

UC Berkeley

UC Berkeley Electronic Theses and Dissertations

Title

Data Augmentation and Generation for Improving Learning-based Autonomous Driving Models

Permalink

<https://escholarship.org/uc/item/4mr5v61f>

ISBN

9798189270048

Author

Li, Yiheng

Publication Date

2026-05-01

Peer reviewed|Thesis/dissertation

Data Augmentation and Generation for Improving Learning-based Autonomous Driving
Models

By

Yiheng Li

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering Science - Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Masayoshi Tomizuka, Chair

Professor David Bogy

Professor Koushil Sreenath

Professor Pieter Abbeel

Spring 2026

Data Augmentation and Generation for Improving Learning-based Autonomous Driving Models

Copyright 2026
by
Yiheng Li

Abstract

Data Augmentation and Generation for Improving Learning-based Autonomous Driving Models

by

Yiheng Li

Doctor of Philosophy in Engineering Science - Mechanical Engineering

University of California, Berkeley

Professor Masayoshi Tomizuka, Chair

Learning-based models have evolved to become the main-stream in autonomous driving research and industry recent years, where data serve an irreplaceable role. However, real-world driving data are costly, requiring expensive data collection instruments and substantial human labors. Therefore, there is a critical need to maximize the informational yield of available datasets and investigate synthetic data generation as a scalable alternative, in order to enable the scaling of learning-based driving models while circumventing the prohibitive costs associated with massive real-world data acquisition.

In this thesis, we first introduce heuristic approaches for the direct augmentation and synthesis of driving data in Chapter 2, aiming to empirically validate the significance of data scaling for autonomous driving models. Specifically, we employ prescriptive planning models to synthesize physically plausible vehicle trajectories and apply geometric transformations to augment existing map topologies. These operations, while simple enough and involve no additional data collections, introduce critical diversity to the motion data, both for the trajectory side and for the map side. By using these augmented data to pre-train a motion prediction model, we observe substantial gains in the motion prediction accuracy.

Beyond direct geometric augmentations, enriching the semantic modalities of existing datasets serves as a critical strategy to extract deeper insights from available driving logs and therefore to maximize data efficiency. In Chapter 3, we augment motion data with natural language descriptors that characterize complex agent interactions. These interactions, especially those governed by intricate traffic rules and subtle human intentions, are often notoriously difficult to distill from raw numerical trajectories alone. Natural language, however, provides a powerful medium to render these latent constraints explicit. This research led to the development of WOMD-Reasoning [107], which, at the time of its release, constituted the most extensive language-annotated dataset for real-world driving interactions. Our results demonstrate that this multimodal enrichment not only catalyzes motion prediction accuracy

but also provides a necessary layer of interpret-ability, allowing models to better utilize the causal logic embedded within existing driving scenarios.

The advent of generative AI has established a new paradigm for synthesizing photorealistic, vision-based driving data, which is imperative for training robust end-to-end autonomous driving systems. While these generative frameworks offer the theoretical capability to produce high-fidelity scenarios, their practical utility is severely constrained by prohibitive training costs. Consequently, achieving training efficiency is a prerequisite for scalable data generation. To circumvent this computational bottleneck, Chapter 4 introduces Immiscible Diffusion, a novel image-noise assignment strategy that fundamentally accelerates the convergence of diffusion models, achieving up to a $3\times$ efficiency gain. Building upon this foundation, Chapter 5 provides a rigorous analysis of the feature-level suboptimalities inherent in standard diffusion processes, while elucidating the underlying mechanisms that drive the efficiency enhancements of our approach. Guided by these insights, we propose generalized implementations of Immiscible Diffusion that further elevate training efficiency to $> 4\times$, establishing a versatile framework adaptable to diverse model architectures and training environments. Collectively, these algorithmic innovations enable cost-effective, high-fidelity data synthesis, serving as the computational backbone for scalable vision-based driving data synthesis and closed-loop driving simulation.

In summary, this dissertation establishes a cohesive framework for data-centric autonomous driving, progressing from heuristic data augmentation and semantic data enrichment to fundamental optimizations in generative modeling for data generation. By systematically resolving bottlenecks in data utilization and generation, this work unlocks the potential of synthetic data as a scalable engine for autonomous driving systems. These contributions lay the essential groundwork for a future where robust driving models are trained within high-fidelity, closed-loop simulations, minimizing the reliance on labor-intensive real-world data collection.

To the epoch 0.

Contents

Contents	ii
List of Figures	v
List of Tables	x
1 Introduction	1
1.1 Background and Motivation	1
1.2 Research Scope and Contributions	2
1.3 Dissertation Organization	4
I Heuristic Driving Data Augmentation with Planner and Geometric Transformations	6
2 Pre-training on Synthetic Driving Data for Trajectory Prediction	7
2.1 Introduction	7
2.2 Related Works	10
2.3 Methods	11
2.4 Experiments	15
2.5 Discussion	18
2.6 Conclusion	20
II Language Augmentation of Driving Data for Featuring Interactions	21
3 WOMD-Reasoning: A Large-Scale Dataset for Interaction Reasoning in Driving	22
3.1 Introduction	23
3.2 Related Work	25
3.3 Method	26
3.4 WOMD-Reasoning Dataset Specifications	28

3.5	Dataset Application and Evaluation with Motion-LLaVA	32
3.6	Dataset Effectiveness in Vehicle Trajectory Prediction	36
3.7	Conclusion	37

III Efficient Training of Vision-based Driving Data Generation Models 38

4	Immiscible Diffusion: Accelerating Diffusion Training with Noise Assignment	39
4.1	Introduction	40
4.2	Related Work	43
4.3	Method	44
4.4	Experiments	49
4.5	Conclusion	53
5	Improved Immiscible Diffusion: Accelerate Diffusion Training by Reducing Its Miscibility	55
5.1	Introduction	55
5.2	Related Works	57
5.3	Improved Immiscible Diffusion	59
5.4	Effectiveness of Immiscible Diffusion	63
5.5	Discussions	67
5.6	Conclusion	68
6	Final Words	70
6.1	Summary of Key Contributions	70
6.2	Implications and Impact	71
6.3	Future Directions to Explore	71
6.4	Concluding Remarks	72

Bibliography 73

A	Appendix for WOMD-Reasoning: A Large-Scale Dataset for Interaction Reasoning in Driving	93
A.1	Details in Building WOMD-Reasoning	93
A.2	Examples of Simulated Vision Modality	100
A.3	Processing Agent IDs for Motion-LLaVA	100
A.4	Hyperparameters for Multipath++	100
A.5	Prompts for evaluating GPT Score	101
A.6	Qualitative Comparison of Benchmarked Multi-modal Models on Answering Questions in WOMD-Reasoning	102
A.7	Additional Baseline Language Models on WOMD-Reasoning	103

A.8	Example Aggregated Context for Chain-of-Thought Reasoning	104
A.9	Example Factual Description Context for Chain-of-Thought Reasoning . . .	105
A.10	Ablations on Motion-LLaVA	105
A.11	WOMD-Reasoning Dataset Efficiency Analysis	107
B	Appendix for Immiscible Diffusion: Accelerating Diffusion Training with Noise Assignmen	108
B.1	Additional Experiment Results	108
B.2	Qualitative Evaluations of Immiscible Diffusion	111
C	Appendix for Improved Immiscible Diffusion: Accelerate Diffusion Training by Reducing Its Miscibility	119
C.1	Experiment Setup Details	119
C.2	Qualitative Comparisons of Image-to-image Tasks	120
C.3	Choice of k in the KNN implementation	120
C.4	tSNE of Noise Point Clusters for Different Methods	120
C.5	Illustrations Comparing Immiscible Diffusion to Other Common Methods . .	121
C.6	Execution Time of Immiscible Diffusion	121

List of Figures

1.1	Sensor-equipped Vehicle for Real-world Data Collection. The figure illustrates the data collection platform utilized by [193]. Retrofitted upon a standard vehicle chassis, the system is instrumented with multiple cameras and LiDARs for robust perception. This complex sensor suite significantly elevates the hardware cost compared to consumer vehicles. © Argo AI, LLC.	2
2.1	Our data synthesis and self-supervised pre-training pipeline enhances prediction performance without extra real-world driving data. The area of each circle is proportional to the number of driving scenes used, including the synthetic and real ones. The numbers in the circles indicate the percentages of real-world driving data from the Argoverse dataset used for fine-tuning. (A lower minimum final displacement error ($\min FDE_6$)[15] is preferable.)	8
2.2	Pipeline of driving data synthesis and utilization. We augment the map and generate trajectory on it to acquire synthetic motion data, which are then used for pre-training via masking and reconstruction. The pre-trained model is used to initialize the backbone model during fine-tuning.	11
2.3	Performance comparison without or with pre-training. The green line and star indicate the true trajectory and its last point, while orange ones are the predicted ones. The orange background shows the possibility of each point being the predicted last point of the trajectory. (a) and (b) show the prediction results without or with trajectory pre-training. (c) and (d) illustrate the performance without or with map pre-training.	16
2.4	Data distribution comparison between synthetic dataset (blue) and the real-world dataset (red) in terms of the speed and direction properties. (a) represents the map representation for MIA city. (b) represents the trajectory distribution of scenes in (a), showing a pattern of divergence in velocity properties. (c) represents the trajectory distribution of scenes in (b) rotated to the same initial direction, demonstrating a pattern of divergence in direction properties. (d)(e)(f) are the counterparts of (a)(b)(c) in PIT city.	17

3.1	Examples of traffic rule-induced interactions in WOMD-Reasoning dataset. (a) captures the traffic rule-induced interaction between the ego agent and agent #0, attributing it correctly to the stop signs. (b) shows the traffic light-controlled yielding interaction between the ego agent and agent #1. The front-view visualization is created by MetaDrive simulator [102]	23
3.2	Selected vocabulary statistics in WOMD-Reasoning. We stat vocabularies strongly related to traffic rule-induced and human intention-induced interactions in WOMD-Reasoning, illustrating that it contains abundant such interaction descriptions and reasoning.	30
3.3	A demonstration of Q&As in each part of WOMD-Reasoning dataset. We show Q&As in all categories regarding the scenario while demonstrating language analysis of overtaking, a human intention-induced interaction in WOMD-Reasoning dataset.	31
3.4	The Motion-LLaVA pipeline fine-tuning a multi-modal model with WOMD-Reasoning. Motion data go through pre-trained motion vector encoders from Multipath++ [179] and a projector layer to serve together with the questions in WOMD-Reasoning as the inputs. The answers in WOMD-Reasoning serve as the supervision.	31
3.5	Interaction predictions made by Motion-LLaVA on various WOMD scenarios.	33
3.6	Traffic rule-compliant planning made by Motion-LLaVA on various WOMD scenarios.	34
3.7	An example set of answers to diverse driving-related questions generated by Motion-LLaVA on a WOMD scenario. We see that Motion-LLaVA is capable of answering all topics of questions covered by WOMD-Reasoning accurately.	34
4.1	Immiscible Diffusion can use a single line of code to efficiently achieve immiscibility by re-assigning a batch of noise to images. This process results in only a 2% reduction in Euclidean distance post-assignment, leading to up to 3× increased training efficiency on top of the Consistency Model [166] for CIFAR Dataset [89] Additionally, Immiscible Diffusion significantly enhances the image quality of Stable Diffusion [149] for both unconditional and conditional generation tasks, and for both training from scratch and fine-tuning training tasks, on ImageNet [31] Dataset within the same number of training steps. FID [63] here refers to Fréchet Inception Distance.	40
4.2	Physics illustration of Immiscible Diffusion. (a) depicts the miscible and Immiscible Diffusion phenomenon in physics, while (b) demonstrates the image-noise pair relation in vanilla (miscible) and Immiscible Diffusion method.	42

4.3	Feature analysis of vanilla (miscible) and immiscible DDIM. Referring to [161], $\tau = S$ represents the layer denoising from the pure noise. We show that while the two sampled noises are similar, the denoised image of immiscible DDIM significantly outperforms that of the traditional one, generating an overall reasonable image. The reason behind this is traditional methods cannot successfully predict noises at noisy layers.	47
4.4	Evaluation of baseline and immiscible Consistency Models on (a) CIFAR-10, (b) CelebA, and (c) tiny-ImageNet dataset. We illustrate the FID of two models with different training steps. Clearly, immiscible Consistency Models have much higher efficiency than the vanilla ones.	50
4.5	Evaluation of baseline and Immiscible DDIM on CIFAR-10 dataset with different inference steps S. We find that Immiscible DDIM outperform the baseline more significantly when the number of inference steps S is smaller.	51
4.6	Evaluation of baseline and immiscible class-conditional Stable Diffusion on ImageNet dataset, using 20 inference steps. (a) FID of two models trained from scratch (b) FID of two models fine-tuned on Stable Diffusion v1.4.	52
5.1	Improved Immiscible Diffusion Theory. (a) While vanilla diffusion trajectories (flows) are mixed (miscible), <i>each noise point is stably correlated to a specific generated image</i> , making many diffusion trajectories irreversible. (b) Those irreversible trajectories would confuse the denoising process. (c) We introduce Immiscible Diffusion to cut mixed (miscible) diffusion trajectories during training for accelerating diffusion training.	56
5.2	Stable correlation between generated images and its noise origins. Here perturbation means another Gaussian noise added to the fixed original Gaussian noise. Note that even with 20% perturbation, images changes are nearly unnoticeable. Only with 30% perturbation, a image object change happens. These demonstrate stable correlation from a noise area to a specific generated image.	59
5.3	Feature analysis of vanilla (miscible) and immiscible DDIM. Referring to [162], $\tau = S$ represents the layer denoising from the pure noise. We show that Immiscible Diffusion activates the noisiest ($\tau \rightarrow S$) layers' denoising functions by clarifying their denoising goals, as shown in the <i>tSNE</i> (t-Distributed Stochastic Neighbor Embedding) of denoised images across τ 's. Such activation results in FID improvements on the denoised images from large τ 's, which leads to better performance and faster convergence of diffusion models.	61
5.4	Implementations of Immiscible Diffusion. (a) Miscible Diffusion pairs the batch of images and noises randomly before adding noise to images. (b)(c)(d) Immiscible Diffusion tries to reduce the miscibility of diffusion by (b) L_2 linear assignment between the batch of images and noises and (c) sampling k noises and pick the nearest one (KNN) to use. (d) scaling images by multiplying their pixel values with a constant > 1 , which reduces overlaps between diffuse-able areas of different images.	62

5.5	Immiscible diffusion boosts training efficiency. We show the training steps required to reach the best FID for vanilla models across three diverse diffusion-based architectures. Results consistently show that Immiscible Diffusion trains significantly faster.	64
5.6	Immiscible diffusion stays effective across diverse image generation tasks. We demonstrate this with figures showing the training steps <i>v.s.</i> FID of (a)(b) Stable Diffusion class-conditional training and fine-tuning with ImageNet, and (c) conditional training with MS-COCO.	65
5.7	Immiscible diffusion boosts the performance of diffusion policy. Here the robot (circle) needs to push the T-shaped object (gray) into the desired place (green).	66
5.8	Analysis on DDIM with images normed to different pixel value STDs.	67
A.1	More visual examples of WOMD-Reasoning through simulations.	100
B.1	Effectiveness of Immiscible DDIM in a selected range of batch sizes.	110
B.2	Qualitative comparison for Immiscible and baseline Consistency Model. We show images generated with the two models trained for 100k steps respectively. Compared to baseline method, immiscible models capture more details and more features of objects.	111
B.3	Generated images from baseline Consistency Models trained on CIFAR-10 Dataset for 100k steps without cherry-picking.	112
B.4	Generated images from immiscible Consistency Models trained on CIFAR-10 Dataset for 100k steps without cherry-picking.	113
B.5	Generated images from baseline Consistency Models trained on CelebA Dataset for 100k steps without cherry-picking.	114
B.6	Generated images from Immiscible Consistency Models trained on CelebA Dataset for 100k steps without cherry-picking.	115
B.7	Images generated by immiscible and baseline Stable Diffusion trained unconditionally on ImageNet for 70k steps. We see that the Immiscible Stable Diffusion presents more reasonable modal and catch more general features and details.	116
B.8	Generated images from immiscible and baseline stable diffusion models trained unconditionally on 10% ImageNet Dataset for 70k steps without cherry-picking .	117
B.9	Generated images from immiscible and baseline stable diffusion models trained conditionally on ImageNet Dataset for 20k steps.	118
C.1	Qualitative Illustration of Immiscible Diffusion in Image-to-image Tasks. We notice that immiscible diffusion can better preserve existed information (i.e. edge information in in-painting and center information in out-painting) so as to provide better overall completed images.	120

C.2	tSNE of noise clusters belonging respectively to vanilla, assign and KNN DDIM.	121
C.3	Comparison between Diffusion Models, Flow Matching and Immiscible Flow Matching.	122

List of Tables

2.1	Effortless Driving Data Generation.	8
2.2	Parameter selection for data synthesis.	13
2.3	Method Comparison for Using Synthetic Data.	16
2.4	Performance of Self-Supervised Pre-Training.	18
2.5	Contributions of map augmentation.	19
2.6	Influence of Information Leak.	20
3.1	Comparison between WOMD-Reasoning Dataset and Previous Real-world Language Datasets for Driving. WOMD-Reasoning covers more comprehensive kinds of interactions, with a significantly higher size supporting a wide range of applications.	25
3.2	Sizes of WOMD-Reasoning dataset.	29
3.3	Human Evaluations of WOMD-Reasoning Dataset.	29
3.4	Benchmarks of fine-tuned multi-modal models with language metrics.	35
3.5	Accuracy of Motion-LLaVA answers to factual questions.	35
3.6	Benefits of Motion-LLaVA Outputs for Prediction Task.	37
4.1	Experiment setting.	49
5.1	FID evaluations of vanilla and Immiscible Diffusion in Image-to-image Tasks. . .	65
5.2	Immiscible diffusion in robotics planning.	66
A.1	An example of the language translation of motion data.	95
A.2	System prompt for GPT.	96
A.3	Responsibility prompt for GPT.	97
A.4	Rule prompt for GPT.	98
A.5	In-context prompt for GPT.	99
A.6	Prompt for GPT Score.	101
A.7	Additional language model baselines evaluated on WOMD-Reasoning. We report the answer quality of several vanilla and WOMD-Reasoning-fine-tuned models, including LLaMA-Adapter, ViTA, and Qwen2.5-VL (all 7B versions), to further validate the effectiveness of WOMD-Reasoning.	104
A.8	Effects of CoT in answering reasoning-related questions.	106

A.9	Benefits of using pre-trained motion vector encoder for fine-tuning multi-modal models on WOMD-Reasoning for factual Q&As.	106
A.10	Refinement of Motion-LLaVA training strategy.	107
A.11	Performance of Motion-LLaVA with different WOMD-Reasoning dataset sizes on factual and interaction reasoning questions. Results confirm the data efficiency of WOMD-Reasoning in improving driving-related reasoning.	107
B.1	Immiscible Diffusion boosts training efficiency for Consistency Model on CIFAR-10 dataset.	108
B.2	FID improvements of Immiscible DDIM with different inference steps	109
B.3	FID of using L1 or L2 norm for noise assignment in immiscible DDIM on CIFAR-10.	109
C.1	Image Generation Experiment setting.	119
C.2	Best k 's in KNN Immiscible Diffusion Implementation.	120
C.3	Execution Time (ms) of Immiscible Diffusion on a single A5000 GPU.	121

Acknowledgments

When I first arrived in Berkeley four years ago, I initially planned to continue my research in mechanics. At that moment, I never anticipated that I would pivot my Ph.D. research entirely to autonomous driving, and graduate in just 4.5 years. These achievements were made possible by the exceptional guidance, help, and support of my advisors, colleagues and family.

I would like to express my deepest gratitude to my advisor, Prof. Masayoshi Tomizuka, for his unwavering support throughout my Ph.D. studies. When I transitioned to Prof. Tomizuka’s MSC lab, I possessed no prior knowledge of controls, robotics, or autonomous driving. Yet, Prof. Tomizuka took a chance on me, welcoming me into a strong team that guided me into this field. He was incredibly patient when I faced the steep learning curve of grasping new concepts in a short time-frame, providing constant encouragement and insightful feedback on my projects. His mentorship allowed me the flexibility to explore cutting-edge topics, such as diffusion-based models. Without Prof. Tomizuka’s generous support, transforming my entire knowledge base to complete my Ph.D. in 4.5 years would have been impossible. I feel incredibly fortunate to have had him as my advisor.

I also wish to extend my sincere and heartfelt thanks to Prof. David Boggy, who served as my advisor when I first joined Berkeley. I am deeply grateful for his continuous mentorship and care. Prof. Boggy granted me the invaluable freedom to explore and identify my true research interests, offering both financially and spiritually critical support during my transition. It was his encouragement that allowed me to find the path I am passionate about. I cannot imagine how I could have navigated the early stages of this journey without his backing.

I am also truly grateful to my dissertation committee members, Prof. Koushil Sreenath and Prof. Pieter Abbeel, for their time and effort in reviewing my dissertation and beyond. I specifically thank Prof. Sreenath for serving as the chair of my qualifying exam, helping me organize the process, and offering his time to review my research progress mid-way. Additionally, Prof. Abbeel’s insights into the development of DDPM [65] greatly inspired my colleagues and me to identify areas for improvement, which resulted in a significant portion of this dissertation [104, 164].

Beyond my faculty advisors, I would like to express my utmost gratitude to my colleagues. First, I must thank the supervisors of my projects, who, fittingly, have all recently secured faculty positions: Prof. Chenfeng Xu, Prof. Chen Tang, and Prof. Mingyu Ding. I still remember starting my first project with absolutely no knowledge of AI or even Python. At that time, Chenfeng and Chen provided thorough, patient, step-by-step guidance, going far beyond any obligation. Later, recognizing my background in mechanics, Chenfeng encouraged me to dive into diffusion-based models, a direction I have come to love. Guided by Mingyu and Chen, we achieved the seemingly impossible task of building a language annotation dataset with very tight funding. I cannot thank them enough for sharing their valuable time with me; I will never forget the nights we strove together until 4 or 5 a.m. to meet conference submission and rebuttal deadlines. These experiences laid the foundation

for my research in autonomous driving and general AI, enabling the successful completion of my degree.

I also thank my co-authors for their brilliant ideas, dedicated collaboration, and generous support: Prof. Kurt Keutzer, Prof. Bolei Zhou, Prof. Huaxiu Yao, Seth Zhihao Zhao, Heyang Jiang, Cunxin Fan, Dr. Chenran Li, Akio Kodaira, Jeff Feng Liang, Dan Kondratyuk, and Dr. Wei Zhan.

In addition, I am grateful for the help and camaraderie provided by my lab-mates at the MSC lab and UC Berkeley: Dr. Liting Sun, Prof. Jiachen Li, Dr. Shiyu Jin, Dr. Yujiao Cheng, Dr. Jessica Leu, Dr. Yiyang Zhou, Dr. Huidong Gao, Dr. Hengbo Ma, Dr. Changhao Wang, Dr. Xinghao Zhu, Dr. Jinning Li, Dr. Lingfeng Sun, Dr. Zheng Wu, Dr. Catherine Weaver, Dr. Wu-te Yang, Dr. Xiang Zhang, Dr. Thomas Ran Tian, Qiyang Qian, Wei-er Chang, Keita Kobashi, Jen-wei Wang, Yichen Xie, Yuxin Chen, Boyuan Liang, John Viljoen, Yixiao Wang, Chensheng Peng, Shuqi Zhao, Yukata Shimizu, Pengcheng Wang, Dr. Qilong Cheng, Dr. Roshan Mathew Tom, and many others not listed due to space limitations.

Finally, and most importantly, I thank my family for their unconditional love and support. Without them, it is hard to imagine how I could have navigated these challenging years. Every night, my parents listened to my stories, including my updates, achievements, and difficulties, and helped me stay on track with their love and care. To my wife, who has walked every step of this journey by my side: I know I often placed the weight of my world onto your shoulders, even as you navigated the immense pressures of your own Ph.D. studies. You quietly absorbed my anxieties and turned them into strength. Your rational clarity and unwavering love became my sanctuary against every obstacle. I am forever indebted to your sacrifice and commitment to me.

I also acknowledge my industry partners at Honda Research Institute and Waymo for their fruitful discussions and insightful suggestions regarding my research.

Chapter 1

Introduction

1.1 Background and Motivation

The pursuit of fully autonomous driving [134, 72, 50, 11] constitutes one of the most transformative engineering challenges of the modern era, which has witnessed significant milestones in recent years. In the realm of L_4 [27] (Level 4) autonomous driving, defined as high driving automation within a specific Operational Design Domain (ODD) without human intervention, large-scale robo-taxi fleets have been successfully deployed across major U.S. cities. These fleets have accumulated millions of autonomous miles, reportedly achieving an 80% reduction in injury-related crash rates [94]. Concurrently, regarding L_2 (Level 2) autonomous driving (*i.e.*, partial automation requiring human supervision), leading manufacturers [39, 40] have widely integrated L_2 modules into consumer vehicles to alleviate the driving burden for users.

Over the past decade, the autonomous driving research field has witnessed a paradigm shift from modular, rule-based pipelines [178, 220] toward robust, learning-based architectures [14, 55, 156, 75, 159]. Whether through modular pipelines that utilize deep learning for perception [218, 138, 28], prediction and planning [56, 47, 14, 156], or recent End-to-End (E2E) frameworks that map raw sensor inputs directly to control commands [69, 159], the efficacy of autonomous driving models is increasingly defined by the scale, semantic diversity and coverage of the data used for training.

However, this reliance on data has introduced a critical bottleneck. While algorithmic architectures have gradually matured, the acquisition of high-quality training data remains prohibitively expensive and logistically complex [28, 106]. Collecting real-world driving data requires fleets of sensor-equipped vehicles (demonstrated in Fig. 1.1) to traverse hundreds of hours to capture statistically significant distributions of traffic scenarios [13, 41, 15]. For example, the Argoverse v1.1 [15] spends 320 driving hours in selecting a motion dataset with 324k scenarios, while its 2nd version [193] further takes 763 driving hours to offer 250k scenarios. Another example is, the Waymo Open Motion Dataset (WOMD) [41] selects 104k scenarios from 574 hours of driving data. Furthermore, raw sensor data must undergo



Figure 1.1: **Sensor-equipped Vehicle for Real-world Data Collection.** The figure illustrates the data collection platform utilized by [193]. Retrofitted upon a standard vehicle chassis, the system is instrumented with multiple cameras and LiDARs for robust perception. This complex sensor suite significantly elevates the hardware cost compared to consumer vehicles. © Argo AI, LLC.

rigorous human annotation to generate the ground truth required for supervised learning, a process that is labor-intensive and difficult to scale [206].

This challenge is exacerbated by the long-tail distribution of driving events [78]. Nominal driving scenarios (e.g., highway cruising, stopping at red lights) are abundant, but safety-critical edge cases, such as aggressive cut-ins, unprotected left-turns and erratic pedestrian behaviors *etc.*, are statistically rare [157]. Relying solely on real-world data collection to capture these rare events is economically unsustainable. Consequently, **there is a critical need to maximize the informational yield of available datasets, and to investigate synthetic data generation as a scalable alternative.**

In this dissertation, we establish a cohesive framework for data scarcity problems in training learning-based autonomous driving models. We address these challenges through three progressive stages: heuristic data augmentation [106], semantic data enrichment with language [107], and optimizations of diffusion-based models’ training efficiency for vision-based driving data generation [104, 105]. These methods help to alleviate the reliance on labor-intensive real-world data acquisition, enabling the construction of a scalable data engine that is not only physically diverse and semantically meaningful, but also computationally efficient for training robust autonomous driving models.

1.2 Research Scope and Contributions

In each chapter of this dissertation, we systematically discuss techniques we develop to improve data utilization and generation for autonomous driving, unlocking the potential

of data enrichment and data generation as a scalable engine for the driving systems. The contributions are organized into three main parts:

1.2.1 Heuristic Augmentation of Driving Data

An intuitive solution to tackle the data scarcity problem is to synthesize more driving data based on existing ones. In Chapter 2, we introduce heuristic approaches for the direct augmentation and synthesis of existing driving data. Through introducing such a simple and straight-forward data augmentation process, we aim to validate the significance of data scaling for improving motion prediction models.

Specifically, we employ prescriptive planning models to synthesize physically plausible vehicle trajectories and apply geometric transformations to augment existing map topologies. Unlike naive noise injection, these operations ensure that the synthesized data remains kinematically valid while introducing critical diversity to both the trajectory and map modalities. By using these augmented data to pre-train a motion prediction model in a self-supervised style, we demonstrate substantial gains in prediction accuracy, validating that heuristic synthesis can effectively densify the training distribution and improve motion predictions without additional data collection costs.

1.2.2 Semantic Enrichment via Natural Language

Beyond data augmentations, existing data often lack semantic depth. Raw numerical trajectories capture how agents move, but not why. In Chapter 3, we address this by enriching motion data with natural language descriptors that characterize high-level agent interactions, defined as those interactions transcend mere spatial proximity, like those interactions induced by traffic rules or human intentions.

