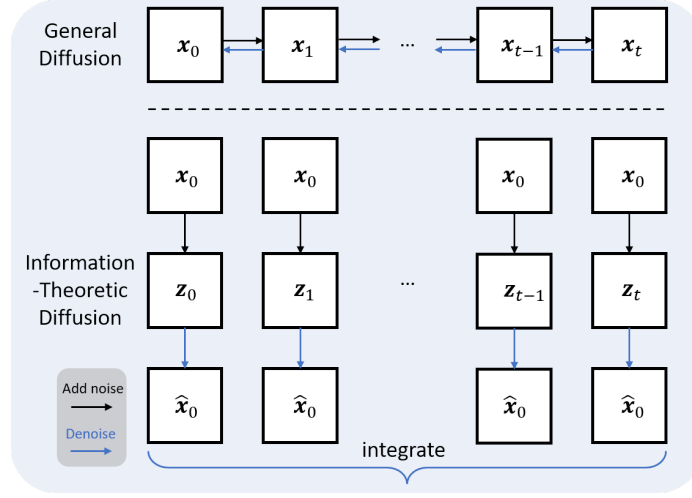


Figure B.4: The diagram of two different diffusion processes.

a complete prompt and the corresponding image, and (2) the conditional mutual information between the object word and the image. Therefore, we filtered the original COCO 2017 validation dataset using the following steps:

- (a) Traverse all the objects in an image.
- (b) Match each object word to the caption containing that object.
- (c) Generate one data point with four contains: [image, caption, context, object].
- (d) If one object doesn't appear in the captions, then omit that data point.

After applying this filter, we acquired a dataset, COCO-IT, comprising 6,927 validation image-text data points and 79 categories. To facilitate more effective visualization, we further randomly selected 10 categories from it, choosing 10 image-text pairs for each to create a smaller dataset, COCO100-IT. Additionally, we constructed a dataset, COCO-WL, for word localization by selecting 10 cases for seven different entities (verb, num., adj., adv., prep., pron., conj.).

B.6 Implementation Details of MIND Training

We follow the implementation of [65] which uses stacked multi-layer perception (MLP) with skip connections. We adopt a simplified version of the same network architecture: this involves three Residual MLP blocks. We use the *Adam optimizer* [119] for training and Exponential moving average (EMA) with a momentum parameter $m = 0.999$. We use the *ReduceLROnPlateau* scheduler with a patience of 200 epochs, reducing the learning rate by half if the training loss does not improve after 200 epochs. We returned the mean estimate on the test data set over 10 runs. All experiments are run on NVIDIA RTX A6000 GPUs. The hyper-parameters are presented in Table B.7 for MIND. Concerning the consistency tests, we independently train an autoencoder for each version of the MNIST dataset with r rows available. We compare MIND against neural estimators, such as MINE [14], INFONCE [175], NWJ [168], DOE [161], f -DIME [135] and MINDE [65].

B.6.1 Ablation Study on Adaptive Sampling

In this section, we conduct an ablation study to specifically evaluate the impact of our adaptive importance sampling strategy. We compare the performance of two non-orthogonal estimator variants:

- **MIND-adaptive ("Adaptive"):** Employs the adaptive importance sampling scheme.
- **MIND ("Baseline"):** Uses a fixed, default importance sampling distribution.

This comparison is designed to isolate the effect of the sampling strategy on estimation accuracy and stability, particularly across a diverse set of distributions. The results are presented in Table B.9.

The results in Table B.9 demonstrate the consistent benefits of the adaptive sampling strategy. The "Adaptive" method frequently achieves a lower standard deviation, indicating improved estimator stability. This is particularly evident in high-dimensional and complex transformation tasks, such as ‘multinormal-dense-50-50-0.5’ and ‘spiral-multinormal-sparse-25-25-2.0’, where the reduction in variance is substantial. While the accuracy of the point estimate is competitive across both methods, the adaptive approach often provides estimates closer to the ground truth in the more challenging settings. Overall, this ablation study validates that tailoring the sampling distribution to the specific data leads to a more robust and reliable MI estimator.

B.6.2 Visual Analysis of the MIND Estimator

This section provides an intuitive, qualitative analysis of the MIND estimator’s integrand to support the core components of our method. By visualizing the integrand on a challenging three-dimensional, Spiral-transformed sparse Multinormal distribution (GT MI = 9.90), we can clearly see the complementary roles of the orthogonal principle for stability and adaptive importance sampling for accuracy. The plots below were generated by densely sampling 10,000 log SNR points and then binning the results (bin=50) to illustrate the underlying trends. Figure ?? directly compares the integrand calculated via direct MMSE subtraction against the one derived from our orthogonal principle.

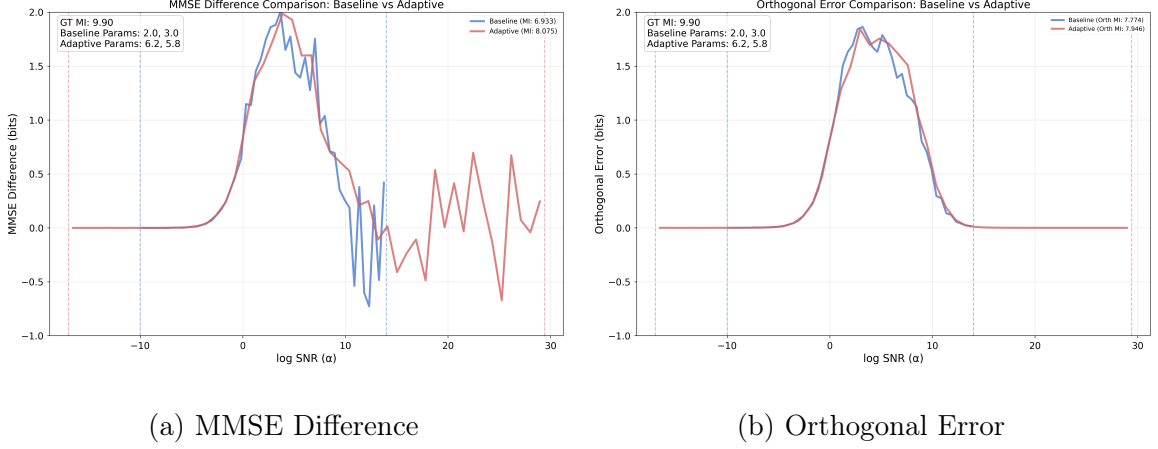


Figure B.16: **Integrand analysis on a Spiral-transformed task (GT MI = 9.90).** Comparison between (a) the volatile direct MMSE subtraction method and (b) the stable orthogonal principle. In both plots, the adaptive sampler (red) outperforms the baseline (blue) by focusing on the most informative region.

Analysis of the Orthogonal Principle’s Contribution The orthogonal principle’s primary contribution is enhancing integrand stability. As shown in Figure B.16b, it replaces the highly volatile direct subtraction method with an exceptionally smooth and non-negative integrand, crucial for reliable numerical integration. This stability, however, introduces a trade-off that becomes particularly evident in **high mutual information estimation**: the orthogonal integrand’s peak is lower, which can lead to a conservative bias by underestimating the true distance between optimal denoisers. While the dramatic reduction in variance makes it a more robust choice for general cases, this systematic underestimation can become a limiting factor when the true MI is large.

Analysis of Adaptive Sampling’s Contribution Adaptive sampling boosts estimation accuracy and efficiency. As illustrated in Figure B.16b, while a fixed sampler may be misaligned with the integrand’s peak, our adaptive method dynamically concentrates samples

in this most informative SNR region. This targeted strategy leads to a demonstrably more accurate MI estimate (**8.075 bits** vs. **6.933 bits** for the direct method), confirming the value of focusing the integration where it matters most.

Table B.6: Fine-grained results in Visual Genome Relation dataset.

	Info. (Unif.) ($\uparrow$)	Info. (Log.) ($\uparrow$)	Disagreement ($\downarrow$)	OpenCLIP ($\uparrow$)	# Samples
Accuracy (%)	68.5	69.1	6.7	51.4	
Spatial Relationships					
above	49.8	53.2	5.6	55.0	269
at	70.7	72.0	9.3	66.7	75
behind	39.9	39.9	4.5	54.4	574
below	49.3	46.4	7.7	49.8	209
beneath	50.0	50.0	0.0	90.0	10
in	76.7	79.9	5.5	51.6	708
in front of	70.2	68.7	7.3	63.1	588
inside	69.0	74.1	8.6	56.9	58
on	75.1	75.6	6.4	51.0	1684
on top of	62.7	63.7	9.0	46.3	201
to the left of	51.1	51.2	7.8	50.5	7741
to the right of	48.6	49.3	7.9	49.8	7741
under	47.0	46.2	3.8	43.9	132
Verbs					
carrying	58.3	66.7	8.3	33.3	12
covered by	36.1	33.3	8.3	55.6	36
covered in	14.3	14.3	14.3	50.0	14
covered with	18.8	18.8	0.0	43.8	16
covering	63.6	72.7	15.2	54.5	33
cutting	91.7	91.7	0.0	66.7	12
eating	85.7	85.7	0.0	57.1	21
feeding	40.0	50.0	10.0	100.0	10
grazing on	60.0	60.0	0.0	30.0	10
hanging on	57.1	71.4	14.3	78.6	14
holding	90.1	87.3	5.6	52.1	142
leaning on	66.7	66.7	0.0	66.7	12
looking at	80.6	83.9	3.2	48.4	31
lying in	100.0	100.0	0.0	33.3	15
lying on	81.7	86.7	5.0	40.0	60
parked on	76.2	71.4	4.8	61.9	21
reflected in	71.4	64.3	7.1	61.9	14
resting on	69.2	84.6	15.4	15.4	13
riding	80.4	76.5	7.8	37.3	51
sitting at	65.4	69.2	3.8	38.5	26
sitting in	82.6	82.6	8.7	65.2	23
sitting on	80.6	78.9	8.6	49.7	175
sitting on top of	80.0	50.0	30.0	60.0	10
standing by	91.7	91.7	16.7	50.0	12
standing in	89.8	91.5	8.5	49.2	59
standing on	78.8	82.7	3.8	55.8	52
surrounded by	64.3	57.1	7.1	42.9	14
using	100.0	100.0	0.0	21.1	19
walking in	90.0	90.0	0.0	70.0	10
walking on	94.7	94.7	0.0	36.8	19
watching	72.7	77.3	4.5	31.8	22
wearing	82.7	84.1	6.4	44.9	949

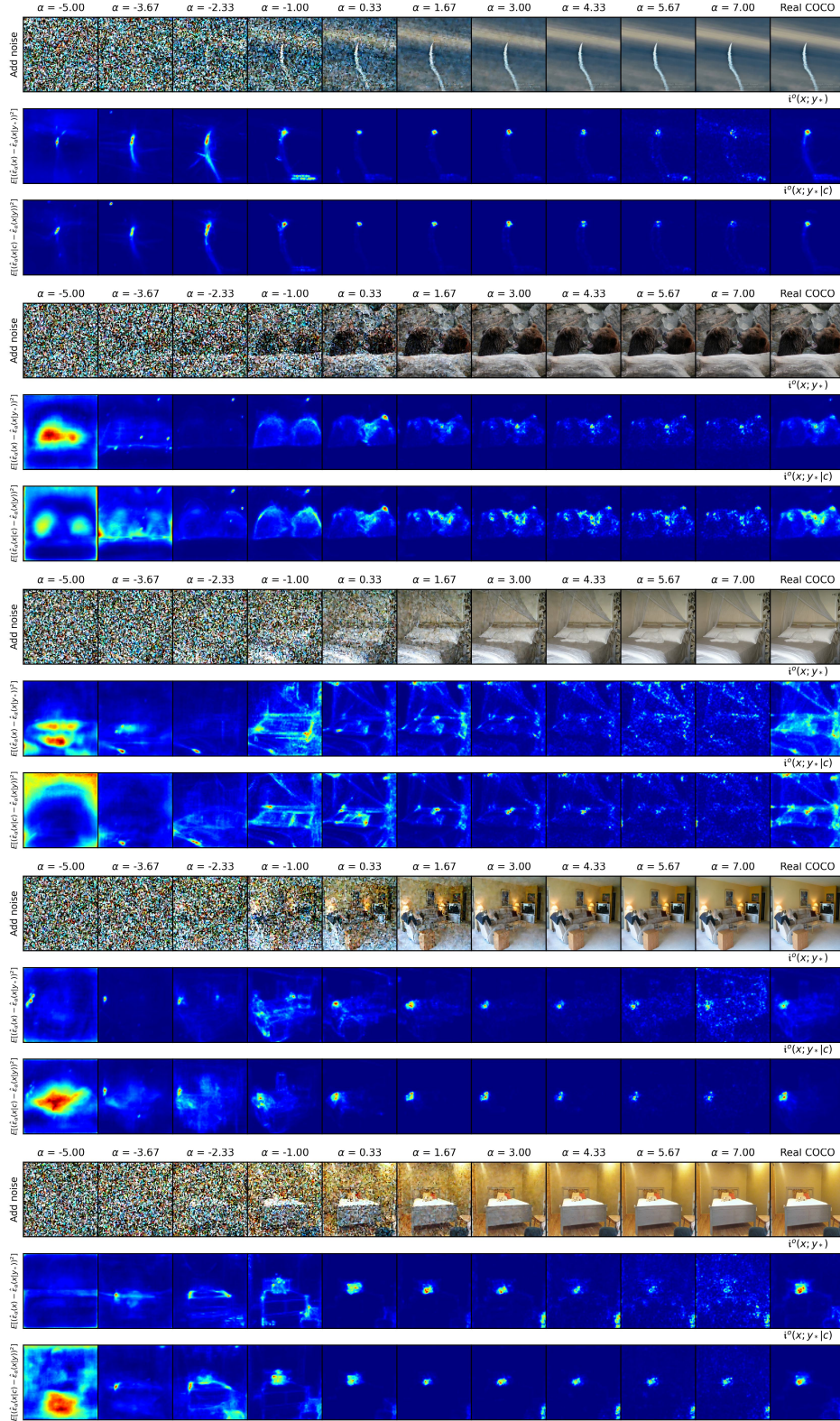


Figure B.5: Examples of pixel-level MMSE visualization.

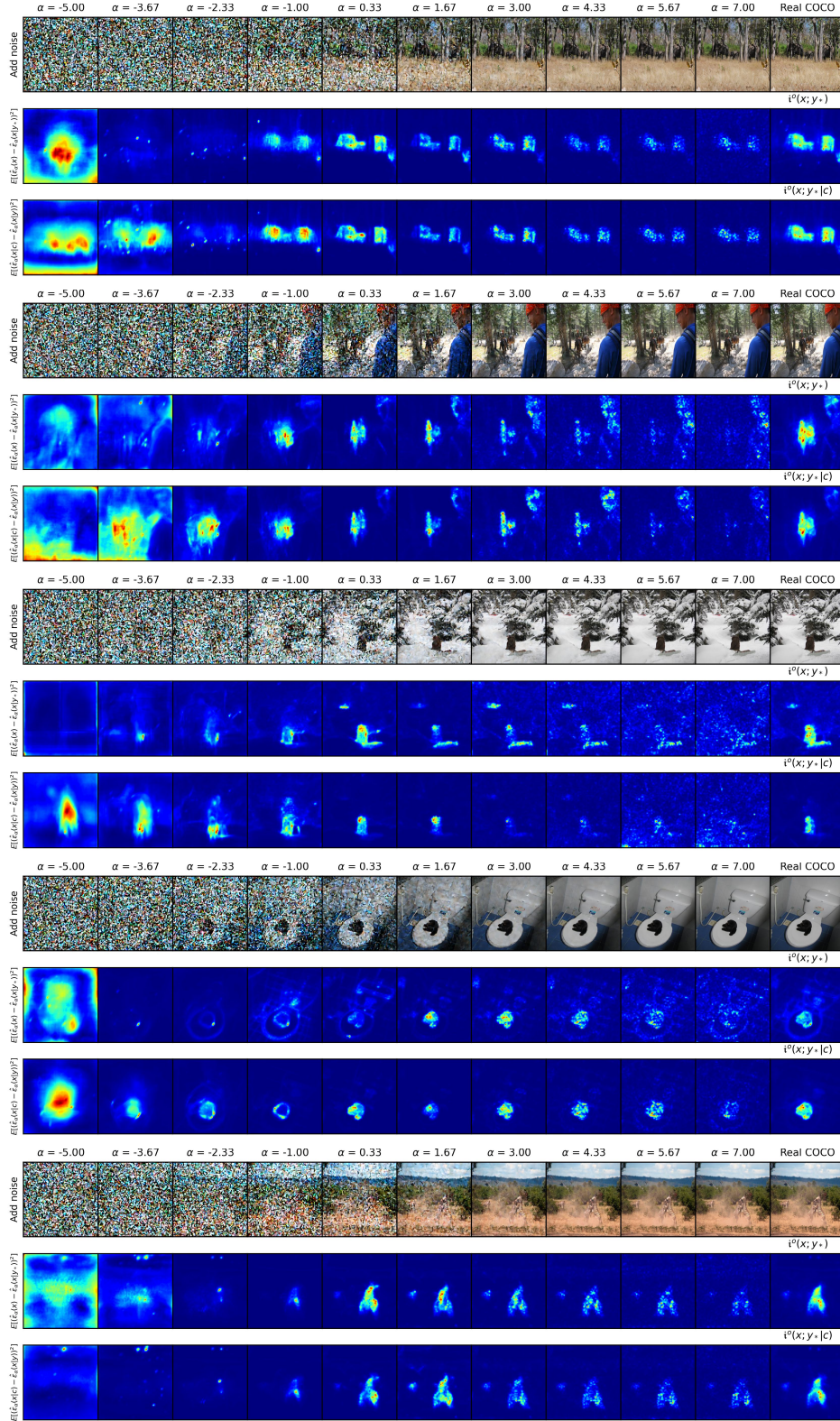


Figure B.6: Examples of pixel-level MMSE visualization.

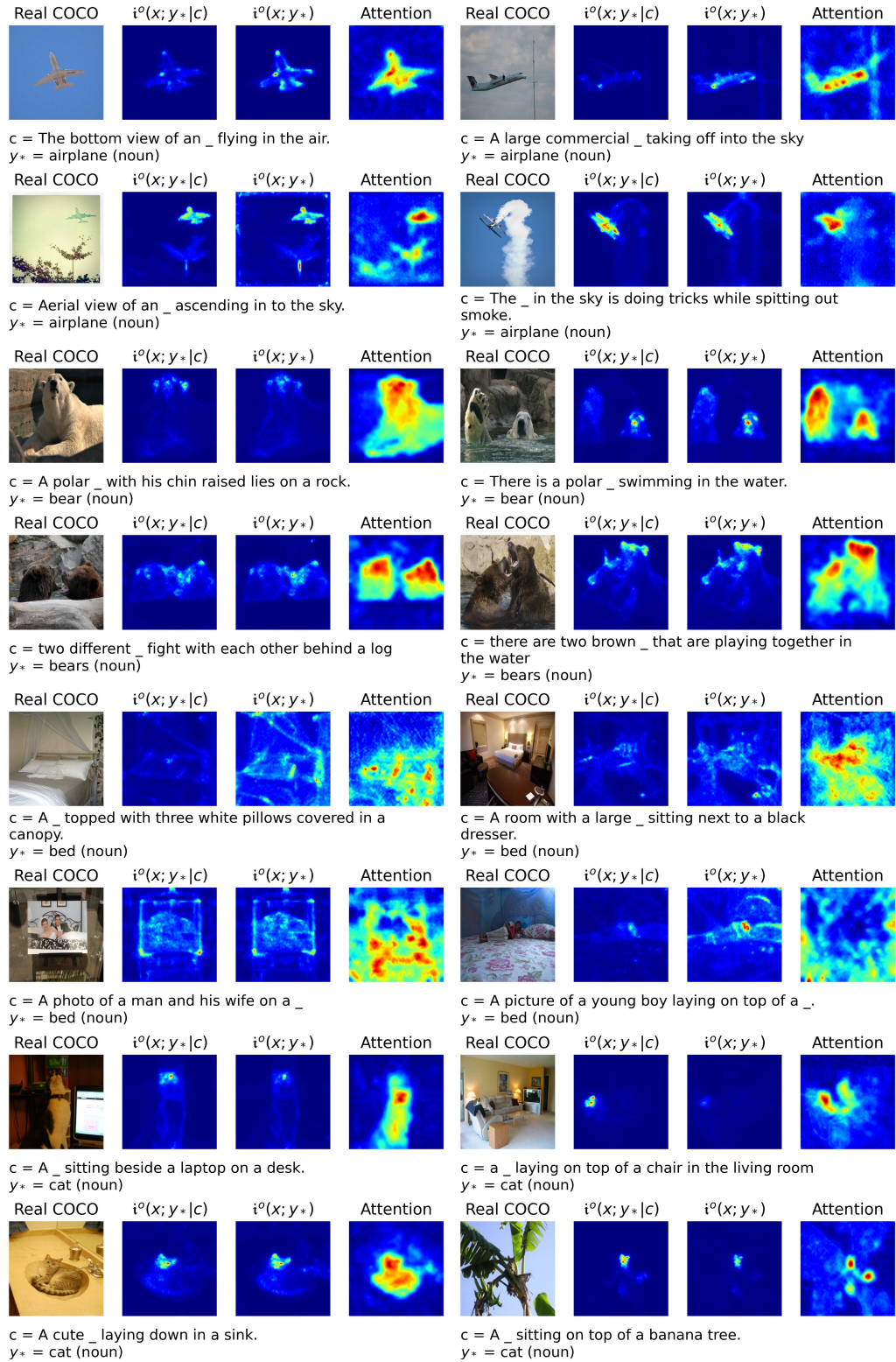


Figure B.7: Examples of localizing noun words in images.

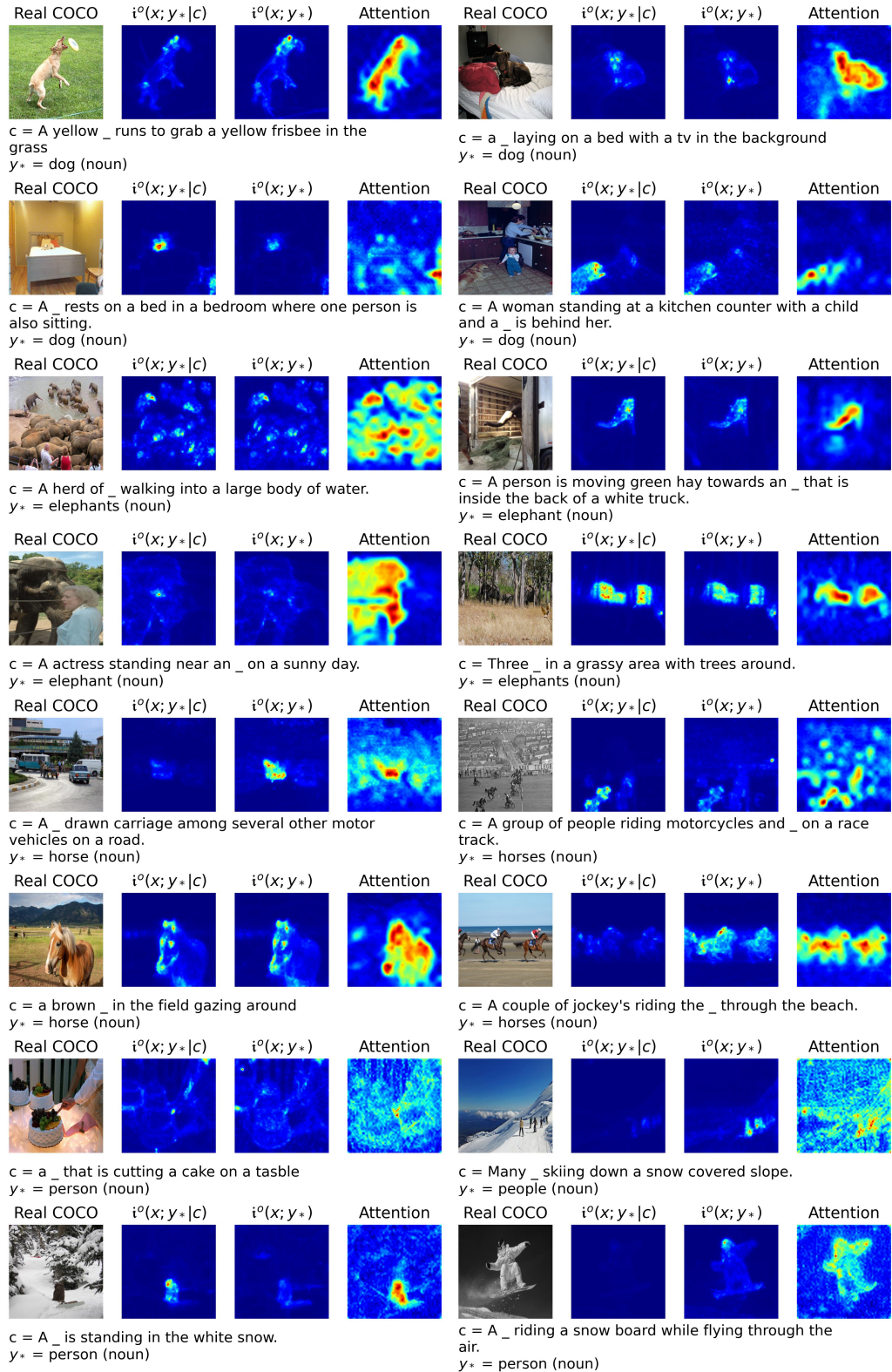


Figure B.8: Examples of localizing noun words in images.

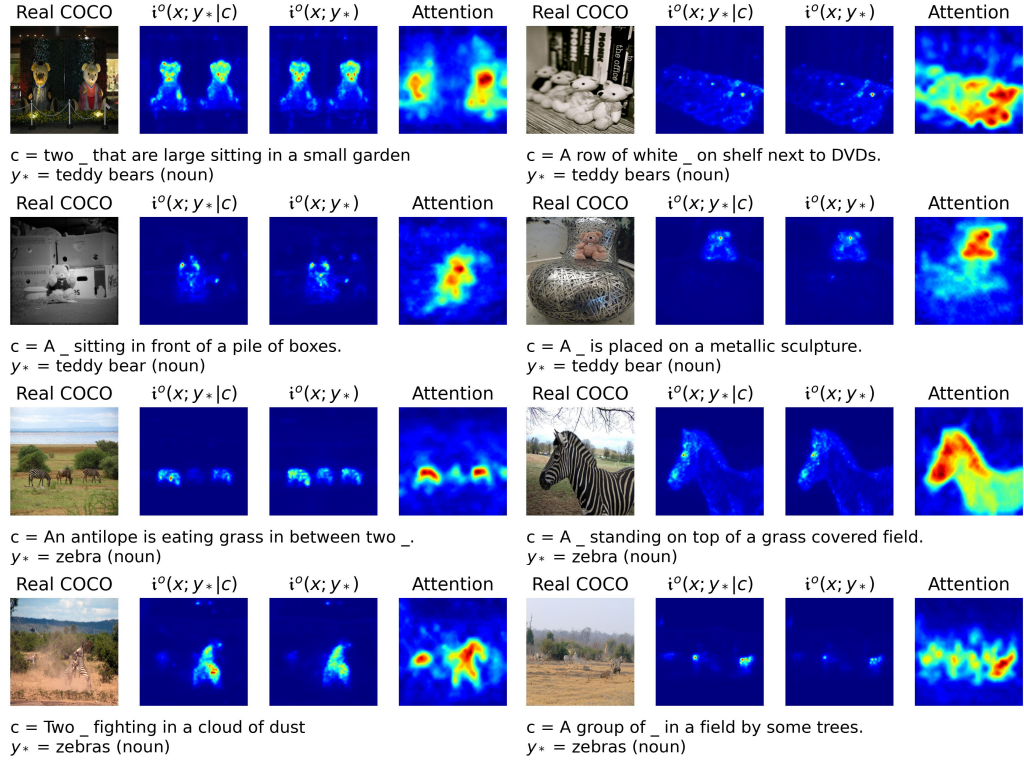


Figure B.9: Examples of localizing noun words in images.

Table B.7: MIND network training hyper-parameters. Dim of the task correspond the sum of the two variables dimensions, and d corresponds to the randomization probability.

Benchmark	Dim	d	Width	Time Emb Size	Batch Size	lr	Iterations	# of Params
≤ 10		0.5	64	64	128	1e-3	390k	55425
50		0.5	128	128	256	2e-3	290k	220810
100		0.5	256	256	256	2e-3	290k	898354
Consistency Tests		0.5	256	256	64	1e-3	390k	1597968

Table B.8: Sampling Hyperparameters

Parameter	Loc	Scale	Clip	EMA Decay	Inference Repeat Times	n_points
-	2.0	3.0	4.0	0.999	10	10000

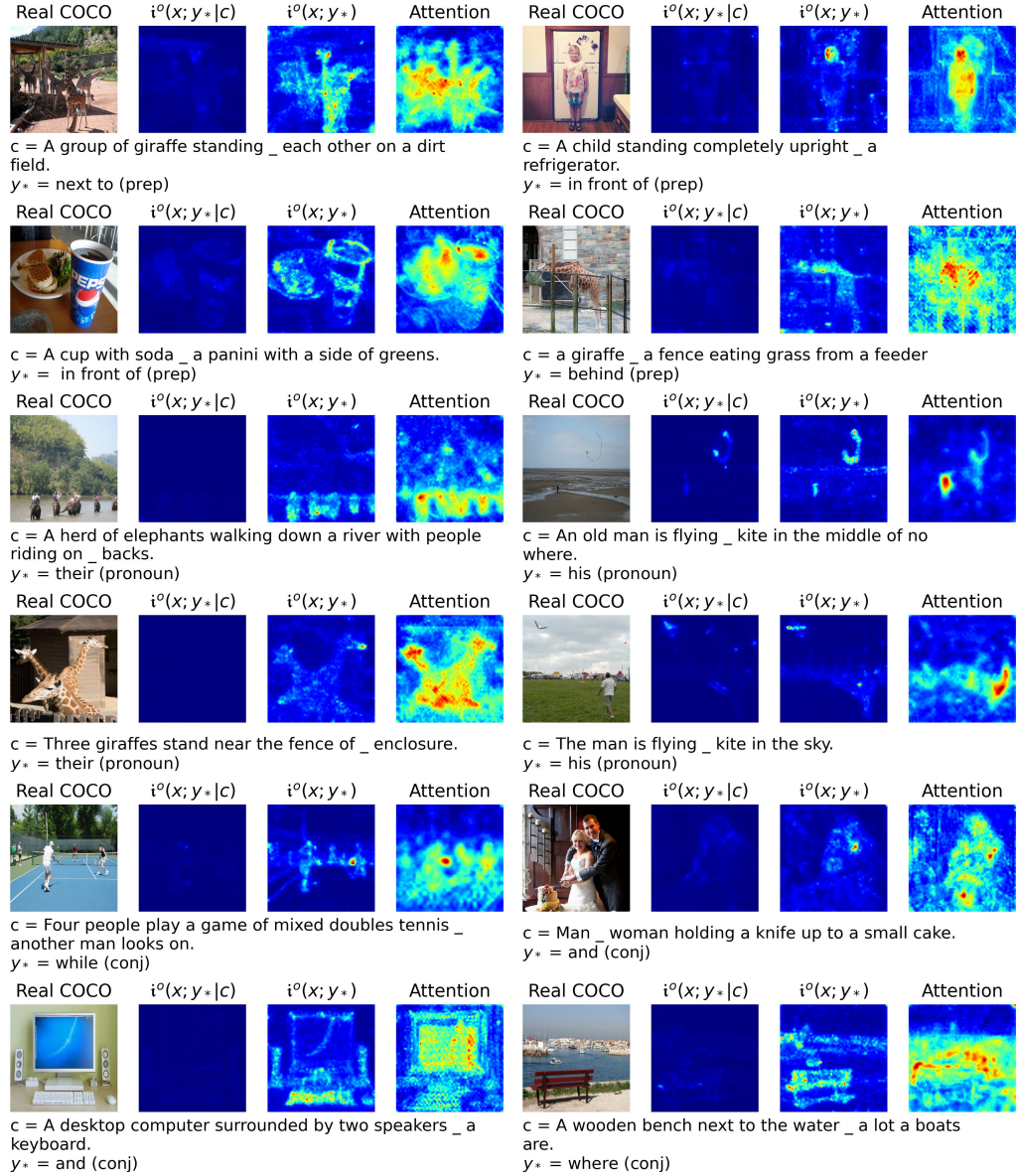


Figure B.10: Examples of localizing abstract words in images.

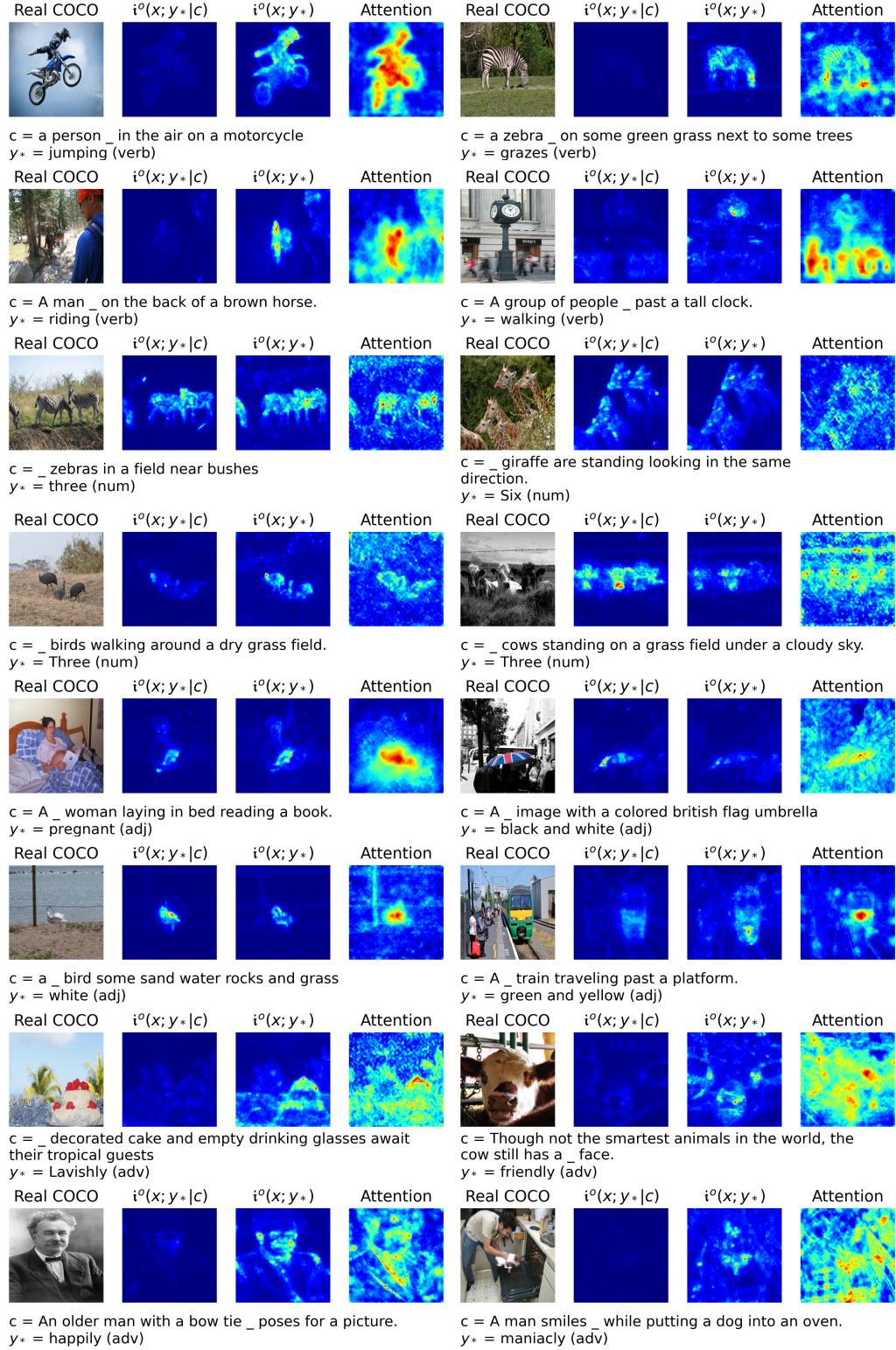


Figure B.11: Examples of localizing abstract words in images.