Interactions governed by intricate traffic rules (e.g., yielding at stop signs) or subtle human intentions (e.g., aggressive merging) are notoriously difficult to be learned from trajectory coordinates alone by either rule-based or learning-based autonomous driving models. Encouragingly, natural language provides a powerful medium to render these latent constraints explicit. This leads to the development of our WOMD-Reasoning [107] dataset, which constituted the most extensive language annotation dataset for real-world driving scenarios featuring interactions at the time of its release. We designed an automatic pipeline, consisting of a rule-based tool translating motion data into raw language descriptions and a Large Language Model (LLM)-based tool to produce scene descriptions and interaction analysis in the form of Q&A’s based on the raw descriptions, which can enrich the semantic information of motion data without additional human labor. Our experiments show that this language enrichment not only catalyzes motion prediction accuracy but also provides a necessary layer of interpretability, enabling models to reason about the causal logic embedded within driving scenarios.

1.2.3 Efficient Generative AI for Data Synthesis

The advent of generative AI has established a new paradigm for synthesizing photo-realistic, vision-based driving data, which is imperative for training robust end-to-end autonomous driving models. However, while generative frameworks offer unprecedented high fidelity, their practical utility is constrained by prohibitive training costs. Chapter 4 and Chapter 5 are to address this computational bottleneck.

In Chapter 4, we introduce Immiscible Diffusion, a novel image-noise assignment strategy for enhancing the training efficiency of diffusion-based vision generative models. We notice that mapping each image to the whole noise space in diffusion-based models may cause confusions in the denoising process, therefore limits the effectiveness of some denoising layers and drags the overall training procedure of diffusion-based models. By optimizing the mapping between images and noise with in-batch image-noise assignment according to their Euclidean (L_2) distance during training, we fundamentally mitigate the ambiguity in the denoising process and accelerate the convergence of diffusion models, achieving up to a 3× efficiency gain [104].

Building upon this, Chapter 5 provides a rigorous analysis of the feature-level suboptimalities inherent in standard diffusion processes. Guided by these insights, we propose generalized implementations of Immiscible Diffusion that further elevate training efficiency to $> 4\times$ across diverse diffusion-based baseline models. Excitingly, we found Immiscible Diffusion also effective for improving the accuracy of diffusion policy. These algorithmic innovations serve as the computational backbone for cost-effective, high-fidelity driving data generator.

1.3 Dissertation Organization

The remainder of this dissertation is organized as follows:

- **Chapter 2** details the heuristic driving data augmentation pipeline, demonstrating how geometric transformations and planning-based data augmentations can scale driving data, and therefore improve the trajectory prediction model’s performance.
- **Chapter 3** presents the WOMD-Reasoning, the dataset we build to extract latent semantic information from driving data and render it into explicit natural language. We detail the automatic pipeline we build to generate the dataset without additional human labor, but instead with LLMs. We further fine-tune a multi-modal LLM, taking motion data as inputs to offer semantic analysis, upon WOMD-Reasoning. The fine-tuned model can analyze interactions in driving scenarios, and these analyses are later proved to be effective in enhancing the accuracy of motion predictions in autonomous driving.

- **Chapter 4** introduces the Immiscible Diffusion, a physics-inspired algorithm to improve the training efficiency of diffusion-based models with few lines of code, which can significantly lower the cost of training for vision-based driving data generations.
- **Chapter 5** further expands the theoretical framework of Immiscible Diffusion, offering generalized implementations and deeper theoretical analysis, making it a flexible method widely work for diverse diffusion baselines. Besides, we also find Immiscible Diffusion can help improve diffusion policy’s action quality, shedding light into its application in diffusion-based motion prediction models.
- **Chapter 6** concludes the dissertation and outlines future directions for solving data scarcity problem in learning-based autonomous driving models.

Part I

Heuristic Driving Data Augmentation with Planner and Geometric Transformations

Chapter 2

Pre-training on Synthetic Driving Data for Trajectory Prediction

Accumulating substantial volumes of real-world driving data proves pivotal in the realm of trajectory forecasting for autonomous driving. Given the heavy reliance of current trajectory forecasting models on data-driven methodologies, we aim to tackle the challenge of learning general trajectory forecasting representations under limited data availability. We propose to augment both HD maps and trajectories and apply pre-training strategies on top of them. Specifically, we take advantage of graph representations of High-Definition (HD) map and apply vector transformations to reshape the maps, to easily enrich the limited number of scenes. Additionally, we employ a rule-based model to generate trajectories based on augmented scenes; thus enlarging the trajectories beyond the collected real ones. To foster the learning of general representations within this augmented dataset, we comprehensively explore the different pre-training strategies, including extending the concept of a Masked AutoEncoder (MAE) for trajectory forecasting. Extensive experiments demonstrate the effectiveness of our data expansion and pre-training strategies, which outperform the baseline prediction model by large margins, *e.g.* 5.04%, 3.84% and 8.30% in terms of Miss Rate (MR_6), minimum average displacement error ($minADE_6$) and minimum final displacement error ($minFDE_6$), whose detailed definitions can be found in [15].

2.1 Introduction

Trajectory forecasting is an important task for safely navigating autonomous vehicles in crowded traffic scenarios. The state-of-the-art trajectory forecasting models are empowered by data-driven supervised learning approaches, whose performance heavily relies on the scale of motion data available for training [42, 206, 196]. However, driving data is expensive and time-consuming to collect and annotate, which hinders cost-efficient scaling of training data as in Natural Language Processing (NLP) and Computer Vision (CV). Data-collection vehicles with sophisticated sensors need to run on public roads to collect traffic data. For

Table 2.1: Effortless Driving Data Generation.

Dataset	Driving Hours	Scenes
Argoverse v1.1	320	324k
Synthetic Dataset	+0	+370k

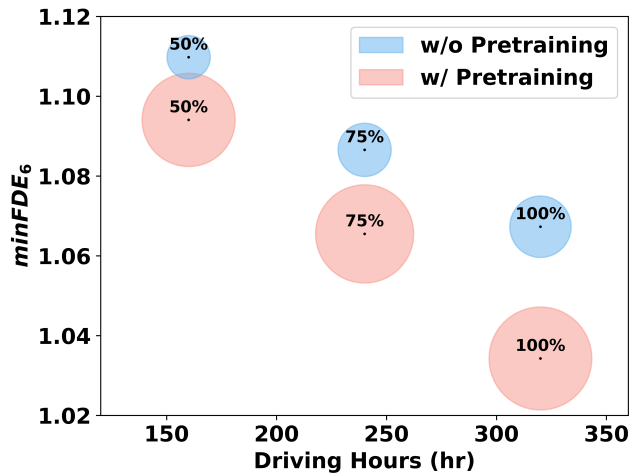


Figure 2.1: **Our data synthesis and self-supervised pre-training pipeline enhances prediction performance without extra real-world driving data.** The area of each circle is proportional to the number of driving scenes used, including the synthetic and real ones. The numbers in the circles indicate the percentages of real-world driving data from the Argoverse dataset used for fine-tuning. (A lower minimum final displacement error ($\min FDE_6$) [15] is preferable.)

instance, the Argoverse V1.1 dataset [15] and the Waymo Motion Dataset [42] consist of 324k and 103k driving scenes, which add up to a total of 320 and 574 hours of driving data, respectively. Notably, a substantially larger amount of raw data needs to be collected in order to filter out the given amount of high-quality training data. In addition to data collection, a significant amount of human labor is required to annotate the raw data into a cohesive dataset—traffic participants need to be annotated; and HD maps need to be aligned and integrated with the traffic data.

In this chapter, we sought *synthetic data* as a solution to break through the data bottleneck in trajectory forecasting. We propose to automatically generate synthetic driving data to complement real-world data, enhancing the overall data diversity and utility. Compared to collecting and processing real-world data, which involve huge labor, generating synthetic data is effortless—as shown in Tab. 2.1, our proposed data generation method only takes 20 computational hours to generate a comparable number of scenes to the real-world dataset.

Yet, as shown in Fig. 2.1, the synthetic data can improve the prediction accuracy and data efficiency when paired with our proposed training pipeline.

Concretely, our proposed data generation consists of two parts: 1) map augmentation; and 2) trajectory generation. For map augmentation, we adopt the vector-transformation method proposed in [6] to convert the linear lanes found in real-world maps into curved lanes within a defined range of sharpness and angle, as shown in Fig. 2.2 (upper-left). The augmented curved roads are introduced to diversify the map data—curved lanes occur at a significantly low frequency in real-world data. Subsequently, we navigate a simulated vehicle on both real-world and augmented maps with a rule-based planner [204] to generate trajectory data. The rule-based planner leverages prior knowledge to facilitate the realism of the generated trajectories.

Despite our efforts to improve the realism of the generated data, there still exists considerable discrepancy between the synthetic and real-world data. One crucial question is how to efficiently utilize the synthetic data to foster prediction performance while minimizing the impact of the domain gap. To answer this question, we conducted an extensive study on different training algorithms to use the synthetic data. In particular, we compare three different paradigms: augmenting the real-world prediction dataset [216, 151, 85], supervised pre-training [60, 123, 192], and self-supervised pre-training [181, 4, 21, 59, 58, 34, 203, 22, 141, 213]. As for directly augmenting the prediction dataset, we disregard the domain gap and mix the synthetic and real data to train the prediction model directly. In our supervised pre-training paradigm, we first pre-train the model for the supervised trajectory prediction task on the synthetic dataset, and then fine-tune the model on the real dataset. While the pre-training and fine-tuning tasks are both supervised prediction tasks, the two-stage scheme is introduced to mitigate the bias due to the domain gap. Further, in self-supervised pre-training, we substitute the supervised pre-training task with self-supervised masked reconstruction tasks [58] tailored to trajectory forecasting, i.e., map reconstruction, trajectory reconstruction, or a combination of both. The goal is to train a generalizable scene representation on the synthetic data that oversees detailed domain-specific information and can benefit cross-domain fine-tuning. Extensive experiments conducted on the Argoverse dataset with DenseTNT backbone demonstrate that self-supervised pre-training outperforms the other two ways. With our synthetic data generation and self-supervised pre-training pipeline, the fine-tuned model outperforms the baseline model by a large margin—5.04%, 3.84%, and 8.30% in terms of MR_6 , $minADE_6$ and $minFDE_6$. Detailed ablation studies are then conducted to shed light on the influence of some subtle factors on prediction performance, such as information leak and map augmentation.

Our contributions are summarized as follows:

- We propose to leverage synthetic map and trajectory data to address the data scarcity issue in trajectory forecasting with minimal cost. In particular, we introduce a novel data generation scheme that consists of geometric-based map augmentation and rule-based trajectory generation.

- We conducted extensive experiments to pinpoint the best training scheme to utilize the synthetic data. Our results show that self-supervised pre-training is the most effective one among the three tested paradigms: augmenting the training dataset, supervised pre-training, and self-supervised pre-training.
- With our synthetic data generation and self-supervised pre-training pipeline, the fine-tuned model outperforms the baseline model by a large margin—5.04%, 3.84%, and 8.30% in terms of MR_6 , $minADE_6$ and $minFDE_6$.

2.2 Related Works

2.2.1 Data Augmentation in Trajectory Prediction

Data scarcity is a bottleneck for exploring the model’s capacity for better performance [196]. To address this issue, numerous data augmentation techniques, such as the application of image transformations [21, 183, 194] and utilization of synthetic data [205, 51, 43, 194], are introduced in the CV community to augment or replicate seldom-seen real-world scenes. While these methods expand the dataset, they also introduce a domain gap between the real-world distribution and their augmented counterparts. In the trajectory prediction problem, the unique properties of trajectory and map data necessitate a specific design for data augmentation or generation that is both reasonable and cost-effective. In the approach detailed in [5], a simple augmentation technique is applied to the vehicle’s velocity properties to mimic unpredictable driver behaviors in anticipation of future trajectories based on past history. Additionally, [6] proposes the use of map augmentation techniques to alter the topological semantics of map information as adversarial attacks on current trajectory prediction models. While these methods expand the dataset size, their additional trajectories are simple geometric transformations of existing lane elements or trajectories, without careful consideration of the feasibility and realism of the transformed trajectories. In our work, we enhance the map data to supplement limited features, like curves and turns, and employ a rule-based simulation to mimic reasonable traffic behaviors for trajectory generation.

2.2.2 Pre-training in Trajectory Prediction

Pre-training is effective for the data shortage problem in trajectory prediction. In [196], contrastive learning was used to cluster similar map patches and establish associations between map patches and the trajectory, thereby fully utilizing the existing map data. In [9], four pre-training tasks were initiated, including randomly masking and recovering parts of the features, to enhance the encoding. Concurrently to our work, [17, 23] used MAE to pre-train the encoder. However, these methods did not introduce any new data. Another work [5] generated a pseudo trajectory that strictly follows the lanes for pre-training. However, this strict compliance hinders the realism of the pseudo trajectories. In contrast, our work aims

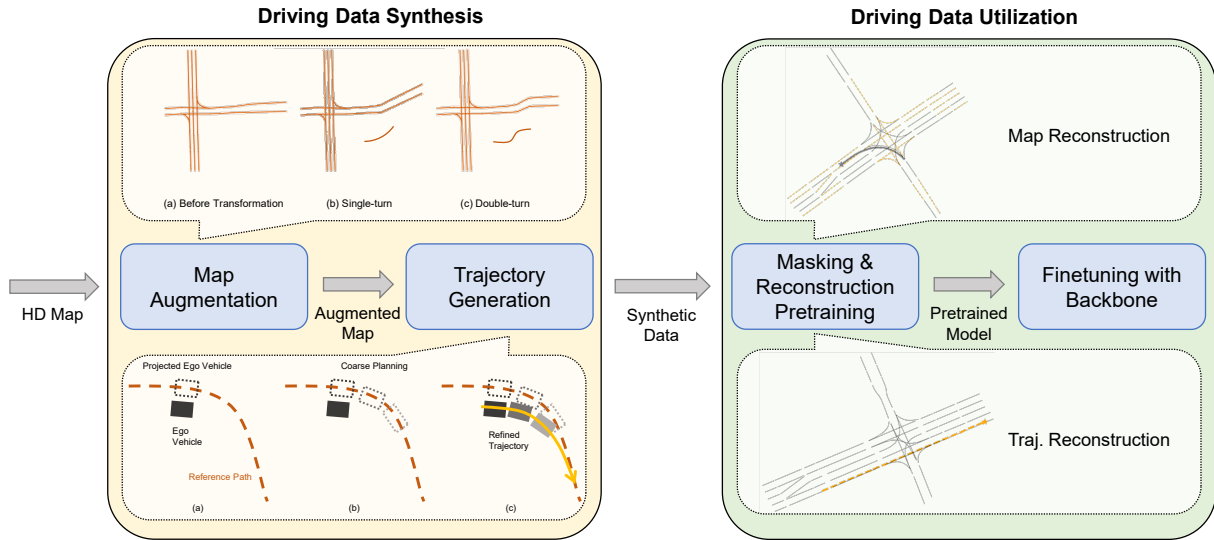


Figure 2.2: **Pipeline of driving data synthesis and utilization.** We augment the map and generate trajectory on it to acquire synthetic motion data, which are then used for pre-training via masking and reconstruction. The pre-trained model is used to initialize the backbone model during fine-tuning.

to generate realistic synthetic trajectories to mitigate the domain gap between the synthetic and real data.

While limited efforts on pre-training exist in the trajectory forecasting literature, some auxiliary tasks have been introduced to regularize trajectory prediction models during training. [47] introduced a feature-masking task to encourage feature interactions, which inspired us to use methods in CV [58] and NLP [34] to stimulate feature interactions in the learned scene representation. Taking cues from these works, we investigated masked reconstruction as a self-supervised pre-training method to utilize the synthetic data.

2.3 Methods

2.3.1 Synthetic Map and Trajectory Data Generation

Map Data Augmentation

Considering the lack of curve trajectory in the real-world training dataset, we add the map augmentation module to our data synthesis pipeline. Inspired by the conditional adversarial scene generation process proposed in [6], we design transformation functions that alter the original map topology, as shown in Fig. 2.2 (upper-left). Given that each scene is composed of a set of scene points, we define the transformation on each scene point (s_x, s_y) in the

following form:

$$\bar{s} = (s_x, s_y + f(s_x - b)) \quad (2.1)$$

where $\bar{s}$ is the transformed point, f is the single-variable transformation function, and b is a parameter that determines the region of applying the transformation, i.e., we only modify the scene points whose x-coordinates surpass b . We further adopt the following two transformation functions for our map augmentation process.

Single-turn introduces a single turning on the roadway. The transformation function for single-turn is defined as:

$$f_{\text{single-turn}}(s_x) = \begin{cases} 0, & s_x < 0, \\ q_\alpha(s_x), & 0 \leq s_x \leq s_t \\ (s_x - s_t)q'_\alpha(s_t) + q_\alpha(s_t), & s_t < s_x \end{cases} \quad (2.2)$$

where s_t represents the length of the turn and q_α is an auxiliary function defined by

$$q_\alpha(s_x) = \frac{\alpha_1}{s_t^{\alpha_2}} s_x^{\alpha_2} \quad (2.3)$$

where α_1, α_2 are the parameters to control the turn's sharpness and angle. Note the q'_α is simply the derivative of q_α . Our formulation of $f_{\text{single-turn}}(s_x)$ is continuously differentiable and thus makes a smooth augmented single-turn curve.

Double-turn introduces two consecutive smooth single-turns in opposite directions to the road. The transformation function for double-turn is based on the that of single-turn:

$$f_{\text{double-turn}}(s_x) = f_{\text{single-turn}}(s_x) - f_{\text{single-turn}}(s_x - \beta) \quad (2.4)$$

where β is the distance between two turns. In practice, we randomly select single-turn or double-turn augmented lanes and randomly assign values in a specific range to ensure a reasonable extent of transformation, as shown in Tab. 2.2.

Trajectory Data Generation

To generate labeled trajectory data on both real-world and augmented maps, we implement a trajectory data generator to synthesize pseudo-expert trajectories in single-car scenarios, as illustrated in Fig. 2.2 (bottom-left). Following the motion data generation pipeline proposed in [204], we began by projecting its position onto the nearest reference path obtained from the training dataset in order to determine the optimal trajectory for the ego vehicle. Subsequently, we utilize the A* planning method on this reference path to generate a preliminary, coarse trajectory. Specifically, we define a node n_i in the search space by a tuple (s_i, v_i, t_i) , which respectively represents the ego vehicle's coordinate, velocity, and time on the reference path. A transition from n_i to n_{i+1} defines the n_{i+1} by:

$$(s_{i+1}, v_{i+1}, t_{i+1}) = (s_i + v_i \delta t + \frac{1}{2} a \delta t^2, v_i + a \delta t, t_i + \delta t) \quad (2.5)$$

Table 2.2: Parameter selection for data synthesis.

Traj. Param.	Value	Map Param.	Value
a	$\in [-2, 1]$	s_x	10
w_1	5	α_1	$\in [1, 10]$
w_2	5	α_2	20
w_3	1	s_t	10
v_d	$\in [6, 15]$	β	20

where a denotes the acceleration and δt denotes the minimum time interval for coarse planning. The cost function of the transition is given as:

$$\mathcal{C}(n_i, a, n_{i+1}) = w_1 a^2 + w_2 \kappa(s_{i+1}) v_{i+1}^2 + w_3 (v_{i+1} - v_d)^2 \quad (2.6)$$

where w_1, w_2, w_3 are weights for each term, $\kappa(s_{i+1})$ denotes curvature of reference path at s_{i+1} , and v_d denotes desired driving velocity set in prior. The planning process concludes when the time associated with the searched node surpasses the end time t_g , resulting in the conversion of the resultant path into a coarse trajectory $s_c(t)$ within the global frame. The hyperparameters adopted for the synthetic trajectory generation pipeline are listed in Tab. 2.2. Finally, we introduce an additional refinement procedure to map the initial state of the final pseudo-expert trajectory to the current state of the ego car. Concretely, we solve the following optimization problem to acquire the refined trajectory $s_r(t)$:

$$\begin{aligned} \min_{s_r} \quad & \sum_{T=\delta\hat{t}}^{t_g} \omega_1 a_r(T)^2 + \omega_2 j_r(T)^2 + \sum_{T=k\delta\hat{t}}^{t_g} \omega_3 (s_r(T) - s_c(T))^2 \\ \text{s.t.} \quad & v_r(T_0) = v_0 \quad \text{and} \quad s_r(T_0) = s_0 \end{aligned} \quad (2.7)$$

where $\omega_1, \omega_2, \omega_3$ are weights for each term, a_r, j_r are refined acceleration and jerk, $\delta\hat{t}$ is the fine-grained time step such that $k\delta\hat{t} = \delta t$. v_0, s_0 corresponds to the ego car's initial velocity and position.

2.3.2 Utilization of Synthetic Data

To make maximum use of synthetic data while avoiding the influence of domain gaps, we comprehensively study several candidate approaches to identify the best strategy. The first approach is to directly combine the synthetic and real data for training, i.e. treating them identically. While this method is fairly straightforward, it also introduces bias into the model due to the domain gap between the synthetic and real data. Another method is supervised pre-training: we first pre-train the model for the supervised trajectory prediction task on

the synthetic dataset, and then fine-tune the model on the real dataset. While the pre-training and fine-tuning tasks are both supervised prediction tasks, the two-stage scheme is introduced to mitigate the bias caused by the domain gap. To further mitigate the influence of the domain gap, finally, we investigate the self-supervised pre-training method which is elaborated in the following subsection.

2.3.3 Self-supervised Pre-training Framework

We adapt MAE [58] into a self-supervised pre-training framework for trajectory forecasting. The goal is to learn a generalizable scene representation that can well capture the interactions between lanes and the ones between lanes and trajectories from the synthetic data. During the pre-training stage, a specific percentage of trajectories or map lanes are masked, challenging the model to reconstruct the masked segments. The masked objects are replaced with mask tokens, which are learned and shared parameters that signal the existence of a masked object to be reconstructed [58]. Positional encoding is added to provide the subordinate and sequential information. Though specific information is removed in mask tokens, they still provide the model with where the masked information is. Such information leak will lead the model to find a shortcut to interpolate missing information instead of to reconstruct them by context understanding [48]. To prevent this, only the unmasked elements are processed through the backbone’s encoder. Thereafter, the masked tokens, in combination with the encoded unmasked parts, are fed into a shallow transformer decoder to reconstruct the masked parts. Upon completion of the pre-training phase, the backbone model is initialized using the encoder parameters of the pre-trained model, and then fine-tuned on real-world data for the prediction task. This approach enables the model to acquire general semantics during the pre-training stage while discarding extraneous details.

2.3.4 Masking Strategy

We propose and compare different design options for masking and reconstruction during the pre-training stage:

Map Reconstruction

Map reconstruction is proposed to foster the encoder to capture the lane features and their interactions. The masking policies and reconstruction procedures are illustrated in Fig. 2.2 (upper-right). We randomly choose 50% lanes as masking lanes, replacing everything but the coordinate information for their first points with mask tokens. These masked objects, together with the encoded features of the unmasked objects, are then fed into the decoder. Utilizing lane-wise interactions, the feature of the masked lanes would be recovered. Point-wise L1 loss is employed across all our experiments. The loss function of map reconstruction l_{map} is defined as the difference between the reconstructed lane and the ground-truth one.

Trajectory Reconstruction

Trajectory reconstruction is introduced to improve the model’s understanding of the trajectory features and the interactions between the trajectory and map elements, as shown in Fig. 2.2 (bottom-right). Since the synthetic scene only consists of one agent, only a single trajectory is masked in each scene. We retain its starting point in the masked object as in map reconstruction. To prevent modal collapse, the decoder is allowed to produce six potential trajectories. We select the one with the minimum loss to define the loss function l_{traj} . Additionally, to ensure that all modes are activated, we add the weighted reconstruction errors of the remaining modes to the loss function as regularization. A weight of 0.05 is used in our experiments.

Combination of Map and Trajectory Reconstructions

We integrate both map and trajectory reconstruction tasks by randomly selecting a percentage of scenes within a mini-batch to conduct either of the reconstructions. This selection is achieved by generating a random number prior to masking each scene. As a result, the same scene could undergo different reconstruction tasks across various epochs. The overall loss function is computed as the average of the individual scene-specific losses previously described.

2.4 Experiments

2.4.1 Experiment Setup

Synthetic Dataset. The pre-training dataset is generated in accordance with the methods outlined in Sec. 2.3.1. The Map is sourced from the HD maps and APIs in the Argoverse v1.1 Motion Forecasting Dataset [15]. We generate trajectories on both the original map and the augmented map. The resultant generated dataset contains 370k scenes, where 205k are generated on the original map and 165k are generated on the augmented map. Each generated scene contains the vehicle’s position spanning 5 seconds, following the format of the Argoverse v1.1 Motion Forecasting Dataset. As visualized in Fig. 2.4, the generated data align well with a vast spectrum of velocities present in the real-world data distribution.

Argoverse v1.1 Motion Forecasting Dataset. The Argoverse v1.1 Motion Forecasting Dataset [15] is a renowned dataset frequently employed in vehicle motion prediction studies. It consists of 324k scenes, each spanning 5 seconds and sampled at 10 Hz. In every scene, the initial 2 seconds serve as the history, while the subsequent 3 seconds are designated for prediction. Additionally, the dataset also includes HD maps collected from Pittsburgh and Miami, covering all the scenes in the dataset. There are 205k and 39k scenes in the training and validation sets, respectively, in which the validation set is used for evaluating our model.

Table 2.3: Method Comparison for Using Synthetic Data.

Method	Pretrain Task	$MR_6(\%)$ ($\downarrow$)	$minFDE_6$ ($\downarrow$)	$minADE_6$ ($\downarrow$)
Baseline	/	9.73	1.0673	0.8052
Dataset Augmentation	/	9.45	1.0567	0.7432
Supervised whole PT	Prediction	9.34	1.0371	0.7551
Supervised encoder PT	Prediction	9.20	1.0369	0.7639
Self-supervised PT	Reconstruction	9.24	1.0263	0.7384

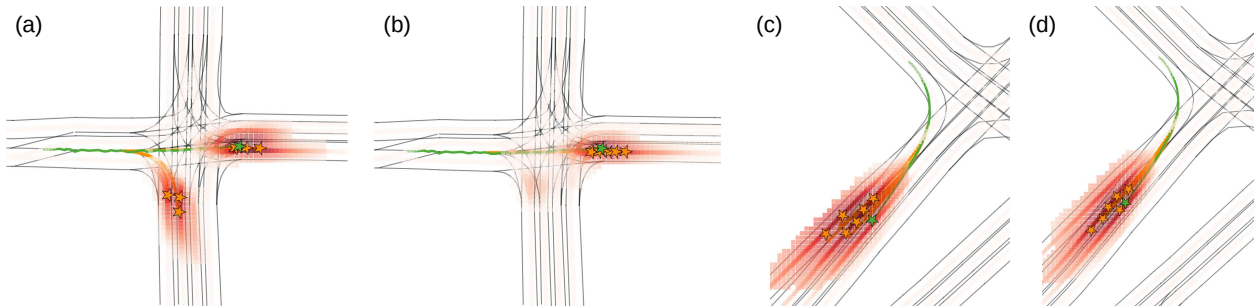


Figure 2.3: **Performance comparison without or with pre-training.** The green line and star indicate the true trajectory and its last point, while orange ones are the predicted ones. The orange background shows the possibility of each point being the predicted last point of the trajectory. (a) and (b) show the prediction results without or with trajectory pre-training. (c) and (d) illustrate the performance without or with map pre-training.

Evaluation Metrics. Following Argoverse’s benchmark, we use miss rate (MR_6), minimum final displacement error ($minFDE_6$), and minimum average displacement error ($minADE_6$) as the evaluation metrics. The subscripts mean that the minima are computed over six predicted trajectories.

Implementation Details. In our experiments, we adopt DenseTNT [55] as the prediction backbone. DenseTNT uses Vectornet [47] as its encoder which adopts a vectorized scene representation, i.e., the lanes and trajectories are represented as vectors, which makes the masking process straightforward. The pre-training is conducted on a Nvidia A6000 GPU with a batch size of 256. The learning rate is 10^{-5} for trajectory reconstruction and 10^{-3} for map reconstruction. For mixed map and trajectory reconstructions, we use a learning rate of 10^{-4} . Except for the GPU, all other fine-tuning settings are the same as in DenseTNT [55]. During evaluation, we choose the 100ms optimization for $minFDE_6$.