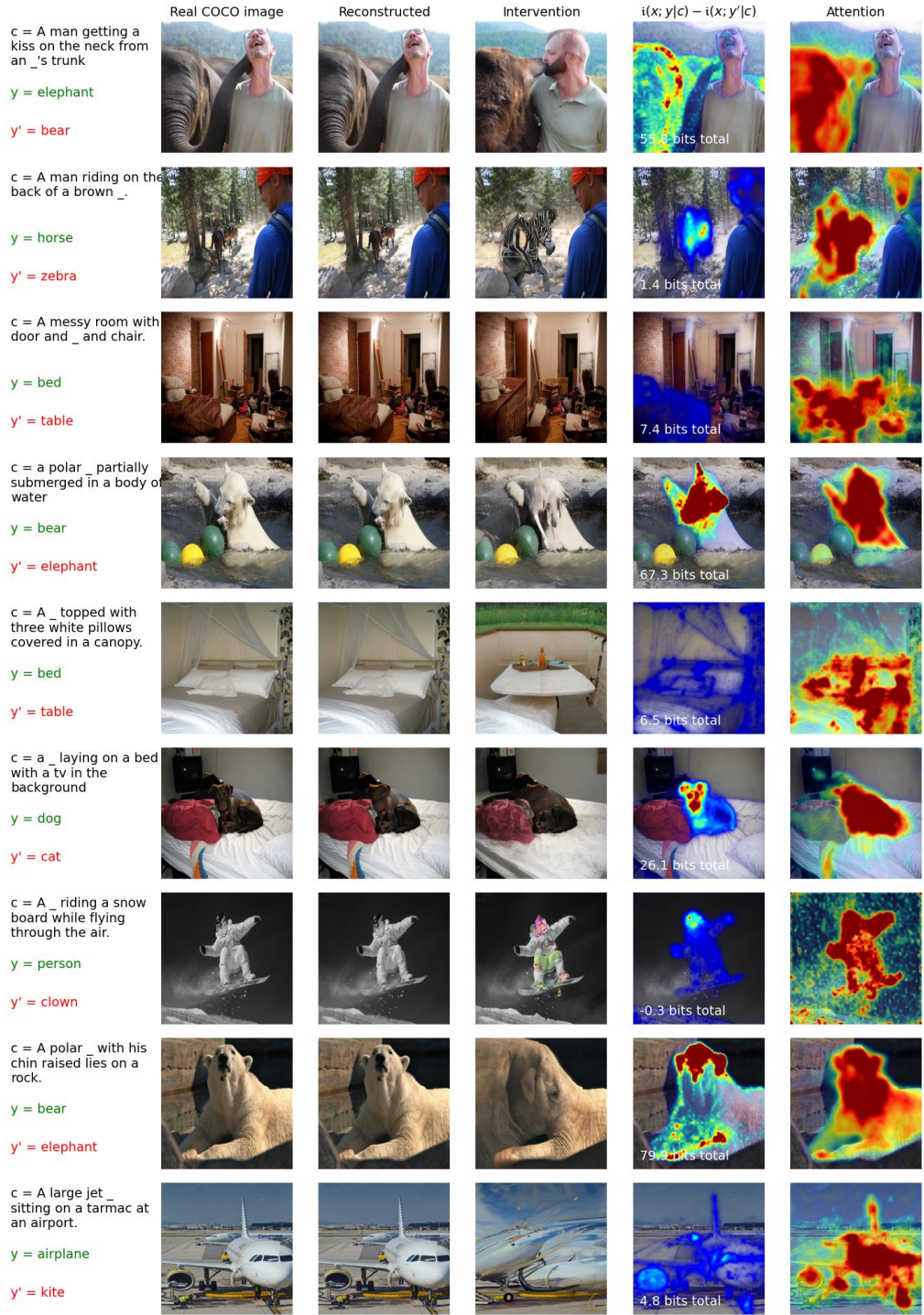


Figure B.12: Examples of word swap interventions

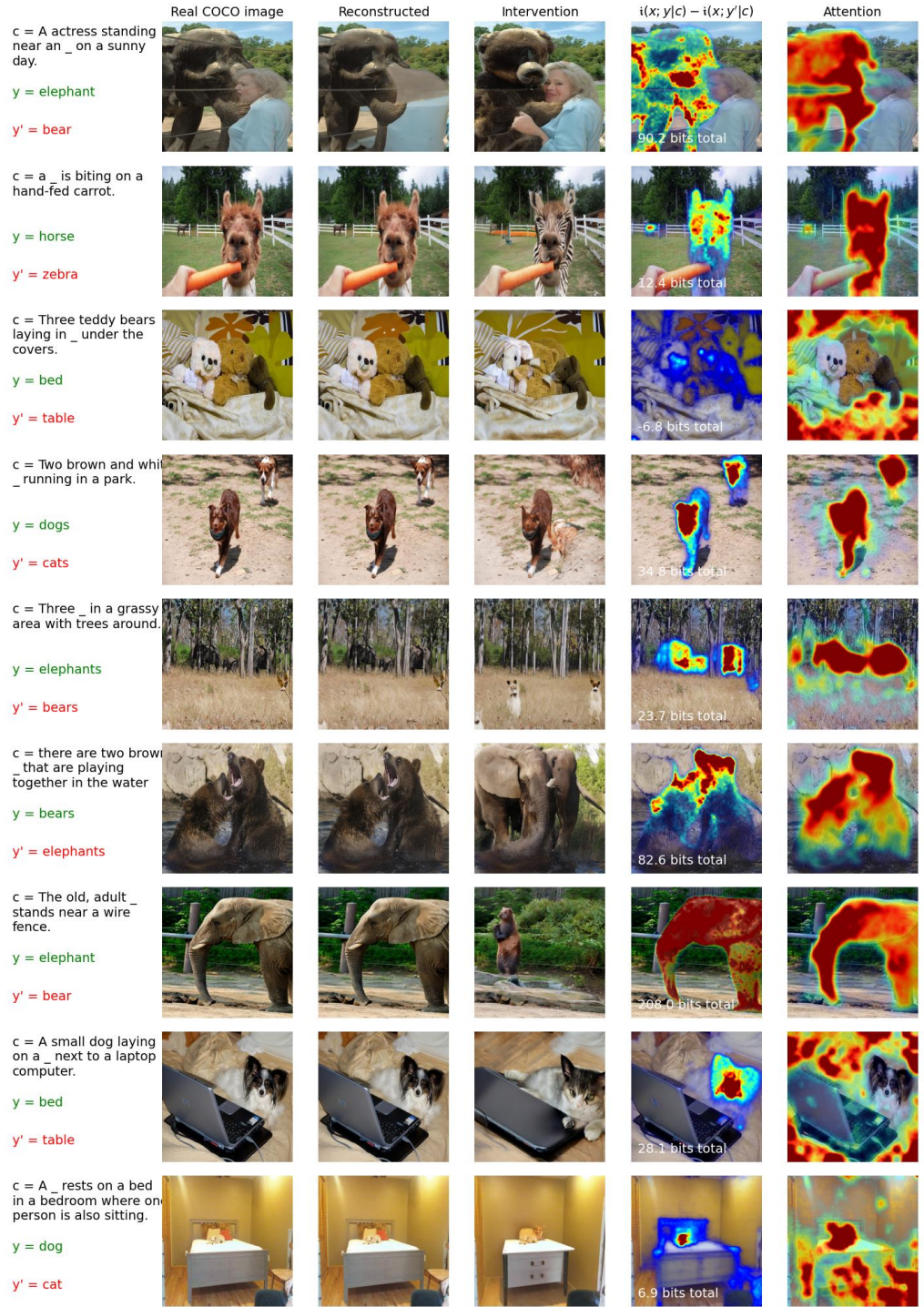


Figure B.13: Examples of word swap interventions

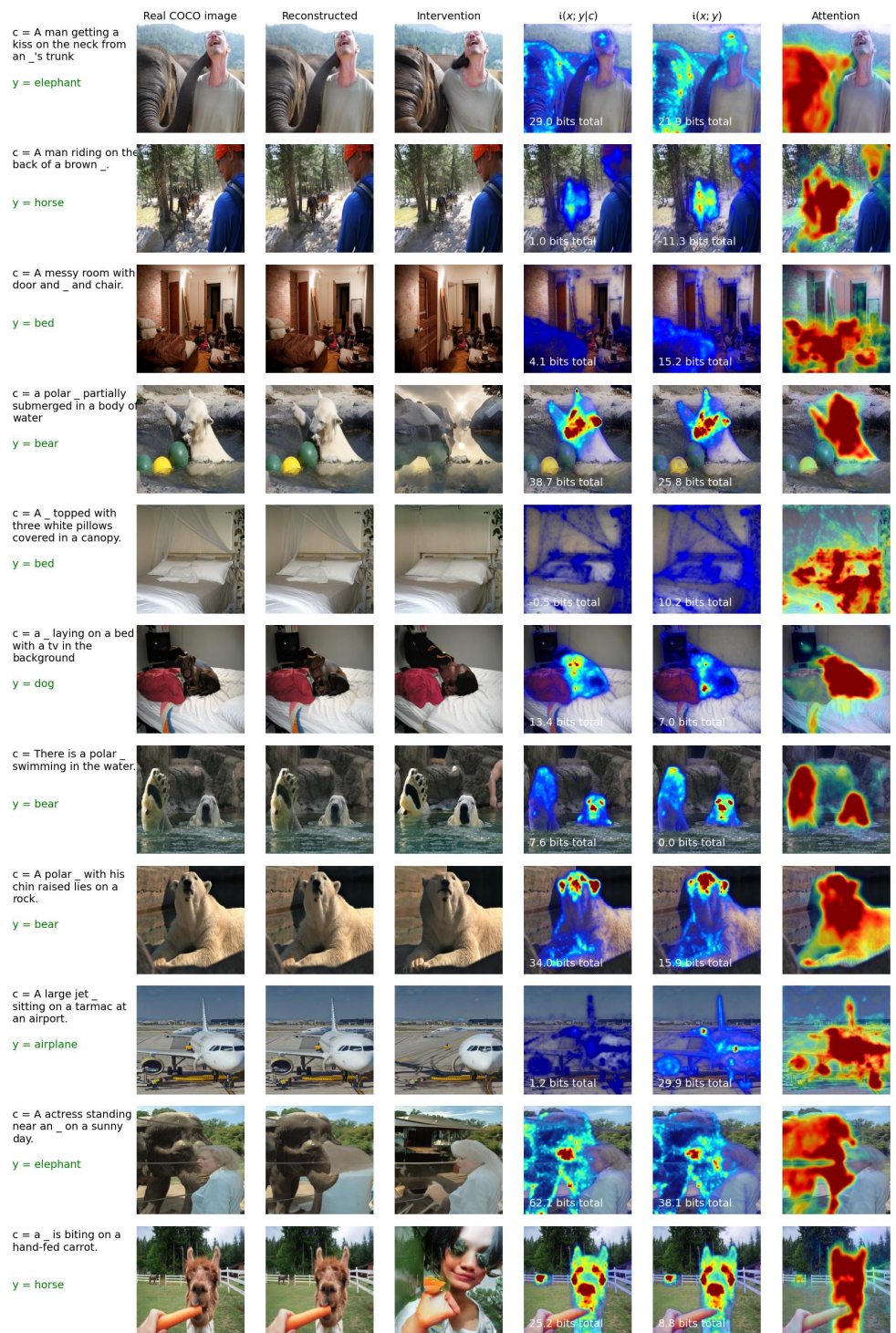


Figure B.14: Examples of word omission interventions

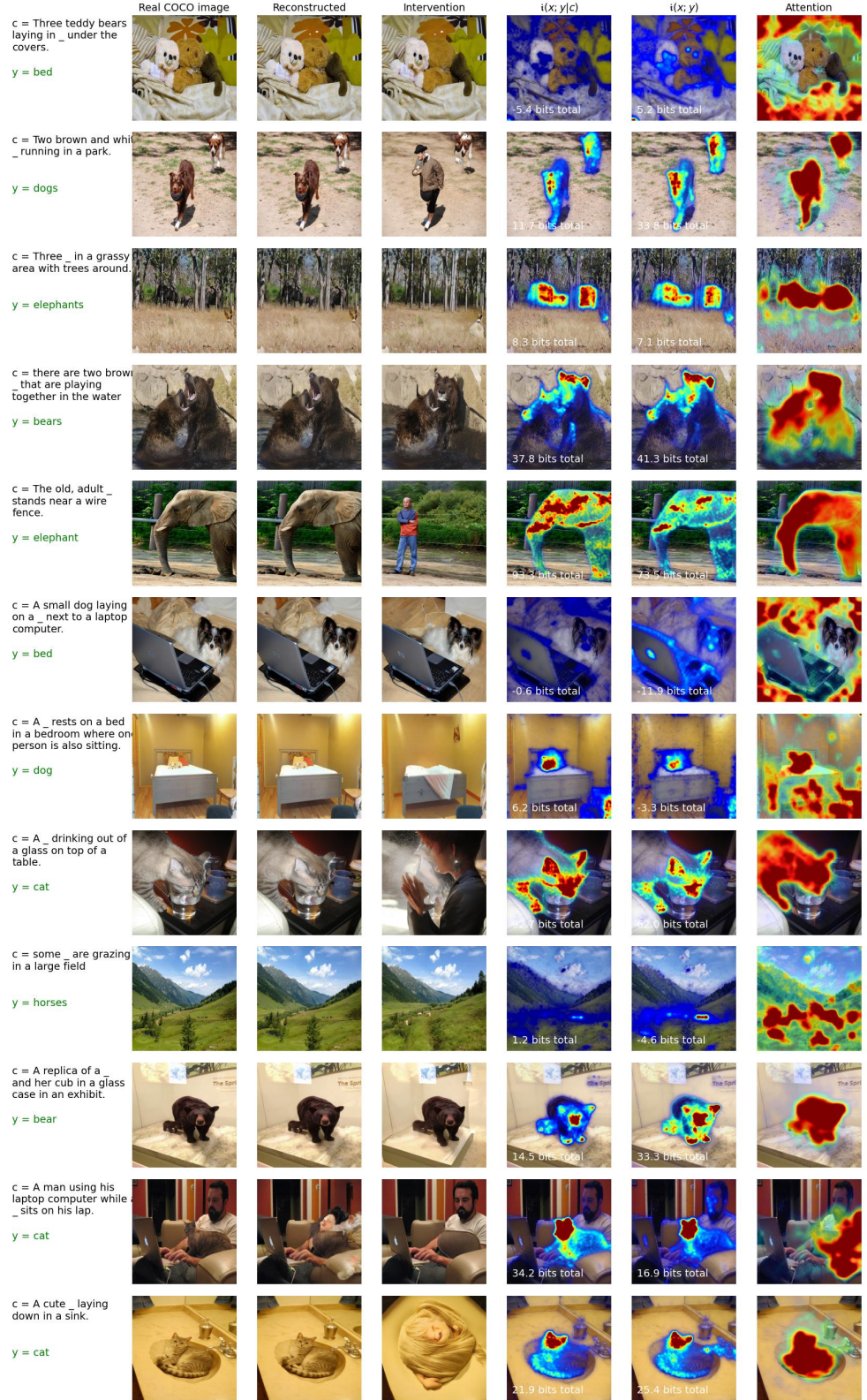


Figure B.15: Examples of word omission interventions

Table B.9: Ablation study comparing MIND-adaptive ("Adaptive") with adaptive importance sampling against the baseline MIND ("Baseline") with a fixed sampling distribution. This analysis uses the non-orthogonal variants to isolate the impact of the sampling strategy. For each task, the estimate closer to the Ground Truth and the lower standard deviation (Std) are independently bolded.

Task	Ground Truth	Adaptive		Baseline	
		Estimate	Std	Estimate	Std
<i>Basic Distribution Tasks</i>					
1v1-normal-0.75	0.4133	0.4160	0.0447	0.4208	0.0754
1v1-additive-0.1	1.7094	1.6929	0.0494	1.6984	0.0677
1v1-bimodal-0.75	0.4133	0.4133	0.0612	0.4201	0.0766
<i>High Dimensional Tasks</i>					
multinormal-dense-25-25-0.5	1.2922	1.2063	0.1636	1.1603	0.2526
multinormal-dense-50-50-0.5	1.6243	1.7926	0.3053	1.8493	0.4398
<i>Non-Gaussian Distribution Tasks</i>					
student-identity-1-1-1	0.2242	0.3644	0.2472	0.3583	0.2884
student-identity-3-3-2	0.2909	0.4901	0.6502	0.5123	0.7055
student-identity-5-5-2	0.4482	0.7747	1.0550	0.7683	1.0492
<i>Complex Transformation Tasks</i>					
spiral-multinormal-sparse-3-3-2.0	1.0217	1.0012	0.0805	1.0152	0.1187
spiral-multinormal-sparse-25-25-2.0	1.0217	0.8713	0.1557	0.8215	0.2683

Appendix C

Appendix for Chapter 5

C.1 Related Work

The landscape of diffusion models [223, 90, 228, 235] has rapidly evolved in the last few years, with modern models trained on web-scale datasets [16, 58, 205, 207]. In contrast to training in pixel space [207, 170], the majority of large-scale diffusion models are trained in a compressed latent space, thus referred to as latent diffusion models [205]. Similar to auto-regressive models, transformer architectures [255] have also been recently adopted for diffusion-based image synthesis. While earlier models commonly adopted a fully convolutional or hybrid UNet network architecture [51, 170, 205, 207], recent works have demonstrated that diffusion transformers [182] achieve better performance [58, 30, 16]. Thus, we also use diffusion transformers for modeling latent diffusion models.

Since the image captions in web-scale datasets are often noisy and of poor quality [25, 212], recent works have started to recaption them using vision-language models [149, 265, 139]. Using synthetic captions leads to significant improvements in the diffusion models’

image generation capabilities [16, 58, 74, 30]. While text-to-image generation models are the most common application of diffusion models, they can also support a wide range of other conditioning mechanisms, such as segmentation maps, sketches, or even audio descriptions [277, 270]. Sampling from diffusion models is also an active area of research, with multiple novel solvers for ODE/SDE sampling formulations to reduce the number of iterations required in sampling without degrading performance [228, 108, 150, 113]. Furthermore, the latest approaches enable single-step sampling from diffusion models using distillation-based training strategies [230, 211]. The sampling process in diffusion models also employs an additional guidance signal to improve prompt alignment, either based on an external classifier [51, 170, 213] or self-guidance [91, 114]. The latter classifier-free guidance approach is widely adopted in large-scale diffusion models and has been further extended to large-language models [280, 210].

Since the training cost of early large-scale diffusion models was noticeably high [205, 195, 163], multiple previous works focused on bringing down this cost. [74] showed that using common tricks from efficient deep learning can bring the cost of stable-diffusion-2 models under \$50K. [30] also reduced this cost by training a diffusion transformer model on a mixture of openly accessible and proprietary image datasets. Cascaded training of diffusion models is also used by some previous works [207, 184, 78], where multiple diffusion models are employed to sequentially upsample the low-resolution generations from the base diffusion model. A key limitation of cascaded training is the strong influence of the low-resolution base model on overall image fidelity and prompt alignment. Most recently, [184] adopted the cascaded training approach (Würstchen) while training the base model in a $42\times$ compressed latent

space. Though Würstchen achieves low training cost due to extreme image compression, it also achieves significantly lower image generation performance on the FID evaluation metric. Alternatively, patch masking has been recently adopted as a means to reduce the computational cost of training diffusion transformers [283, 69], taking inspiration from the success of patch masking in contrastive models [81]. Patch masking is straightforward to implement in transformers, and the diffusion transformer successfully generalizes to unmasked images in inference, even when patches for each image were masked randomly during training.

C.2 Additional Details on Experimental Setup

We use DiT-Tiny/2 and DiT-XL/2 diffusion transformer architectures [182] for small and large scale training setups, respectively. We use four and six transformer blocks in the patch-mixer for the DiT-Tiny/2 and DiT-XL/2 architectures, respectively. The patch-mixer comprises approximately 13% of the parameters in the backbone diffusion transformer. We use half-precision layernorm normalization and SwiGLU activation in the feedforward layers for all transformers. Initially, half-precision layernorm led to training instabilities after 100K steps of training. Thus, we further apply layer normalization to query and key embeddings in the attention layers [45], which stabilizes the training. We reduce the learning rate for expert layers by half, as each expert now processes a fraction of all patches. We provide an exhaustive list of our training hyperparameters in Table C.2. We use the default configuration for EDM [113] diffusion framework ($\sigma_{\max} = 80, \sigma_{\min} = 0.002, S_{\max} = \infty, S_{\text{noise}} = 1, S_{\min} = 0, S_{\text{churn}} = 0$), except that we increase σ_{data} to match the standard deviation of our image datasets in latent space. We generate images using deterministic sampling from Heun’s 2^{nd} order

method [64, 113] with 30 sampling steps. Unless specified, we use classifier-free guidance of 3 and sample 30K images in quantitative evaluation on the cifar-captions dataset. We reduce the guidance scale to 1.5 for large-scale models. We find that these guidance values achieve the best FID score under the trade-off of FID and Clip-score. In qualitative generations, we recommend using a guidance scale of 5 for better photorealism and prompt adherence. By default, we use 512×512 pixel resolution when generating synthetic images from large-scale models and 256×256 pixel resolution with small-scale models trained on the cifar-captions dataset.

We consider the computation of text embeddings for captions and latent compression of images as a one-time cost that amortizes over multiple training runs of a diffusion model. Thus, we compute them offline and do not account for these costs in our estimation of training costs. We use a four-channel variational autoencoder (VAE) from the Stable-Diffusion-XL [185] model to extract image latents. We also consider the latest 16-channel VAEs to test their performance in large-scale micro-budget training [58]. We use the EDM framework from [113] as a unified training setup for all diffusion models.

We conduct all experiments on an 8×H100 GPU node. We train and evaluate all models using `bfloat16` mixed-precision mode. We did not observe a significant speedup when using FP8 precision with the Transformer Engine library¹. We use PyTorch 2.3.0 [180] with PyTorch native flash-attention-2 implementation. We use the DeepSpeed [197] flops profiler to estimate total FLOPs in training. We also use just-in-time compilation (`torch.compile`)

Table C.1: Comparing the computational cost and storage overhead of CLIP [188, 59] and T5 [192, 240] text encoders. We use the state-of-the-art CLIP model from [59]. We report the compute and storage cost for one million image captions. Even though T5 embeddings achieve better generation quality, especially for text generation, computing them is an order of magnitude slower than CLIP embeddings and even precomputing them for our dataset (37M images) poses high storage overheads (36.4 TB). In this table, we use one H100 GPU to estimate the time to process 1M image captions. We save the embeddings in `float16` precision.

Text encoder	Sequence length	Embedding size	Time (min:sec)	Storage (GB)
CLIP (ViT-H-14)	77	1024	3:20	157
T5 (T5-xxl)	120	4096	33:04	983

to achieve a 10-15% speedup in training time. We use StreamingDataset [246] to enable fast data loading.

Table C.2: **Training hyperparameters.** Hyperparameters across both phases of our large-scale training setup.

Resolution	256 × 256 (Phase-I)		512 × 512 (Phase-II)	
Masking ratio	0.75	0	0.75	0
Training steps	250000	30000	50000	5000
Batch size	2048	2048	2048	2048
Learning rate	2.4×10^{-4}	8×10^{-5}	8×10^{-5}	8×10^{-5}
Weight decay	0.1	0.1	0.1	0.1
Optimizer	AdamW	AdamW	AdamW	AdamW
Momentum	$\beta_1 = 0.9, \beta_2 = 0.999$	$\beta_1 = 0.9, \beta_2 = 0.999$	$\beta_1 = 0.9, \beta_2 = 0.999$	$\beta_1 = 0.9, \beta_2 = 0.999$
Optimizer epsilon	1×10^{-8}	1×10^{-8}	1×10^{-8}	1×10^{-8}
Lr scheduler	Cosine	Constant	Constant	Constant
Warmup steps	2500	0	500	0
Gradient clip norm	0.25	0.25	0.5	0.5
EMA	no ema	no ema	0.99975	0.9975
Hflip	False	False	False	False
Precision	bf16	bf16	bf16	bf16
Layernorm precision	bf16	bf16	bf16	bf16
QK-normalization	True	True	True	True
$(P_{\text{mean}}, P_{\text{std}})$	(−0.6, 1.2)	(−0.6, 1.2)	(0, 0.6)	(0, 0.6)

Choice of text encoder. To convert discrete token sequences in captions to high-dimensional feature embeddings, two common choices of text encoders are CLIP [188, 102] and T5 [192],

¹<https://github.com/NVIDIA/TransformerEngine>

with T5-xxl² embeddings narrowly outperforming equivalent CLIP model embeddings [207]. However, T5-xxl embeddings pose both compute and storage challenges: 1) Computing T5-xxl embeddings costs an order of magnitude more time than a CLIP ViT-H/14 model while also requiring $6\times$ more disk space for pre-computed embeddings. Overall, using T5-xxl embeddings is more demanding (Table C.1). Following the observation that large text encoders trained on higher quality data tend to perform better [207], we use state-of-the-art CLIP models as text encoders [59]³.

Training process. We use a DiT-XL/2 transformer with eight experts in alternate transformer blocks for our large-scale training setup. We provide details on training hyperparameters in Table C.2. We conduct the training in following two phases. We refer to any diffusion transformer trained using our micro-budget training as MicroDiT.

- *Phase-1:* In this phase, we pretrain the model on 256×256 resolution images. We train for 250K optimization steps with 75% patch masking and finetune for another 30K steps without any patch masking.
- *Phase-2:* In this phase, we finetune the Phase-1 model on 512×512 resolution images. We first finetune the model for 50K steps with 75% patch masking followed by another 5K optimization steps with no patch masking.

Cost analysis. We translate the wall-clock time of training to financial cost using a \$30/hour budget for an $8\times$ H100 GPU cluster. Since the cost of H100 fluctuates significantly across vendors, we base our estimate on the commonly used cost estimates for A100 GPUs [30], in particular \$1.5/A100/hour⁴. We observe a 2.5x reduction in wall-clock time on H100 GPUs,

²https://huggingface.co/DeepFloyd/t5-v1_1-xxl

³We use the DFN-5B text encoder: <https://huggingface.co/apple/DFN5B-CLIP-ViT-H-14-378>

⁴<https://cloud-gpus.com/>

thus assume a \$3.75/H100/hour cost. We also report the wall-clock time to benchmark the training cost independent of the fluctuating GPU costs in the AI economy.

Datasets. We use the following five datasets, comprising a total of 37 million images, to train our large-scale models.

- *Conceptual Captions (real)*. This dataset was released by Google and includes 12M pairs of image URLs and captions [27]. Our downloaded version of the dataset includes 10.9M image-caption pairs.
 - *Segment Anything-1B (real)*. Segment Anything comprises 11.1M high-resolution images originally released by Meta for segmentation tasks [123]. Since the images do not have corresponding real captions, we use synthetic captions generated by the LLaVA model [30, 149].
 - *TextCaps (real)*. This dataset comprises 28K images with natural text in the images [220]. Each image has five associated descriptions. We combine them into single captions using the Phi-3 language model [1].
 - *JourneyDB (synthetic)*. JourneyDB is a synthetic image dataset comprising 4.4M high-resolution Midjourney image-caption pairs [238]. We use the train subset (4.2M samples) of this dataset.
 - *DiffusionDB (synthetic)*. DiffusionDB is a collection of 14M synthetic images generated primarily by Stable Diffusion models [266]. We filter out poor quality images from this dataset, resulting in 10.7M samples, and use this dataset only in the first phase of training.
- expand these details further.

CIFAR-Captions. We construct a text-to-image dataset, named cifar-captions, that closely resembles the existing web-curated datasets and serves as a drop-in replacement of existing datasets in our setup. Cifar-captions is closed-domain and only includes images of ten classes (airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks), imitating the widely used CIFAR-10 classification dataset [130]. In contrast to other small-scale text-to-image datasets that are open-world, such as subsets of conceptual captions [27], we observe fast convergence of diffusion models on this dataset. We create this dataset by first downloading 120M images from the coyo-700M [25] dataset. We observe a success rate of approximately 60% at the time of downloading the dataset. Next, we use a ViT-H-14 [59] to measure CLIP score by averaging it over the eighteen prompt templates (such as ‘a photo of a {}.’) used in the original CLIP model [188]. We select images with CLIP scores higher than 0.25 (1.25% acceptance rate) that results in a total of 1.3M images. As the real captions are highly noisy and uninformative, we replace them with synthetic captions generated with an LLaVA-1.5 model [149].

Billion-image synthetic dataset. To capture the asymptotic performance of training solely on synthetic images, we also consider a dataset comprising 1B images generated from diffusion models⁵. It mainly includes generations from various versions of stable diffusion models and their adaptations.

Evaluation metrics. We use the following evaluation metrics to assess the quality of synthetic images generated by the text-to-image models.

⁵<https://huggingface.co/datasets/bigdata-pw/Diffusion1B>

- *FID*. Fréchet Inception Distance (FID) measures the 2-Wasserstein distance between real and synthetic data distributions in the feature space of a pretrained vision model. We use the clean-fid⁶ [179] implementation for a robust evaluation of the FID score.
- *Clip-FID*. Unlike FID that uses an Inception-v3 [242] model trained on ImageNet [46], Clip-FID uses a CLIP [188] model, trained on a much larger dataset than ImageNet, as an image feature extractor. We use the default ViT-B/32 CLIP model from the clean-fid library to measure Clip-FID.
- *Clip-score*. It measures the similarity between a caption and the generated image corresponding to the caption. In particular, it measures the cosine similarity between normalized caption and image embeddings calculated using a CLIP text and image encoder, respectively.

Human preference based evaluation. We ask each individual to choose one of the two given images based on the following criteria: 1) choose based on image quality in the absence of the prompt, and 2) choose based on the alignment between the prompt and the generated images. We use the identical prompt to sample from both models. We report the win rate based on the percentage of comparisons in which the image from the winning model is preferred over the other model. We further follow the evaluation pipeline of [16] and use DrawBench extended prompts to generate samples. We slightly compress these prompts to accommodate the 77-token context window of CLIP text encoders. We list these prompts at the end of the paper.

⁶<https://github.com/GaParmar/clean-fid>

hidden dimension of feedforward layers by a factor m_f , thus linearly increasing the number of parameters and FLOPs of the feedforward block. Similarly, layer-wise scaling dynamically increases the number of heads in the attention layers by scaling the embedding size by a factor of m_a . By default, $m_a = 1.0$ and $m_f = 4.0$ lead to a canonical transformer network. We use $m_a \in [0.5, 1.0]$ and $m_f \in [0.5, 4.0]$ to reduce the size of transformer blocks. Note that we do not apply any scaling to cross-attention layers ($m_a = 1.0$, $m_f = 4.0$) to avoid degrading the controllability of captions on image generation.

Mixture-of-experts. Mixture-of-experts enable the construction of much larger transformers, referred to as sparse transformers, with minimal impact on the training and inference cost, thus making them highly applicable for micro-budget training. A sparse model modifies the transformer block to include replicas of the feedforward layer, referred to as experts. The input patch embeddings to the feedforward layers are first fed into a router that determines the configuration of patches processed by each expert. We use the expert-choice (EC) routing mechanism [284] where each expert selects the top-k patches using the importance score determined by the routing network for each expert (Figure C.2). We favor EC routing over conventional routing due to its simplicity, as it does not require any auxiliary loss to balance the load across experts [219, 286]. We use a linear layer as a router and it is trained jointly with the rest of the network.

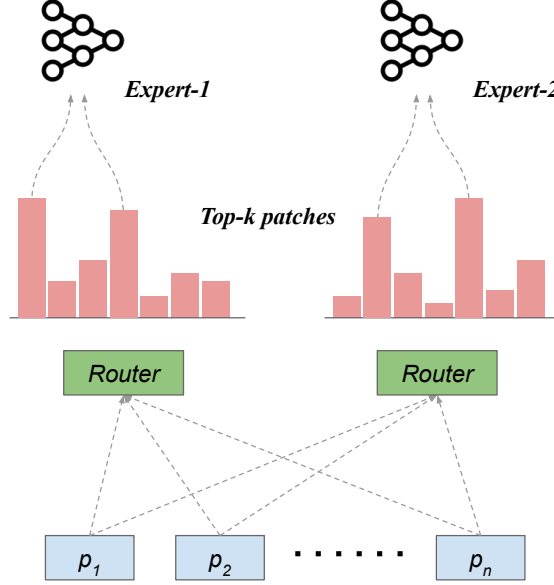


Figure C.2: **Mixture-of-experts (MoE)**. Expert-choice routing based mixture-of-experts [284]. Each patch is passed to a patch router that determines the top-k patches routed to each expert. Make this figure nicer.

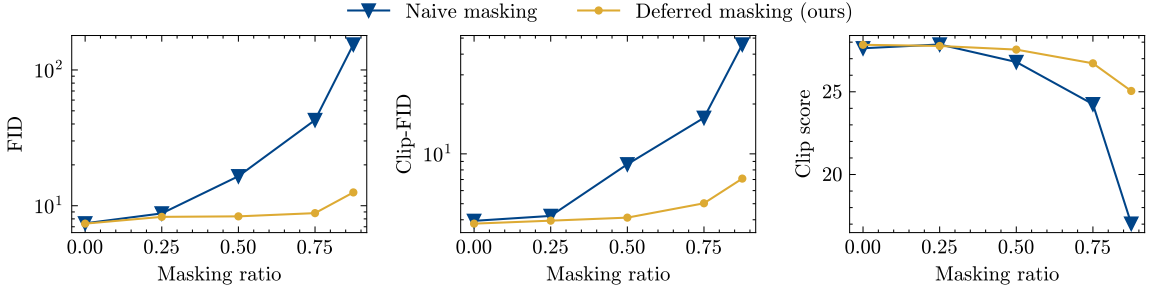


Figure C.3: **Out-of-the-box performance of deferred masking**. Without any hyperparameter optimization, we compare the performance of our deferred masking with a naive masking strategy. We find that deferred masking, i.e., using a patch-mixer before naive masking, tremendously improves image generation performance, particularly at high masking ratios.

C.4 Additional Results on Effectiveness of Deferred Masking

In this section, we supplement the results and discussion from the main paper on evaluating the effectiveness of deferred masking. We use the DiT-Tiny/2 model and the cifar-captions dataset (256×256 image resolution) for all experiments in this section. We

train each model for 60K optimization steps using the AdamW optimizer and exponential moving average with a 0.995 smoothing coefficient enabled for the last 10K steps.

C.4.1 Out-of-the-box performance: Making high masking ratios feasible with deferred masking

In this section, we provide further details on the out-of-the-box performance of deferred masking. We evaluate its out-of-the-box performance with common training parameters for up to 87.5% masking ratios. As a baseline, we train a network with no patch-mixer, i.e., naive masking (Figure 4.2c) for each masking ratio.

We use commonly used default hyperparameters to simulate the out-of-the-box performance for both our approach and the baseline. We train both models with the AdamW optimizer with an identical learning rate of 1.6×10^{-4} , 0.01 weight decay, and a cosine learning rate schedule. We set $(P_{\text{mean}}, P_{\text{std}})$ to $(-1.2, 1.2)$ following the original work [113]. We provide the results in Figure C.3.

C.4.2 Ablation study of our training pipeline

In this section, we further discuss the improvements over out-of-the-box performance (Figure C.3). When ablating the learning rate, we observe better performance with higher learning rates. However, we run into training instabilities after approximately 2K steps, despite using qk-normalization [45], with mixed precision training. Overall, we recommend ablating the choice of learning rate for each network and using the maximum attainable value without causing training instability. We also consider a fast training strategy using cyclic learning schedules [221]. Cyclic learning has been previously observed to improve the

convergence rate for image classifiers [221, 222]. We consider a cyclic cosine schedule with base cycle time t and each subsequent cycle duration multiplied by a multiplier (m_t). We only observe a marginal benefit of cyclic schedules (Table C.3(g)). In favor of simplicity, we continue to use a single cycle schedule.

Comparison with training hyperparameters of LLMs. Since the diffusion transformer architectures are very similar to transformers used in large language models (LLMs), we compare the hyperparameter choices across the two tasks. Similar to common LLM training setups [249, 106, 40], we find that SwiGLU activation [218] in the feedforward layer outperforms the GELU [83] activation function. Similarly, higher weight decay leads to better image generation performance. However, we observe better performance when using a higher running average coefficient for the AdamW second moment (β_2), in contrast to large-scale LLMs where $\beta_2 \approx 0.95$ is preferred. As we use a small number of training steps, we find that increasing the learning rate to the maximum possible value until training instabilities also significantly improves image generation performance.

Design choices in masking and patch-mixer. We observe a consistent improvement in performance with a larger patch mixer. However, despite the better performance of larger patch-mixers, we choose to use a small patch mixer to lower the computational budget spent by the patch mixer in processing the unmasked input. We also update the noise distribution ($P_{\text{mean}}, P_{\text{std}}$) to $(-0.6, 1.2)$ as it improves the alignment between captions and generated images. In ablating the choice of masking, we shift from masking patches randomly to retaining a continuous region of image patches with block masking 4.3b. We perform this ablation on 256×256 image resolution with a corresponding latent resolution of 32×32

Table C.3: **Ablating individual components in our training pipeline.** In each subsequent ablation, we use the overall best performing model from previous ablation. In each ablation, we train a DiT-Tiny/2 model for 60K training steps using 75% masking ratio.

(a) *Ablating the choice of β_2 in AdamW optimizer.* Unlike the LLM training where β_2 is often set to 0.95 [249, 24], we find that image generation quality consistently degrades as we reduce β_2 .

β_2	FID ($\downarrow$)	Clip-FID ($\downarrow$)	Clip-score ($\uparrow$)
0.999	8.53	4.85	26.88
0.99	8.63	4.94	26.75
0.95	8.71	5.02	26.71
0.9	8.81	5.13	26.61

(c) *Ablating the parameters of noise distribution.* We used patch-mixer with 0.999 β_2 and 0.1 weight decay. We observe a tradeoff between FID and Clip-score in first two choices and set (m, s) to $(-0.6, 1.2)$ in all followup experiments.

(m, s)	FID ($\downarrow$)	Clip-FID ($\downarrow$)	Clip-score ($\uparrow$)
$(-1.2, 1.2)$	8.38	4.90	27.00
$(-0.6, 1.2)$	8.49	4.93	27.47
$(-0.6, 0.6)$	9.05	6.72	26.95
$(-0.25, 0.6)$	10.44	7.51	27.46
$(0.0, 0.6)$	12.76	9.00	27.40

(e) *Ablating the size of patch-mixer.* We increase the width of path embeddings, while also varying the multipliers for attention layers (m_a) and feed-forward layers (m_f). We also report the total wall-clock time (in hours:minutes) for training.

(w, m_a, m_b)	Time	FID ($\downarrow$)	Clip-FID ($\downarrow$)	Clip-score ($\uparrow$)
(384, 0.5, 0.5)	3:19	8.49	4.93	27.47
(384, 1.0, 4.0)	3:20	7.72	4.80	27.72
(512, 1.0, 4.0)	3:24	7.40	4.57	27.79
(768, 1.0, 4.0)	3:39	7.09	4.39	27.76

(g) *Testing the cyclic learning rate scheduler.* We consider the cosine learning rate with warm restarts, while varying the duration of the base cycle (t) and the multiplier (m_t), where the duration of each subsequent cycle increases by the multiplier factor.

lr schedule	FID ($\downarrow$)	Clip-FID ($\downarrow$)	Clip-score ($\uparrow$)
1-cycle ($t = 60,000, m_t = 1$)	7.40	4.57	27.79
3-cycles ($t = 20,000, m_t = 1$)	7.64	4.52	27.75
4-cycles ($t = 4,000, m_t = 2$)	7.49	4.43	27.81

(b) *Ablating the choice of weight-decay in AdamW optimizer.* In resonance with transformer training in LLMs, we observe improvement in performance with increase in weight decay regularization.

wd	FID ($\downarrow$)	Clip-FID ($\downarrow$)	Clip-score ($\uparrow$)
0.00	8.77	5.03	26.82
0.01	8.73	5.03	26.82
0.03	8.53	4.85	26.88
0.10	8.38	4.90	27.00

(d) *Ablating the block-size in block sampling.* We observe consistent performance degradation with block masking (at 75% masking ratio). At block size of 8 (latent image with $16 \times 16 = 256$ patches), block sampling sampling collapse to sampling a quadrant, thus we sample a single continuous square patch for it.

Block-size	FID ($\downarrow$)	Clip-FID ($\downarrow$)	Clip-score ($\uparrow$)
1	8.49	4.93	27.47
2	8.89	4.64	27.42
4	9.80	5.09	26.91
square	12.71	7.11	26.14

(f) *Ablating the choice of feedforward layers: GELU vs SwiGLU activation.* We find that replacing GELU activation with a SwiGLU based activation in feedforward layer improves all three performance metrics. SwiGLU activation is also commonly used in transformers for large language models [249].