2.4.2 Comparison of Different Training Methods

We compare the performances of multiple methods described in Sec. 2.3.2. The results are shown in Tab. 2.3. While directly augmenting the training dataset can slightly improve

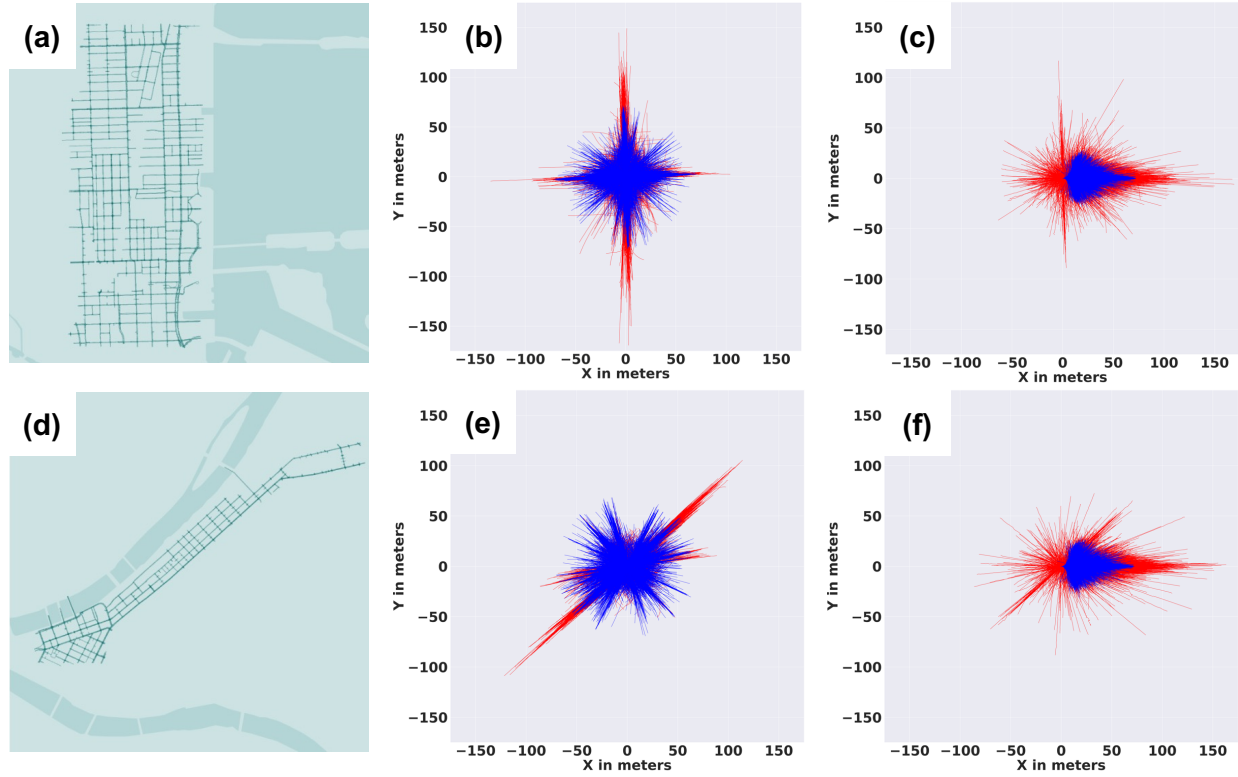


Figure 2.4: **Data distribution comparison between synthetic dataset (blue) and the real-world dataset (red) in terms of the speed and direction properties.** (a) represents the map representation for MIA city. (b) represents the trajectory distribution of scenes in (a), showing a pattern of divergence in velocity properties. (c) represents the trajectory distribution of scenes in (b) rotated to the same initial direction, demonstrating a pattern of divergence in direction properties. (d)(e)(f) are the counterparts of (a)(b)(c) in PIT city.

the prediction accuracy, it does not perform as well as pre-training approaches. The result is consistent with our motivation behind adopting pre-training methods. For supervised pre-training methods, we compared two alternative fine-tuning strategies: initializing the entire model with the pre-trained model or only initializing the *encoder* with the pre-trained one. We found that the decoder does not benefit from loading the parameters from the pre-trained model. Our hypothesis is that the domain-specific nuances learned by the decoder during supervised pre-training might not generalize well to real-world scenarios. It implies that the synthetic data is most suitable for improving representation learning rather than directly improving the entire backbone model for trajectory prediction. To this end, self-supervised pre-training should be the most efficient way to utilize the synthetic data, as the training tasks are deliberately designed for learning generalizable scene representation, which

Table 2.4: Performance of Self-Supervised Pre-Training.

Method	$MR_6(\%)$ ($\downarrow$)	$minFDE_6$ ($\downarrow$)	$minADE_6$ ($\downarrow$)
Baseline	9.73	1.0673	0.8052
Map Reconstruction	9.20 (-5.45%)	1.0343 (-3.09%)	0.7571 (-5.97%)
Trajectory Reconstruction	9.24 (-5.04%)	1.0263 (-3.84%)	0.7384 (-8.30%)
Combined Reconstruction	9.27 (-4.73%)	1.0349 (-3.04%)	0.7284 (-9.54%)

is supported by the experimental results—The self-supervised pre-trained model achieves the best prediction accuracy after fine-tuning.

2.4.3 Comparison of Self-Supervised Pre-training Tasks

We present the experimental results of different self-supervised pre-training tasks in Tab. 2.4. The results show that all of the self-supervised pre-training tasks can substantially improve the prediction accuracy compared with the baseline. Specifically, with map reconstruction, an MR_6 of 9.20% is observed, which is 5.04% lower than the baseline. Trajectory reconstruction pre-training helps improve the $minFDE_6$ by 3.84% compared to the baseline. Combining them together, when we allocate 70% scenes in a batch to map reconstruction and 30% scenes to trajectory reconstruction, we achieve a $minADE_6$ of 0.7284, marking an improvement of 9.54% against the baseline.

To elucidate the impact of our synthetic data and the pre-training, we provide qualitative examples in Fig. 2.3. Fig. 2.3 (a) and (b) contrast prediction outcomes without and with trajectory reconstruction pre-training. We observe that the model without pre-training predicts potential right-turn trajectories. However, since the target vehicle is not in the right-turn lane, its intention to go straight should be able to be identified, if the spatial relation between the historical trajectory and the lanes is captured by the encoder. In contrast, trajectory pre-training strengthens the model’s confidence in discerning the go-straight intention. Sub-figures (c) and (d) compare the prediction outcomes of the models without and with map reconstruction pre-training. In these sub-figures, the right-turn lane eventually merges with the 2nd lane from the right. We observe that with map pre-training, the model is able to more accurately predict that the vehicle will not merge into the 1st lane from the right. This result aligns with expectations—map reconstruction enhances the model’s understanding of lane connectivity, thereby improving prediction accuracy.

2.5 Discussion

In this section, we elaborate on some detailed design choices we made. In Sec. 2.5.1, we validate the necessity of map augmentation. In Sec. 2.5.2, we analyze the synthetic data distribution to further validate the necessity of self-supervised pre-training. In Sec. 2.5.3,

Table 2.5: Contributions of map augmentation.

Pretrain Dataset	$MR_6(\%)$	$minFDE_6$	$minADE_6$
Baseline	9.73	1.0673	0.8052
Original Map	9.40	1.0517	0.7694
Original+Augmented Map	9.20	1.0343	0.7571

we look into the information leak problem and validate our pre-training model architecture choice, which avoids information leak and effectively improve fine-tuned performance.

2.5.1 Benefits of Map Augmentation

We assess the performance of models pre-trained with and without scenes generated on the augmented maps. As map augmentation influences the distribution of map elements more significantly, map reconstruction is selected as the pre-training task here for evaluation. The model trained on data with augmented maps significantly outperforms the other, as shown in Tab. 2.5, underscoring the value of map augmentation in improving prediction models.

2.5.2 Properties of Synthetic Data

Fig. 2.4 illustrates the disparity between the *synthetic data* and *real-world data* distributions concerning velocity and direction. While our synthetic dataset encompasses a broader spectrum of velocity than the real-world data, it is important to note the presence of outliers representing extreme velocity cases within the real-world data. Additionally, the real-world dataset encompasses a multitude of scenarios with pronounced turning angles, such as U-turns, which pose challenges in terms of accurate generation within the synthetic dataset. The discrepancy in data distributions validates that self-supervised pre-training is necessary to mitigate the impact of distributional shifts.

2.5.3 Information Leak in Self-supervised Pre-training

In our work, we adopt a method similar to MAE [58], where the mask tokens are *not* passed to the encoder. While this eliminates the information leak from mask tokens, the encoder also loses the first point coordinates attached to masked objects. We, therefore, compare the prediction results both with or without information leak. In both scenarios, we ensure that the mask tokens pass through an equal number of transformer layers (i.e. one in our case) to exchange information. As shown in Tab. 2.6, we observe that the configuration without information leakage significantly outperforms its counterpart, affirming that the risk of information leakage outweighs the drawback of losing some positional data.

Table 2.6: Influence of Information Leak.

Pretrain Method	$MR_6(\%)$	$minFDE_6$	$minADE_6$
Baseline	9.73	1.0673	0.8052
No Info Leak Traj PT	9.24	1.0263	0.7384
W/ Info Leak Traj PT	9.45	1.0520	0.7581
No Info Leak Map PT	9.20	1.0343	0.7571
W/ Info Leak Map PT	9.84	1.0735	0.7614

2.6 Conclusion

To expand and diversify the scant motion data for trajectory prediction, we efficiently generate vehicle trajectory using both the original map and its augmented variant in this chapter. We evaluate strategies for optimizing data utilization and find out that self-supervised pre-training is the most effective strategy. With our synthetic data generation and self-supervised pre-training pipeline, the fine-tuned prediction model outperforms the baseline by a large margin. Our research offers a comprehensive pipeline for data generation and utilization, providing a promising direction to alleviate the data scarcity problem in trajectory forecasting.

Part II

Language Augmentation of Driving Data for Featuring Interactions

Chapter 3

WOMD-Reasoning: A Large-Scale Dataset for Interaction Reasoning in Driving

In addition to direct data augmentation, maximizing the utility of existing datasets offers a complementary solution to data scarcity. Conventionally, driving models exhibit a bias toward proximal interactions, often neglecting long-range dependencies. However, these distant interactions are critical, as they are frequently governed by explicit traffic rules and latent human intentions. Excitingly, language models uncover unprecedented abilities in analyzing driving scenarios, owing to their limitless knowledge accumulated from text-based pre-training. Naturally, they should particularly excel in analyzing rule-based interactions, such as those triggered by traffic laws, which are well documented in texts. However, such interaction analysis remains underexplored due to the lack of dedicated language datasets that address it. Therefore, we propose **Waymo Open Motion Dataset Reasoning (WOMD-Reasoning)**, a comprehensive large-scale Q&As dataset built on WOMD focusing on describing and reasoning traffic rule-induced interactions in driving scenarios. WOMD-Reasoning also presents by far the largest multi-modal Q&A dataset, with 3 million Q&As on real-world driving scenarios, covering a wide range of driving topics from map descriptions and motion status descriptions to narratives and analyses of agents’ interactions, behaviors, and intentions. To showcase the applications of WOMD-Reasoning, we design Motion-LLaVA, a motion-language model fine-tuned on WOMD-Reasoning. Quantitative and qualitative evaluations are performed on WOMD-Reasoning dataset as well as the outputs of Motion-LLaVA, supporting the data quality and wide applications of WOMD-Reasoning, in interaction predictions, traffic rule compliance plannings, etc.

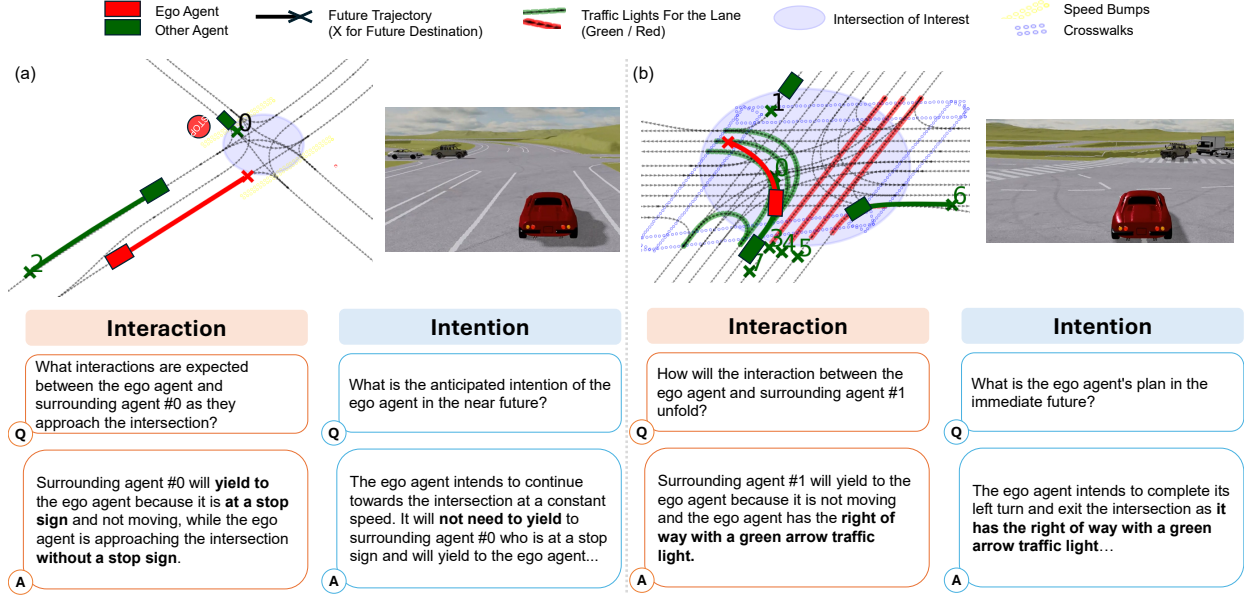


Figure 3.1: **Examples of traffic rule-induced interactions in WOMD-Reasoning dataset.** (a) captures the traffic rule-induced interaction between the ego agent and agent #0, attributing it correctly to the stop signs. (b) shows the traffic light-controlled yielding interaction between the ego agent and agent #1. The front-view visualization is created by MetaDrive simulator [102]

3.1 Introduction

Language-guided models improve explainability, controllability, and performance of autonomous driving tasks [177, 12, 20, 45, 124, 199, 99, 2, 132, 108, 30, 122, 207, 54, 35, 57, 187], owing to their enormous text-based knowledge acquired through the pre-training. To help language models elevate abilities on driving-related tasks, various language-based datasets [159, 139, 84, 152, 124, 71, 219, 173, 2, 132, 108, 122] have been proposed to fine-tune these LLMs. However, due to the complexity and variations in summarizing traffic rule-based interactions in real driving scenarios, existing datasets often focus on interactions based on spatial proximity. For example, in BDD-X [84], interactions are often limited to the spatial blocking, like ‘*The car moves back into the left lane because the school bus in front of it is stopping.*’; while in another example, DriveLM [159] attributes interaction to ‘*keep a safe distance*’ without further detailed explanations. Likewise, most existing works only cover interactions induced by proximity, as shown in our analysis in Table 3.1. While reasoning about many non-proximate interactions, like those caused by traffic rules, are vital for the safety-critical decision-making in autonomous driving systems [188, 212, 214, 37, 144], these interactions are rarely included in previous datasets, leading to suboptimal performance of LLMs fine-tuned on these datasets. Additionally, most datasets in driving are not large

enough to support multi-modal fine-tuning across diverse tasks such as scene description, prediction, and planning.

For the sake of interaction descriptions and reasoning, this work introduces WOMD-Reasoning dataset, a large-scale multi-modal dataset centered on language Q&As based on WOMD [41]. To incorporate interactions induced by traffic rules and human intentions, we build an automated data curation pipeline by prompting ChatGPT [12, 130] with a set of well-designed prompts. Together with a rule-based translator to convert initial motion data into language, we create an automated pipeline to generate the language dataset and achieve an approximate accuracy rate of 90%. WOMD-Reasoning dataset covers interactions induced by traffic rules and human intentions with 409k Q&As, like those shown in Figure 3.1. As indicated in Table 3.1, WOMD-Reasoning contains the highest number of total Q&As and interaction-specific Q&As, supporting a wide range of applications, including scene description, prediction and planning. Figure 3.2 further demonstrates its uniqueness of interaction descriptions in WOMD-Reasoning with vocabulary statistics. In addition to comprehensive and in-depth interactions, WOMD-Reasoning is also the largest language dataset for real-world driving data, to the best of our knowledge and by its publication. Additionally, to facilitate multi-modal driving-related tasks like training end-to-end driving models, in addition to the LiDAR [19] info provided by Waymo, we render both the corresponding bird’s eye view and front view videos of covered driving scenarios by the MetaDrive simulator [102], which are attached to the dataset.

To showcase the applications of WOMD-Reasoning, we fine-tune our Motion-LLaVA, which is built upon LLaVA [113], on WOMD-Reasoning. With careful design and a set of effective training and inference strategies ablated, Motion-LLaVA achieves high accuracy in both driving scenario understanding and interaction analysis, which unequivocally advocate the effectiveness of WOMD-Reasoning in boosting the interaction prediction abilities of driving language models.

Our contributions can be summarized as follows:

- We propose WOMD-Reasoning, the largest multi-modal dataset with 3 million Q&A pairs centered on interaction reasoning in driving. It provides extensive insights into critical but previously overlooked interactions induced by traffic rules and human intentions. Quantitative and qualitative evaluations verify its wide coverage in interaction descriptions and reasoning, as well as its accuracy.
- Fine-tuned with WOMD-Reasoning, our Motion-LLaVA is capable of providing detailed and insightful interaction predictions on various driving scenarios, supporting the effectiveness of WOMD-Reasoning. Extensive evaluations with various metrics validate the wide coverage and high quality of the answers of Motion-LLaVA fine-tuned on WOMD-Reasoning, ranging from driving scenario descriptions to traffic rule-compliant planning.
- To optimize the performance of Motion-LLaVA, a motion-language model designed for WOMD-Reasoning with robust capabilities, we benchmark task performance across

various configurations, including input modalities, reasoning techniques, and network architectures.

Table 3.1: **Comparison between WOMD-Reasoning Dataset and Previous Real-world Language Datasets for Driving.** WOMD-Reasoning covers more comprehensive kinds of interactions, with a significantly higher size supporting a wide range of applications.

Dataset	Data Source	Statistics			Interactions			Applications		
		Total Scenes	Total Q&As	Interaction Q&As	Distance-induced	Traffic Rule-induced	Human Intention-induced	Scene Descriptions	Motion Prediction	Motion Planning
nuScenes-QA[139]	nuScenes[13]	34k	460k	0	✗	✗	✗	✓	✗	✗
BDD-X[84]	BDD[198]	26k	26k	26k	✓	✗	✗	✓	✗	✓
DriveLM[159]	nuScenes[13]	5k	443k	~199k	✓	✗	✗	✓	✓	✓
Rank2Tell[152]	Rank2Tell[152]	116	>116	>116	✓	✓	✓	✓	✓	✓
DRAMA[124]	DRAMA[124]	18k	102k	<18k	✓	✓	✗	✓	✗	✓
NuInstruct[36]	nuScenes[13]	850	91k	<46k	✓	✗	✗	✓	✓	✓
nuCaption[201]	nuScenes[13]	<1k	420k	~140k	✓	✗	✗	✓	✗	✓
Tod3Cap[77]	nuScenes[13]	850	~2,300k	0	✗	✗	✗	✓	✗	✗
WOMD-Reasoning	WOMD [41]	63k	2,940k	409k	✓	✓	✓	✓	✓	✓

3.2 Related Work

3.2.1 Application of Language Modality in Autonomous Driving.

In recent years, language has been widely explored in autonomous driving, as incorporating language into implicit features could enhance the explainability [202, 145, 169, 18, 125, 211]. For example, Kuo *et al.* [93] employ language features in the trajectory prediction model, ensuring both explainability and consistency between language and predictions. Similarly, Zhong *et al.* [217] use language input and LLMs to generate conditions in diffusion models, thereby controlling the generation of driving scenarios. Recently, general driving agents have been developed [199, 20, 159, 126], aiming to integrate all functions—perception, prediction, and planning—into a single language agent. However, we observe that their performance in interaction analysis is not entirely satisfying, especially in non-proximal interactions like those induced by traffic rules. This shortcoming is largely attributed to the lack of interaction analysis in language datasets specific to driving scenarios.

3.2.2 Language Datasets in Autonomous Driving.

Language datasets for driving scenarios have recently been developed to support related LLM-based work. Due to the controllability of simulations, several simulation-based datasets have been created [20, 159]. However, simulations often fail to capture real-world interactions. To address this, recent studies [84, 152, 124, 186] incorporate human labeling of real-world driving scenarios to ensure the inclusion of genuine interactions. Nevertheless,

obtaining high-quality human labels is labor-intensive and costly, limiting the size and coverage of these datasets. Moreover, interactions driven by traffic rules and human intentions have often been overlooked. To streamline the labeling process, some studies [33, 127, 139, 159] have generated rule-based labels, although these typically cover only basic language elements like scene descriptions, leaving interaction analysis to human labelers. To minimize human labor, we first employ manual rules and then utilize ChatGPT-4 [130] to build our dataset automatically. The resulting dataset, WOMD-Reasoning, is rich in interaction details, particularly those stemming from traffic rules and human intentions. The designed automated data-curation pipeline enables the dataset to be significantly larger. We also create a high-quality human-verified subset. Preliminary human evaluation provides positive feedback with an approximate accuracy of 90%.

3.2.3 Fine-tuning LLM for Driving Tasks.

Recent years have witnessed a surge of research on leveraging LLMs and vision-language models (VLMs) for driving applications [156, 20, 76, 200, 159, 158, 125, 172]. However, handling heterogeneous inputs such as vectorized motion representation remains challenging due to their domain gaps to most existing VLMs and LLMs. Notable prior work, such as [125], designed natural language prompt structures to accommodate numerical motion data, while another approach [20] proposed training LLMs with custom-built motion vector encoders.

Our Motion-LLaVA approach is designed for benchmarking the performance of WOMD-Reasoning. Inspired by the design of [20], Motion-LLaVA augments existing LLM architectures with tailored components. We employ a motion prediction model as a motion vector encoder, utilizing its prior knowledge to streamline the training pipeline. To enhance reasoning accuracy, we adopt a Chain-of-Thought (CoT) [191] approach, allowing the model to base its reasoning on factual grounding, mitigating potential hallucinations. Based on WOMD-Reasoning, our Motion-LLaVA demonstrates the practicality of applying VLMs to real-world vectorized motion data to accurately answer diverse driving-related questions correctly.

3.3 Method

In this section, we first introduce how to build the WOMD-Reasoning dataset. To verify the effectiveness of WOMD-Reasoning, we introduce Motion-LLaVA, a multi-modal fine-tuning approach, to build motion language models upon WOMD-Reasoning. The Motion-LLaVA method as well as the fine-tuning process are also included here.

3.3.1 Building WOMD-Reasoning Dataset

Manually constructing autonomous driving datasets with fine-grained natural-language labels regarding the interactions among road participants would involve intense human labor, which is one of the main reasons why previous datasets have insufficient interaction analysis and a very limited overall size [153]. To reduce human labor while obtaining reasonably useful data on vehicle interactions, we propose a fully automatic data-curation pipeline to label the WOMD dataset with language: We first develop a rule-based program to interpret the motion dataset, which contains trajectories and the HD map, into language; and then build a set of prompts to utilize the reasoning abilities of ChatGPT-4 [130] to generate interaction analysis and reasoning, and organize the results into the target Q&A format. Details of each step are attached in the Appendix A.1. Codes for interpreting the motion dataset as well as the full prompts for ChatGPT-4 are available in supplemental materials.

Our analysis is performed on Microsoft Azure ChatGPT-4-Turbo API, which costs around 12,750 USD. The details of WOMD-Reasoning will be provided in Sec. 3.4.

3.3.2 Fine-tuning Multi-modal Model on WOMD-Reasoning

To showcase the application of WOMD-Reasoning dataset, we use our Motion-LLaVA approach to fine-tune upon LLaVA [113]. LLaVA is chosen to accommodate motion data and text prompts input at the same time, with the vision encoder adapted to a motion data encoder. Inspired by [158], we utilize encoders from motion prediction models MultiPath++ [179] for encoding motion data. Ablations in Table A.9 prove the usefulness of this encoder. Some detailed fine-tuning strategies are in Appendix A.10. Details on how we process agent IDs in Motion-LLaVA are listed in Appendix A.3. The overall Motion-LLaVA pipeline is illustrated in Figure 3.4.

During training, we unfreeze all components including the motion vector encoder, and train on all Q&A pairs in WOMD-Reasoning simultaneously to avoid the potential catastrophic forgetting. To prevent Motion-LLaVA from forgetting its inherent LLM’s knowledge, we mix up a few questions irrelevant to the current driving scenario into the training process [208]. During inference, we adopt Chain-of-Thought (CoT) Reasoning strategy [191] to reduce the hallucination. Specifically, we first prompt Motion-LLaVA to answer the **Factual** questions, which include questions in map environment, ego, and other agents’ motion status. Then, we aggregate these answers and feed them into Motion-LLaVA as the contexts for answering **Reasoning** questions in interactions and intentions analysis. Examples of these contexts can be found in the Appendix A.8. We ablate in Sec. A.10 that such a CoT strategy can effectively improve the quality of answering reasoning questions.

Specifically, we take LLaVA-v1.5-7b [113] as the pre-trained VLM. Since MultiPath++ [179] did not release their codes nor checkpoints, we take the implementations by [88]. Our multi-modal fine-tuning takes 2 GPU days (~ 1 day on 2xNVIDIA A6000 GPUs) to train 1 epoch on the entire training set of WOMD-Reasoning. For quantitative evaluations, we randomly select $\sim 1,000$ QA pairs from the validation set of WOMD-Reasoning.

To benchmark Motion-LLaVA model, we utilize a suite of standard language metrics, including BLEU-4 [131], METEOR [8], CIDEr [180], SPICE [1], and ROUGE-L [109], along with GPT Score [140] to assess the quality of the generated content. Details on the prompts used to calculate the GPT Score can be found in Appendix A.5.

3.4 WOMD-Reasoning Dataset Specifications

In this section, we provide detailed information about WOMD-Reasoning. We first present the organization and statistics of the dataset, then provide quality analysis based on case studies and preliminary human evaluations. Finally, we compare our dataset to previous ones to show the strengths of WOMD-Reasoning in diverse interactions, huge size, and wide potential applications.

3.4.1 Dataset Statistics

Table 3.2 provides the statistics of Q&As in WOMD-Reasoning dataset, as well as in every Q&A class. Our Q&As can be classified into two parts: The **Factual** contents and the **Reasoning** contents, where the first part serves as the context for the second part. The factual contents contain three topics: *Map Environment*, which includes the existence and category of intersections, existence and places of stop signs and crosswalks, and count of lanes, etc. *Ego Agent’s Motion Status* and *Other (Surrounding) Agents’ Motion status*, which describe each involved agent’s speed, acceleration, direction, related traffic light and sign, and relative position to the intersection center as well as to the ego agent. The factual descriptions alone are also good for training language scene describers.

Our key part, the Reasoning contents, first comes with Q&As on *Interactions*, which includes a summary of interactions between each surrounding agent and the ego agent and the reasoning behind such interactions. Then by summing up all interaction information, we offer the *Intentions*, which give the predicted intention of the ego agent, considering its responses to all the interactions. The interaction and intention reasoning parts provide an unprecedented tool for training language-based prediction and planning agents.

As shown in Table 3.1, WOMD-Reasoning contains **346k** unique traffic rule-induced and intention-induced interaction Q&As, as well as **63k** comprehensive intention predictions of the ego agent containing responses to the interactions, which rarely exist in previous datasets. We will show these features through examples and statistics in the following section. Furthermore, we also claim that to the best of our knowledge, WOMD-Reasoning is the largest real world language data set for driving, since our WOMD-Reasoning provides roughly **3 million** Q&A pairs for **63k** scenes in WOMD, which are 6 – 113 times greater than those of other datasets we compare to.

Compared with existing data sets, as shown in Table 3.1, our WOMD-Reasoning is the **largest** real-world dataset in terms of total scenes, total Q&As, and the total interactions included. Also, WOMD-Reasoning supports Q&As in scene descriptions, prediction, and

Table 3.2: Sizes of WOMD-Reasoning dataset.

Set	Map	Ego Agent	Other Agents	Interactions	Intentions	Total	Scenes
Training	188k	268k	1,635k	287k	52k	2,430k	52k
Validation	41k	58k	341k	59k	11k	510k	11k
Total	229k	326k	1,976k	346k	63k	2,940k	63k
Per Scene Ave.	3.58	5.09	30.88	5.41	0.98	45.95	-

Table 3.3: Human Evaluations of WOMD-Reasoning Dataset.

Q&A Class	All	Interactions	Intentions
Accuracy	91.99%	91.03%	87.50%

planning. Together with the simulated video of each covered scenario, it is suitable for training VLMs to perform nearly all major tasks in autonomous driving. Diving even deeper into the interaction reasoning in these datasets, we summarize that most interactions covered in previous datasets are caused by a very near distance. However, as we show in Sec. 3.4.3, many interactions happen when the two counterparts are initially far away, but they establish interactions by following the traffic rules or by human intentions. Our WOMD-Reasoning contains a wealth of this interaction information, as shown in Figure 3.2, the vocabulary statistics of the interaction and intention part of our dataset.

3.4.2 Interaction Contents

To testify to the quality of WOMD-Reasoning, we show examples from it containing rich traffic rule-induced and human intention-induced interactions.

WOMD-Reasoning contains traffic rule-induced interaction information. Firstly, in Figure 3.1 (a), we show a traffic scenario of an intersection controlled by the stop sign. In this case, the agent #0 is quite far away from the ego agent at the starting (current) moment, thus their interactions would rarely be considered in previous datasets, as they mainly cover distance-induced interactions. However, the two agents do have a significant interaction: Controlled by the stop signs, the agent #0 has to yield to the ego agent, even though the ego agent is still far from the intersection. This case vividly proves the remarkable importance of the traffic rule-induced interactions we cover.