Block-size	FID ($\downarrow$)	Clip-FID ($\downarrow$)	Clip-score ($\uparrow$)
GELU	7.82	4.61	27.33
SwiGLU	7.40	4.57	27.79

(h) *Using higher learning rate for each batch for better performance.* We observe training instabilities after 2K steps for learning rates higher than 3.2×10^{-4} in mixed-precision training.

lr	FID ($\downarrow$)	Clip-FID ($\downarrow$)	Clip-score ($\uparrow$)
1.6×10^{-4}	7.40	4.57	27.79
3.2×10^{-4}	7.09	4.10	28.24

and 256 patches with a patch size of two in the DiT-Tiny/2 model. Note that at a block size of 8 and a 75% masking ratio, block sampling collapses to masking quadrants of image patches. Thus, for this configuration, we resort to sampling a single continuous square patch of non-masked patches. Overall, we find that any amount of block masking degrades performance compared to random masking of each patch. It is likely because despite latent compression and patching, there exists some redundancy in visual information between neighboring patches, and the diffusion transformer model receives more global semantic information about the image with random masking.

C.4.3 Validating improvements in diffusion transformer architecture

Layer-wise scaling. We investigate the impact of this design choice in a standalone experiment where we train two variants of a DiT-Tiny architecture, one with a constant width transformer and the other with layer-wise scaling of each transformer block. We use naive masking for both methods. We select the width of the constant-width transformer such that its computational footprint is identical to layer-wise scaling. We train both models for an identical number of training steps and wall-clock time. We find that the layer-wise scaling approach outperforms the baseline constant width approach across all three performance

Table C.4: **Using layer-wise scaling.** Layer-wise scaling of transformer architecture is a better fit for masked training in diffusion transformers. We validate its effectiveness in the canonical naive masking with 75% masking ratio.

Arch	FID (↓)	Clip-FID (↓)	Clip-score (↑)
Constant width	19.6	9.9	26.7
Layer-wise scaling	15.9	7.4	27.1

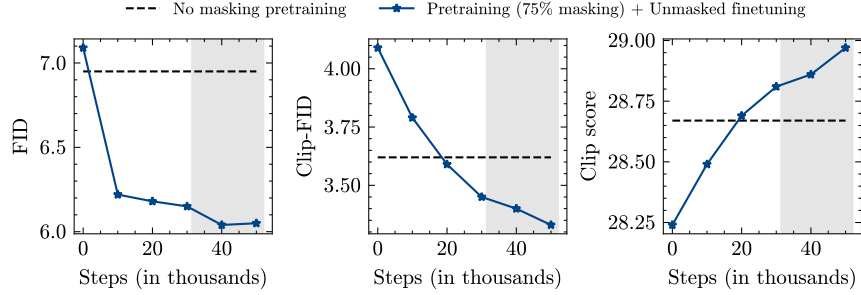


Figure C.4: **Deferred masking in pretraining.** We test the advantage of using deferred masking with finetuning over training a model with no masking for a given computational budget. After pretraining with 75% deferred masking, we increase the number of unmasked finetuning steps and compare its performance with another model trained completely without masking. The shaded region represents steps after the isoflops threshold, where deferred masking pretraining and finetuning have higher computation cost. At the isoflops threshold, the finetuned model achieves better performance than training the diffusion model without any masking.

metrics (Table C.4), demonstrating a better fit of the layer-wise scaling approach in masked training of diffusion transformers.

Mixture-of-experts layers. We train a DiT-Tiny/2 transformer with expert-choice routing based mixture-of-experts (MoE) layers in each alternate transformer block. On the small-scale setup, it achieves similar performance to baseline model trained without MoE blocks. While it slightly improves Clip-score from 28.11 to 28.66, it degrades FID score from 6.92 to 6.98. We hypothesize that the lack of improvement is due to the small number of training steps (60K), as using multiple experts effectively reduces the number of samples observed by each router. In top-2 routing for 8-experts, each expert is trained effectively for one fourth number of epochs over the dataset.

C.4.4 Deferred masking as pretraining + unmasked finetuning.

We find that deferred masking also acts as a strong pretraining objective, and using it with an unmasked finetuning schedule achieves better performance than training an unmasked network under an identical computational budget. We first train a network without any masking and another identical network with 75% deferred masking. We finetune the latter with no masking and measure performance as we increase the number of finetuning steps. We mark the *isoflops* threshold when the combined cost of masked pretraining and unmasked finetuning is identical to the model trained with no masking. We find that at the isoflops threshold, the finetuned model achieves superior performance across all three performance metrics. The performance of the model also continues to improve with unmasked finetuning steps beyond the isoflops threshold.

Table C.5: **IsoFLOPs training.** Both patch masking and model downscaling, i.e., reducing the size of the model, reduce the computational cost in training and can be complementary to each other. However, it is natural to ask how the two paradigms compare to each other in training diffusion transformers. Under identical training setup and wall-clock time, we compare the effectiveness of model downscaling and our deferred masking approaches. We find that except at extremely high masking ratios, deferred masking achieves better performance across at least two performance metrics. Based on this finding, we do not use a masking ratio higher than 75% in our models.

Masking ratio	Masking ratio	FID	Clip-FID	Clip-score
Downscaled network	-	6.60	3.85	28.49
Deferred masking	0.5	6.74	3.45	28.86
Downscaled network	-	7.09	4.56	27.68
Deferred masking	0.75	6.96	4.17	28.16
Downscaled network	-	7.52	5.03	26.98
Deferred masking	0.875	8.35	5.26	27.00

C.5 Additional Results on Micro-budget Training of Large-scale Models

Training MicroDiT in higher dimensional latent space. We replace the default four-channel autoencoder with one that has sixteen channels, resulting in a $4\times$ higher dimensional latent space. Recent large-scale models have adopted high dimensional latent space as it provides significant improvements in image generation abilities [44, 58]. Note that the autoencoder with higher channels itself has superior image reconstruction capabilities, which further contributes to overall success. Intriguingly, we find that using a higher dimensional latent space in micro-budget training hurts performance. For two MicroDiT models trained with identical computational budgets and training hyperparameters, we find that the model trained in four-channel latent space achieves better FID, Clip-score, and GenEval scores (Table C.7). We hypothesize that even though an increase in latent dimensionality allows better modeling of data distribution, it also simultaneously increases the training budget required to train higher-quality models.

Table C.6: **Breakdown of computational cost.** Computational cost of the two-stage training of our large-scale model. Our total computational cost is 3.45×10^{20} FLOPs, amounting to a total cost of \$1,890 and 2.6 training days on an $8\times$ H100 GPU machine. Report CLIP score at CFG 5. For the 4-ch encoder, put a table in the appendix. Move FID values to a separate table in the appendix, as they are confusing here. Instead, add $8\times$ A100 and $8\times$ H100 days.

Resolution	Masking ratio	Training steps	Total FLOPs	$8\times$ A100 days	$8\times$ H100 days	Cost (\$)
256×256	0.75	250000	1.47×10^{20}	2.77	1.11	800
	0.00	30000	4.53×10^{19}	0.94	0.38	271
512×512	0.75	50000	1.18×10^{20}	2.18	0.88	630
	0.00	5000	3.48×10^{19}	0.65	0.26	189

Table C.7: **Why prefer 4-channel image encoders over 16-channel image encoders in micro-budget training?** We ablate the dimension of latent space by training our MicroDiT models in four and sixteen channel latent spaces, respectively. Even though training in higher dimensional latent space is being adopted across large-scale models [58, 44], we find that it underperforms when training on a micro-budget.

(a) Measuring fidelity and prompt alignment on COCO dataset. (b) Performance on the GenEval benchmark.

Channels	FID-30K ($\downarrow$)	Clip-FID-30K ($\downarrow$)	Clip-score ($\uparrow$)	Attribution							
4	12.65	5.96	28.14	Single		Two					
16	13.04	6.84	25.63	4	0.46	0.97	0.47	0.33	0.78	0.09	0.20
				16	0.40	0.96	0.36	0.27	0.72	0.07	0.09

Table C.8: **Zero-shot FID on COCO2014 validation split.** We report total training time in terms of number of days required to train the model on a machine with eight A100 GPUs. We observe a $2.5\times$ reduction in training time when using H100 GPUs. Our micro-budget training takes $14.2\times$ less training time than state-of-the-art low-cost training approach while simultaneously achieving competitive FID compared to some open-source models.

Model	Params ($\downarrow$)	Sampling steps ($\downarrow$)	Open-source	Training images($\downarrow$)	8×A100 GPU days ($\downarrow$)	FID-30K ($\downarrow$)
CogView2 [52]	6.00B	—	—	—	—	24.0
Dall-E [196]	12.0B	256	—	—	—	17.89
Glide [170]	3.50B	250	—	—	—	12.24
Parti-750M [273]	0.75B	1024	—	3690M	—	10.71
Dall-E.2 [195]	6.50B	—	—	650M	5208.3	10.39
Make-a-Scene [68]	4.00B	1024	—	—	—	11.84
GigaGAN [109]	1.01B	1	—	980M	597.8	9.09
ImageN [207]	3.00B	—	—	860M	891.5	7.27
Parti-20B [273]	20.0B	1024	—	3690M	—	7.23
eDiff-I [10]	9.10B	25	—	11470M	—	6.95
Stable-Diffusion-2.1 ^a [205]	0.86B	50	✓	3900M	1041.6	9.12
Stable-Diffusion-1.5 [205]	0.86B	50	✓	4800M	781.2	11.18
Würstchen [184]	0.99B	60	✓	1420M	128.1	23.60
PixArt- α [30]	0.61B	20	✓	25M ^b	94.1 ^c	7.32
MicroDiT (our work)	1.16B	30	✓	37M	6.6	12.66

C.5.1 Challenge with canonical evaluation metrics in determining effect of synthetic data.

To determine the effect of synthetic data, we train two models under identical training setup and cost: 1) model trained only on real images (total 22M images) 2) model trained on the combined real and synthetic images (total 37M images).

Table C.9: **Comparing performance on compositional image generation using GenEval [73] benchmark.** Table adopted from [58]. Higher performance is better. Must provide some visualization on where our methods performs well.

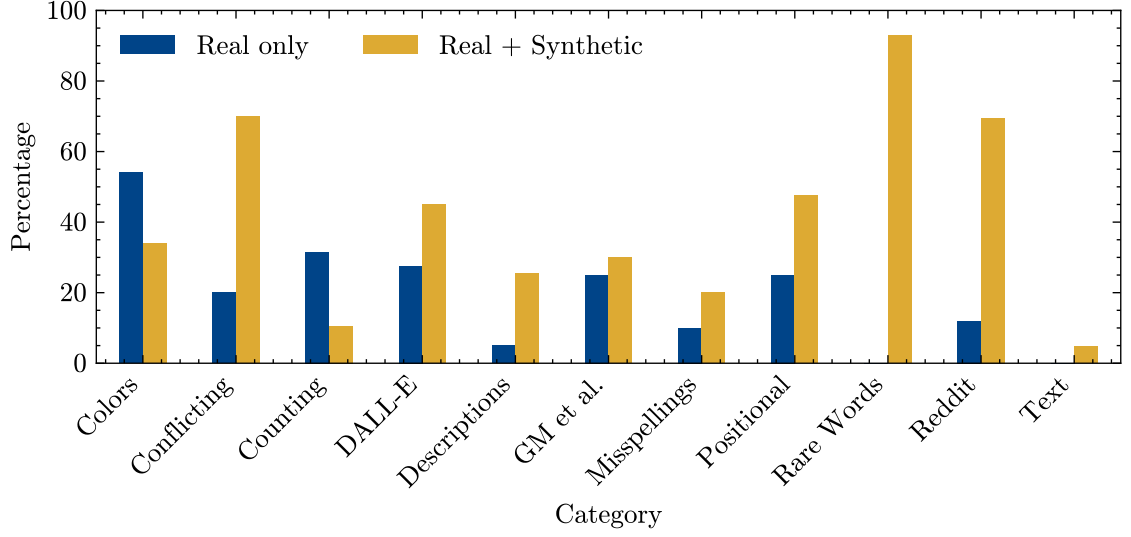
Model	Open-source	Overall	Objects			Counting	Colors	Position	Color attribution
			Single	Two					
DaLL-E.2 [195]	—	0.52	0.94	0.66	0.49	0.77	0.10	0.19	
DaLL-E.3 [16]	—	0.67	0.96	0.87	0.47	0.83	0.43	0.45	
minDALL-E [118]	✓	0.23	0.73	0.11	0.12	0.37	0.02	0.01	
Stable-Diffusion-1.5 [205]	✓	0.43	0.97	0.38	0.35	0.76	0.04	0.06	
PixArt- α [30]	✓	0.48	0.98	0.50	0.44	0.80	0.08	0.07	
Stable-Diffusion-2.1 [205]	✓	0.50	0.98	0.51	0.44	0.85	0.07	0.17	
Stable-Diffusion-XL [185]	✓	0.55	0.98	0.74	0.39	0.85	0.15	0.23	
Stable-Diffusion-XL-Turbo [211]	✓	0.55	1.00	0.72	0.49	0.80	0.10	0.18	
IF-XL	✓	0.61	0.97	0.74	0.66	0.81	0.13	0.35	
Stable-Diffusion-3 [58]	✓	0.68	0.98	0.84	0.66	0.74	0.40	0.43	
MicroDiT (our work)	✓	0.46	0.97	0.47	0.33	0.78	0.09	0.20	

Table C.10: **Scaling synthetic dataset to half billion images.** Comparing performance on compositional image generation using GenEval [73] benchmark for MicroDiT models trained with different configuration of training dataset.

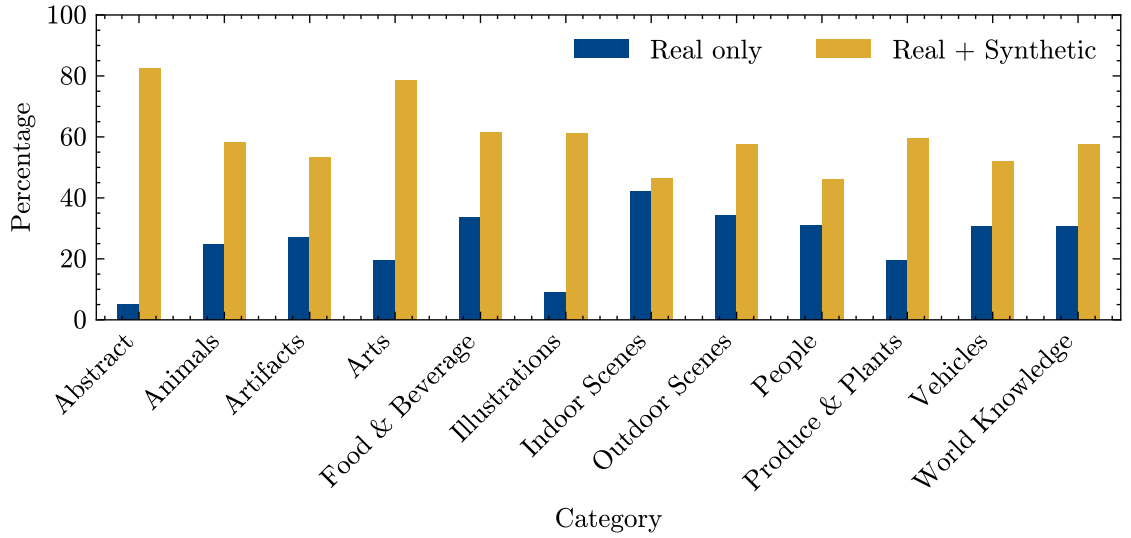
Model (training dataset)	Overall	Objects			Counting	Colors	Position	Color attribution
		Single	Two					
MicroDiT (37M - combined real and synthetic images)	0.46	0.97	0.47	0.33	0.78	0.09	0.20	
MicroDiT (490M - only synthetic images)	0.52	0.98	0.61	0.31	0.89	0.11	0.17	

Under canonical performance metrics, both models apparently achieve similar performance. For example, the model trained on real-only data achieved an FID score of 12.72 and a CLIP score of 26.67, while the model trained on both real and synthetic data achieved an FID score of 12.66 and a CLIP score of 28.14. Even on GenEval [73], a benchmark that evaluates the ability to generate multiple objects and modelling object dynamics in images, both models achieved an identical score of 0.46. These results seemingly suggest that incorporating a large amount of synthetic samples didn’t yield any meaningful improvement in image generation capabilities.

However, we argue that this observation is heavily influenced by the limitations of our existing evaluation metrics. In a qualitative evaluation, we found that the model



(a) *DrawBench*



(b) *PartiPrompts*

Figure C.6: **Assessing image quality with GPT-4o.** We supply the following prompt to the GPT-4o model: *Given the prompt ‘{prompt}’, which image do you prefer, Image A or Image B, considering factors like image details, quality, realism, and aesthetics? Respond with ‘A’ or ‘B’ or ‘none’ if neither is preferred.* For each comparison, we also shuffle the order of images to remove any ordering bias. We evaluate on two prompt databases: DrawBench [207] and PartiPrompts [273]. The y-axis in the bar plots indicates the percentage of comparisons in which an image from a model is preferred; we also break down results by category.

trained on the combined dataset achieved much better image quality (Figure C.7). The real data model often fails to adhere to the prompt, frequently hallucinating key details and often failing to synthesize the correct object. Metrics, such as FID, fail to capture this difference because they predominantly measure distribution similarity [184]. Thus, we focus on using human visual preference as an evaluation metric. To automate the process, we use GPT-4o [176], a state-of-the-art multimodal model, as a proxy for human preference. We supply the following prompt to the model: *Given the prompt ‘{prompt}’, which image do you prefer, Image A or Image B, considering factors like image details, quality, realism, and aesthetics? Respond with ‘A’ or ‘B’ or ‘none’ if neither is preferred.* For each comparison, we also shuffle the order of images to remove any ordering bias. We generate samples using DrawBench [207] and PartiPrompts (P2) [273], two commonly used prompt databases (Figure C.6). On the P2 dataset, images from the combined data model are preferred in 63% of comparisons while images from the real data model are only preferred 21% of the time (16% of comparisons resulted in no preference). For the DrawBench dataset, the combined data model is preferred in 40% of comparisons while the real data model is only preferred in 21% of comparisons. Overall, using a human preference-centric metric clearly demonstrates the benefit of additional synthetic data in improving overall image quality.