Another example is a traffic light-controlled intersection shown in Figure 3.1 (b). Similar to the previous one, agent #0 is initially not close to the ego agent. However, thanks to our analysis of traffic rules, our language dataset effectively captures the interaction that the agent #1 would yield to the ego agent who is turning left and having right-of-way due to

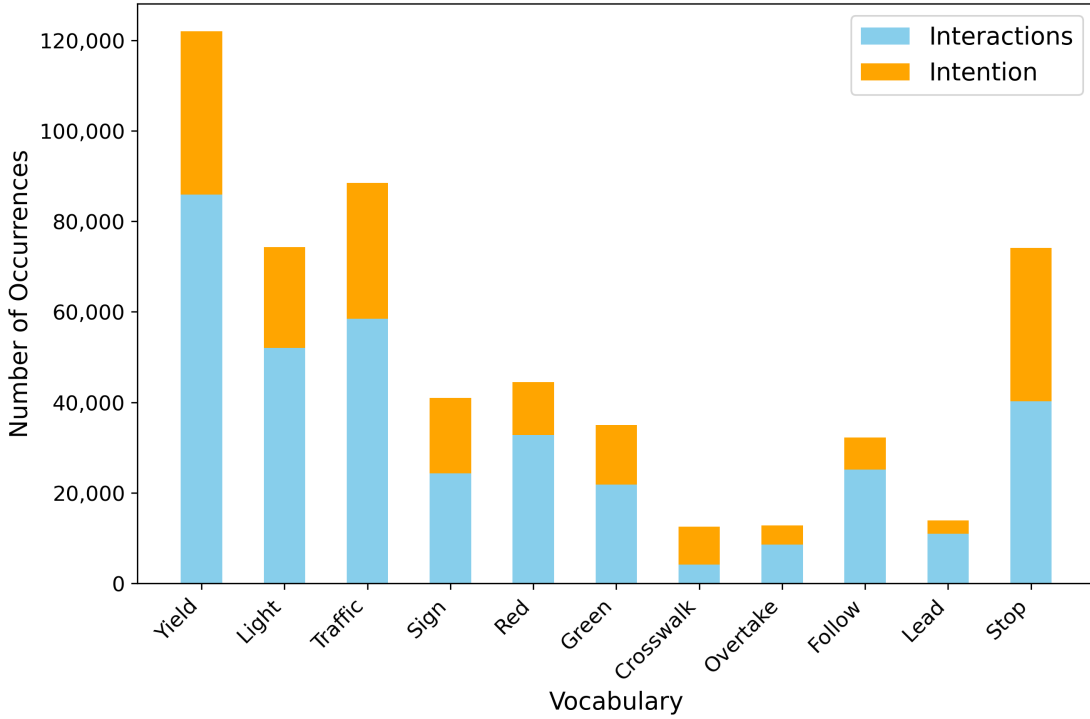


Figure 3.2: **Selected vocabulary statistics in WOMD-Reasoning.** We stat vocabularies strongly related to traffic rule-induced and human intention-induced interactions in WOMD-Reasoning, illustrating that it contains abundant such interaction descriptions and reasoning.

an arrow green light. This further proves that traffic rules are essential in determining the interactions.

WOMD-Reasoning contains human intention-induced interaction information. Beside traffic rules, our dataset also covers human intention-induced interactions. We provide an additional example to show this, as well as to demonstrate a whole picture of each section of the dataset. In Figure 3.3, WOMD-Reasoning provides fruitful information on the human intention-induced interaction - overtaking, based on pattern recognition abilities built in the GPT. We also show 1-2 Q&As for each category of the dataset for completeness.

3.4.3 Dataset Quality Evaluation by Human

To grab a sense of the accuracy of WOMD-Reasoning, we execute a human evaluation process. Consent was obtained from all the people interviewed. We ask interviewers to evaluate whether the correct answer to the question is included in the answer provided in

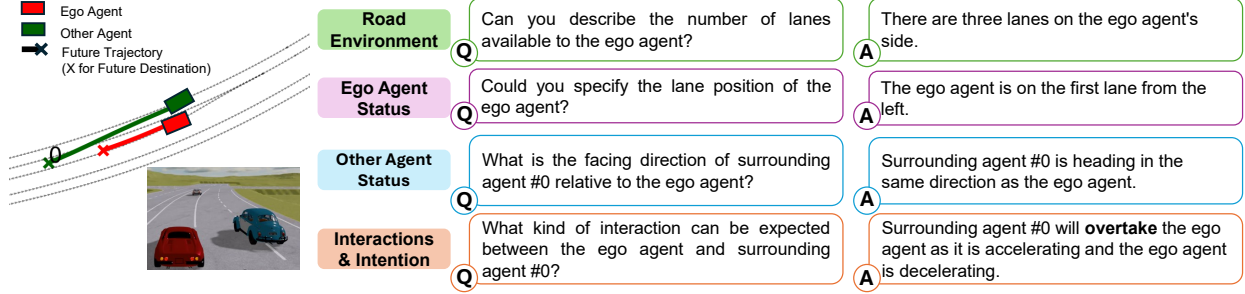


Figure 3.3: **A demonstration of Q&As in each part of WOMD-Reasoning dataset.** We show Q&As in all categories regarding the scenario while demonstrating language analysis of overtaking, a human intention-induced interaction in WOMD-Reasoning dataset.

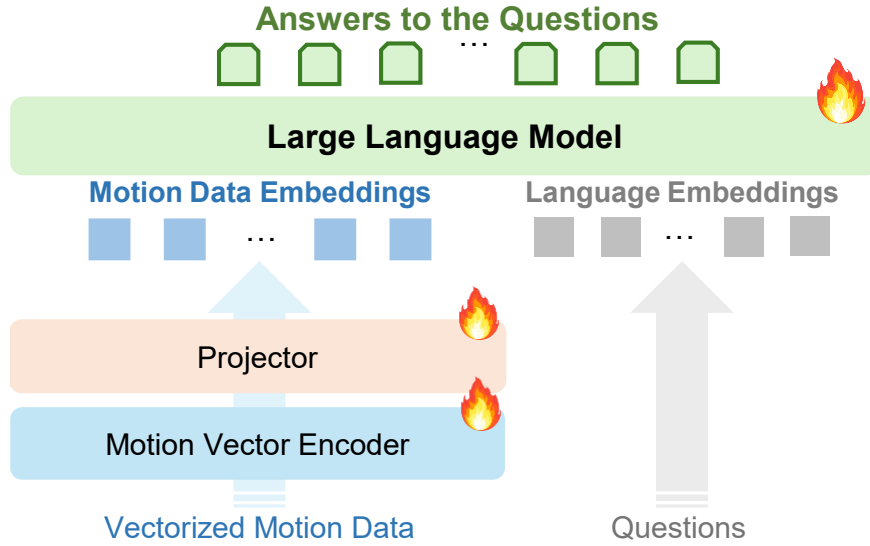


Figure 3.4: **The Motion-LLaVA pipeline fine-tuning a multi-modal model with WOMD-Reasoning.** Motion data go through pre-trained motion vector encoders from Multipath++ [179] and a projector layer to serve together with the questions in WOMD-Reasoning as the inputs. The answers in WOMD-Reasoning serve as the supervision.

the dataset. We invite 4 people to judge 1,610 Q&As in total. The results are shown in Table 3.3. We conclude that WOMD-Reasoning achieves a satisfying accuracy, especially considering it is automatically generated. Further human-based data cleaning is ongoing.

3.4.4 Vision Extension with Simulations

To support applications requiring visual information, we provide an API to extend WOMD-Reasoning scenario descriptions to the vision domain, compensating for the absence of raw camera data in the original WOMD dataset. We achieve this by utilizing an open-source traffic scenario simulation platform, namely ScenarioNet [101], and employing the MetaDrive [102] simulator to replay WOMD-Reasoning scenarios in both Bird’s Eye View (BEV) layout and ego-view images through 3D rendering. This approach enables visual inputs to support the application of WOMD-Reasoning for future research in the vision and language domain. For video generation, we produce a 90-frame 10 Hz video aligned with Waymo’s trajectory data, including 10 frames of historical footage and 80 frames of future trajectory. Users may choose to use the historical portion for prediction tasks.

3.5 Dataset Application and Evaluation with Motion-LLaVA

To demonstrate a real-world application of WOMD-Reasoning, we design Motion-LLaVA, a multi-modal model taking motion data as input, and fine-tune it on WOMD-Reasoning. We show that the model gains significant abilities from WOMD-Reasoning in predicting interactions, especially those caused by traffic rules, and in planning right-of-way compliant route for the ego. Furthermore, its abilities extend to answer various questions related to driving across factual and reasoning categories, which is supported by both qualitative showcases and quantitative evaluations on language metrics. These highlight the broad applications of WOMD-Reasoning for the autonomous driving industry.

3.5.1 Interaction Prediction

Since WOMD-Reasoning is proposed to focus on the interaction analysis, one of the key abilities of Motion-LLaVA fine-tuned on it is to predict interactions, especially the traffic rule-induced interactions. We present a few interaction prediction examples answered by Motion-LLaVA in Figure 3.5. These results confirm that by fine-tuning on WOMD-Reasoning, Motion-LLaVA unleashes the potential of language models for predicting traffic rule-related interactions in unseen real-world scenarios.

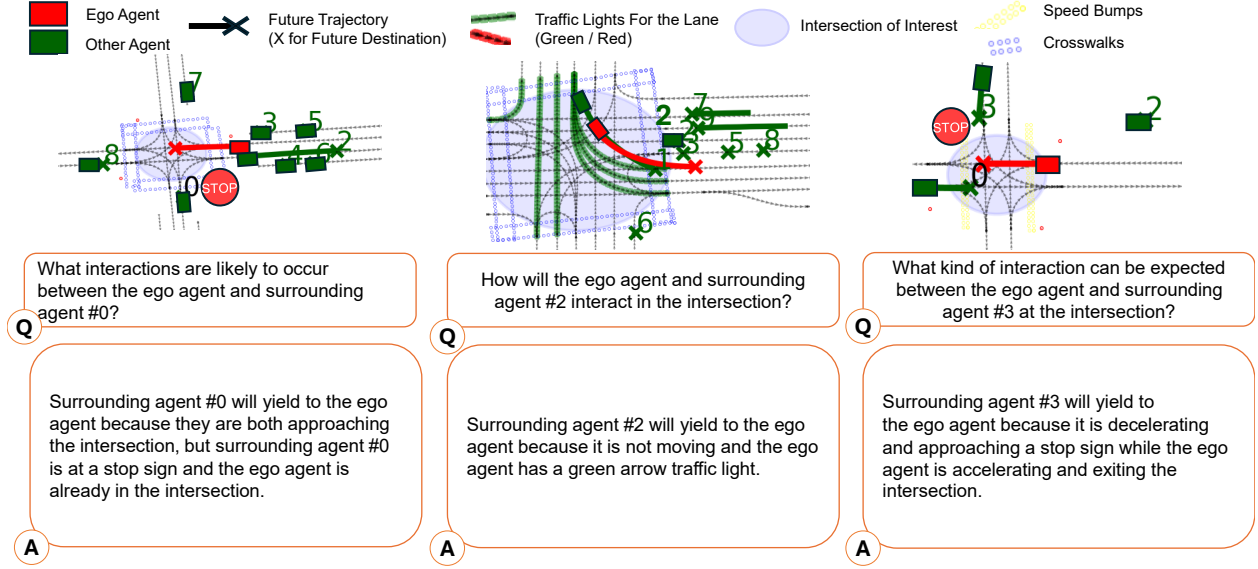


Figure 3.5: Interaction predictions made by Motion-LLaVA on various WOMD scenarios.

3.5.2 Traffic Rule Compliant Planning

Based on thorough interaction analysis, traffic rule compliant planning can naturally be achieved by summarizing interactions, as we do in producing ego agents' intentions in WOMD-Reasoning. Motion-LLaVA also perfectly learns this ability. In Figure 3.6, we demonstrate that Motion-LLaVA fine-tuned on WOMD-Reasoning can offer traffic rule-compliant future trajectory plans, which clearly extends the applications of WOMD-Reasoning into planning.

3.5.3 Answering Various Driving-related Questions

We further present a set of Q&A examples answered by Motion-LLaVA covering each category of WOMD-Reasoning in one specific driving scenario, shown in Figure 3.7. We observe that Motion-LLaVA model is capable of providing accurate narratives and analysis on diverse driving-related contents thanks to the wide coverage of WOMD-Reasoning.

3.5.4 Validation of Motion-LLaVA

To validate the effectiveness of Motion-LLaVA and the fine-tuning process, we present evaluation metrics of all generated answers, as well as those in the two categories, i.e. the **Facts** which include the road environment and agents' motion status, and the **Reasonings** which cover analysis on interactions and intentions. Benchmark results are shown in Table

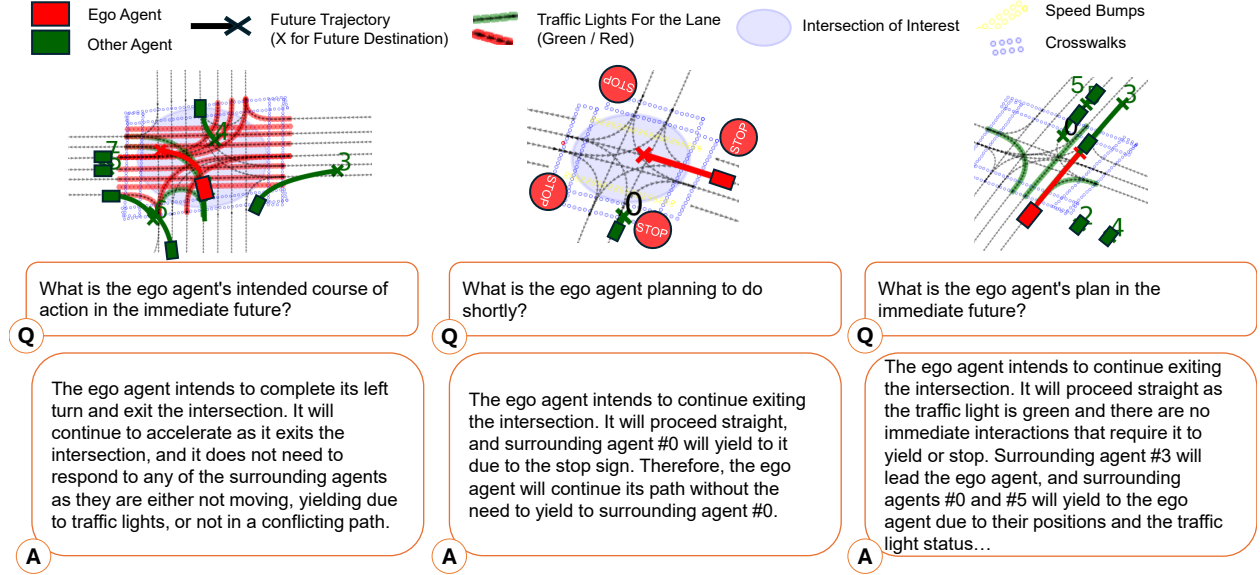


Figure 3.6: Traffic rule-compliant planning made by Motion-LLaVA on various WOMD scenarios.

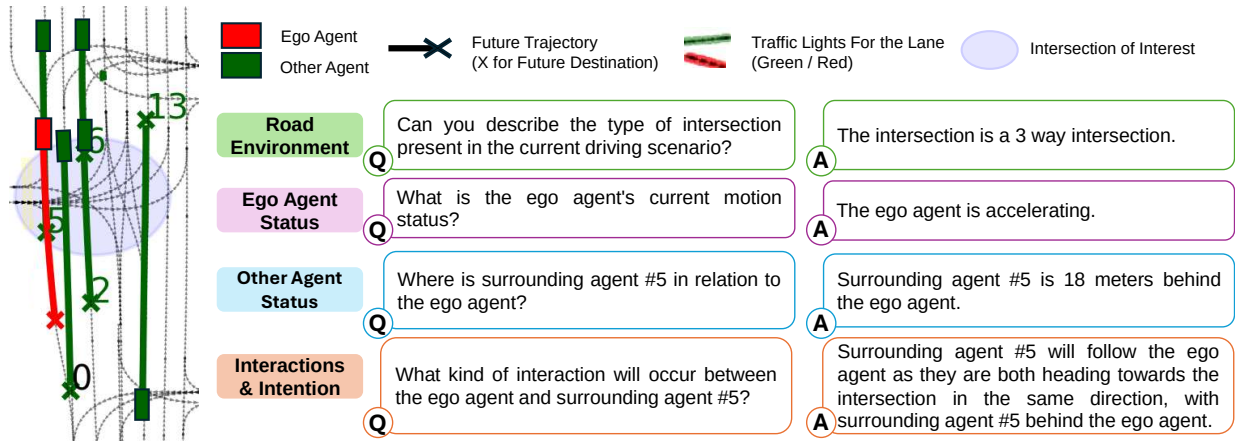


Figure 3.7: An example set of answers to diverse driving-related questions generated by Motion-LLaVA on a WOMD scenario. We see that Motion-LLaVA is capable of answering all topics of questions covered by WOMD-Reasoning accurately.

Table 3.4: Benchmarks of fine-tuned multi-modal models with language metrics.

Questions	Model	ROUGE (↑)	BLEU (↑)	METEOR (↑)	CIDEr (↑)	SPICE (↑)	GPT Score (↑)
All	Shuffled Ground-truths	0.481	0.311	0.263	1.68	0.375	1.48
	Non-Fine-tuned LLaVA [113]	0.512	0.211	0.275	1.36	0.455	2.31
	Fine-tuned LLaVA [113]	0.779	0.581	0.439	5.51	0.735	6.88
	Motion-LLaVA (Ours)	0.792	0.616	0.449	5.69	0.744	7.02
Facts Only	Shuffled Ground-truths	0.528	0.349	0.273	2.26	0.401	2.06
	Non-Fine-tuned LLaVA [113]	0.561	0.322	0.326	1.50	0.483	2.46
	Fine-tuned LLaVA [113]	0.821	0.702	0.495	6.05	0.783	6.91
	Motion-LLaVA (Ours)	0.840	0.736	0.516	6.35	0.794	7.09
Reasonings Only	Shuffled Ground-truths	0.479	0.326	0.281	1.81	0.462	1.87
	Non-Fine-tuned LLaVA [113]	0.262	0.097	0.201	0.06	0.283	0.91
	Fine-tuned LLaVA [113]	0.596	0.452	0.357	2.26	0.557	6.14
	Motion-LLaVA (Ours)	0.614	0.474	0.366	2.52	0.571	6.76

3.4. We first verify the effectiveness of these language metrics with shuffled ground-truths, which attach shuffled true answers to questions randomly. Results confirm that low scores are granted for this model, indicating that a similar structure with wrong answer content would not be able to cheat our metrics. Then we benchmark the off-the-shelf LLaVA [113], where BEV plots are used as the vision input for the driving scenario. This non-fine-tuned version earns a very low score in all metrics, confirming that LLaVA itself cannot answer driving-related questions correctly. Then, we perform fine-tuning on WOMD-Reasoning. We compare two models with different modalities as motion input: LLaVA using the BEV plot of the driving scene as the vision input [113], and Motion-LLaVA which uses motion data as input. We observe that both fine-tuned multi-modal models significantly outperform previous baselines across all metrics and question types, underscoring the value of fine-tuning on WOMD-Reasoning for addressing driving-related queries. Compared to the fine-tuned LLaVA, Motion-LLaVA achieves superior performance across all metrics and question types, highlighting the advantages of utilizing the motion data as input. We hypothesize that this performance gap is due to the information loss that occurs when converting motion data into a BEV plot with a scale bar. We also visualize and compare the answers of Motion-LLaVA and these baselines on the same driving scene in Appendix A.6.

In addition to language metrics, we also provide objective metrics to the answers to questions regarding **Facts**. We report the root median squared error (Median Error), and accuracy within a 1.0-meter tolerance ($\text{Acc}_{1.0}$) in Table 3.5, following similar metrics used in [76]. Results confirm that Motion-LLaVA is capable of providing numerically accurate answers to factual questions.

Table 3.5: Accuracy of Motion-LLaVA answers to factual questions.

Model	$\text{Acc}_{1.0}$ (↑)	Median Error (↓)
Fine-tuned LLaVA [113]	10.9%	5.0
Motion-LLaVA (Ours)	56.2%	1.0

3.6 Dataset Effectiveness in Vehicle Trajectory Prediction

In this section, we evaluate the effectiveness of WOMD-Reasoning in a real-world driving task - vehicle trajectory prediction. Our experiments indicate that with our WOMD-Reasoning dataset involved, the metrics of vehicle trajectory predictions are significantly improved.

3.6.1 Language-assisted Trajectory Prediction Experiment Setup

We adapt the Multipath++ [179] trajectory prediction model to accommodate the language input. The adaption mainly contains two steps:

Language Acquisition and Encoding The language we use comes from Motion-LLaVA’s output. We run Motion-LLaVA on the cases covered by WOMD-Reasoning dataset in the training and validation set of WOMD. For each scenario included, Motion-LLaVA would provide a comprehensive set of Q&As, where we pick the interaction Q&As for this experiment. Each of these Q&As includes the interaction info between a specific agent and the ego agent. We pick the answer parts of these Q&As to use. These answers are then fed into a T5 encoder [142] followed by a few MLP layers to be encoded. In the end, for each scenario included, we have a set of language embeddings, each contains info on the interaction between the ego agent and one specific agent.

Introducing Languages into Multipath++ Our next step is to introduce these encoded languages into each agent’s feature. For Multipath++, we use MPA [88], an open-source Multipath++ implementation, as our code base. Our language introduction module is inserted right after their "Other Agents History Encoder", and before their "MCG (multi-context gating) encoder". Our module is a cross-attention block, which takes each agent’s history encoding as queries, letting them cross-attend to their corresponding language embeddings containing info about the interaction between the ego agent and that specific agent. In this way, the interaction information in Motion-LLaVA is introduced into the features of each agent.

3.6.2 Enhancing Trajectory Prediction with Motion-LLaVA Outputs

As shown in Table 3.6, we observe significant metric improvements with language involved, which strongly supports WOMD-Reasoning’s ability to help downstream tasks like predictions. It can also help to understand the driving behaviors, and to improve explainability of vehicle trajectory prediction models.

Table 3.6: Benefits of Motion-LLaVA Outputs for Prediction Task.

Model	$minFDE_6$	MR_6
Multipath++ [88]	1.27	12.59
+ Motion-LLaVA Outputs	1.18	11.69
Δ	-7.35%	-7.10%

3.7 Conclusion

We first build WOMD-Reasoning, a language-centered multi-modal dataset upon WOMD, focusing on analyzing and reasoning interactions. Specifically, our WOMD-Reasoning emphasizes interactions induced by traffic rules and human intentions, such as interactions governed by traffic lights and signs, and interactions like overtaking and following, which are both rarely covered in previous datasets. Besides, WOMD-Reasoning provides 3 million Q&A pairs, which makes it the largest real-world language dataset in autonomous driving to the best of our knowledge, covering a range of topics from scene descriptions, prediction, to planning. To demonstrate the applications of WOMD-Reasoning, we propose Motion-LLaVA, which fine-tunes a multi-modal model to accurately answer questions covering both driving scenario descriptions and interaction reasonings. Motion-LLaVA realizes the function of interaction prediction and traffic rule-compliant planning, taking advantages of the WOMD-Reasoning and the text-based knowledge intrinsically built in language models. Motion-LLaVA proves the effectiveness of WOMD-Reasoning in assisting real-world driving through supporting various language-involved downstream tasks. Our WOMD-Reasoning and Motion-LLaVA frameworks distill latent semantic information from driving data, particularly interactions governed by traffic rules or human intentions, into explicit natural language representations. These linguistic descriptors are subsequently demonstrated to be effective in enhancing the performance of autonomous driving systems, such as motion prediction models.

Part III

Efficient Training of Vision-based Driving Data Generation Models

Chapter 4

Immiscible Diffusion: Accelerating Diffusion Training with Noise Assignment

Recent advancements in diffusion-based models [65, 162, 111] have paved the way for the large-scale generation of vision-based driving data. However, the training of these models is characterized by exorbitant computational costs and prolonged convergence times, which significantly impedes their practical scalability for driving data synthesis. In this chapter, we point out that suboptimal noise-data mapping leads to slow training of diffusion models. During diffusion training, current methods diffuse each image across the entire noise space, resulting in a mixture of all images at every point in the noise layer. We emphasize that this random mixture of noise-data mapping complicates the optimization of the denoising function in diffusion models. Drawing inspiration from the immiscibility phenomenon in physics, we propose **Immiscible Diffusion**, a simple and effective method to improve the random mixture of noise-data mapping. In physics, miscibility can vary according to various intermolecular forces. Thus, immiscibility means that the mixing of molecular sources is distinguishable. Inspired by this concept, we propose an assignment-then-diffusion training strategy to achieve *Immiscible Diffusion*. As one example, prior to diffusing the image data into noise, we assign diffusion target noise for the image data by minimizing the total image-noise pair distance in a mini-batch. The assignment functions analogously to external forces to expel the diffuse-able areas of images, thus mitigating the inherent difficulties in diffusion training. Our approach is remarkably simple, requiring only **one line of code** to restrict the diffuse-able area for each image while preserving the Gaussian distribution of noise. In this way, each image is preferably projected to nearby noise. To address the high complexity of the assignment algorithm, we employ a quantized assignment strategy, which significantly reduces the computational overhead to a negligible level (*e.g.* 22.8ms for a large batch size of 1024 on an A6000). Experiments demonstrate that our method can achieve up to $3\times$ faster training for unconditional Consistency Models on the CIFAR dataset, as well as for DDIM and Stable Diffusion on CelebA and ImageNet dataset, and in class-conditional training

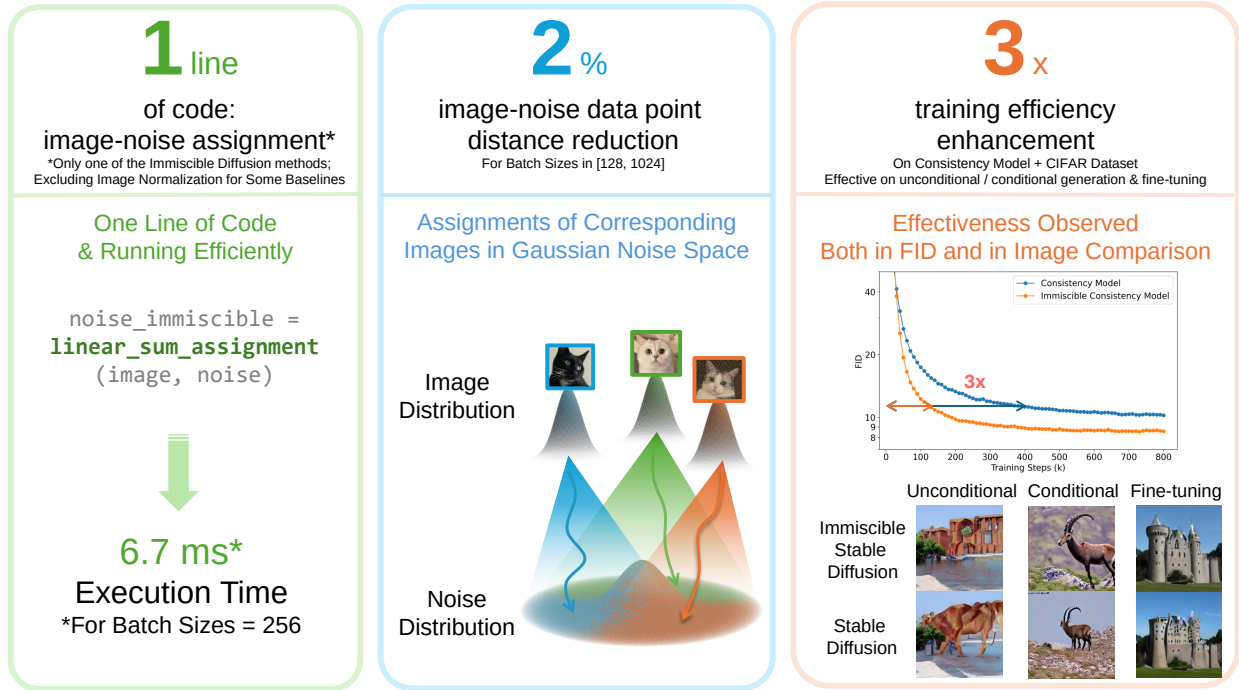


Figure 4.1: **Immiscible Diffusion** can use a single line of code to efficiently achieve immiscibility by re-assigning a batch of noise to images. This process results in only a 2% reduction in Euclidean distance post-assignment, leading to up to 3 $\times$ increased training efficiency on top of the Consistency Model [166] for CIFAR Dataset [89]. Additionally, Immiscible Diffusion significantly enhances the image quality of Stable Diffusion [149] for both unconditional and conditional generation tasks, and for both training from scratch and fine-tuning training tasks, on ImageNet [31] Dataset within the same number of training steps. FID [63] here refers to Fréchet Inception Distance.

and fine-tuning. In addition, we conducted a thorough analysis that sheds light on how it improves diffusion training speed while improving fidelity.

4.1 Introduction

Diffusion models have made impressive progress in image generation by framing the process as a phase of denoising random Gaussian noise into the final image. Despite the advancements, training a diffusion model is resource intensive. For example, even in the primary image dataset CIFAR-10, the representative few-step diffusion model, Consistency Model [167], requires training for 10 days on 4 A6000 GPUs to reach a desired FID score of around 10. Similarly, with fewer model parameters, multiple-step diffusion model DDIM [163] still requires 24 hours on an A5000 GPU on the CIFAR-10 dataset. Although recent remarkable

achievements in accelerating the inference of diffusion models [87, 121, 167, 115, 118, 119] have been accomplished, the inefficiency of diffusion training remains a significant bottleneck, hindering the iterative development of vision generative AI.