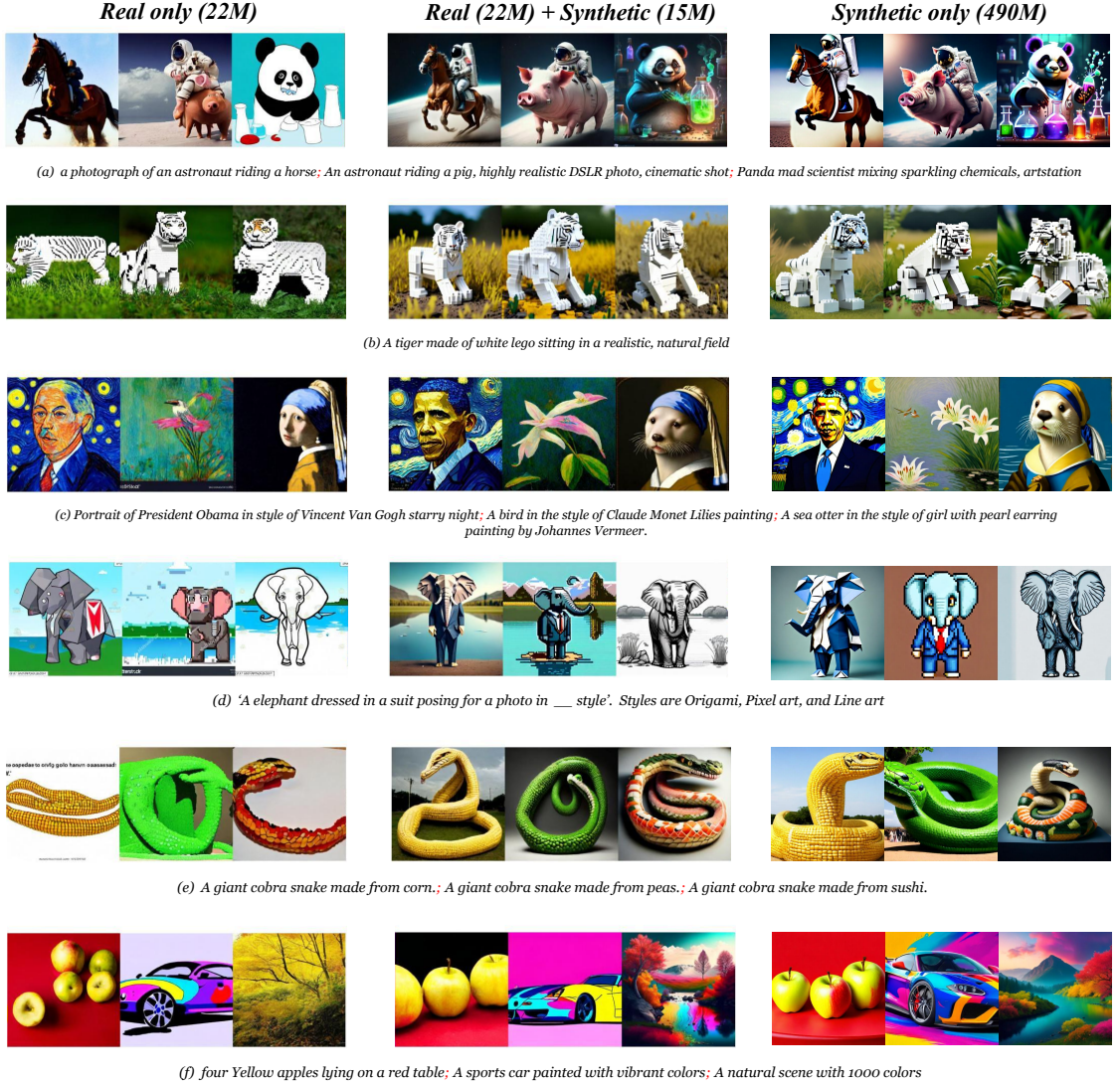


Figure C.7: **Comparing generations from our three micro-budget models.** We compare synthesized images from our large-scale models trained with different configurations of the training dataset but with identical computational costs.

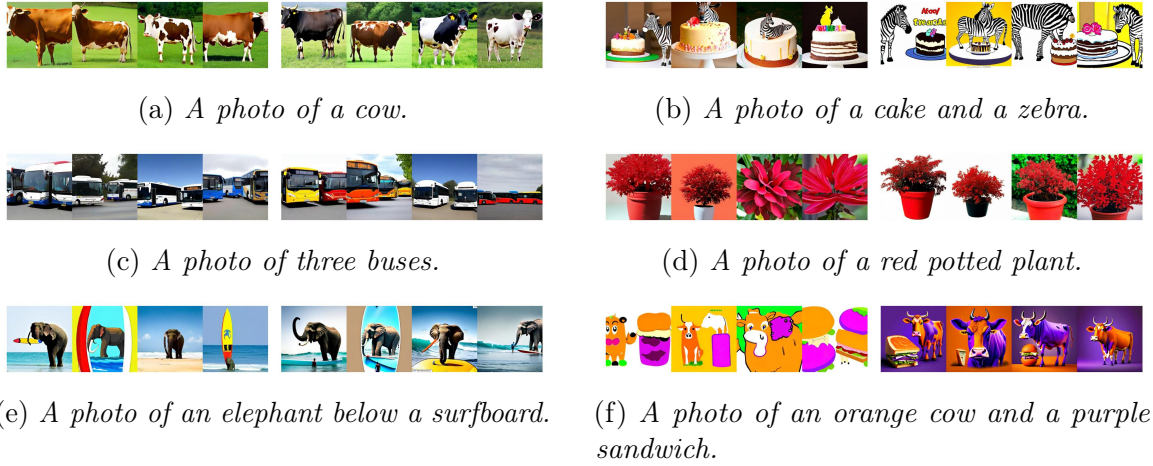
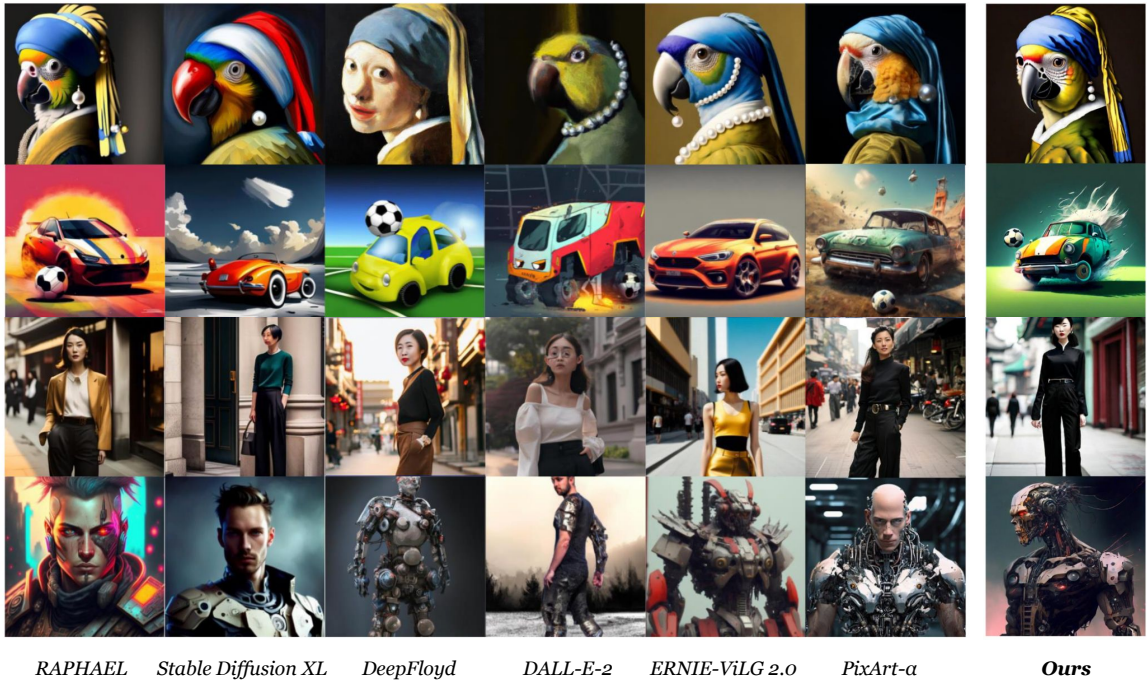
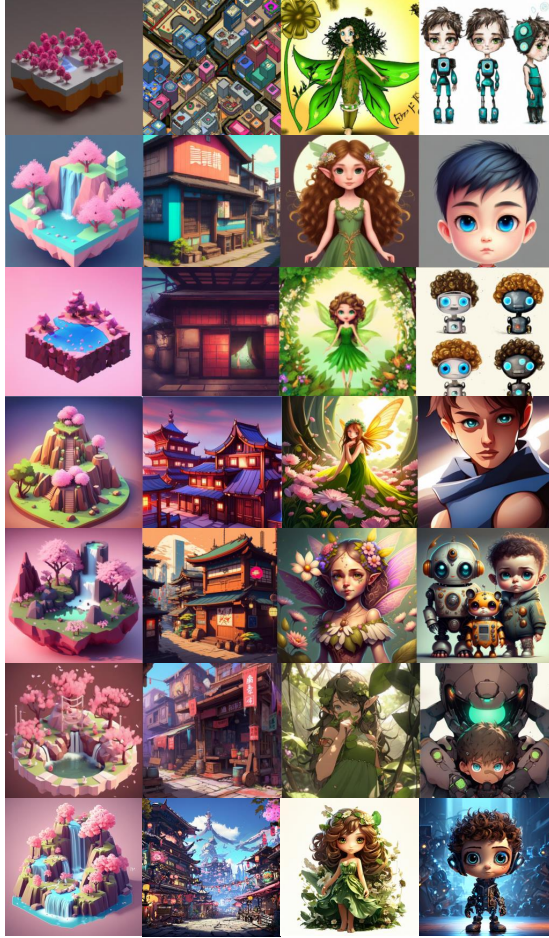


Figure C.8: **Comparing generations from our first two micro-budget models.** Generations from a model trained only on real data (left) and combined real + synthetic data (right) on GenEval benchmark prompts. Both models use identical random seeds for generation.

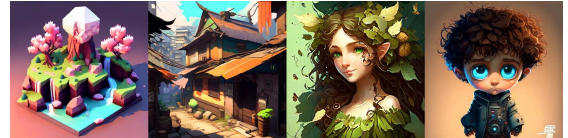


- A parrot with a pearl earring, Vermeer style.
- A car playing soccer, digital art.
- Street shot of a fashionable Chinese lady in Shanghai, wearing black high-waisted trousers.
- Half human, half robot, repaired human, human flesh warrior, mech display, man in mech, cyberpunk.

Figure C.9: Comparison with previous works. Figure adapted from [30].



(a) *Previous works: RAPHAEL, Stable Diffusion XL, DeepFloyd, DALL-E 2, ERNIE-ViLG 2.0, PixArt- α .*



(b) *Our work.*

Prompts used:

- A cute little matte low poly isometric cherry blossom forest island, waterfalls, lighting, soft shadows, trending on Artstation, 3D render, Monument Valley, Fez video game.
- A shanty version of Tokyo, new rustic style, bold colors with all colors palette, video game, Genshin, tribe, fantasy, Overwatch.
- Cartoon characters, mini characters, figures, illustrations, flower fairy, green dress, brown hair, curly long hair, elf-like wings, many flowers and leaves, natural scenery, golden eyes, detailed light and shadow, a high degree of detail.
- Cartoon characters, mini characters, hand-made, illustrations, robot kids, color expressions, boy, short brown hair, curly hair, blue eyes, technological age, cyberpunk, big eyes, cute, mini, detailed light and shadow, high detail.

Figure C.10: **Comparison with previous works.** Figure adapted from [30].

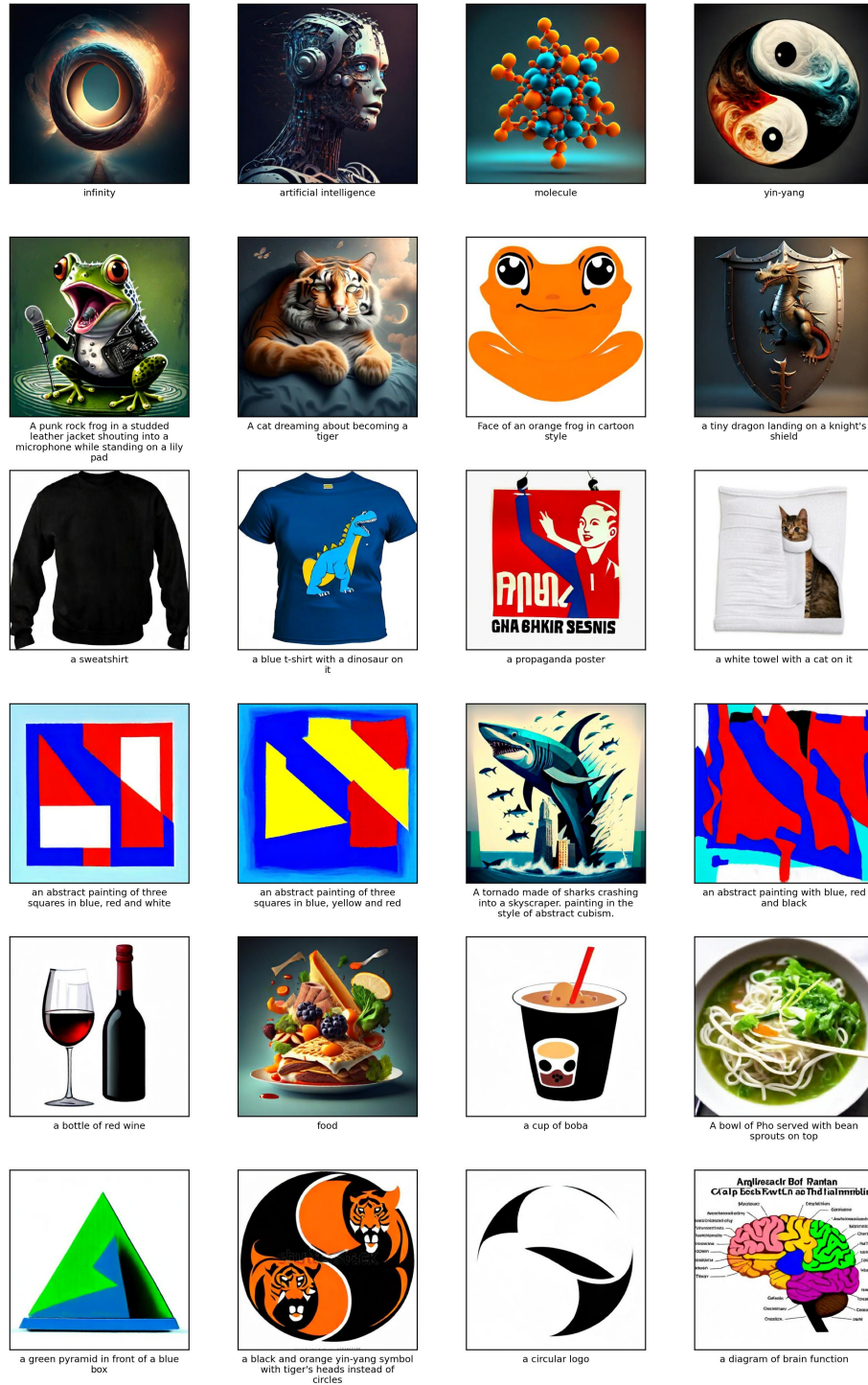
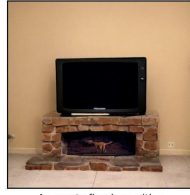


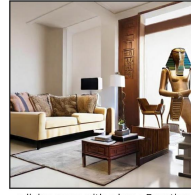
Figure C.11: **Evaluating on PartiPrompts.** Synthesized images by our model using randomly selected prompts from PartiPrompts [273]. Rows correspond to following categories: Abstract, Animals, Artifacts, Arts, Food & Beverage, and Illustrations.



a very fancy French restaurant



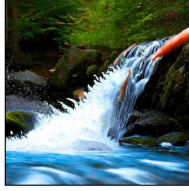
An empty fireplace with a television above it. The TV shows a lion hugging a giraffe.



a living room with a large Egyptian statue in the corner



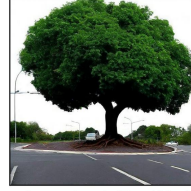
a kitchen



a mountain stream with salmon leaping out of it



a marina



a tree growing out of the middle of an intersection



a crescent moon viewed between tree branches at night



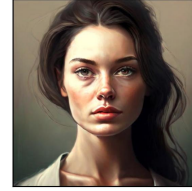
a photograph of the mona lisa drinking coffee as she has her breakfast. her plate has an omelette and croissant.



a man in a business suit on a ladder that is leaning up against the side of a white house



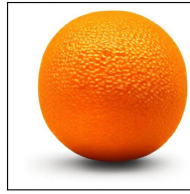
a team



a woman



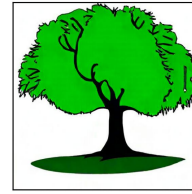
a flower with large yellow petals



an orange



A photo of a palm tree made of water.



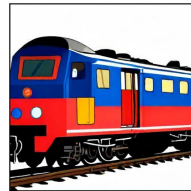
a tree



Three-quarters front view of a blue 1977 Corvette coming around a curve in a mountain road and looking over a green valley on a cloudy day.



a red sport bike



a train



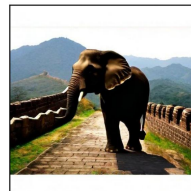
a motorcycle



a prop plane flying low over the Great Wall



A photo of an Athenian vase with a painting of toucans playing basketball in the style of Egyptian hieroglyphics

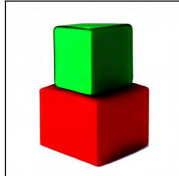


an elephant walking on the Great Wall



A photograph of a portrait of a statue of a pharaoh wearing steampunk glasses, white t-shirt and leather jacket.

Figure C.12: **Evaluating on PartiPrompts.** Synthesized images by our model using randomly selected prompts from PartiPrompts [273]. Rows correspond to following categories: Indoor scenes, Outdoor scenes, People, Produce & Plants, Vehicles, and World knowledge.



A stack of 3 cubes. A red cube is on the top, sitting on a red cube. The red cube is in the middle, sitting on a green cube. The green cube is on the bottom.



An emoji of a baby panda wearing a red hat, green gloves, red shirt, and green pants.



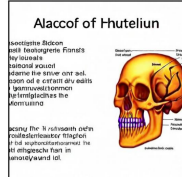
New York Skyline with 'Hello World' written with fireworks in the sky.



A keyboard made of water, the water is made of light, the light is turned off.



A small blue book sitting on a large red book.



An organ of soft nervous tissue contained in the skull of vertebrates, functioning as the coordinating center of sensation and intellectual and nervous activity.



A carrot on the left of a broccoli.



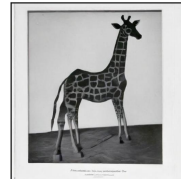
A giraffe underneath a microwave.



A storefront with 'Text to Image' written on it.



An umbrella on top of a spoon.



A photocopy of a photograph of a painting of a sculpture of a giraffe.



A fluffy baby sloth with a knitted hat trying to figure out a laptop, close up, highly detailed, studio lighting, screen reflecting in its eyes.



Painting of the orange cat Otto von Garfield, Count of Bismarck-Schönhausen, Duke of Lauenburg, Minister-President of Prussia. Depicted wearing a Prussian Pickelhaube and eating his favorite meal - lasagna.



A medieval painting of the wifi not working.



A sign that says 'Deep Learning'.



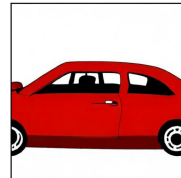
A black colored car.



A bird scaring a scarecrow.



A yellow and black bus cruising through the rainforest.



A red colored car.



Abraham Lincoln touches his toes while George Washington does chin-ups. Lincoln is barefoot. Washington is wearing boots.



Three dogs on the street.



A cat on the right of a tennis racket.



A blue cup and a green cell phone.



An ancient Egyptian painting depicting an argument over whose turn it is to take out the trash.

Figure C.13: **Evaluating on DrawBench.** Synthesized images by our model using randomly selected prompts from DrawBench [207].

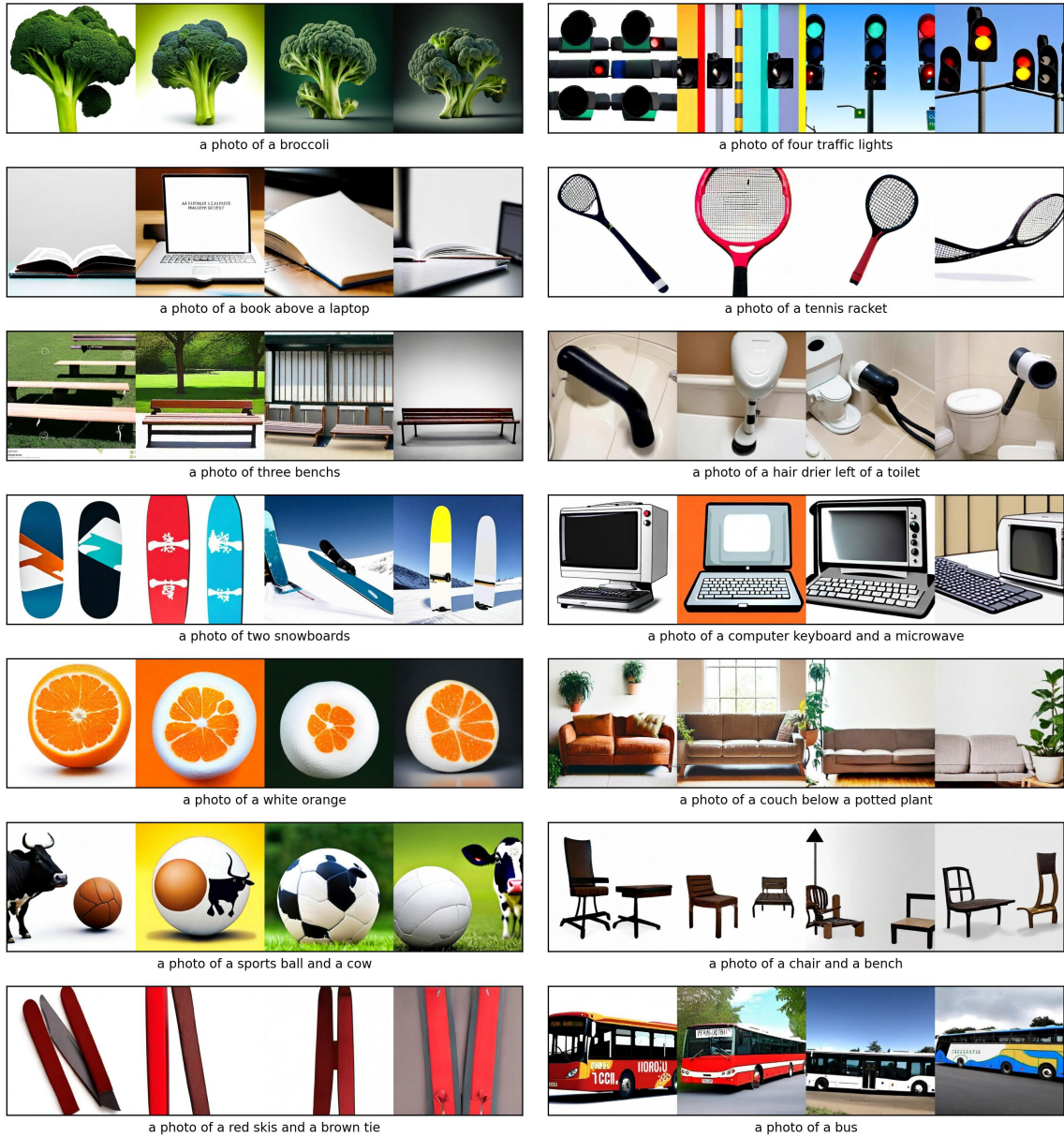
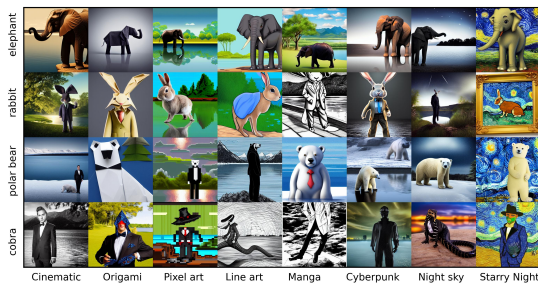
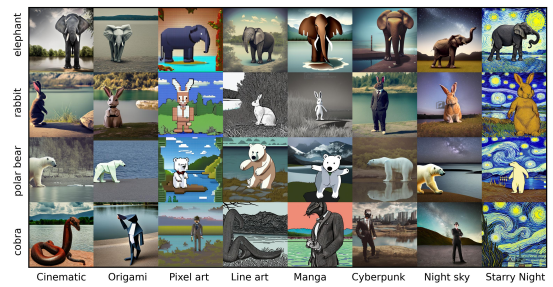


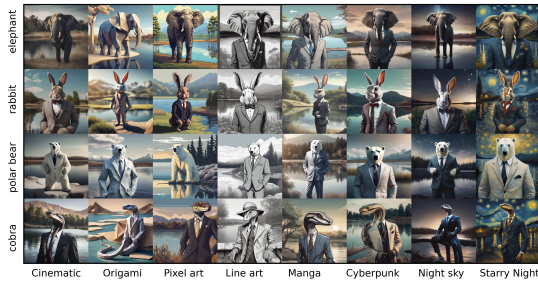
Figure C.14: **Evaluating on GenEval.** Synthesized images by our model using randomly selected prompts from GenEval [73].



(a) *Stable-Diffusion-1.5*



(b) *Stable-Diffusion-2.1*

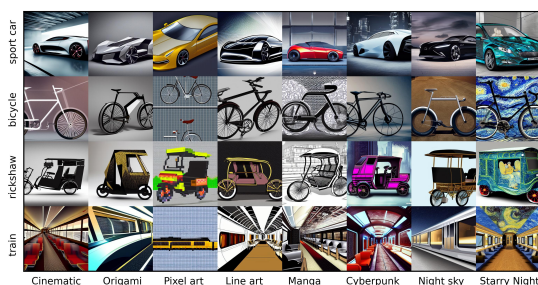


(c) *Stable-Diffusion-XL*

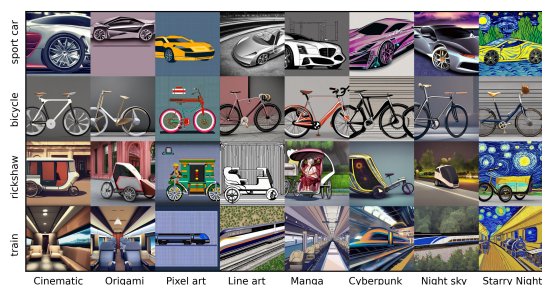


(d) *Ours*

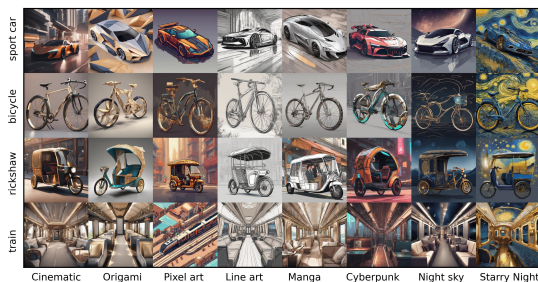
Figure C.15: **Control and diversity in image styles.** Evaluating the ability to generate diverse styles. *Prompt:* A ___ dressed in a suit posing for a photo in ___ style. Natural lake landscape in background, detailed light and shadow, high detail.



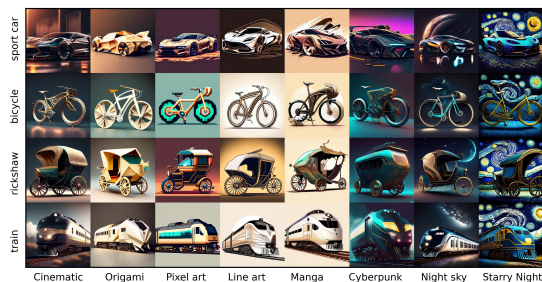
(a) *Stable-Diffusion-1.5*



(b) *Stable-Diffusion-2.1*



(c) *Stable-Diffusion-XL*

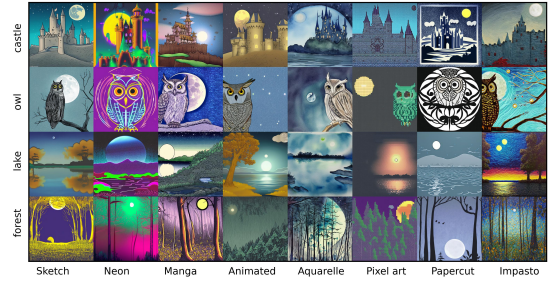


(d) *Ours*

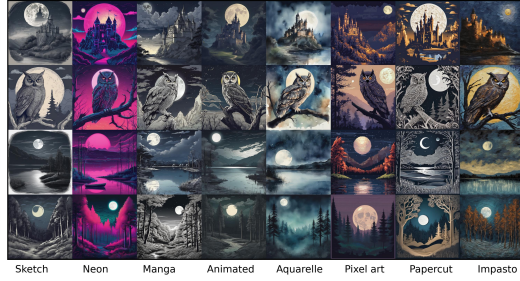
Figure C.16: **Control and diversity in image styles.** Evaluating the ability to generate diverse styles. *Prompt:* A luxurious — with a next-generation modern design illustrated in — style.



(a) *Stable-Diffusion-1.5*



(b) *Stable-Diffusion-2.1*

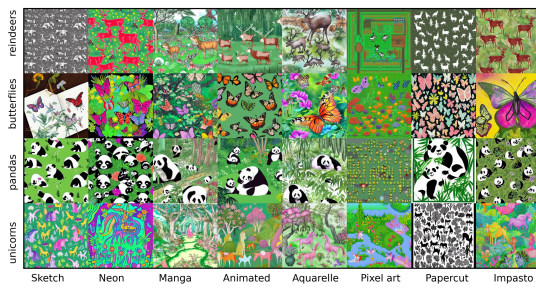


(c) *Stable-Diffusion-XL*

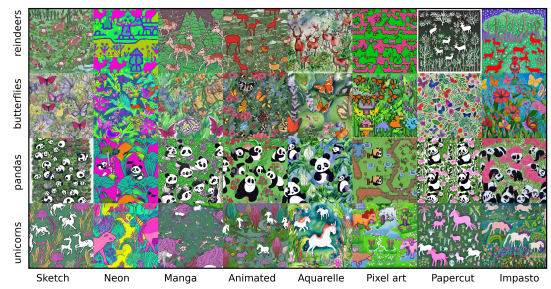


(d) *Ours*

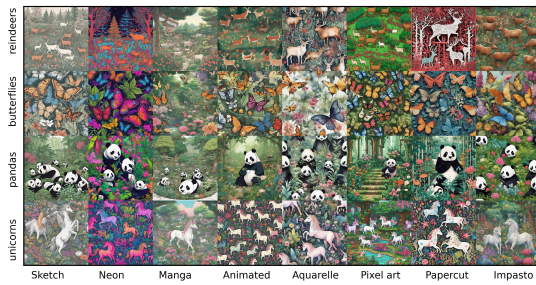
Figure C.17: **Control and diversity in image styles.** Evaluating the ability to generate diverse styles. *Prompt:* A moonlit night over a ____, mysteriously rendered in ____ style.



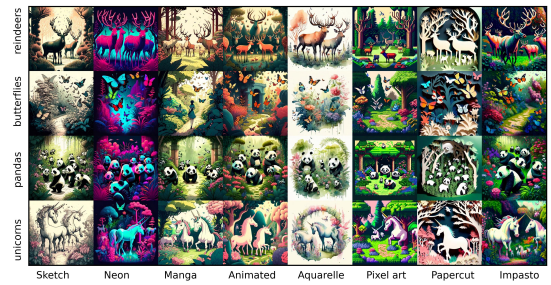
(a) *Stable-Diffusion-1.5*



(b) *Stable-Diffusion-2.1*



(c) *Stable-Diffusion-XL*



(d) *Ours*

Figure C.18: **Control and diversity in image styles.** Evaluating the ability to generate diverse styles. *Prompt:* A lush garden full of ____, vividly illustrated in ____ style.



flat vector, in a circle, Niagara Falls, Canada & USA, 8k, high resolution, adobe illustration color, no words



The Knight's Red Sword Cuts the Angel's Shining Wings.



Lo-fi style anime manga pirate and pirate ship, kawaii sticker, vaporwave, sparklecure, vibrant holographic gradient, iridescent highlights, flat vector, minimalist, [♥].



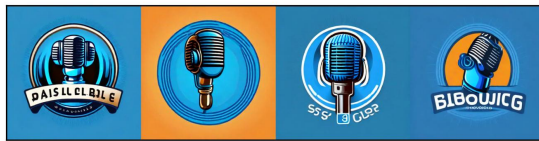
gorgeous girl white glossy female android covered in moss and wires in the style of robot, laughing facial expression, colorful flowers, solarpunk, detailed portrait, glamor, hi-resolution 4k, sharp focus, 8k, octane render



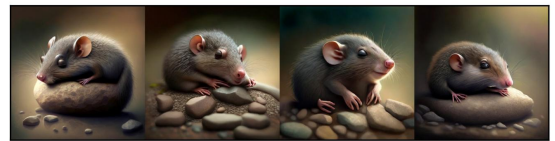
photo portrait of a blonde female racing driver in a black and purple race suit. She has a very determined expression on her face. Foggy background, dramatic lighting



The god of dreams, endovelico, endovelico, with a beard and white robes with a crown of palm leaves, peering into a dark abyss with his hand outstretched behind him, beckoning you to hold his hand and peer into the stygian blackness, celtic, gallic, gaulish, extremely detailed etching, celtiberian



a 2D professional logo for a podcast, blue background, microphone, no text



shrew feeling wonderful and blissed out, calm, stoned, sleepy, curled up



time machine, cinematic lighting, fine-art photography, photorealistic, Annie Leibovitz style, aesthetic, dynamic lighting, digital illustration, cinematic lighting, 8k,



symmetrical ancient woven fabric, minimal, animals circling the sun in the center



rose garden, lamp, photorealistic, insane details, cinematic lighting, canon 5d mk4, EF200mm F2L IS USM, iso 200



Two criminals, one male and one female, stand in silhouette with their backs to camera as they watch the city of London burn behind them + ultra realistic + photography + Full Perspective + crime + cinematography by Roger Deakins + award winning realism + The image is shot in a dark style + full body perspective + The composition is balanced and dynamic + The image is shot with a Arri Alexa, 18 mm lens 8k

Figure C.19: **Evaluating on JourneyDB.** Synthesized images by our model using randomly selected prompts from the test set of JourneyDB dataset [73].



first day of work in a big administration with a lot of scary workers, 70s, horror style, cinematic lighting



"A hijab woman is petting a dog in front of her house. The woman has green eyes and is wearing glasses. The dog has a black head and a white body.":: charcoal style::1



journalism photo of a futuristic alien ufo ring in the sky dramatic sky over a valley cinematic lighting realistic flying saucer detailed minimal ring, historic, breaking news, dramatic, fog, cinematic, old kodak photo, grainy, film scratches, news archive photo, polaroid photo



The mystical gate of Valhalla. The door is almost half open. Worn stairs leading to the door. Viking Runes glowing blue in the door frames. hyper detailed. Hyper quality. Hyper reality. 8k



concept art of a male hispanic mech pilot



bearded sorcerer Benedict Wong playing magical guitar with colorful lights in a dark tavern



8k. The most realistic photo of the Sahara desert terrain, with a black hole in the middle



2d game monster design, wizardry style, beautiful female werewolf, full body drawing, nendoroid, anime, black background



Fantasy concept art of an anthropomorphic bear shaman



anime concept art of a male field explorer



computer virus and hacker,



mob boss, mafia, red tie, cyberpunk, epic, dark, mysterious

Figure C.20: **Evaluating on JourneyDB.** Synthesized images by our model using randomly selected prompts from the test set of JourneyDB dataset [73].



hyperrealistic detailed mk2 vw golf gti in metallic blue, manchester city in background , caption, watermark, logo, or signature 8k



Jesus Christ Jehovah strong peaceful happy



Glass Cityscape in a recursive intricacy , sunrise



Full-length view, oil painting, woman morphing into colorful bird, gossamer wings, flowing robes, feminine, exquisite, delicate, ethereal, mystical, fantastical, ar3:4



The Starry night composed by cosmetics



queen of tooth fairies character, solid white background



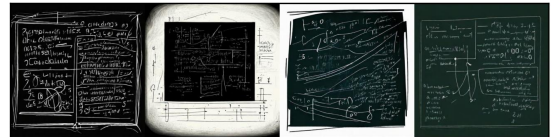
boxes of liquor chocolates, smashed, 1930s, prohibition, police officers, candy store



a traditional Chinese watercolor painting of a Chinese village with several bamboo houses in the sub-tropical mountains of Yunnan with a tea plantation in the background, traditional chinese dressed people, cobblestone path, in a round frame, three black chinese characters on a white background q2



a flatlay of a single Japanese watch face, 1943, no strap, against white background , writing implements, paper, hands



abstract, mathematical, chalkboard annotations a professor would use to illustrate complex financial derivatives in front of a classroom, clean vector lines, in the style of cy twombly



watercolore,3d render,cute moth fairy doll



photorealistic, HD, steampunk, owl with open wings, celtic designs on wings

Figure C.21: **Evaluating on JourneyDB.** Synthesized images by our model using randomly selected prompts from the test set of JourneyDB dataset [73].



coffee cup roast heart love wall art, Brian Allen, flat, sticker, black and white svg vector



steampunk percolator coffee pot sign which says steampunk, steampunk style, neon lights, smoky, transparent background



Stunning 4K Realistic Fog Bright Light neon pink blue fairy fantasy beach



corgi dressed as queen of england, in royal attire, pen and ink



Inscrutable wraith ghost hyperborean resistance research athaeneum, ubiquitous darkness enigma diffused moonlight



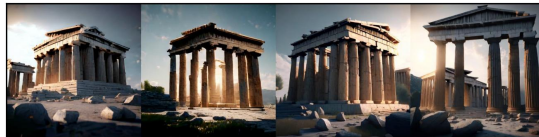
liquid girl, graffiti style,



T-shirt with short sleeves, printed in a climate of extreme sports, beautiful graphics



**Victorian candle, dark roses and peonies arrangement, surrounded by red floating fairy lights, hyper realistic, hyper detailed, low light, dark, gothic, witchy, 8d,



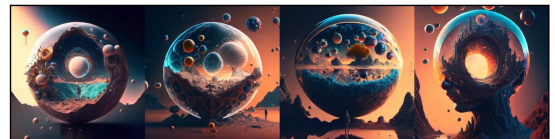
the Parthenon, Photo realistic, unreal engine, cinematic lighting, .5



black evil goblin, red eyes



young person from the 1920's, with a antique PC from 1920's, in their home shopping online in the 1920's , hyper realistic



The Multiverse:: Max Detailed Foreground:: In A Multi-Perspective Surreal Aesthetic, DMT, Psychedelic Style:: With a Cinematic Imaginative Background That Cuts Into The Foreground With Realistic Light Reflections and Perfect Scene Lighting:: Insanely detail, photomanipulation, photorealistic, C4D, Octane, Blender::

Figure C.22: **Evaluating on JourneyDB.** Synthesized images by our model using randomly selected prompts from the test set of JourneyDB dataset [73].