Previous methods for improving diffusion training have focused on various strategies, such as balancing the impact of activation layers and neural weights [79], modifying hyperparameters and design choices [165], and leveraging patchifying strategies [189] etc. Specifically, Karras *et al.* [79] modifies the activation magnitude, neural weight standardization, and group normalization, achieving significant acceleration in diffusion training. Besides, previous work [165] proposes a customized method for the Consistency Model to improve the performance and diffusion training. Our method is orthogonal to these previous methods. We got inspired by the Immiscible Diffusion in physics. As illustrated in Fig. 4.2 (a) left, miscible particles tightly jumble together after the diffusion process, making it difficult to separate them individually during the denoising phase. However, when the particles are rendered immiscible, they can still achieve a similar overall distribution while remaining clearly distinguishable (see Fig. 4.2 (a) right). This insight inspires our strategy for improving the disentanglement of diffused data.

We draw an analogy from the phenomenon of Immiscible Diffusion and relate the distribution of image data to the behavior of particles discussed above. In traditional diffusion processes, each image can be diffused to any point in the noise space, and conversely, each point in the noise space can be denoised to any source image, as illustrated in the left image of Fig. 4.2 (b). We hypothesize that the jumbled image-noise mapping creates a miscible diffusion effect and makes the optimization of the diffusion model difficult. Inspired by the Immiscible Diffusion, we are motivated to make the mixed diffusion phase distinguishable.

We propose one simple **Immiscible Diffusion** method. Note that we still sample Gaussian noise but perform a batch-wise assignment of noise to each image based on the distance between them during training. This approach ensures that each image is only diffused to surrounding areas while maintaining the overall Gaussian distribution of all noises. This technique was also used in flow matching-based methods [135, 174] for optimizing the image-noise flow. Nevertheless, we find that the image-noise distance is reduced by only $\sim 2\%$ after assignment, as provided in Part 4.4.3. This motivates us to ask which factors dominate the performance improvement? To investigate it, we propose another method that qualifies Immiscible Diffusion in Part 4.4.3. Experiments show that the training efficiency improvement is comparable to the function of flow optimization, demonstrating that immiscibility is the dominant factor.

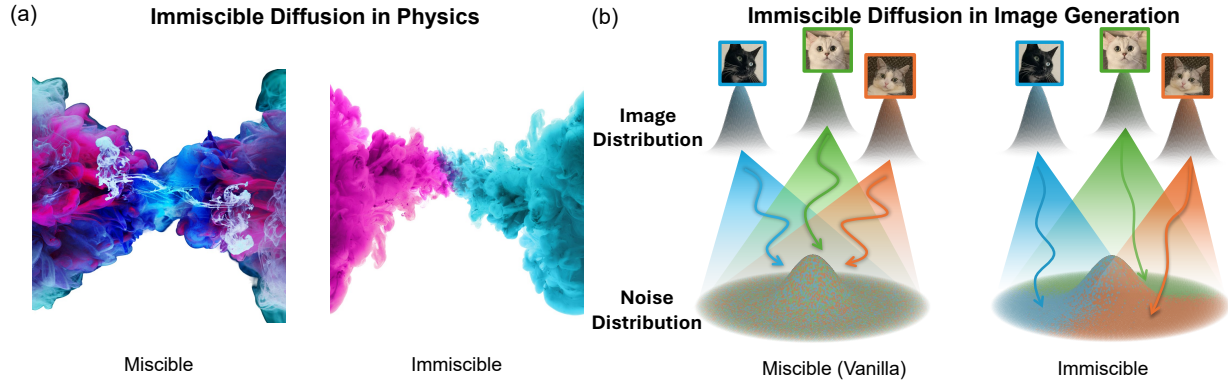


Figure 4.2: **Physics illustration of Immiscible Diffusion.** (a) depicts the miscible and Immiscible Diffusion phenomenon in physics, while (b) demonstrates the image-noise pair relation in vanilla (miscible) and Immiscible Diffusion method.

However, technically, to achieve immiscibility with the image data-noise assignment has an $O(N^2 \log N)$ - $O(N^3)$ complexity. This introduces significant overhead during training, especially for large-scale training with huge batch sizes and high-resolution images. To address this, we employ a novel quantization method during assignment. We quantize the noise and image data into low-precision formats (e.g., 16-bit) during conducting the assignment algorithm. We highlight that this assignment operation only involves **one line of code**, and is performed only during the training phase without modifying the model architecture, the noise scheduler, the sampler, or the method of inference.

We conduct extensive experiments on three common modes: unconditional, conditional, and fine-tuning on three diffusion baselines: Consistency Models, DDIM and Stable Diffusion and three datasets: CIFAR-10, CelebA and ImageNet datasets. Results show that our proposed method significantly improves the training efficiency in all experiments. Specifically, we achieve $3\times$ training efficiency for the CIFAR-10 dataset with immiscible unconditional Consistency Model compared to the original Consistency Model. Furthermore, we show that the FID is even lower with our method used, confirming the fidelity of our generated images. We also provide images generated from models trained with vanilla and immiscible models experiencing the same training steps, where we see that those from immiscible models are much more complete and clearer, further proving the training efficiency enhancement resulted from the Immiscible Diffusion. Examples are shown on the right of Fig. 4.1. Deeper analysis shows that our method, although with only one line of code and involving $\sim 2\%$ image-noise datapoint distance changes, achieves all the benefits above in negligible running time.

To sum up, our contributions are as follows:

- We clearly and specifically identify the miscibility issue in noisy diffusion steps, which leads to slow convergence of diffusion training.

- To tackle the miscibility issue, we propose a simple and effective method, Immiscible Diffusion, a strategy that can only requires one-line of code, to improve training efficiency for diffusion training.
- Experiments demonstrate the effectiveness of our proposed method on several popular diffusion models across multiple datasets, and across unconditional, conditional, and fine-tuning tasks. In addition, we conducted thorough analyses and ablation studies to elucidate how our method works and dominates the effectiveness.

4.2 Related Work

4.2.1 Diffusion Model with Efficient Inference

Diffusion models [168, 66, 150, 133] have been attracting huge attentions because of their high-fidelity image and video generation [68, 67, 129], data-efficient perception [171, 120, 197], and even representation abilities for robotics [25, 137, 10]. However, slow inference is one of the key bottlenecks for diffusion models. To address this issue, various approaches have been proposed. For instance, techniques such as DDIM [163] have reduced the number of denoising steps from 1000 to 10, significantly speeding up the process. Furthermore, the introduction of Consistency Models [167] and LCM [121], which utilize the properties of self-consistency, enables denoising in as few as 1-4 steps, further enhancing the generation speed of diffusion models. Subsequently, the development of SD-turbo [154], which leverages GAN [53] loss for high-definition image generation in a single step, has occurred. The Consistency Trajectory Models [83, 143] improve the generation quality of Consistency Models and accelerate research on efficient inference for diffusion models. Additionally, beyond reducing denoising steps, efforts to improve the inference efficiency of single function evaluation are being explored in various ways, including model quantization [103] and partitioning the generative components [100]. Moreover, StreamDiffusion [87] streamlines denoising steps to achieve real-time inference at the pipeline level optimization. The improvement of the inference efficiency significantly pushes forward real applications based on diffusion models. Yet accelerating diffusion training is still under-explored.

4.2.2 Diffusion Model with Efficient Training

Improving the training efficiency of diffusion models is crucial. Various strategies have been proposed, including architectural modifications [79], approximating the diffusion phase with flow [115], and designing parameter choices [165] *etc.* Specifically, in [79], the authors discover that the magnitude of activation and the magnitude of neural weights significantly impact the training dynamics of diffusion models. They propose adjusting the activation magnitude and standardizing neural weights, as well as modifying the normalization layers to make diffusion training in a more smooth dynamic. Besides, Song *et al.* [165] aims to enhance the training efficiency of Consistency Models through customized design choices,

significantly improving both training speed and fidelity. Furthermore, leveraging approximation strategies based on Ordinary Differential Equation (ODE) assumptions [115] improves not only inference efficiency but also training efficiency since diffusion trajectories are prone to deterministic. Beyond improving diffusion training with either architecture adjustment or selection parameters, Wang *et al.* [189] introduce a novel patch strategy to control the ease of diffusion training, achieving both training and data efficiency. Simultaneously, Wang *et al.* [185] notices that the denoising of some noisy diffusion steps contains little information and is too easy to learn, so focusing more on other steps would significantly improve training efficiency. Our method differs from previous works by clearly highlighting an under-explored problem: the miscibility problem of image data in the noise space, which plays a crucial role in training diffusion models. Our proposed Immiscible Diffusion is extremely simple yet significantly improves training efficiency.

4.2.3 Image-noise Optimal Transport in Generative Models

In ODE-based methods such as flow matching [112], straightening the flow with optimal transport (OT) has been used as a tool to improve the generation performance. Specifically, optimizing image-noise transport in a batch [135, 174] has been found to be an effective way to improve performance. Training efficiency improvement was also observed [135, 174], and explained or posited with reduction of the variance of the training goal. However, the standard deviation reduction is only $\sim 4\%$ in [135]. Moving forward, [97] pointed out the curvature problem in the ODE paths caused by the collapse of the reverse trajectories in the average direction. However, they replaced Gaussian noise sampling with a Variational Autoencoder (VAE) encoder-style structure to eliminate such curvatures, which destroys the strict Gaussian distribution in the noise space. Concurrent to our work, [82] applies batch-wise OT to diffusion models to achieve better FIDs, making posits in curvature reduction for the enhancement. Several methods were proposed to improve the speeds and effectiveness of OT, such as pre-training with PF-ODE [195], using Schrodinger bridging [32], utilizing conditional Wasserstein distance [16] and generator-induced coupling in Consistency Models [73]. However, we are the first to emphasize that the dominant reason for training efficiency enhancement is the miscibility problem in noisy layers, which we prove in Part 4.4.3. Furthermore, most of previous methods are for unconditional flow matching methods only. Our work demonstrates the effectiveness in multiple diffusion models, datasets, conditional generation, and fine-tuning experiments.

4.3 Method

4.3.1 Physics Intuition

Diffusion models mimic the reverse thermodynamic diffusion phenomenon [160] to ease the denoising process. However, when the sources are **miscible**, as shown in the left of Fig. 4.2

(a), they end up messily mixed. Predicting the reversal process from such a random mixture encounters significant difficulties, and unfortunately, this is a problem diffusion model always facing during denoising.

However, we notice that mixing can also be organized when sources are **immiscible**. Under that circumstance, the sources would take different continuous areas after diffusion, while the whole diffused area remains the same, as shown in the right image of Fig. 4.2 (a). Thereafter, the reversal process becomes smooth. Inspired by Immiscible Diffusion, we then introduce it to the diffusion models, with the aim of making the optimization easier and to achieve a higher training efficiency.

4.3.2 Immiscible Diffusion Model

Similar to the physics phenomenon, we find that for diffusion models, any images are diffused to every corner of the noise space, which also means that each noise point can go back to any image. This would cause the denoising model to be confused on which image to go to, as shown in the left of Fig. 4.2 (b).

Mimicking the immiscible phenomenon in physics, we hope to design similar processes where each noise point is only matched to limited images, so as to avoid the confusion for the denoising model. However, the noise space must remain Gaussian to help the sampling process. Therefore, we propose our first implementation way of **Immiscible Diffusion**, which assigns the batch of noises to the batch of images during training according to the image-noise distance in their shared space. We minimize the total distance of the image-noise pairs in a batch during assignment. Here we use the L2 distance for assignment, which is ablated in Part B.1.3 in the Supplemental Materials. After assignment, the noise is still Gaussian, while each noise is assigned to nearer images like what happens in the immiscibility phenomenon, which significantly eases the difficulties for the denoising. Fig. 4.2 (b) right illustrates an extreme example of the Immiscible Diffusion, where the noise corresponding to each image is relatively separated.

For implementation, all we need to do is to perform a linear assignment [91] between the batch of images and noises according to their distances. This can be achieved in only one line of code using Scipy [182]. The algorithm is shown below:

Algorithm 1 Batch-wise Image-Noise Assignment

- 1: **Input:** Image batch x_b , random noise batch $n_{rand,b}$, sampled diffusion steps t_b and diffusion schedule α
 - 2: $\text{assign_mat} \leftarrow \text{scipy.optimize.linear_sum_assignment}(\text{dist}(x_b, n_{rand,b}))$
 - 3: $x_{t,b} \leftarrow \sqrt{\alpha_{t_b}}x_b + \sqrt{1 - \alpha_{t_b}} \cdot n_{rand,b}[\text{assign_mat}]$
 - 4: **Output:** Diffused image batch $x_{t,b}$
-

While linear assignment qualifies Immiscible Diffusion, Immiscible Diffusion can be achieved with multiple paths. In part 4.4.3, we ablate the linear assignment by letting it remain im-

miscible while disqualifying optimal transport, proving that immiscibility plays the dominant role in performance improvement.

4.3.3 Mathematical Illustration

In this section, we mathematically elucidate the denoising difficulty for traditional diffusion models based on DDIM [163, 66] and how our proposed Immiscible Diffusion reduces such difficulties.

In DDIM, we know for any image data-point x_0 , when it is diffused to the *last* diffusion step T , i.e. $t = T$, the image is sufficiently wiped out and nearly only gaussian noise is remaining, therefore,

$$q(x_T | x_0) \approx \mathcal{N}(x_T; 0, I) \approx p(x_T), \text{ where } x_T(x_0, \epsilon) = \sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, 1), \quad (4.1)$$

where $q(x_T | x_0)$ refers to the the forward diffusion process [65] (or inference distribution [162]). Utilizing Bayes' Rule and Equation 4.1, we can find that for a specific x_T :

$$p(x_0 | x_T) = \frac{q(x_T | x_0) \cdot p(x_0)}{p(x_T)} \approx p(x_0) \quad , \quad (4.2)$$

which indicates that the distributions of the corresponding images for any noise data-point are the same as the distribution of all images.

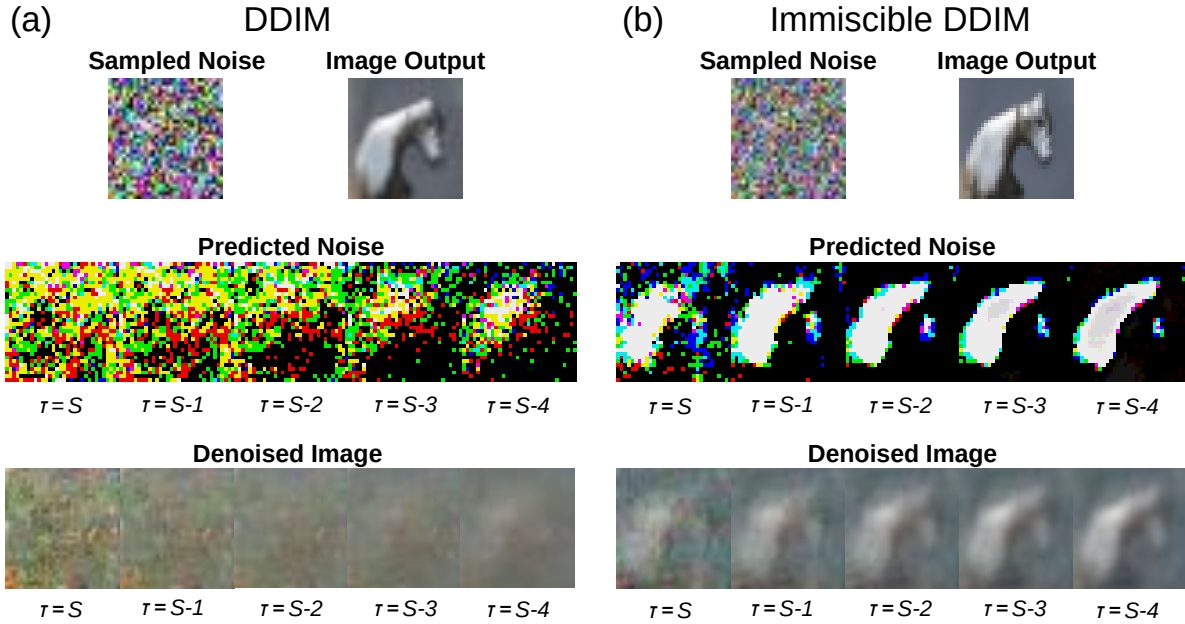


Figure 4.3: **Feature analysis of vanilla (miscible) and immiscible DDIM.** Referring to [161], $\tau = S$ represents the layer denoising from the pure noise. We show that while the two sampled noises are similar, the denoised image of immiscible DDIM significantly outperforms that of the traditional one, generating an overall reasonable image. The reason behind this is traditional methods cannot successfully predict noises at noisy layers.

The simplified training objective in DDIM [163, 66] is the added noise $\epsilon(x_t, t)$ at each diffusion step t and at the point x_t . However, we find that for a specific point x_T in the noise space at diffusion step T ,

$$\begin{aligned}
 \epsilon(x_T, T) &= ax_0 + bx_T = \sum_{x_0} (ax_0 + bx_T) p(x_0 | x_T) = a \sum_{x_0} x_0 p(x_0 | x_T) + bx_T \sum_{x_0} p(x_0 | x_T) \\
 &= a \sum_{x_0} x_0 p(x_0) + bx_T = a\bar{x}_0 + bx_T
 \end{aligned}
 \tag{4.3}$$

where $a = -\frac{\sqrt{\alpha_t}}{\sqrt{1-\alpha_t}}$ and $b = \frac{1}{\sqrt{1-\alpha_t}}$ are constants, and $\bar{x}_0$ is an average of images in the dataset. When the number of images is large enough, $\bar{x}_0$ contains little meaningful information.

As shown in Fig. 4.3 (a), we study the predicted noises of different denoising layers in a DDIM model with total inference steps of 20, to illustrate our discovery. Here we refer to [161], specifying the sampling step $\tau = S$ as the layer that denoises the pure noise, which is equal to the diffusing step $t = T$. We can see that the predicted noise for DDIM at $\tau = S$ (equal to $t = T$ in diffusing) does not provide much useful information, while denoising with

this "noise" does not provide any distinguishable image, which all support our hypothesis for the difficulty in denoising when τ in sampling (or t in diffusing) is large. Similar observations are also shown in the concurrent work [185]. However, in our Immiscible Diffusion, while for each batch, we still have

$$p(x_T) = \mathcal{N}(x_T; 0, I), \quad (4.4)$$

for each specific data point x_T or x_0 , the *conditional* noise distribution does not follow the Gaussian distribution because of the batch-wise noise assignment

$$p(x_T | x_0) \neq \mathcal{N}(x_T; 0, I). \quad (4.5)$$

Instead of the Gaussian distribution, we assume that the predicted noise with noise assignment has a distribution described as follows

$$p(x_T | x_0) = f(x_T - x_0, bs, \dots) \mathcal{N}(x_T; 0, I), \quad (4.6)$$

where f is a function denoting the influence of assignment on the conditional distribution of x_T , and bs is the training batch size. Apparently, according to the definition of linear assignment problem [91], f decreases when $x_T - x_0$ increases its norm, specifically the L2 norm as in our default setting.

Therefore, from Equation 4.2 and 4.6, we have

$$p(x_0 | x_T) = f(x_T - x_0, \dots) p(x_0), \quad (4.7)$$

which means that for a specific noise data-point, the possibility of denoising it to the nearby image data-point would be higher than to a far-away image.

For the noise prediction task, we see that

$$\begin{aligned} \epsilon(x_T, T) &= \sum_{x_0} (ax_0 + bx_T) p(x_0 | x_T) \\ &= a \sum_{x_0} f(x_T - x_0, \dots) x_0 p(x_0) + bx_T \\ &= \overline{ax_0 f(x_T - x_0, \dots)} + bx_T \end{aligned} \quad (4.8)$$

where $\overline{ax_0 f(x_T - x_0, \dots)}$ is the weighted average of x_0 with more weights on image data-points closer to the noisy data-point x_T itself. Therefore, the noise predicted would lead to the average of nearby image data-points, which makes more sense than pointing to a constant. Indeed, in Fig. 4.3, we see that even for the pure noise layer, immiscible DDIM can predict the noise effectively pointing to the shape of the horse image, and the prediction in one step by subtracting the predicted noise shows the outline of the horse correctly.

4.3.4 Accelerating Assignment in Immiscible Diffusion

The assignment problem has been studied extensively for decades [38, 92, 52]. In this chapter, we use the Hungarian algorithm [92] as our main assignment method. However, Hungarian matching has high complexity with $O(N^3)$, which drastically slows down the training especially when we have high-dimensional image data (*e.g.*, even using the mini image data $32 \times 32 \times 3 = 3072$). To mitigate this issue, we make a novel use of quantization for image data and noise, *that is*, we quantize the *fp32* image and noise data to *fp16* to carry out the assignment, while maintaining the same precision input to diffusion models. This trick significantly reduces the overhead to a negligible level.

To efficiently perform Immiscible Diffusion when running on multiple GPUs, we assign the image-noise distance matrix computation to each process, and then gather them to execute the assignment. This is particularly important as high resolutions and large batch sizes are frequently required in applications.

4.4 Experiments

4.4.1 Experiment Settings

To elaborate the performance of Immiscible Diffusion, we utilize the proposed method on Consistency Models [167], DDIM [161] and Stable Diffusion [148], and using CIFAR-10 [90], CelebA [117], tiny, randomly picked 10% and the full ImageNet [31] datasets due to the limitation of computation resource. The training hyperparameters are shown in Tab. 4.1. Unspecified hyperparameters are taken the same as those in their baseline methods’ original papers. For evaluations, we compare the results generated by our Immiscible Diffusion method and the baseline using both the quantitative evaluation metric FID [64] and qualitative assessments.

Note that for Consistency Models, we use the single-step generation consistency training. For DDIM, we add no noise during the sampling and use linear scheduling for picking sampling steps. For Stable Diffusion, we directly use the implementation from Diffusers of Huggingface team [146]. For fine-tuning, we use Stable Diffusion v1.4 [146] as the pre-trained model.

Table 4.1: Experiment setting.

Model	Consistency Model	Consistency Model	Consistency Model	DDIM	Stable Diffusion Unconditional	Stable Diffusion Class-conditional	Stable Diffusion Fine-tuning
Dataset	CIFAR-10	CelebA	Tiny ImageNet	CIFAR-10	10% ImageNet	Full ImageNet	Full ImageNet
Batch Size	512	1024	2048	256	512	2048	512
Resolution	32×32	64×64	64×64	32×32	256×256	256×256	256×256
Devices	$4 \times A6000$	$8 \times A800$	$16 \times A800$	$1 \times A5000$	$4 \times A6000$	$8 \times A800$	$4 \times A6000$

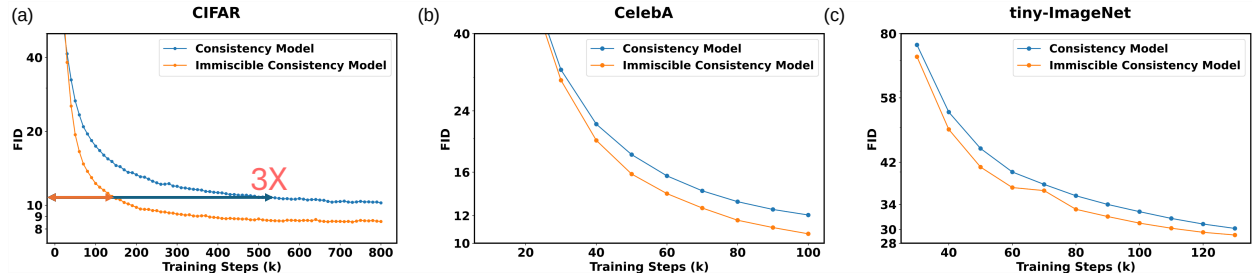


Figure 4.4: **Evaluation of baseline and immiscible Consistency Models on (a) CIFAR-10, (b) CelebA, and (c) tiny-ImageNet dataset.** We illustrate the FID of two models with different training steps. Clearly, immiscible Consistency Models have much higher efficiency than the vanilla ones.

4.4.2 Training Efficiency Improvement with Linear Assignment

Unconditional generation with Consistency Models: In Fig. 4.4, we show the FIDs of images generated with baseline and immiscible Consistency Models trained with different training steps on the CIFAR-10 dataset, the CelebA dataset and the tiny ImageNet dataset, respectively. We observe that the immiscible Consistency Model trains much faster than the baseline Consistency Model, and converges to a significantly lower FID on all these datasets. We also show the images generated by immiscible and baseline Consistency Models trained for 100k steps in Fig. B.2 in the Supplemental Materials, where we find that the images generated by the Immiscible Consistency Model are much more complete and realistic. Tab. B.1 in the Supplemental Materials further presents the training steps necessary to achieve specific reasonable FID thresholds. We find that the immiscible Consistency Model significantly improves the training efficiency by around 3 $\times$, proving the effectiveness of Immiscible Diffusion in training accelerations.

In the main experiment, we observe that our method on top of the Consistency Model is effective across the datasets varying from different data sizes and resolutions. Indeed, the Consistency Model is a few-step diffusion model, and our proposed Immiscible Diffusion especially works on improving the denoising effect when the noise level is high, as shown in Fig. 4.3. The improvement of the training efficiency on such a few-step diffusion model further validates our findings.

One characteristic of the Consistency Model is that it approximates the Stochastic Differential Equation (SDE) diffusion model with the ODE approximation. Thus, the original image-noise mapping is highly jumbled together since it is highly possible that closed image data points are diffused to distant noise points. Our Immiscible Diffusion improves this issue by adjusting the trajectories of image-noise mapping and making them more distinguishable.

More baselines: To show the generalization of Immiscible Diffusion for more baselines, we further conduct experiments on two baselines: DDIM [161] and Stable Diffusion [150] on the CIFAR-10 and the randomly picked 10% ImageNet dataset, respectively. As shown in

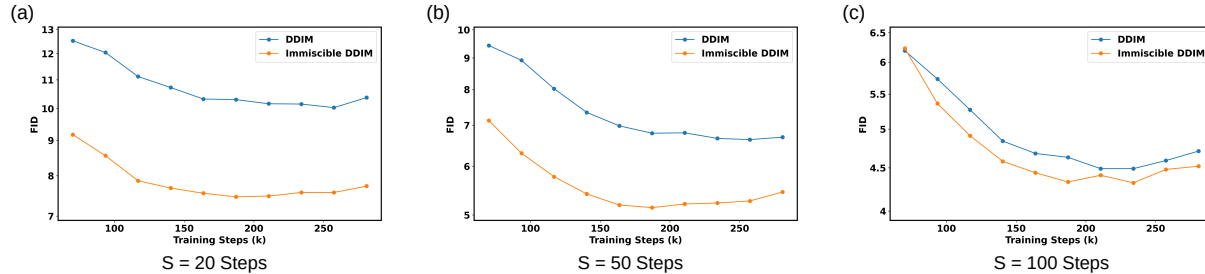


Figure 4.5: **Evaluation of baseline and Immiscible DDIM on CIFAR-10 dataset with different inference steps S .** We find that Immiscible DDIM outperform the baseline more significantly when the number of inference steps S is smaller.

Fig. 4.5, and detailed in Tab. B.2 in Supplemental Materials, we find that our immiscible DDIM significantly improves the training speed and the FID compared to those of the baseline DDIM on the CIFAR-10 dataset, and the improvement is more significant when the sampling step is lower. This demonstrates the effectiveness of our proposed method works beyond the Consistency Model and can be generalized to more few-step denoising models. We also provide a discussion in Part B.1.4, showing that the effectiveness of Immiscible Diffusion can persist in a wide range of batch sizes. To further evaluate generalizability on the popular baseline, Stable Diffusion [150], we also conduct unconditional generation experiments on the ImageNet dataset. We observe that immiscible Stable Diffusion and baseline Stable Diffusion achieve similar FID without significant gap, yet our immiscible Stable Diffusion is able to generate much higher quality images from a subjective human judgement. For example, Fig. B.7 in the Supplemental Materials shows that our proposed method generates significantly clearer images compared to the baseline. More visualization without any cherry-picking can be seen in Fig. B.8 in the supplementary materials. We indicate that even though FID is the primary metric and is remarkably successful, the metric is known to sometimes disagree with human judgement [96].

Class-conditional Generation: We extend Immiscible Diffusion to class-conditional generations on ImageNet dataset with Stable Diffusion [148], to explore the performance of Immiscible Diffusion in conditional generations. Results are shown in Fig. 4.6 (a), where we observe that in 20k training steps, the FID for immiscible class-conditional Stable Diffusion is 16.43, which is 1.49 lower than our Stable Diffusion baseline. We further confirm such improvements on CMMD [74], where the immiscible and vanilla models get 1.385 and 1.436 respectively. Additional evaluation on CLIPScore [61] shows that both the immiscible and the baseline models generate images with CLIPScores of 28.55, with a standard deviation of 0.01 and 0.02 respectively, indicating that Immiscible Diffusion does not hurt the image-prompt correspondence in complicated ImageNet dataset. Qualitative comparisons in Fig. B.9 in Supplemental Materials further prove such performance enhancements, which augment the effectiveness of Immiscible Diffusions into more commonly-used conditional generations.

Fine-tuning: Our Immiscible Diffusion can also be applied to enhance the fine-tuning process where numerous applications fall in. We fine-tune the stable diffusion v1.4 model [148] on ImageNet dataset, finding that immiscible fine-tuning achieves an FID of 10.28 compared to 11.45 for vanilla fine-tuning with 5k training steps. Detailed results are shown in Fig. 4.6. This further broadens the application of Immiscible Diffusion.

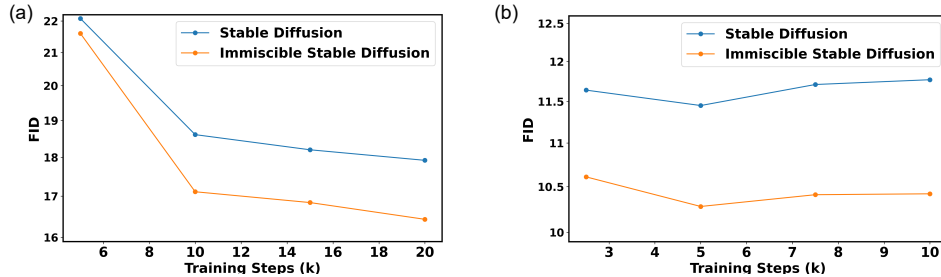


Figure 4.6: **Evaluation of baseline and immiscible class-conditional Stable Diffusion on ImageNet dataset, using 20 inference steps.** (a) FID of two models trained from scratch (b) FID of two models fine-tuned on Stable Diffusion v1.4.

4.4.3 Discussion

To further understand the proposed Immiscible Diffusion method, we delve into several key questions to ablate our approach:

How much does image-noise distance reduce in the assignment? Tab. 4.2 shows the reduction in distance after the image-noise assignment. We find that the L2 distance only reduces by about 2%, with a slight increase observed at higher batch sizes. However, as shown in Fig. 4.3, the assignment is sufficient to effectively activate denoising at high noise levels, significantly boosting training efficiency, even though the distance change is low. We attribute the low distance reduction rate after the assignment to the extremely high dimensionality (3072 for each image of the CIFAR-10 dataset) of the image and noise space.

How much time does image-noise assignment cost? In Tab. 4.2, we indicate that our assignment method does not introduce significant extra overhead due to our utilization of quantized assignment in our practical implementation. Even for a large batch size per GPU of 1024, our algorithm only brings in an additional 22.8 ms, demonstrating the potential of utilization for future applications.

Immiscible and OT: who dominates the training efficiency enhancement? Our Immiscible Diffusion claims to enhance training efficiency by improving miscibility in noisy diffusion steps. However, the method we take towards Immiscible Diffusion, i.e. linear assignment between image and noise, also serve as a roughly approximate OT between image and noise, which might intuitively benefit the diffusion through straightening the diffusion

paths [135, 174]. However, the previous section shows that the image-noise distance is only reduced by $\sim 2\%$, motivating us to ask if OT is really the dominant factor?

To answer this question, we ablate these two factors: OT and immiscibility. We design a non-OT Immiscible Diffusion experiment which keeps the immiscible property while not involving the OT. This is achieved by assigning images to the flipped noise whose all dimensions are reversed, while using the original noise during diffusion. In such a way, the image-noise pair no longer follows OT, but still qualifies the Immiscible Diffusion - i.e. images are still assigned to a limited area. Interestingly, we observe that the non-OT Immiscible Diffusion can still accelerate and enhance the diffusion training, which is nearly comparable to the OT Immiscible Diffusion in final stages, as shown in in Fig. 4.7. Considering that the non-OT version introduce miscibility in middle diffusion layers, which we posit for its difference to OT version, we conclude that Immiscible Diffusion is dominant in enhancing the diffusion model’s performance, compared to the benefits from OT.

Table 4.2: Image-noise data-point L2 distance reduction after the assignment for minimizing it and the time cost for the assignment.

Batch Size	128	256	512	1024
$\Delta\text{Dist.}$	-1.93%	-2.16%	-2.32%	-2.44%
Assignment Time (ms)	5.4	6.7	8.8	22.8

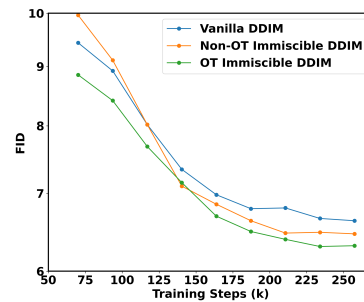


Figure 4.7: **Ablation of OT in Immiscible Diffusion.** FIDs of OT and non-OT Immiscible Diffusion indicates that it is the Immiscible Diffusion rather than OT that dominate the performance enhancement.

4.5 Conclusion

Inspired by the immiscibility phenomenon in physics, we introduce Immiscible Diffusion, a method to improve image-noise mapping to accelerate diffusion training. Specifically, Immiscible Diffusion is an assignment-then-diffusion strategy. One way of it is to minimize the image-noise pair distance within a mini-batch so that each image is diffused to nearby noise areas. This simple approach requires only one line of code and includes a quantized-assignment strategy to reduce computational overhead.

Experiments show our Immiscible Diffusion approach speeds up Consistency Model’s training by approximately $3\times$ on the CIFAR-10 dataset, $1.3\times$ on the CelebA dataset, and $1.2\times$ on the tiny-ImageNet dataset, as well as in conditional generation and fine-tuning on Stable Diffusion. Thus, we show that Immiscible Diffusion can generalize to across datasets, baselines and tasks. Further analysis is provided to explain how this works.

Limitation. The assignment strategy is one straightforward way for Immiscible Diffusion, but not necessarily optimal. Due to the limited computational resources, our experiments are mainly conducted on small-scale datasets, so we lack the validation on larger-scale datasets such as LAION [155]. In future work, we will improve the assignment strategy to cater to practical utilization of conditional generation such as accelerating the general text-to-image or text-to-video diffusion training.

Chapter 5

Improved Immiscible Diffusion: Accelerate Diffusion Training by Reducing Its Miscibility

The substantial training cost of diffusion models hinders their deployment, like in driving data synthesis. Immiscible Diffusion [104] recently showed that reducing diffusion trajectory mixing in the noise space via linear assignment accelerates training by simplifying denoising. To extend Immiscible Diffusion beyond the inefficient linear assignment under high batch sizes and high dimensions, we refine this concept to a broader miscibility reduction at any layer and by any implementation. Specifically, we empirically demonstrate the bijective nature of the denoising process with respect to Immiscible Diffusion, ensuring its preservation of generative diversity. Moreover, we provide thorough analysis and show step-by-step how immiscibility eases denoising and improves efficiency. Extending beyond linear assignment, we propose a family of implementations including K-nearest neighbor (KNN) noise selection and image scaling to reduce miscibility, achieving up to $> 4\times$ faster training across diverse models and tasks including unconditional/conditional generation, image editing, and robotics planning. Furthermore, our analysis of immiscibility offers a novel perspective on how optimal transport (OT) enhances diffusion training. By identifying trajectory miscibility as a fundamental bottleneck, we believe this work establishes a potentially new direction for future research into high-efficiency diffusion training.

5.1 Introduction

Training diffusion-based models is costly and time-consuming, and such training costs are continuously growing. For example, Stable Diffusion V1.1 [147] was trained for more than 24 days on 256 GPUs, while its V2’s training time grew to 32 days with the same GPUs. Consequently, the training efficiency problem attracts intensive explorations from diverse aspects [149, 111, 136, 175].

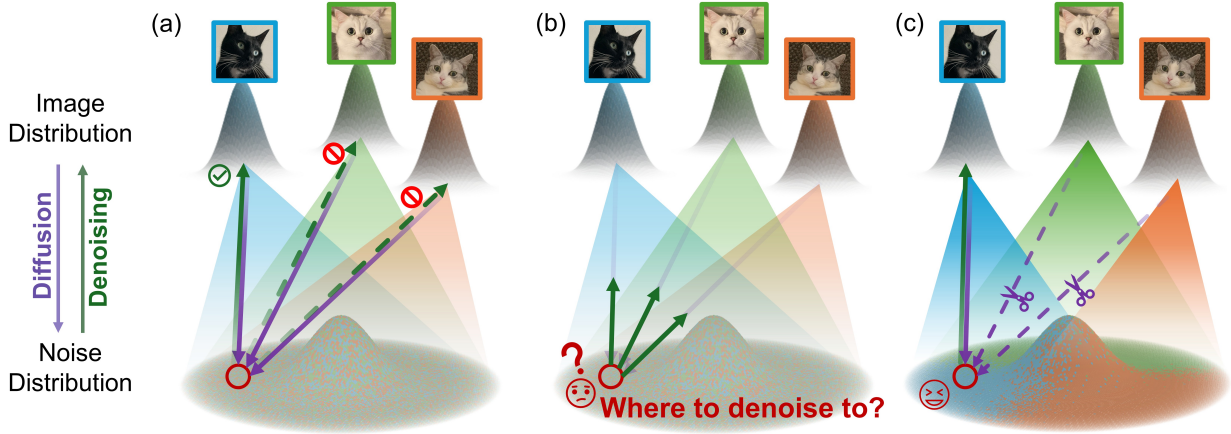


Figure 5.1: **Improved Immiscible Diffusion Theory.** (a) While vanilla diffusion trajectories (flows) are mixed (miscible), *each noise point is stably correlated to a specific generated image*, making many diffusion trajectories irreversible. (b) Those irreversible trajectories would confuse the denoising process. (c) We introduce Immiscible Diffusion to cut mixed (miscible) diffusion trajectories during training for accelerating diffusion training.

Recent works [136, 175] found that applying batch-wise optimal transport (OT) to pair the images and the noises can boost the training efficiency of flow matching [111] by shortening flow trajectories and lowering variance in denoising. However, [104] shows that such approximate OT reduces the image-noise distance by only $\sim 2\%$, and [136] shows that the standard deviation (std) of the denoising function reduces only $\sim 4\%$. On the other hand, Immiscible Diffusion [104] assigns each image to a relatively separated noise area to boost the training efficiency, attributing such performance improvements to better denoising performances in noisy layers. However, its implementation is limited to image-noise linear assignment, which not only reduces image-noise distance for just $\sim 2\%$ as well, but also has a time complexity of $O(n^3)$, questioning its efficiency optimality. Additionally, the concept of Immiscible Diffusion itself raises questions on the diversity of its generated images, as it stops images from being diffused to some far-away noise areas.

In this chapter, we first refine Immiscible Diffusion to a broader concept: a diffusion process with reduced mixtures of diffusion trajectories (flows) from different images. We find that generated images are stably correlated to their noise origins, which makes the mixing of diffusion trajectories unnecessary. This resolves the diversity concerns of Immiscible Diffusion. We then explore Immiscible Diffusion’s benefits step-by-step, demonstrating how the miscibility reduction eventually leads to boosts on diffusion models. With the refined definition, we accordingly offer a few new Immiscible Diffusion implementations including KNN and image scaling, which satisfy the improved concept but either do not qualify for an assignment or even involve no image-noise pairings. Extensive experiments are performed to examine the performance of the Immiscible Diffusion family, where we observe consis-

tent training efficiency boosts on various baseline models including consistency models [166], flow matching [111] and DDIM [162]. Further experiments see similar boosts across diverse image generation tasks including unconditional and conditional ones, different stages (training and fine-tuning), and various image datasets such as CIFAR-10 [89], ImageNet [31] and MSCOCO [110]. The Immiscible Diffusion family is subsequently extended to tasks outside image generation such as image in-painting and out-painting and robotics planning [24], where the coherent performance enhancements support its robustness. Thorough discussion is provided to distinct Immiscible Diffusion to other training efficiency improvement methods, and to compare between the implementations. Our contributions are summarized as follows,

- We extend Immiscible Diffusion to an implementation-agnostic concept, characterized by reduced mixture (miscibility) of diffusion trajectories from different images. Our experiments show that generated images are stably correlated with their noise origins, relieving concerns on Immiscible Diffusion’s generation diversity. Systematic feature analysis clarifies how Immiscible Diffusion enhances diffusion training.
- Based on improved Immiscible Diffusion, we design a family of implementations, including the KNN, image scaling. These methods are more efficient to linear assignment, and feature analysis shows that they both effectively reduce the miscibility of diffusion trajectories.
- The Immiscible Diffusion family is applied to various image generation tasks, including unconditional and conditional image generation training and fine-tuning, and across diverse datasets and baseline methods. Unanimous effectiveness of Immiscible Diffusion is observed, and additional experiments extend its benefits to applications such as image editing and robotics planning. The miscibility problem we established points out a potential direction for future research in efficient diffusion training.

5.2 Related Works

5.2.1 Training Efficiency of Diffusion-based Models

As training efficiency limits the large-scale deployment of diffusion-based models, actions are taken to trigger diverse abundant parts inside them. For the feature dimensions, Stable Diffusion [149] shrinks image sizes with VAE [86], downsizing the image dimension by 16-64 times. [190] introduces a novel patch strategy to control the ease of diffusion training, achieving both training and data efficiency. For the noise space, [70] modifies the noise used in diffusion models to boost the performance. For training dynamics, [80] discovers that the magnitude of activation and the magnitude of neural weights can impact the diffusion’s training speed. [164] demonstrates the importance of training dynamics on training efficiency, including the usage of exponential moving average (EMA), training loss and the

noise schedule. [215, 184] notices that denoising some noisy steps helps little, so focusing more on other steps can improve training efficiency. However, these works hardly alter the diffusion trajectory, therefore cannot solve the miscibility problem Immiscible Diffusion aims to tackle.

5.2.2 Diffusion Paths and Training Efficiency

Recent works redesign diffusion trajectories for faster training. Notably, based on the rectified flow [114], [116] improves training efficiency by making diffusion trajectories deterministic and straight. [111] further straightens the diffusion trajectory by making the image-noise mixture linear. Probing more deeply, several studies began exploring alternatives to mapping each image to the full noise space. [98] pointed out the curvature problem in the ODE paths caused by the collapse of the denoising trajectories pointing to the average direction. However, they replace the noise sampling with a VAE encoder-style structure to eliminate such curvatures, which destroys the strict Gaussian of the noise. In order to make diffusion trajectory paths even straighter and shorter, [136, 175] applies batch-wise OT to assign noise to closer images before performing flow matching, followed up by a few further improvements on them [195, 32, 16, 73]. However, [104] shows that the OT only reduces average image-noise distance by $\sim 2\%$, and [136] claims that the standard deviation of denoising reduces only $\sim 4\%$ after applying OT, which doubts the contributions of OT in the training efficiency boosts. More recently, Immiscible Diffusion [104] assigns noise to some preferred noise areas, aiming to avoid the difficulty of denoising in noisy layers. However, its implementation of batch-wise linear assignment is still similar to OT, so its theory needs to be further justified versus OT.

5.2.3 Image-Noise Correlation in Diffusion-based Models

While most diffusion-based models diffuse each image to the whole noise space, some diffusion inversion works indicate that learned diffusion model’s inversion does not follow this. [170] indicates that the diffusion and denoising is not symmetric, and [209] demonstrates that nearly the same image would be generated with the same noise but diverse diffusion models. These imply that generated images and their noise origins are somehow correlated. [3] further illustrates some similarity between the generated images and their noise origin, so as [128]. Quantitatively, [81, 210, 95] suggests that DDPM’s inversion is an L_2 OT process. However, the correlation strength between generated images and their noise origin, *i.e.* how much perturbation can such correlation resist, was not thoroughly discussed. Our work proves a **stable** relation exists between the generated images and their noise origins, which supports the generation diversity of Immiscible Diffusion.

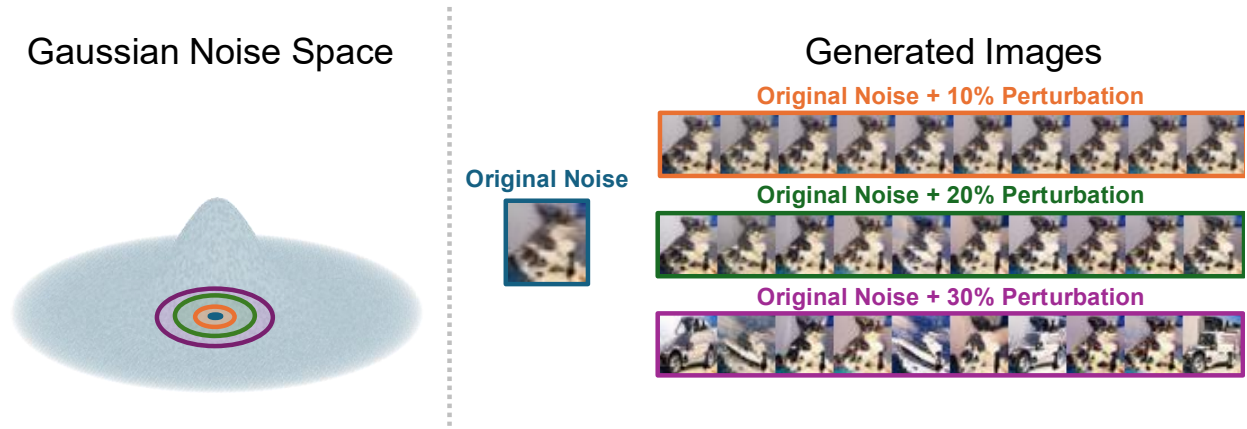


Figure 5.2: **Stable correlation between generated images and its noise origins.** Here perturbation means another Gaussian noise added to the fixed original Gaussian noise. Note that even with 20% perturbation, images changes are nearly unnoticeable. Only with 30% perturbation, a image object change happens. These demonstrate stable correlation from a noise area to a specific generated image.

5.3 Improved Immiscible Diffusion

[104] borrows Immiscible Diffusion, a physics concept describing solutes which cannot mix homogeneously, to assign images with near noise points via batch-wise linear assignment. While this accelerates diffusion training, the linear assignment is weak with only $\sim 2\%$ image-noise distance drop afterwards, and its computational complexity is $O(n^3)$, which scales up quickly with the batch size.

Therefore, to improve Immiscible Diffusion, we need to break the limit of using linear assignments. However, involving image-noise correlation generally triggers concerns on the diversity of generated images, as images are not diffused image to uncorrelated noises. Consequently, we need to take a deeper look into the generation diversity, to justify building correlations between images and noises.

5.3.1 Image-noise Correlation in the Denoising Processes

To evaluate the image-noise correlation during denoising, we generate images on a trained vanilla DDIM with a specific noise point and its ambient noise area. We first sample a specific noise point N_{orig} . Then we add diverse perturbations N_{pert} onto it, respectively. Specifically, we sample 10 independent Gaussian N_{pert} 's. The perturbed total noise can be expressed as,

$$N_{tot} = N_{orig} + W \cdot N_{pert} \quad (5.1)$$

Where W is the weight of the perturbation. We present images generated by the original and perturbed total noise in Figure 5.2. For example, we see N_{orig} generate a cat image. Surprisingly, with $W = 10\%$ on all 10 perturbation, we see that the generated images are still all cats without noticeable differences. Further, we see that $W = 20\%$ of perturbation only results in very slight image differences (like the background difference in the 1st and 2nd images from left), while no modal changes are observed. Only with 30% weight of perturbation, we observe image object changes in a few generated images. However, we can clearly find pixel-level similarities between these images despite modal differences. These results unequivocally prove that generated images are stably correlated with the sampled noise. As a result, while vanilla (miscible) diffusion models diffuse each image equally to the whole noise area, hoping to see that every image can be generated from a noise point or a small noise area, this goal cannot be achieved, nullifying the motivation of miscible diffusion in generation diversity, so as the concerns on Immiscible Diffusion’s generation diversity. In Section 5.4.3, we further quantitatively show that Immiscible Diffusion does not negatively impact the diversity of generated images.

5.3.2 Step-by-step Feature-level Benefits Analysis of Immiscible Diffusion

We now dive deeper to understand more clearly on how Immiscible Diffusion accelerates diffusion training, in order to help to accordingly design more implementations of it.

[104] suggests and mathematically proves that miscible diffusion causes difficulty in denoising at noisy layers. As shown in Figure 5.3 (a), the noisiest layer, referred as $\tau = S$ in DDIM [162], does not effectively predict the noise added. To confirm the ubiquity of such denoising difficulty, in Figure 5.3 (c) we show the $tSNE$ of the predicted images from each denoising layer $\tau = 0, \dots, S$. We collect 128 noise points, generating images with them, and logging the predicted images ($x_{pred,0}$ ’s) from each denoising layer, which are computed with the layer’s feature x_τ and the predicted noise $n_{pred,\tau}$ according to the noise schedule α_τ :

$$x_{pred,0} = \frac{1}{\sqrt{\alpha_\tau}}x_\tau - \frac{\sqrt{1-\alpha_\tau}}{\sqrt{\alpha_\tau}}n_{pred,\tau} \quad (5.2)$$

Points on the same line represent $x_{pred,0}$ ’s from the same initial noise on different τ ’s. Apparently, though different lines start from different noise points, they are chaotically tangled, suggesting that the same denoising goals are frequently shared between different denoising paths, which implies that the denoising difficulty commonly happens in the denoising. That is not surprising, as the vanilla DDIM takes miscible diffusion. To quantitatively express the miscibility, we stat the average L_2 distance between noise clusters (i.e. 10,000 noise points) assigned to each image. The distance for vanilla DDIM is only 0.92 ± 0.06 , proving that the diffusion process is truly miscible.

However, Immiscible Diffusion substantially overturns these difficulties. With the linear assignment implementation, we find that the average distance between noise clusters assigned

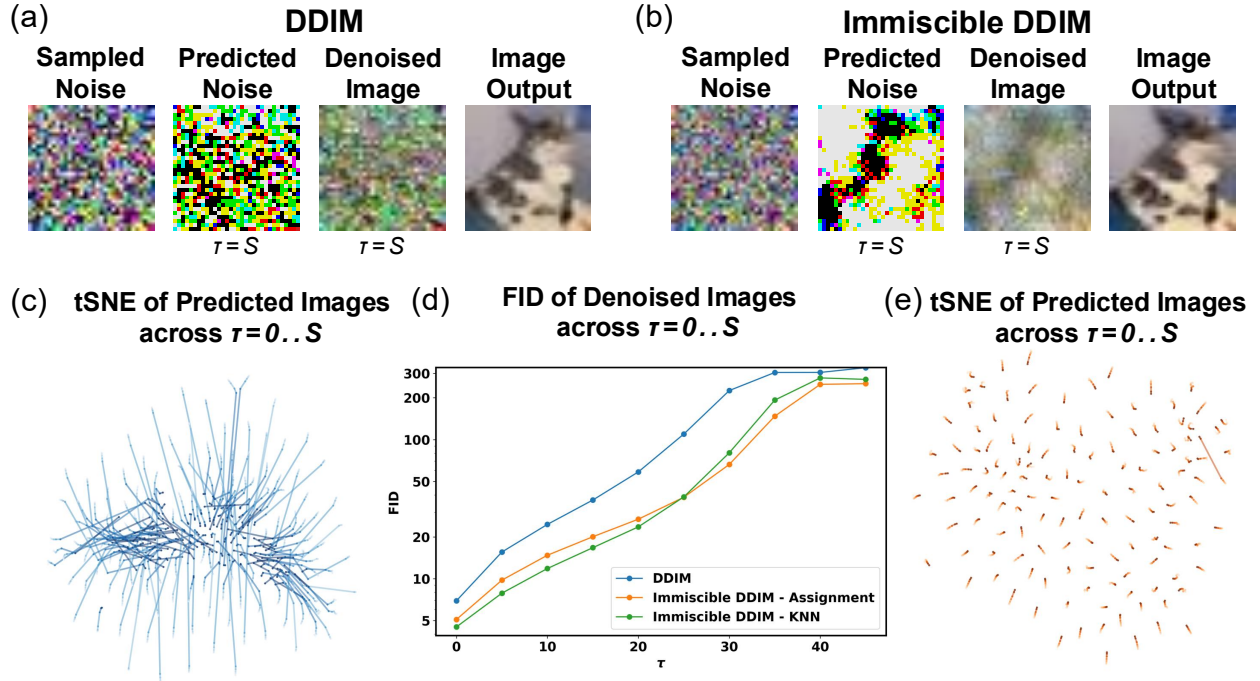


Figure 5.3: **Feature analysis of vanilla (miscible) and immiscible DDIM.** Referring to [162], $\tau = S$ represents the layer denoising from the pure noise. We show that Immiscible Diffusion activates the noisiest ($\tau \rightarrow S$) layers’ denoising functions by clarifying their denoising goals, as shown in the *tSNE* (t-Distributed Stochastic Neighbor Embedding) of denoised images across τ ’s. Such activation results in FID improvements on the denoised images from large τ ’s, which leads to better performance and faster convergence of diffusion models.

to each image is increased significantly to 4.11 ± 0.37 , suggesting that the noise clusters are more separate than the miscible diffusion, and thus is more immiscible by the definition. Figure 5.3 (b) illustrates that under Immiscible Diffusion, even the noisiest layer $\tau = S$ can effectively predict the noise and can denoise the image, and Figure 5.3 (e) shows that with Immiscible Diffusion, the *tSNE* figure shows much less intersections, implying that each noise point has its own stable denoising goal, which corresponds to the goal of easing denoising. As a result, as shown in Figure 5.3 (d), the FID of the denoised images from noisy layers exhibit significant improvement, which finally leads to the performance and training efficiency boost of Immiscible Diffusion.

The analysis above clearly figures out that the reduced miscibility helps to clarify the denoising goal, easing the denoising and helping it to be effective, and finally improve the performance of the miscible layers, and the final outputs. These step-by-step benefits of Immiscible Diffusion help us to extract the essence of it, and to design additional implemen-

tations in the following section.

5.3.3 Improved Immiscible Diffusion

As the benefits of Immiscible Diffusion can be traced back to the reduction of miscibility in diffusion, we naturally propose that the concept of Immiscible Diffusion should also reflect only the reduction of miscibility in diffusion, without unnecessary bounds to image-noise pairing. Under this improved concept, we argue that such assignment is only *one* way to achieve Immiscible Diffusion. In this chapter, we additionally propose two new Immiscible Diffusion implementations, which do not need linear assignment, or even image-noise correlations. Nevertheless, both methods exhibit excellent immiscibility, and therefore boost the training efficiency significantly.

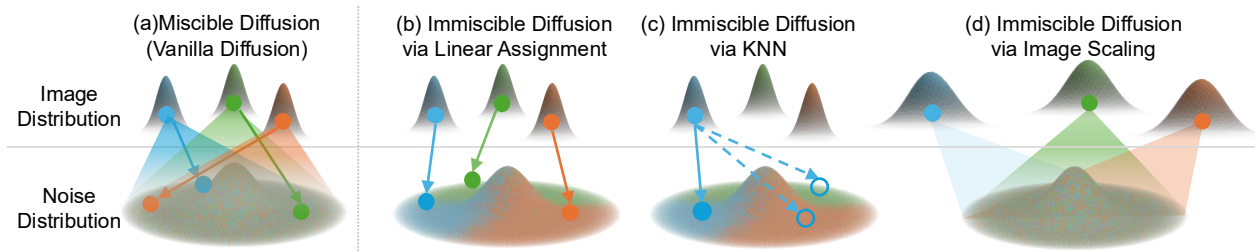


Figure 5.4: **Implementations of Immiscible Diffusion.** (a) **Miscible Diffusion** pairs the batch of images and noises randomly before adding noise to images. (b)(c)(d) **Immiscible Diffusion** tries to reduce the miscibility of diffusion by (b) L_2 linear assignment between the batch of images and noises and (c) sampling k noises and pick the nearest one (KNN) to use. (d) scaling images by multiplying their pixel values with a constant > 1 , which reduces overlaps between diffuse-able areas of different images.

Batch-wise Linear Assignment

Batch-wise linear assignment in [104] still qualifies Immiscible Diffusion. As shown in Figure 5.4 (b), this method performs a linear assignment [29] between the batch of images and noise points sampled. Such an assignment preferably assigns noise to nearby images while keeping the general distribution of all noises Gaussian. However, this trades off the running speed for each step when the batch size scales up, as the linear assignment is an $O(n^3)$ operation.

KNN Noise Selection

To avoid the scaling-up of execution time with larger batch sizes, we provide the second implementation of Immiscible Diffusion, the KNN method, where we sample k Gaussian noise points for each image and pick the one L_2 -closest to the image, as illustrated in Figure 5.4 (c). This method is very efficient - its execution time is only 0.2 ms for a batch size of

256 and would not scale up quickly when larger batches are used, as it is an $O(n)$ operation. The algorithm is shown in Algorithm 2.

Algorithm 2 Immiscible Diffusion Implementation - KNN Sampling

- 1: **Input:** Image x , k random noises $\{n_1, \dots, n_k\}$, noise schedule α_t
 - 2: $n \leftarrow \arg \min_{n_j \in \{n_1, n_2, \dots, n_k\}} \text{dist}(x, n_j)$
 - 3: $x_t \leftarrow \sqrt{\alpha_t}x + \sqrt{1 - \alpha_t} \cdot n$
 - 4: **Output:** Diffused image batch x_t
-

While the distribution of overall noise points used in training KNN-implemented Immiscible Diffusion is not guaranteed to be Gaussian, as some sampled noise points are selectively dropped, we argue that such discrepancy is negligible. To prove this, we sample $50k$ Gaussian noise points in the size of CIFAR-10 [89] images, i.e. $3 \times 32 \times 32$. Comparing them with the Gaussian distribution results in a KL divergence of 48.25. We then collect another $50k$ noise points which are *selected* in the KNN Immiscible Diffusion ($k = 8$), finding that their KL divergence to Gaussian is 48.60, which is only very slightly higher than noise points from a strict Gaussian distribution. Therefore, our KNN implementation does not significantly alter the Gaussian noise distribution.