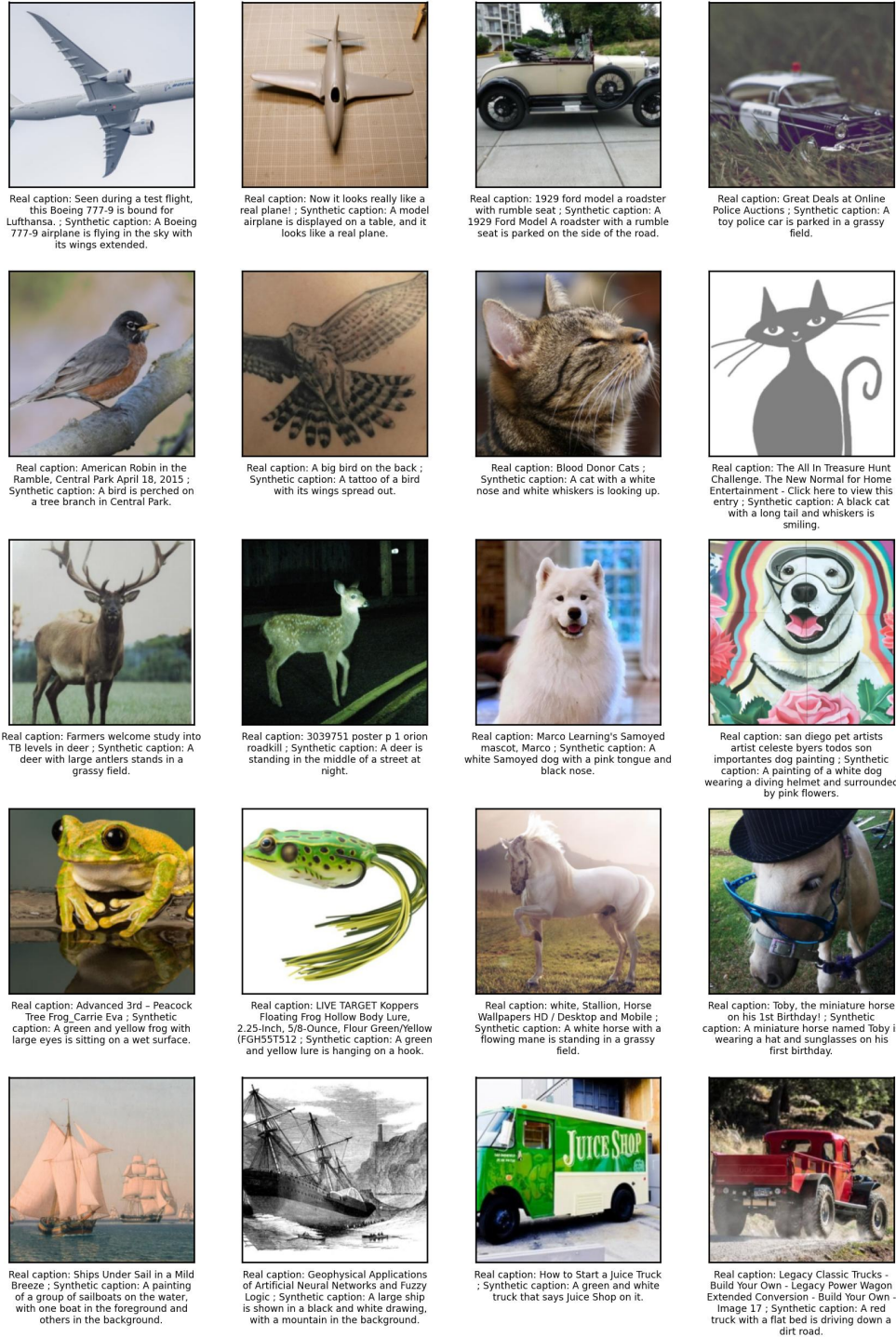


Figure C.23: **Samples from our cifar-captions dataset.** Selected images with corresponding real and synthetic captions from the cifar-captions dataset. We created the cifar-captions dataset, imitating the widely used CIFAR-10 dataset [130], to enable small scale experimentation on the text-to-image generative models.

List of extended DrawBench [207] prompts used in human-centric evaluation.

Sleek red car reflects cityscape, admired by passersby.

Black car cruises rainy streets, reflecting vibrant neon lights.

Glossy pink car cruising a sunny highway, surrounded by lush greenery, chrome gleaming, catching admiring glances.

Majestic black dog stands alert in a meadow, wildflowers sway, sunlight glinting on its sleek coat.

Curly red dog chases a butterfly in a windy, flower-filled meadow, frolicking playfully in the sunshine.

Vibrant blue dog prances in a sunny field, tail wagging, butterflies fluttering nearby.

A green banana contrasts with purple and yellow fruit, sunlight streaming in.

A red banana sits on a wooden table, surrounded by sunlight filtering through the leaves.

A black banana lies on white marble, hinting at overripe mystery beneath its dark peel.

White bread sandwich filled with cream cheese, cucumber, and turkey on a pristine plate, bathed in soft, warm light.

Black sandwich with charcoal bread, lettuce peeking out, set on a white plate beside lemonade.

Orange sandwich of roasted pumpkin and cheddar on whole wheat bread, sitting on a rustic table.

Pink giraffe in a wildflower field, plucking flowers with its long neck, butterflies swirling around.

Bright yellow giraffe nibbles leaves from tall trees under the warm savannah sun.

Brown giraffe strides across a sunlit savannah, its unique pattern glowing in the vibrant landscape.

Red car parked near a white sheep on a peaceful country road, hills rolling in the distance.

Blue bird perched on a brown bear's shoulder in a wildflower meadow, sharing a serene moment.

Green apple atop a black backpack on a park bench, with trees and children playing in the background.

Green cup of steaming coffee sits beside a blue cell phone on a rustic table in warm sunlight.

Yellow book and red vase filled with flowers sit on a sunlit wooden table, radiating elegance.

White car parked near a red sheep in a lush meadow, under a blue sky.

Brown bird sings from a cherry blossom branch while a blue bear gazes up, surrounded by flowers.

Black apple beside a green backpack on a rustic table, sunlight streaming through a window.

Blue ceramic cup filled with hot coffee sits next to a green cell phone, casting steam into the air.

Red book and yellow vase sit together on an antique table, illuminated by sunlight.

A horse wearing a spacesuit stands triumphantly on an astronaut's shoulders on a moon-like surface.

Pepperoni pizza bakes in a brick oven, surrounded by dancing flames and delicious aromas.

Cardinal startles a scarecrow in a cornfield, sending straw-filled arms flailing comically.

A blue pizza topped with blueberries, blackberries, and edible flowers sits on a rustic table.

A hovering cow abducts aliens with a tractor beam, while a spaceship glows in the night sky.

Panda making latte art in a cozy café, skillfully shaping a bamboo leaf in a cup of espresso.

A great white shark swims through desert dunes, its fin slicing through the sand like water.

An elephant swims under the sea, wearing a bubble-shaped helmet, surrounded by coral reefs and fish.

Rainbow-colored penguin waddles joyfully across a vibrant arctic landscape, reflecting the sky's colors.

A large fish leaps from the ocean to devour a startled pelican mid-flight during a storm.

One red sports car parked on a quiet cobblestone street in a picturesque European village.

Two vintage cars, one red and one blue, race down a charming cobblestone street.

Three colorful vintage cars line up along a cobblestone street, bathed in the setting sun's glow.

Four vintage cars in vibrant colors cruise down a bustling cobblestone street in retro style.

Five unique cars, from vintage to sleek sports models, line a lively city street.

A golden retriever trots along a cobblestone street, wearing a red bandana, as the sun sets.

Two dogs, a retriever and a beagle, walk together in the city, proudly sporting red and green bandanas.

Three dogs—a retriever, terrier, and dalmatian—walk down a lively street, leashes loosely held.

Four dogs—retriever, poodle, dachshund, and bulldog—strut down a bustling street in colorful bandanas.

Five playful dogs of various breeds chase each other on a cobblestone street, wagging tails happily.

A dog and a cat sit together on a grassy hill, gazing at the sunset over a lake.

A cat and two dogs sit peacefully on a grassy patch in a sunlit park, surrounded by flowers.

A regal cat sits with three cheerful dogs in a meadow, wearing a tiny gold crown.

Two cats and a dog relax on the grass as the sunset bathes them in warm colors.

Two cats and two dogs sit on a grassy hill, basking in the sunlight and gentle breeze.

Two cats and three dogs lounge together under a tree, enjoying the sunny meadow.

Three cats and a dog sit peacefully in the grass, enjoying each other's company.

Three cats and two dogs share a sunlit meadow, lounging gracefully on the soft grass.

Three cats and three dogs form a harmonious group, lounging together on a grassy hill.

Purple triangular flower pot brimming with green plants sits on a sunlit windowsill.

Orange triangular picture frame holds a serene landscape painting, set against a pale blue background.

A bright pink triangular stop sign stands at the edge of a forest road, glowing in the sunlight.

A denim-textured cube sits on a wooden table, surrounded by sewing needles and spools of thread.

A sphere made of kitchen tiles reflects sunlight in a mesmerizing pattern.

A large cube made of red bricks sits in a grassy field, vines creeping up its side.

A collection of nails neatly arranged on a wooden table in a cozy workshop.

A grand brass clock sits on an antique wooden table in a warmly lit room.

Two ornate crystal glasses sit on a polished wooden table, casting gentle shadows.

A red elephant sits atop a small blue mouse in a whimsical, flower-filled field.

A green elephant stands behind a large red mouse in a bright, grassy meadow.

A small blue book sits on top of a large red book in an elegant library.

Three stacked plates—two blue, one green—sit neatly on a rustic wooden table.

Three stacked cubes—two red, one green—rest on a wooden surface in vibrant colors.

A stack of three books—green, red, and blue—sits on a polished table in a sunlit room.

Baby panda emoji in a red hat and green pants grins adorably.

Baby panda wearing a red hat and blue gloves sits happily in its colorful outfit.

A turtle sits in a lush forest, captured through a fisheye lens, with the trees curving around.

A majestic owl perches in a wildflower field, its feathers ruffled by the gentle breeze.

A detailed cross-section of a brain showcases intricate structures and pathways.

A vintage bicycle rests against a brick wall, flowers in its wicker basket.

A modern bus cruises through a vibrant city, picking up passengers.

A small wooden boat glides peacefully across a serene lake, leaving a gentle wake.

A red fire hydrant stands at a busy city intersection as firefighters connect hoses.

A sleek parking meter stands beside a bustling city street, displaying the time left.

A vintage umbrella with lace patterns protects from a gentle rain in a sunlit park.

A vintage wooden chair with intricate carvings sits alone in a softly lit room.

An old-fashioned icebox sits in a cozy rustic kitchen, filled with fresh produce.

A steampunk-inspired clock with exposed gears sits on an ornate table, pendulum swinging.

A pair of ornate scissors rests on a patterned tablecloth, ready to cut delicate fabrics.

A majestic chestnut horse grazes in a sunlit meadow, its mane blowing in the breeze.

A bunch of ripe, yellow bananas hangs from a tree in a vibrant jungle scene.

A sleepy calico cat lounges on a windowsill, bathed in sunlight.

A well-groomed dog with a long snout stands proudly in a sunlit backyard, sniffing the air.

A colorful human brain sits within a transparent skull, pulses of energy flowing through it.

A futuristic office of a multinational tech company buzzing with innovation.

A grand piano with intricate carvings stands in a sunlit room, strings vibrating with music.

A golden coin symbolizes the decentralized nature of cryptocurrency, floating above a computer.

A thick-skinned hippopotamus stands at the edge of a calm river, basking in the sun.

A humanoid robot mimics a scientist's movements in a futuristic lab filled with machinery.

Customer pays for a tiny pizza with a giant quarter, both grinning in surprise.

An elegant couple in formal wear caught in a downpour, sharing a tender moment.

Pint cartons of milk sit neatly on the top shelf of a grocery store refrigerator.

A man stands in the shadow of a maple tree on a crisp January afternoon in New England.

An elephant hides behind a tree, trunk visible on one side, back legs on the other.

A tomato sits atop a pumpkin on a kitchen stool, a fork stuck in the side.

A pear cut into seven even pieces is arranged in a ring on a rustic table.

A donkey and octopus play tug-of-war while a cat leaps over the rope at sunset.

Supreme Court justices face off against FBI agents in a friendly game of baseball.

Abraham Lincoln touches his toes while George Washington does chin-ups in a meadow.

Tennis racket rests against a wooden bench on a sunlit clay court.

A well-worn baseball glove lies on a patch of grass, ready for the next catch.

A retro red refrigerator stands in a cozy kitchen filled with vintage décor.

A polished dining table set with fine china and silverware, beneath a glowing chandelier.

A vintage parking meter stands on a bustling street, surrounded by classic cars.

A small boat propelled by oars floats peacefully on a lake at sunset.

A fluffy cat rests in a cozy living room, intently watching a dangling toy mouse.

A pair of elegant stainless steel scissors with engraved handles rests on a patterned table.

A happy, well-groomed dog stands in a grassy backyard, sniffing the air.

A grand piano with wooden carvings stands in a sunlit room, with strings and hammers visible.

A red steam locomotive rides atop a surfboard, slicing through ocean waves at sunset.

A golden retriever balances a wine glass on its head, sitting proudly by a warm fireplace.

A vintage bicycle rests on top of a small wooden boat floating near the shore.

A delicate umbrella balances on top of a polished silver spoon on a wooden table.

A fluffy teddy bear supports a sleek laptop, its kind eyes peeking over the top.

A curious giraffe stands beneath an oversized microwave hanging from a tree branch.

A pink frosted donut lies beneath a white porcelain toilet in a whimsical bathroom scene.

A hair dryer blows warm air beneath a sheep standing calmly in a grassy field.

A tennis racket rests beneath a traffic light in a bustling city street.

A zebra stands under a gigantic broccoli tree, its stripes contrasting with the green leaves.

A banana rests on a wooden table to the left of a shiny red apple.

A red velvet couch sits beside a vintage leather chair in a sunlit living room.

A red sports car cruises beside a double-decker bus on a busy city street.

A sleek black cat lounges next to a tennis racket on a sunlit tennis court.

A stop sign leans against a refrigerator in a cozy, vintage-inspired kitchen.
 A fluffy white sheep stands beside a wine glass filled with red wine in a grassy meadow.
 A zebra stands next to a red fire hydrant in a bustling city street.
 Acersecomicke, a majestic creature, flies through a vibrant sky, searching for adventure.
 A family gathers for a jentacular feast in a warm kitchen, sharing laughter and food.
 Matutinal sun rises over a peaceful valley, casting golden light on the dewy grass.
 Pigeons socialize on a red-bricked rooftop in a village, creating a peristeronic scene.
 The mystical artophagous feasts on colorful paintings, absorbing their creative energy.
 An abandoned backlot of a film studio, overgrown and filled with forgotten treasures.
 The octothorpe, a massive metallic creature, roams through an abandoned city.
 Stained glass windows of a church depict a hamburger and fries, casting colorful rays inside.
 Otto von Garfield, Duke of Lauenburg, eats lasagna in an elegant painting.
 A baby fennec fox sneezes onto a strawberry, backlit ears glowing in detail.
 A confused grizzly bear attends calculus class, staring at the chalkboard in puzzlement.
 Egyptian painting shows two figures arguing over who should take out the trash.
 A baby sloth in a knitted hat stares at a laptop, trying to figure it out.
 A tiger in a lab coat works a science machine in a 1980s Miami-style laboratory.
 Animals dressed as humans pose for a 1960s yearbook photo in vintage clothing.
 A Lego version of Arnold Schwarzenegger stands confidently, holding a mini gun.
 A yellow and black bus cruises through the dense foliage of a rainforest.
 Medieval scholars gather around a broken Wi-Fi router, trying to restore the connection.
 An IT guy struggles with tangled cables like Laocoön while fixing a PC tower.
 A handful of colorful Skittles scattered across a smooth surface.
 A gothic-style McDonald's church with stained glass windows stands proudly in a village.
 An athletic cat addresses a scandal at a press conference, surrounded by journalists.
 A marble statue shows a man tripping over a surprised cat in a lush garden.
 A 1920s airship shaped like a pig floats over a golden wheat field.
 A tuxedo cat sings in a barbershop quartet, wearing a straw boater hat and bow tie.
 Astronaut couple poses in American Gothic style, holding a flag and a space helmet.
 The regal Burger King poses with a Whopper in an opulent oil painting.
 A keyboard made of water shimmers as the light is turned off, blending into darkness.
 Mona Lisa viewed from behind, her hair cascading down, gazing at a Tuscan landscape.
 A hyper-realistic photo of an abandoned industrial site during a storm.
 An iOS app screen shows different types of milk available for ordering.
 Super Mario leaps through a bustling city in a realistic 8K Ultra HD photograph.
 Cats climb the Eiffel Tower in a futuristic cyberpunk coloring page.
 A mega Lego space station towers inside a child's galaxy-themed bedroom.
 A spider with a moustache greets a gentlemanly grasshopper as they cross paths.
 A framed photo of a photocopy of a painting of a giraffe statue.
 Bird's-eye view of a bridge connecting Europe and North America across the Atlantic Ocean.
 A maglev train plunges vertically downward at high speed in New York City.
 A magnifying glass reveals a page from a 1950s Batman comic, highlighting a dramatic scene.
 A futuristic car plays soccer on a digital field, sending a holographic ball flying.

Darth Vader plays with a raccoon on Mars at sunset, red sky glowing around them.

A 1960s poster warns against climate change with vibrant, psychedelic imagery.

A mouse uses a mushroom as an umbrella during a gentle rain, ripples forming in puddles.

A Pomeranian dressed as a 1980s wrestler strikes a pose in neon wrestling tights.

A pyramid of falafel stands in the desert under a partial solar eclipse.

A storefront displays "Hello World" in large, welcoming letters across the glass window.

A charming storefront with "Diffusion" written elegantly across the window.

A storefront with "Text to Image" written above the door, surrounded by framed pictures.

A sleek storefront with "NeurIPS" etched on the glass, reflecting the busy street outside.

A quaint storefront with "Deep Learning" written in gold script on the window.

A modern storefront with "Google Brain Toronto" etched above the glass doors.

A rustic storefront with "Google Research Pizza Cafe" warmly inviting passersby inside.

A wooden sign in a meadow says "Hello World," surrounded by colorful wildflowers.

A weathered wooden sign reading "Diffusion" sways gently in the breeze.

A sign in a grassy field says "Text to Image," with butterflies fluttering around.

A large conference center sign displays "NeurIPS" in bold letters, lit by string lights.

A wooden sign engraved with "Deep Learning" stands in a meadow, glowing in the sunset.

A sleek metal sign says "Google Brain Toronto," standing in front of a glass building.

A colorful sign reading "Google Research Pizza Cafe" hangs outside a charming café.

The New York skyline glows as fireworks spell out "Hello World" in the evening sky.

Fireworks over the New York skyline spell out "Diffusion" against the night sky.

The New York skyline at dusk with "Text to Image" written in dazzling fireworks.

Fireworks over New York spell out "NeurIPS," illuminating the cityscape below.

"Deep Learning" is written in fireworks above the New York skyline, reflected in the water.

"Google Brain Toronto" written with fireworks above the New York skyline at night.

The New York skyline shines as fireworks spell "Google Research Pizza Cafe."

"Hello World" spelled out in brilliant fireworks over New York's famous skyline.

"Diffusion" written with fireworks above the New York skyline, creating a dazzling display.

The New York skyline lights up with "Text to Image" written in vibrant fireworks.