Image Scaling

Though image-noise correlation is effective in building Immiscible Diffusion, there are also ways to reduce miscibility without it. A typical example is image scaling, *i.e.* to multiply the normed images' pixel values by a factor greater than 1, like 2 or 4. Such action has no influence on the noise space. However, the L_2 distance between each image is farther after the scaling, making the centers of diffused areas ($t < t_{max}$) of different images farther away from each other. Considering the noise amplitude at each diffusion step t is kept consistent, the diffused areas for different images at every step $t < t_{max}$ are naturally having less intersections, which constitutes Immiscible Diffusion. We will defer the immiscibility and performance experiments of this method to Section 5.4.6.

5.4 Effectiveness of Immiscible Diffusion

With the improved Immiscible Diffusion concept and the new and existing implementations, we perform a large series of experiments to systematically examine the benefits and the generalization ability of Immiscible Diffusion across various image generation tasks, models, and datasets, as well as extended tasks including image in-painting, out-painting, and robotics planning.

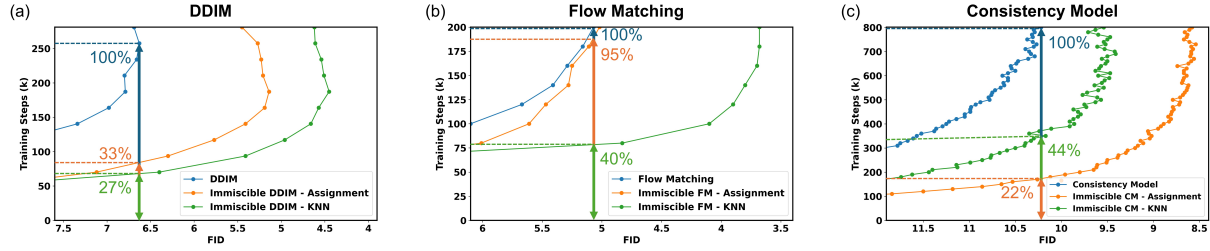


Figure 5.5: **Immiscible diffusion boosts training efficiency.** We show the training steps required to reach the best FID for vanilla models across **three** diverse diffusion-based architectures. Results consistently show that Immiscible Diffusion trains significantly faster.

5.4.1 Experiment Setups

We implement Immiscible Diffusion on diverse diffusion-based methods, including Consistency Models [166], DDIM [162], Stable Diffusion [149], and Flow Matching [111]. We train these implementations on a variety of popular datasets, including CIFAR-10 [89], ImageNet-1k [31] and MS-COCO [110]. The training hyper-parameters are discussed in Section C.1 and Table C.1.

5.4.2 Unconditional Image Generation Training

We compare the unconditional image generation training steps necessary to reach the best FID of the vanilla diffusion-based models in Figure 5.5. Astonishing and consistent training efficiency enhancement is observed across all diffusion-based models, despite their significant differences in diffusion trajectories (flows), denoising solvers and sampling step picking strategies. Across all experiments, we generally observe that the baseline diffusion-based models need a maximum of **2.5 to 4.5** times of training steps to achieve the same performance of their Immiscible Diffusion counterparts. These results strongly support the robust ability of Immiscible Diffusion in improving the training efficiency of diverse diffusion-based models.

5.4.3 Conditional Image Generation Training and Fine-tuning

Class-conditional Image Generation from Scratch. We conduct class-conditional image generation training from scratch on Stable Diffusion [149] and ImageNet-1k [31], whose result is shown in Figure 5.6 (a). Similar to the unconditional generations, Immiscible Diffusion exhibits much faster training.

However, since Immiscible Diffusion does not diffuse each image equally to the noise space, questions on whether Immiscible Diffusion models can still follow the prompts as good as vanilla models can be raised. Simply put, the answer is *Yes, they can*. In Section 5.3.1, we have shown that vanilla diffusion-based models yet have strong image-noise correla-

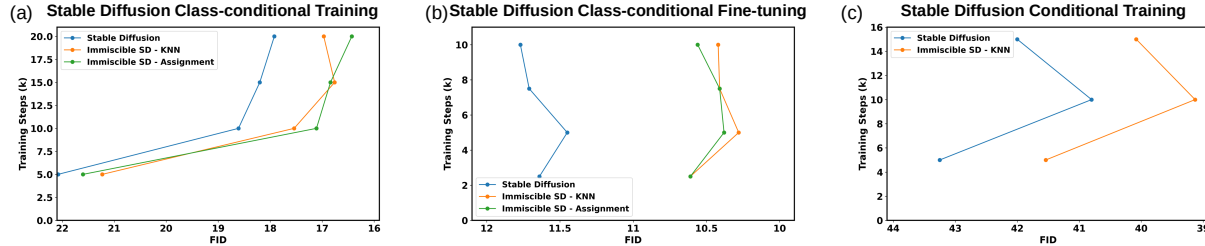


Figure 5.6: **Immiscible diffusion stays effective across diverse image generation tasks.** We demonstrate this with figures showing the training steps *v.s.* FID of (a)(b) Stable Diffusion class-conditional training and fine-tuning with ImageNet, and (c) conditional training with MS-COCO.

tions, so the diversity of generated images should not be influenced solely by the miscibility of the diffusion process. We further confirm this by evaluating the diversity of the generated images with CLIPScore [62], which shows that both the immiscible and the baseline models generate images with CLIPScores of 28.55, with a standard deviation of 0.01 and 0.02 respectively, indicating that Immiscible Diffusion does not hurt the image-prompt correspondence in complicated ImageNet dataset.

Class-conditional Image Generation Fine-tuning. Immiscible diffusion can also benefit the fine-tuning. We confirm this intuition with a class-conditional image generation fine-tuning experiment, which use ImageNet to fine-tunes Stable Diffusion which is pre-trained on LAION [155] by [149]. Results in Figure 5.6 show significant and consistent performance enhancements of Immiscible Diffusion compared to vanilla baselines. Note that our class-conditional generation uses class names as prompts instead of the class number, so the class-conditional fine-tuning and the conditional pre-training do not conflict in the form of the conditions.

Models	Vanilla SD	Immiscible SD KNN
In-painting	18.35	17.32
Out-painting	29.34	27.57

Table 5.1: FID evaluations of vanilla and Immiscible Diffusion in Image-to-image Tasks.

Free-prompt Conditional Image Generation from Scratch. To test Immiscible Diffusion’s effects on free prompts other than limited classes of prompts, we train Stable Diffusion [149] from scratch on MSCOCO [110]. Results also suggest a performance enhancement with Immiscible Diffusion.

5.4.4 Image Editing

We perform in-painting and out-painting, two classic image-to-image tasks, with vanilla and Immiscible Diffusion respectively. Both models are Stable Diffusion (SD) models trained on ImageNet dataset. For the in-painting,

we load images from ImageNet, replacing its center with Gaussian noise, and we let the model to repair the missing parts of the images. The out-painting task is similar, but everything other than the center part is replaced with Gaussian noise. We compare the completed images with the ImageNet dataset to calculate the FID, as shown in Table 5.1. We see that in both tasks the immiscible models exhibit better completed images. We further qualitatively provide a few comparisons in Figure C.1, where Immiscible Diffusion enjoys a better overall output which we ascribe to its better preserving of source information.

5.4.5 Robotics Planning



Figure 5.7: **Immiscible diffusion boosts the performance of diffusion policy.** Here the robot (circle) needs to push the T-shaped object (gray) into the desired place (green).

Experiment	Vanilla (Miscible)	Immiscible KNN k=2
Ave. Coverage for Last 10 Ckpts (%)	79.56%	82.83%
Max Coverage (%)	85.71%	86.74%

Table 5.2: Immiscible diffusion in robotics planning.

To further demonstrate the generalization ability of Immiscible Diffusion, we apply it onto PushT, a task in diffusion policy [24] for robotics, which let the robot to push a T-shaped object to a desired place. Our experiments are performed on the simulated PushT task explained in Figure 5.7, and with the data provided in [24]. Each experiment is trained for 3,000 epochs with 3 seeds, where the averages are taken as the results. We take the average area coverage

by the T-shaped object onto the desired destination as the metric, which is the same as [24].

The experiment results are shown in Table 5.2. We see that area coverage is significantly improved with Immiscible Diffusion. To illustrate this improvement more directly, we show a typical improvement case in Figure 5.7. We observe a more accurate pushing process without errors for Immiscible Diffusion policy where the vanilla one fails due to errors during the pushing.

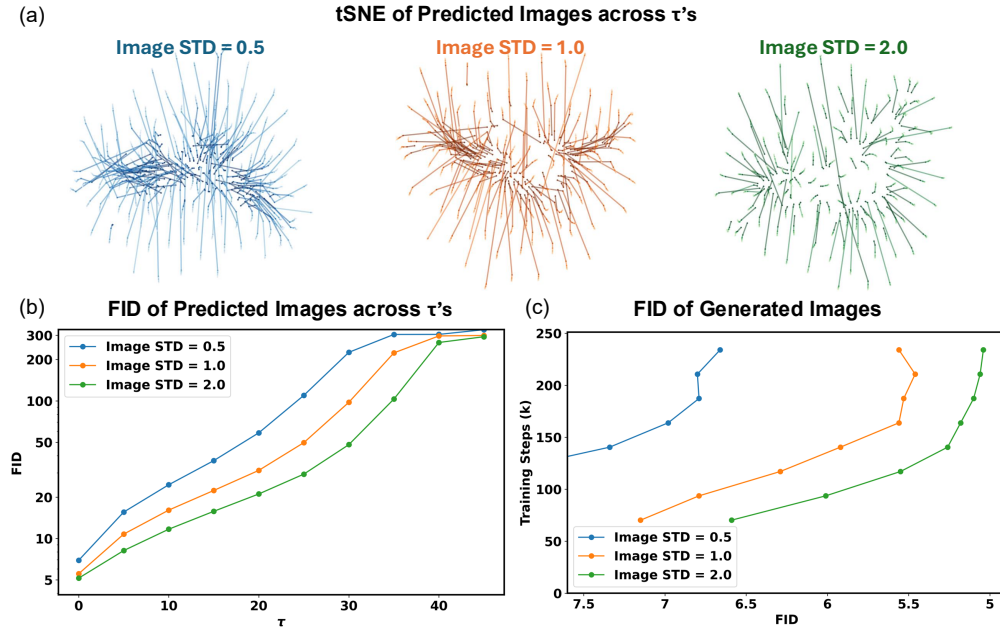


Figure 5.8: Analysis on DDIM with images normed to different pixel value STDs.

5.4.6 Immiscible Diffusion Beyond Image-noise Correlations

As discussed in Section 5.3.3, image scaling can intuitively achieve Immiscible Diffusion without enforcing image-noise correlations. Indeed, in Figure 5.8, we compare scaling pixel values of normed images to different STD's before performing diffusion with DDIM, whose default is to norm image to a pixel value $STD = 0.5$. Indeed, Figure 5.8 (a) shows that larger STD helps to reduce the confusion in denoising caused by miscible diffusion. Consequently in Figure 5.8 (b), we observe that experiments with larger image STD enjoys lower FIDs of predicted images during denoising, and finally in Figure 5.8 performs better with faster training and better convergence performance. These results show that Immiscible Diffusion is not necessarily bounded with image-noise correlations.

5.5 Discussions

In this section, we discuss the distinction of Immiscible Diffusion to some seemingly similar methods, and compare our Immiscible Diffusion implementations to highlight their respective advantages.

What is the difference between Immiscible Diffusion, flow matching [111] and rectified flow [114]?

We compare Immiscible Diffusion with flow matching [111] and vanilla DDIM [162] in Figure C.3. Compared to DDIM, flow matching linearizes the diffusion trajectory. However,

its diffusion trajectories (flows) can still arrive at the same noise point from different images, so flow matching is still miscible. Immiscible flow matching aims to reduce such miscibility to let each diffusion trajectory mix less, so as to ease the denoising.

Rectified flow [114] can also make *denoising* paths distinct. However, in their method, each image is still *diffused* to all the noise space, which means their diffusion is still miscible.

What is the relation between Immiscible Diffusion and batch-wise OT?

Batch-wise OT (linear assignment) can be *one* of the Immiscible Diffusion implementations. It preferably assign noise to closer images, so images would not be diffused to the whole noise space. Therefore, the mixing of diffusion trajectories from different images reduces, and the diffusion is more immiscible. Step-by-step feature analysis in Section 5.3.1 clearly demonstrate how the linear assignment makes diffusion immiscible and boost the diffusion’s performance. At the same time, batch-wise OT only reduces the image-noise average distance by $\sim 2\%$ [104] and the denoising STD by $\sim 4\%$ [136]. Therefore, we attribute batch-wise OT’s training efficiency boosts mainly to its realization of Immiscible Diffusion.

Which implementation should I use, batch-wise Linear assignment or KNN?

While there are numerous ways to achieve Immiscible Diffusion, here we compare the implementations we take for the reader’s reference. Since the setting of image scaling depends heavily on the image normalization method taken by the baseline diffusion models, we focus on comparison between batch-wise linear assignment and KNN. While both methods help to enforce the image-noise correlation during diffusion, batch-wise linear assignment keeps the noise space strictly Gaussian, and achieve better immiscibility in the noise space. The average L_2 distance between noise clusters assigned to each image for batch-wise linear assignment is 4.11 ± 0.37 , which is higher than that of KNN, which is 2.17 ± 0.35 . Therefore, as shown in Figure 5.3 (d), it achieves better FID in noisy layers than KNN. However, batch-wise linear assignment suffers from computational cost of $O(n^3)$, which is higher than the KNN’s $O(n)$. Furthermore, in Figure C.2, we show the noise points assigned to the same image using batch-wise assignment and KNN. We observe that the KNN noise points distribute in a more continuous manner, which we posit to contribute to KNN’s better performance in denoising un-noisy layers, as also indicated in Figure 5.3 (d).

5.6 Conclusion

Conclusion. In this chapter, we systematically revisit Immiscible Diffusion, a physics-inspired method aiming to boost diffusion training efficiency. Our experiments firstly show that image-noise correlation introduced by Immiscible Diffusion empirically does not alter the diversity of generated images due to the intrinsic image-noise correlation in vanilla diffusion models. Detailed feature analysis shows how Immiscible Diffusion step-by-step enhances the denoising outputs. Based on these findings, we improve the Immiscible Diffusion concept, which does not require doing image-noise pairing nor even image-noise correlations. We offer a few new Immiscible Diffusion implementations, achieving training efficiency boosts up to $> 4\times$, across diverse baseline diffusion models and on various image generation tasks, image

editing and robotics planning. Our method points out the diffusion trajectory miscibility problem, a generally existing problem in diffusion training dragging its efficiency, which could be a fundamental direction to explore towards high-efficiency diffusion training, and towards large-scale vision-based driving data synthesis.

Limitations. While we propose diverse methods to reduce miscibility in diffusion training, there should be more methods to handle the miscibility problem, which will be a broader area to explore.

Chapter 6

Final Words

6.1 Summary of Key Contributions

This dissertation has presented a comprehensive framework for addressing the critical challenge of data scarcity in autonomous driving. The core contribution of this work lies not in direct physical data collection, but in paving diverse pathways to maximize the utility of existing data and to generate new data automatically: through augmentation, semantic enrichment, and efficient generation, all without any additional human labor. Together, these works constitute a robust toolbox for overcoming the data bottlenecks inherent in learning-based autonomous systems.

Chapter 2 introduces a driving data augmentation method that leverages simple geometric transformations for augmenting maps and existing planner on generating vehicle trajectories. We demonstrate that performing self-supervised pre-training on this synthesized data results in a significant improvement in the accuracy of motion prediction models.

Chapter 3 presents WOMD-Reasoning, a language annotation dataset automatically generated from the Waymo Open Motion Dataset. This work distills interactions governed by traffic rules and human intentions into explicit natural language representations. The resulting dataset significantly enriches the semantic density of existing motion data, enabling downstream models to comprehend complex interactions and incorporate them into the prediction loop. Building on this, we introduce Motion-LLaVA, a motion-language model fine-tuned on WOMD-Reasoning. With the integration of Motion-LLaVA, we again observe a substantial enhancement in the performance of motion prediction models.

Chapter 4 and Chapter 5 identify miscibility, an unnecessary inductive bias in diffusion-based models, as a critical factor that compromises training efficiency and impedes the training of large-scale generation model for vision-based driving data. We theoretically prove that miscibility not only complicates the denoising process but also contributes little to generation diversity. Based on this theoretical insight, we propose immiscible diffusion, a method designed to reduce miscibility, *i.e.*, the mixing of diffusion paths during the diffusion process. We provide several implementations that achieve significant training efficiency gains

across diverse diffusion backbones. These contributions pave the way for training generative models efficiently, positioning them as powerful data engines for the next generation of autonomous driving.

6.2 Implications and Impact

The research presented in this dissertation offers significant implications for tackling data scarcity in autonomous driving, with broader impacts extending to the representation learning, Large Language Model (LLM), and generative modeling communities. By leveraging heuristic augmentation, extracting latent semantic information, and optimizing the training of vision data generation models, vast quantities of high-fidelity driving data can be accessed without incurring the costs of physical mileage or expensive sensor instrumentation. The author hopes these methods can provide a road-map for solving learning-based problems under strict data budgets, offering profound insights for the general AI community regarding data efficiency and scalability.

6.3 Future Directions to Explore

While this dissertation has made significant strides in addressing data scarcity, it also illuminates several promising avenues for future research:

- Integrating data augmentation, generative synthesis, and semantic enrichment into a unified pipeline to achieve state-of-the-art driving performance under extremely limited data budgets. As demonstrated in Chapter 2, data augmentation methods can significantly expand existing datasets. These datasets can be further enriched with language annotations using the techniques proposed in Chapter 3. Furthermore, vision modalities can be synthesized using models trained with the acceleration methods introduced in Chapters 4 and 5. By leveraging this multi-modal augmented data, autonomous driving models can be trained to achieve superior accuracy and safety.
- Exploring high-reliability, automated human-aligned evaluation metrics to continuously elevate the quality of generated language datasets, such as WOMD-Reasoning, ensuring their suitability for safety-critical tasks. While human evaluation is effective for quality control, it is often prohibitively expensive and hard to scale. However, without rigorous validation, the quality of language datasets falls short of the strict requirements for safety-critical models. Therefore, further exploration into high-accuracy automated validation of language annotations is imperative.
- Designing semantically meaningful image-noise correlations in diffusion-based models to enhance the controllability of the generation process. In Chapters 4 and 5, images are assigned to noise points based on proximity in the pixel space. While this improves training efficiency, such assignment lacks semantic alignment, thereby not benefiting

the controllability of the diffusion models. Future research on mapping images to noise regions based on semantic meanings will enable models to generate images with specific semantic attributes, greatly enhancing their prompt-following capabilities.

- Utilizing diffusion as a unified foundation model to perform holistic multi-modal predictions, encompassing motion trajectories, video prediction, and high-level linguistic predictive reasoning. By sharing information across modalities within a unified diffusion-based backbone, one can foster stronger cross-modality interactions and establish a mutual relationship between vision, motion, and language predictions.
- Leveraging large-scale driving data synthesized by diffusion models to perform unsupervised pre-training, further boosting the robustness and generalizability of driving models. Since diffusion-based models are trained on vast corpora of general-purpose images and videos, they encode broad world knowledge that driving-specific models could lack. Utilizing these models to synthesize visual driving data offers a promising avenue for transferring this general knowledge into driving agents, thereby improving their robustness and generalization abilities.

6.4 Concluding Remarks

In conclusion, this dissertation establishes a diverse and effective set of methodologies to augment expensive driving data without the need for real-world driving costs. The methods introduced, ranging from geometric augmentations and planner-based synthesis to semantic language annotations, have been demonstrated to be highly effective in improving motion prediction accuracy, a cornerstone task in autonomous driving. Furthermore, our immiscible diffusion series renders the training of diffusion models computationally affordable, establishing them as a viable data engine in the era of generative AI. These works collectively offer a comprehensive solution to the data scarcity problem, which persists as one of the most significant obstacles in the autonomous driving field. Through the continuous expansion of data quantity and richness, these approaches have proven and will continue to prove instrumental in advancing autonomous driving technologies toward a stage that is safer, more reliable, and more explainable.

Bibliography

- [1] Peter Anderson, Xiaodong He, Dhruv Batra, Devi Parikh, Michael Lee, Alina Ehsan, and C. Lawrence Zitnick. “SPICE: Semantic Propositional Image Caption Evaluation”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2016, pp. 382–398.
- [2] Hidehisa Arai, Keita Miwa, Kento Sasaki, Yu Yamaguchi, Kohei Watanabe, Shunsuke Aoki, and Issei Yamamoto. “CoVLA: Comprehensive Vision-Language-Action Dataset for Autonomous Driving”. In: *arXiv preprint arXiv:2408.10845* (2024).
- [3] Andrea Asperti, Davide Evangelista, Samuele Marro, and Fabio Merizzi. “Image embedding for denoising generative models”. In: *Artificial Intelligence Review* 56.12 (2023), pp. 14511–14533.
- [4] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. “Self-Supervised Learning From Images With a Joint-Embedding Predictive Architecture”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2023, pp. 15619–15629.
- [5] Caio Azevedo, Thomas Gilles, Stefano Sabatini, and Dzmitry Tsishkou. “Exploiting map information for self-supervised learning in motion forecasting”. In: *arXiv preprint arXiv:2210.04672* (2022).
- [6] Mohammadhossein Bahari, Saeed Saadatnejad, Ahmad Rahimi, Mohammad Shaverdikondori, Amir-Hossein Shahidzadeh, Seyed-Mohsen Moosavi-Dezfooli, and Alexandre Alahi. “Vehicle trajectory prediction works, but not everywhere”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022.
- [7] Shuai Bai et al. “Qwen2.5-vl technical report”. In: *arXiv preprint arXiv:2502.13923* (2025).
- [8] Satanjeev Banerjee and Alon Lavie. “METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments”. In: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. 2005, pp. 65–72.

- [9] Prarthana Bhattacharyya, Chengjie Huang, and Krzysztof Czarnecki. “SSL-lanes: Self-supervised learning for motion forecasting in autonomous driving”. In: *Conference on Robot Learning*. PMLR. 2023, pp. 1793–1805.
- [10] Kevin Black, Mitsuhiro Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. “Zero-shot robotic manipulation with pretrained image-editing diffusion models”. In: *arXiv preprint arXiv:2310.10639* (2023).
- [11] Mariusz Bojarski et al. “End to end learning for self-driving cars”. In: *arXiv preprint arXiv:1604.07316* (2016).
- [12] Tom Brown et al. “Language models are few-shot learners”. In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.
- [13] Holger Caesar et al. *nuScenes: A multimodal dataset for autonomous driving*. 2020. arXiv: [1903.11027](https://arxiv.org/abs/1903.11027) [cs.LG].
- [14] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. “MultiPath: Multiple Probabilistic Anchor Trajectory Hypotheses for Behavior Prediction”. In: *Proceedings of the Conference on Robot Learning*. Ed. by Leslie Pack Kaelbling, Dancica Kragic, and Komei Sugiura. Vol. 100. Proceedings of Machine Learning Research. PMLR, Oct. 2020, pp. 86–99. URL: <https://proceedings.mlr.press/v100/chai20a.html>.
- [15] Ming-Fang Chang et al. “Argoverse: 3d tracking and forecasting with rich maps”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 8748–8757.
- [16] Jannis Chemseddine, Paul Hagemann, Gabriele Steidl, and Christian Wald. *Conditional Wasserstein Distances with Applications in Bayesian OT Flow Matching*. 2024. arXiv: [2403.18705](https://arxiv.org/abs/2403.18705) [cs.LG]. URL: <https://arxiv.org/abs/2403.18705>.
- [17] Hao Chen, Jiaze Wang, Kun Shao, Furui Liu, Jianye Hao, Chenyong Guan, Guangyong Chen, and Pheng-Ann Heng. “Traj-MAE: Masked Autoencoders for Trajectory Prediction”. In: *arXiv preprint arXiv:2303.06697* (2023).
- [18] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snaveley, and Yoav Artzi. “Touchdown: Natural language navigation and spatial reasoning in visual street environments”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12538–12547.
- [19] Kan Chen et al. *WOMD-LiDAR: Raw Sensor Dataset Benchmark for Motion Forecasting*. 2024. arXiv: [2304.03834](https://arxiv.org/abs/2304.03834) [cs.CV]. URL: <https://arxiv.org/abs/2304.03834>.
- [20] Long Chen, Oleg Sinavski, Jan Hünemann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. *Driving with LLMs: Fusing Object-Level Vector Modality for Explainable Autonomous Driving*. 2023. arXiv: [2310.01957](https://arxiv.org/abs/2310.01957) [cs.R0].

- [21] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. “A simple framework for contrastive learning of visual representations”. In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607.
- [22] Ting Chen, Saurabh Saxena, Lala Li, David J. Fleet, and Geoffrey Hinton. “Pix2seq: A Language Modeling Framework for Object Detection”. In: *International Conference on Learning Representations 2022*. 2022.
- [23] Jie Cheng, Xiaodong Mei, and Ming Liu. “Forecast-MAE: Self-supervised Pre-training for Motion Forecasting with Masked Autoencoders”. In: *arXiv preprint arXiv:2308.09882* (2023).
- [24] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. “Diffusion Policy: Visuomotor Policy Learning via Action Diffusion”. In: *Proceedings of Robotics: Science and Systems (RSS)*. 2023.
- [25] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. “Diffusion policy: Visuomotor policy learning via action diffusion”. In: *arXiv preprint arXiv:2303.04137* (2023).
- [26] Zheng Chu et al. “A survey of chain of thought reasoning: Advances, frontiers and future”. In: *arXiv preprint arXiv:2309.15402* (2023).
- [27] SAE On-Road Automated Vehicle Standards Committee et al. “Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems”. In: *SAE Standard J 3016* (2014), p. 1.
- [28] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. “The cityscapes dataset for semantic urban scene understanding”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3213–3223.
- [29] David F Crouse. “On implementing 2D rectangular assignment algorithms”. In: *IEEE Transactions on Aerospace and Electronic Systems* 52.4 (2016), pp. 1679–1696.
- [30] Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, and Ziran Wang. “Drive As You Speak: Enabling Human-Like Interaction With Large Language Models in Autonomous Vehicles”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*. Jan. 2024, pp. 902–909.
- [31] 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*. Ieee. 2009, pp. 248–255.
- [32] Wei Deng, Weijian Luo, Yixin Tan, Marin Biloš, Yu Chen, Yuriy Nevmyvaka, and Ricky T. Q. Chen. *Variational Schrödinger Diffusion Models*. 2024. arXiv: [2405.04795](https://arxiv.org/abs/2405.04795) [cs.LG]. URL: <https://arxiv.org/abs/2405.04795>.

- [33] Thierry Deruyttere, Simon Vandenhende, Dusan Grujicic, Luc Van Gool, and Marie-Francine Moens. “Talk2car: Taking control of your self-driving car”. In: *arXiv preprint arXiv:1909.10838* (2019).
- [34] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *Proceedings of naacL-HLT*. Vol. 1. 2019, p. 2.
- [35] Xinpeng Ding, Jianhua Han, Hang Xu, Xiaodan Liang, Wei Zhang, and Xiaomeng Li. “Holistic Autonomous Driving Understanding by Bird’s-Eye-View Injected Multi-Modal Large Models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2024, pp. 13668–13677.
- [36] Xinpeng Ding, Jianhua Han, Hang Xu, Xiaodan Liang, Wei Zhang, and Xiaomeng Li. “Holistic autonomous driving understanding by bird’s-eye-view injected multi-modal large models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 13668–13677.
- [37] Zhezhang Ding and Huijing Zhao. “Incorporating Driving Knowledge in Deep Learning Based Vehicle Trajectory Prediction: A Survey”. In: *IEEE Transactions on Intelligent Vehicles* 8.8 (2023), pp. 3996–4015. DOI: [10.1109/TIV.2023.3266446](https://doi.org/10.1109/TIV.2023.3266446).
- [38] Lawrence W Dowdy and Derrell V Foster. “Comparative models of the file assignment problem”. In: *ACM Computing Surveys (CSUR)* 14.2 (1982), pp. 287–313.
- [39] Ashok Elluswamy. *Building an Autonomous Future: End-to-End Foundation Models for Self-Driving*. Keynote at ICCV 2025 Workshop on Foundation Models for Autonomous Driving (WDFM-AD). Honolulu, HI, Oct. 2025.
- [40] Ashok Elluswamy. *Foundation Models for Autonomy*. Keynote at CVPR 2023 Workshop on End-to-End Autonomous Driving. June 2023.
- [41] Scott Ettinger et al. *Large Scale Interactive Motion Forecasting for Autonomous Driving: The Waymo Open Motion Dataset*. 2021. arXiv: [2104.10133](https://arxiv.org/abs/2104.10133) [cs.CV].
- [42] Scott Ettinger et al. “Large Scale Interactive Motion Forecasting for Autonomous Driving: The Waymo Open Motion Dataset”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 9710–9719.
- [43] Weiyu Feng, Seth Z. Zhao, Chuanyu Pan, Adam Chang, Yichen Chen, Zekun Wang, and Allen Y. Yang. “Digital Twin Tracking Dataset (DTTD): A New RGB+Depth 3D Dataset for Longer-Range Object Tracking Applications”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2023, pp. 3288–3297.
- [44] Chaoyou Fu et al. “Vita-1.5: Towards gpt-4o level real-time vision and speech interaction”. In: *arXiv preprint arXiv:2501.01957* (2025).

- [45] Daocheng Fu, Xin Li, Licheng Wen, Min Dou, Pinlong Cai, Botian Shi, and Yu Qiao. *Drive Like a Human: Rethinking Autonomous Driving with Large Language Models*. 2023. arXiv: [2307.07162 \[cs.R0\]](#).
- [46] Xiaotang Gai, Chenyi Zhou, Jiayang Liu, Yang Feng, Jian Wu, and Zuozhu Liu. *MedThink: Explaining Medical Visual Question Answering via Multimodal Decision-Making Rationale*. 2024. arXiv: [2404.12372 \[cs.CV\]](#). URL: <https://arxiv.org/abs/2404.12372>.
- [47] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. “Vectornet: Encoding hd maps and agent dynamics from vectorized representation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11525–11533.
- [48] Peng Gao, Teli Ma, Hongsheng Li, Ziyi Lin, Jifeng Dai, and Yu Qiao. “MCMAE: Masked Convolution Meets Masked Autoencoders”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 35632–35644.
- [49] Peng Gao et al. “LLaMA-Adapter V2: Parameter-Efficient Visual Instruction Model”. In: *arXiv preprint arXiv:2304.15010* (2023).
- [50] Andreas Geiger, Philip Lenz, and Raquel Urtasun. “Are we ready for autonomous driving? the kitti vision benchmark suite”. In: *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, pp. 3354–3361.
- [51] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, and Barret Zoph. “Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation”. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 2917–2927. DOI: [10.1109/CVPR46437.2021.00294](#).
- [52] Paul C Gilmore. “Optimal and suboptimal algorithms for the quadratic assignment problem”. In: *Journal of the society for industrial and applied mathematics* 10.2 (1962), pp. 305–313.
- [53] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. “Generative adversarial nets”. In: *Advances in neural information processing systems* 27 (2014).
- [54] Ross Greer and Mohan Trivedi. “Towards explainable, safe autonomous driving with language embeddings for novelty identification and active learning: Framework and experimental analysis with real-world data sets”. In: *arXiv preprint arXiv:2402.07320* (2024).
- [55] Junru Gu, Chen Sun, and Hang Zhao. “Densetnt: End-to-end trajectory prediction from dense goal sets”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 15303–15312.

- [56] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. “Social gan: Socially acceptable trajectories with generative adversarial networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2255–2264.
- [57] Wencheng Han, Dongqian Guo, Cheng-Zhong Xu, and Jianbing Shen. “Dme-driver: Integrating human decision logic and 3d scene perception in autonomous driving”. In: *arXiv preprint arXiv:2401.03641* (2024).
- [58] 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 (CVPR)*. June 2022, pp. 16000–16009.
- [59] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. “Momentum contrast for unsupervised visual representation learning”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 9729–9738.
- [60] Kaiming He, Ross Girshick, and Piotr Dollar. “Rethinking ImageNet Pre-Training”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019.
- [61] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. *CLIP-Score: A Reference-free Evaluation Metric for Image Captioning*. 2022. arXiv: [2104.08718](https://arxiv.org/abs/2104.08718) [cs.CV]. URL: <https://arxiv.org/abs/2104.08718>.
- [62] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. “CLIP-Score: A Reference-free Evaluation Metric for Image Captioning”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2021, pp. 7514–7528.
- [63] 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*. 2018. arXiv: [1706.08500](https://arxiv.org/abs/1706.08500) [cs.LG]. URL: <https://arxiv.org/abs/1706.08500>.
- [64] 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*. 2018. arXiv: [1706.08500](https://arxiv.org/abs/1706.08500) [cs.LG].
- [65] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising diffusion probabilistic models”. In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851.
- [66] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising diffusion probabilistic models”. In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851.
- [67] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. “Video diffusion models”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 8633–8646.

- [68] Jonathan Ho et al. “Imagen video: High definition video generation with diffusion models”. In: *arXiv preprint arXiv:2210.02303* (2022).
- [69] Yihan Hu et al. “Planning-oriented autonomous driving”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 17853–17862.
- [70] Xingchang Huang, Corentin Salaun, Cristina Vasconcelos, Christian Theobalt, Cengiz Oztireli, and Gurprit Singh. “Blue noise for diffusion models”. In: *ACM SIGGRAPH 2024 Conference Papers*. SIGGRAPH ’24. Denver, CO, USA: Association for Computing Machinery, 2024. ISBN: 9798400705250. DOI: [10.1145/3641519.3657435](https://doi.org/10.1145/3641519.3657435). URL: <https://doi.org/10.1145/3641519.3657435>.
- [71] Yuichi Inoue, Yuki Yada, Kotaro Tanahashi, and Yu Yamaguchi. “NuScenes-MQA: Integrated Evaluation of Captions and QA for Autonomous Driving Datasets Using Markup Annotations”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*. Jan. 2024, pp. 930–938.
- [72] Petros Ioannou. *Automated highway systems*. Springer Science & Business Media, 1997.
- [73] Thibaut Issenhuth, Ludovic Dos Santos, Jean-Yves Franceschi, and Alain Rakotomamonjy. *Improving Consistency Models with Generator-Induced Coupling*. 2024. arXiv: [2406.09570](https://arxiv.org/abs/2406.09570) [cs.LG]. URL: <https://arxiv.org/abs/2406.09570>.
- [74] Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. *Rethinking FID: Towards a Better Evaluation Metric for Image Generation*. 2024. arXiv: [2401.09603](https://arxiv.org/abs/2401.09603) [cs.CV]. URL: <https://arxiv.org/abs/2401.09603>.
- [75] Chiyu Max Jiang, Andre Cornman, Cheolho Park, Ben Sapp, Yin Zhou, and Dragomir Anguelov. *MotionDiffuser: Controllable Multi-Agent Motion Prediction using Diffusion*. 2023. arXiv: [2306.03083](https://arxiv.org/abs/2306.03083) [cs.RO].
- [76] Bu Jin, Xinyu Liu, Yupeng Zheng, Pengfei Li, Hao Zhao, Tong Zhang, Yuhang Zheng, Guyue Zhou, and Jingjing Liu. *ADAPT: Action-aware Driving Caption Transformer*. 2023. arXiv: [2302.00673](https://arxiv.org/abs/2302.00673) [cs.CV]. URL: <https://arxiv.org/abs/2302.00673>.
- [77] Bu Jin et al. “Tod3cap: Towards 3d dense captioning in outdoor scenes”. In: *European Conference on Computer Vision*. Springer. 2024, pp. 367–384.
- [78] Nidhi Kalra and Susan M Paddock. “Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?” In: *Transportation research part A: policy and practice* 94 (2016), pp. 182–193.
- [79] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. “Analyzing and improving the training dynamics of diffusion models”. In: *arXiv preprint arXiv:2312.02696* (2023).

- [80] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. “Analyzing and improving the training dynamics of diffusion models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 24174–24184.
- [81] Valentin Khrulkov, Gleb Ryzhakov, Andrei Chertkov, and Ivan Oseledets. “Understanding DDPM Latent Codes Through Optimal Transport”. In: *The Eleventh International Conference on Learning Representations*. 2023. URL: <https://openreview.net/forum?id=6PIrhAx1j4i>.
- [82] Daegyu Kim, Jooyoung Choi, Chaehun Shin, Uiwon Hwang, and Sungroh Yoon. *Improving Diffusion-Based Generative Models via Approximated Optimal Transport*. 2024. arXiv: 2403.05069 [cs.CV]. URL: <https://arxiv.org/abs/2403.05069>.
- [83] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. “Consistency Trajectory Models: Learning Probability Flow ODE Trajectory of Diffusion”. In: *arXiv preprint arXiv:2310.02279* (2023).
- [84] Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. *Textual Explanations for Self-Driving Vehicles*. 2018. arXiv: 1807.11546 [cs.CV].
- [85] Sihyeon Kim, Sanghyeok Lee, Dasol Hwang, Jaewon Lee, Seong Jae Hwang, and Hyunwoo J. Kim. “Point Cloud Augmentation With Weighted Local Transformations”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 548–557.
- [86] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2022. arXiv: 1312.6114 [stat.ML]. URL: <https://arxiv.org/abs/1312.6114>.
- [87] Akio Kodaira et al. “StreamDiffusion: A Pipeline-level Solution for Real-time Interactive Generation”. In: *arXiv preprint arXiv:2312.12491* (2023).
- [88] Stepan Konev. *MPA: MultiPath++ Based Architecture for Motion Prediction*. 2022. DOI: 10.48550/ARXIV.2206.10041. URL: <https://arxiv.org/abs/2206.10041>.
- [89] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. Tech. rep. University of Toronto, 2009. URL: <https://www.cs.toronto.edu/~kriz/cifar.html>.
- [90] Alex Krizhevsky, Geoffrey Hinton, et al. “Learning multiple layers of features from tiny images”. In: *Toronto, ON, Canada* (2009).
- [91] H. W. Kuhn. “The Hungarian method for the assignment problem”. In: *Naval Research Logistics Quarterly* 2.1-2 (1955), pp. 83–97. DOI: <https://doi.org/10.1002/nav.3800020109>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/nav.3800020109>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109>.

- [92] Harold W Kuhn. “The Hungarian method for the assignment problem”. In: *Naval research logistics quarterly* 2.1-2 (1955), pp. 83–97.
- [93] Yen-Ling Kuo, Xin Huang, Andrei Barbu, Stephen G. McGill, Boris Katz, John J. Leonard, and Guy Rosman. *Trajectory Prediction with Linguistic Representations*. 2022. arXiv: [2110.09741](https://arxiv.org/abs/2110.09741) [cs.R0].
- [94] Kristofer D Kusano, John M Scanlon, Yin-Hsiu Chen, Timothy L McMurry, Ruoshu Chen, Tilia Gode, and Trent Victor. “Comparison of Waymo rider-only crash data to human benchmarks at 7.1 million miles”. In: *Traffic Injury Prevention* 25.sup1 (2024), S66–S77.
- [95] Dohyun Kwon, Ying Fan, and Kangwook Lee. “Score-based Generative Modeling Secretly Minimizes the Wasserstein Distance”. In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho. 2022. URL: <https://openreview.net/forum?id=oPzICxVFqVM>.
- [96] Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. “The role of imagenet classes in fr\’echet inception distance”. In: *arXiv preprint arXiv:2203.06026* (2022).
- [97] Sangyun Lee, Beomsu Kim, and Jong Chul Ye. *Minimizing Trajectory Curvature of ODE-based Generative Models*. 2023. arXiv: [2301.12003](https://arxiv.org/abs/2301.12003) [cs.LG]. URL: <https://arxiv.org/abs/2301.12003>.
- [98] Sangyun Lee, Beomsu Kim, and Jong Chul Ye. “Minimizing Trajectory Curvature of ODE-based Generative Models”. In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, July 2023, pp. 18957–18973. URL: <https://proceedings.mlr.press/v202/lee23j.html>.
- [99] Boyi Li, Yue Wang, Jiageng Mao, Boris Ivanovic, Sushant Veer, Karen Leung, and Marco Pavone. “Driving everywhere with large language model policy adaptation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 14948–14957.
- [100] Muyang Li, Ji Lin, Chenlin Meng, Stefano Ermon, Song Han, and Jun-Yan Zhu. *Efficient Spatially Sparse Inference for Conditional GANs and Diffusion Models*. 2023. arXiv: [2211.02048](https://arxiv.org/abs/2211.02048) [cs.CV].
- [101] Quanyi Li, Zhenghao Peng, Lan Feng, Zhizheng Liu, Chenda Duan, Wenjie Mo, and Bolei Zhou. “ScenarioNet: Open-Source Platform for Large-Scale Traffic Scenario Simulation and Modeling”. In: *Advances in Neural Information Processing Systems* (2023).

- [102] Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. “Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).
- [103] Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. “Q-Diffusion: Quantizing Diffusion Models”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2023, pp. 17535–17545.
- [104] Yiheng Li, Heyang Jiang, Akio Kodaira, Masayoshi TOMIZUKA, Kurt Keutzer, and Chenfeng Xu. “Immiscible Diffusion: Accelerating Diffusion Training with Noise Assignment”. In: *Advances in Neural Information Processing Systems*. Ed. by A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang. Vol. 37. Curran Associates, Inc., 2024, pp. 90198–90225. URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/a422a2f016c14406a01ddba731c0969a-Paper-Conference.pdf.
- [105] Yiheng Li, Feng Liang, Dan Kondratyuk, Masayoshi Tomizuka, Kurt Keutzer, and Chenfeng Xu. “Improved Immiscible Diffusion: Accelerate Diffusion Training by Reducing Its Miscibility”. In: *arXiv preprint arXiv:2505.18521* (2025).
- [106] Yiheng Li, Seth Z Zhao, Chenfeng Xu, Chen Tang, Chenran Li, Mingyu Ding, Masayoshi Tomizuka, and Wei Zhan. “Pre-training on synthetic driving data for trajectory prediction”. In: *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2024, pp. 5910–5917.
- [107] Yiheng Li et al. “WOMD-Reasoning: A Large-Scale Dataset for Interaction Reasoning in Driving”. In: *Proceedings of the 42nd International Conference on Machine Learning*. Ed. by Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu. Vol. 267. Proceedings of Machine Learning Research. PMLR, July 2025, pp. 34288–34311. URL: <https://proceedings.mlr.press/v267/li251.html>.
- [108] Yuchen Li, Luxi Li, Zizhang Wu, Zhenshan Bing, Zhe Xuanyuan, Alois Christian Knoll, and Long Chen. “UnstrPrompt: Large language model prompt for driving in unstructured scenarios”. In: *IEEE Journal of Radio Frequency Identification* (2024).
- [109] Chin-Yew Lin. “ROUGE: A Package for Automatic Evaluation of Summaries”. In: *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*. 2004, pp. 74–81.
- [110] 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*. Springer. 2014, pp. 740–755.

- [111] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. *Flow Matching for Generative Modeling*. 2023. arXiv: [2210.02747](https://arxiv.org/abs/2210.02747) [cs.LG]. URL: <https://arxiv.org/abs/2210.02747>.
- [112] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. *Flow Matching for Generative Modeling*. 2023. arXiv: [2210.02747](https://arxiv.org/abs/2210.02747) [cs.LG]. URL: <https://arxiv.org/abs/2210.02747>.
- [113] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. “Visual instruction tuning”. In: *Advances in neural information processing systems* 36 (2024).
- [114] Xingchao Liu, Chengyue Gong, et al. “Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow”. In: *The Eleventh International Conference on Learning Representations*. 2023.
- [115] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. “Instaflow: One step is enough for high-quality diffusion-based text-to-image generation”. In: *The Twelfth International Conference on Learning Representations*. 2023.
- [116] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and qiang liu. “InstaFlow: One Step is Enough for High-Quality Diffusion-Based Text-to-Image Generation”. In: *The Twelfth International Conference on Learning Representations*. 2024. URL: <https://openreview.net/forum?id=1k4yZbbDqX>.
- [117] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. “Deep learning face attributes in the wild”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 3730–3738.
- [118] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. “Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 5775–5787.
- [119] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. “Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models”. In: *arXiv preprint arXiv:2211.01095* (2022).
- [120] Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. “Diffusion hyperfeatures: Searching through time and space for semantic correspondence”. In: *Advances in Neural Information Processing Systems* 36 (2024).
- [121] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. *Latent Consistency Models: Synthesizing High-Resolution Images with Few-Step Inference*. 2023. arXiv: [2310.04378](https://arxiv.org/abs/2310.04378) [cs.CV].
- [122] Yunsheng Ma et al. “LaMPilot: An Open Benchmark Dataset for Autonomous Driving with Language Model Programs”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2024, pp. 15141–15151.

- [123] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. “Exploring the Limits of Weakly Supervised Pretraining”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Sept. 2018.
- [124] Srikanth Malla, Chiho Choi, Isht Dwivedi, Joon Hee Choi, and Jiachen Li. “DRAMA: Joint Risk Localization and Captioning in Driving”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2023, pp. 1043–1052.
- [125] Jiageng Mao, Yuxi Qian, Junjie Ye, Hang Zhao, and Yue Wang. *GPT-Driver: Learning to Drive with GPT*. 2023. arXiv: [2310.01415](https://arxiv.org/abs/2310.01415) [cs.CV]. URL: <https://arxiv.org/abs/2310.01415>.
- [126] Jiageng Mao, Junjie Ye, Yuxi Qian, Marco Pavone, and Yue Wang. *A Language Agent for Autonomous Driving*. 2023. arXiv: [2311.10813](https://arxiv.org/abs/2311.10813) [cs.CV].
- [127] Ming Nie, Renyuan Peng, Chunwei Wang, Xinyue Cai, Jianhua Han, Hang Xu, and Li Zhang. “Reason2drive: Towards interpretable and chain-based reasoning for autonomous driving”. In: *arXiv preprint arXiv:2312.03661* (2023).
- [128] Matthew Niedoba, Berend Zwartsenberg, Kevin Murphy, and Frank Wood. “Towards a Mechanistic Explanation of Diffusion Model Generalization”. In: *arXiv preprint arXiv:2411.19339* (2024).
- [129] OpenAI. <https://openai.com/sora>. 2024.
- [130] OpenAI et al. *GPT-4 Technical Report*. 2024. arXiv: [2303.08774](https://arxiv.org/abs/2303.08774) [cs.CL].
- [131] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. “BLEU: a Method for Automatic Evaluation of Machine Translation”. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. 2002, pp. 311–318.
- [132] SungYeon Park, MinJae Lee, JiHyuk Kang, Hahyeon Choi, Yoonah Park, Juhwan Cho, Adam Lee, and DongKyu Kim. “Vlaad: Vision and language assistant for autonomous driving”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024, pp. 980–987.
- [133] William Peebles and Saining Xie. “Scalable Diffusion Models with Transformers”. In: *arXiv preprint arXiv:2212.09748* (2022).
- [134] Dean A Pomerleau. “Alvin: An autonomous land vehicle in a neural network”. In: *Advances in neural information processing systems* 1 (1988).
- [135] Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T. Q. Chen. *Multisample Flow Matching: Straightening Flows with Minibatch Couplings*. 2023. arXiv: [2304.14772](https://arxiv.org/abs/2304.14772) [cs.LG]. URL: <https://arxiv.org/abs/2304.14772>.

- [136] Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T. Q. Chen. *Multisample Flow Matching: Straightening Flows with Minibatch Couplings*. 2023. arXiv: [2304.14772](https://arxiv.org/abs/2304.14772) [cs.LG]. URL: <https://arxiv.org/abs/2304.14772>.
- [137] Aaditya Prasad, Kevin Lin, Jimmy Wu, Linqi Zhou, and Jeannette Bohg. *Consistency Policy: Accelerated Visuomotor Policies via Consistency Distillation*. 2024. arXiv: [2405.07503](https://arxiv.org/abs/2405.07503) [cs.R0].
- [138] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. “Pointnet: Deep learning on point sets for 3d classification and segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 652–660.
- [139] Tianwen Qian, Jingjing Chen, Linhai Zhuo, Yang Jiao, and Yu-Gang Jiang. *NuScenes-QA: A Multi-modal Visual Question Answering Benchmark for Autonomous Driving Scenario*. 2024. arXiv: [2305.14836](https://arxiv.org/abs/2305.14836) [cs.CV].
- [140] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. “Improving Language Understanding by Generative Pre-Training”. In: *OpenAI Blog* (2018). URL: <https://openai.com/research/language-unsupervised>.
- [141] Alec Radford et al. “Learning Transferable Visual Models From Natural Language Supervision”. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 8748–8763.
- [142] 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*. 2023. arXiv: [1910.10683](https://arxiv.org/abs/1910.10683) [cs.LG].
- [143] Yuxi Ren, Xin Xia, Yanzuo Lu, Jiacheng Zhang, Jie Wu, Pan Xie, Xing Wang, and Xuefeng Xiao. *Hyper-SD: Trajectory Segmented Consistency Model for Efficient Image Synthesis*. 2024. arXiv: [2404.13686](https://arxiv.org/abs/2404.13686) [cs.CV].
- [144] Rebecca Roelofs, Liting Sun, Ben Caine, Khaled S. Refaat, Ben Sapp, Scott Ettinger, and Wei Chai. *CausalAgents: A Robustness Benchmark for Motion Forecasting using Causal Relationships*. 2022. arXiv: [2207.03586](https://arxiv.org/abs/2207.03586) [cs.LG].
- [145] Junha Roh, Chris Paxton, Andrzej Pronobis, Ali Farhadi, and Dieter Fox. “Conditional driving from natural language instructions”. In: *Conference on Robot Learning*. PMLR. 2020, pp. 540–551.
- [146] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. “High-Resolution Image Synthesis With Latent Diffusion Models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 10684–10695.

- [147] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. “High-Resolution Image Synthesis With Latent Diffusion Models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 10684–10695.
- [148] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2021. arXiv: [2112.10752](https://arxiv.org/abs/2112.10752) [cs.CV].
- [149] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2022. arXiv: [2112.10752](https://arxiv.org/abs/2112.10752) [cs.CV]. URL: <https://arxiv.org/abs/2112.10752>.
- [150] 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*. 2022, pp. 10684–10695.
- [151] Kwonyoung Ryu, Soonmin Hwang, and Jaesik Park. “Instant Domain Augmentation for LiDAR Semantic Segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2023, pp. 9350–9360.
- [152] Enna Sachdeva, Nakul Agarwal, Suhas Chundi, Sean Roelofs, Jiachen Li, Mykel Kochenderfer, Chiho Choi, and Behzad Dariush. “Rank2Tell: A Multimodal Driving Dataset for Joint Importance Ranking and Reasoning”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2024, pp. 7513–7522.
- [153] Enna Sachdeva, Nakul Agarwal, Suhas Chundi, Sean Roelofs, Jiachen Li, Mykel Kochenderfer, Chiho Choi, and Behzad Dariush. “Rank2Tell: A Multimodal Driving Dataset for Joint Importance Ranking and Reasoning”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2024, pp. 7513–7522.
- [154] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. “Adversarial Diffusion Distillation”. In: *arXiv preprint arXiv:2311.17042* (2023).
- [155] Christoph Schuhmann et al. “Laion-5b: An open large-scale dataset for training next generation image-text models”. In: *Advances in neural information processing systems* 35 (2022), pp. 25278–25294.
- [156] Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S. Refaat, Rami Al-Rfou, and Benjamin Sapp. *MotionLM: Multi-Agent Motion Forecasting as Language Modeling*. 2023. arXiv: [2309.16534](https://arxiv.org/abs/2309.16534) [cs.CV]. URL: <https://arxiv.org/abs/2309.16534>.
- [157] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. “On a formal model of safe and scalable self-driving cars”. In: *arXiv preprint arXiv:1708.06374* (2017).

- [158] Hao Shao, Yuxuan Hu, Letian Wang, Steven L. Waslander, Yu Liu, and Hongsheng Li. *LMDrive: Closed-Loop End-to-End Driving with Large Language Models*. 2023. arXiv: [2312.07488](https://arxiv.org/abs/2312.07488) [cs.CV]. URL: <https://arxiv.org/abs/2312.07488>.
- [159] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Ping Luo, Andreas Geiger, and Hongyang Li. *DriveLM: Driving with Graph Visual Question Answering*. 2023. arXiv: [2312.14150](https://arxiv.org/abs/2312.14150) [cs.CV].
- [160] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. *Deep Unsupervised Learning using Nonequilibrium Thermodynamics*. 2015. arXiv: [1503.03585](https://arxiv.org/abs/1503.03585) [cs.LG].
- [161] Jiaming Song, Chenlin Meng, and Stefano Ermon. “Denoising Diffusion Implicit Models”. In: *International Conference on Learning Representations*. 2021. URL: <https://openreview.net/forum?id=StlgiaarCHLP>.
- [162] Jiaming Song, Chenlin Meng, and Stefano Ermon. *Denoising Diffusion Implicit Models*. 2022. arXiv: [2010.02502](https://arxiv.org/abs/2010.02502) [cs.LG]. URL: <https://arxiv.org/abs/2010.02502>.
- [163] Jiaming Song, Chenlin Meng, and Stefano Ermon. “Denoising diffusion implicit models”. In: *arXiv preprint arXiv:2010.02502* (2020).
- [164] Yang Song and Prafulla Dhariwal. “Improved Techniques for Training Consistency Models”. In: *The Twelfth International Conference on Learning Representations*. 2024. URL: <https://openreview.net/forum?id=WNzy9bRDvG>.
- [165] Yang Song and Prafulla Dhariwal. “Improved techniques for training consistency models”. In: *arXiv preprint arXiv:2310.14189* (2023).
- [166] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. “Consistency Models”. In: *International Conference on Machine Learning*. PMLR. 2023, pp. 32211–32252.
- [167] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. *Consistency Models*. 2023. arXiv: [2303.01469](https://arxiv.org/abs/2303.01469) [cs.LG].
- [168] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. “Score-based generative modeling through stochastic differential equations”. In: *arXiv preprint arXiv:2011.13456* (2020).
- [169] NN Sriram, Tirth Maniar, Jayaganesh Kalyanasundaram, Vineet Gandhi, Brojeshwar Bhowmick, and K Madhava Krishna. “Talk to the vehicle: Language conditioned autonomous navigation of self driving cars”. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 5284–5290.
- [170] Łukasz Staniszewski, Łukasz Kuciński, and Kamil Deja. *There and Back Again: On the relation between Noise and Image Inversions in Diffusion Models*. 2025. arXiv: [2410.23530](https://arxiv.org/abs/2410.23530) [cs.CV]. URL: <https://arxiv.org/abs/2410.23530>.
- [171] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. “Emergent correspondence from image diffusion”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 1363–1389.

- [172] Ran Tian, Boyi Li, Xinshuo Weng, Yuxiao Chen, Edward Schmerling, Yue Wang, Boris Ivanovic, and Marco Pavone. “Tokenize the World into Object-level Knowledge to Address Long-tail Events in Autonomous Driving”. In: *arXiv preprint arXiv:2407.00959* (2024).
- [173] Xiaoyu Tian et al. “Drivevlm: The convergence of autonomous driving and large vision-language models”. In: *arXiv preprint arXiv:2402.12289* (2024).
- [174] Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrod Rector-Brooks, Guy Wolf, and Yoshua Bengio. *Improving and generalizing flow-based generative models with minibatch optimal transport*. 2024. arXiv: [2302.00482 \[cs.LG\]](#). URL: <https://arxiv.org/abs/2302.00482>.
- [175] Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrod Rector-Brooks, Guy Wolf, and Yoshua Bengio. *Improving and generalizing flow-based generative models with minibatch optimal transport*. 2024. arXiv: [2302.00482 \[cs.LG\]](#). URL: <https://arxiv.org/abs/2302.00482>.
- [176] S. M Towhidul Islam Tonmoy, S M Mehedi Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. *A Comprehensive Survey of Hallucination Mitigation Techniques in Large Language Models*. 2024. arXiv: [2401.01313 \[cs.CL\]](#). URL: <https://arxiv.org/abs/2401.01313>.
- [177] Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: [2302.13971 \[cs.CL\]](#).
- [178] Chris Urmson et al. “Autonomous driving in urban environments: Boss and the urban challenge”. In: *Journal of field Robotics* 25.8 (2008), pp. 425–466.
- [179] Balakrishnan Varadarajan et al. *MultiPath++: Efficient Information Fusion and Trajectory Aggregation for Behavior Prediction*. 2021. arXiv: [2111.14973 \[cs.CV\]](#).
- [180] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. “CIDEr: Consensus-based Image Description Evaluation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 4566–4575.
- [181] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.” In: *Journal of machine learning research* 11.12 (2010).
- [182] Pauli Virtanen et al. “SciPy 1.0: fundamental algorithms for scientific computing in Python”. In: *Nature Methods* 17.3 (Feb. 2020), pp. 261–272. ISSN: 1548-7105. DOI: [10.1038/s41592-019-0686-2](#). URL: <http://dx.doi.org/10.1038/s41592-019-0686-2>.
- [183] Jason Wang, Luis Perez, et al. “The effectiveness of data augmentation in image classification using deep learning”. In: *Convolutional Neural Networks Vis. Recognit* 11.2017 (2017), pp. 1–8.

- [184] Kai Wang, Mingjia Shi, Yukun Zhou, Zekai Li, Zhihang Yuan, Yuzhang Shang, Xiaojiang Peng, Hanwang Zhang, and Yang You. “A closer look at time steps is worthy of triple speed-up for diffusion model training”. In: *arXiv preprint arXiv:2405.17403* (2024).
- [185] Kai Wang, Mingjia Shi, Yukun Zhou, Zekai Li, Zhihang Yuan, Yuzhang Shang, Xiaojiang Peng, Hanwang Zhang, and Yang You. “A closer look at time steps is worthy of triple speed-up for diffusion model training”. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 12934–12944.
- [186] Shihao Wang, Zhiding Yu, Xiaohui Jiang, Shiyi Lan, Min Shi, Nadine Chang, Jan Kautz, Ying Li, and Jose M Alvarez. “Omnidrive: A holistic llm-agent framework for autonomous driving with 3d perception, reasoning and planning”. In: *arXiv preprint arXiv:2405.01533* (2024).
- [187] Tsun-Hsuan Wang, Alaa Maalouf, Wei Xiao, Yutong Ban, Alexander Amini, Guy Rosman, Sertac Karaman, and Daniela Rus. “Drive Anywhere: Generalizable End-to-end Autonomous Driving with Multi-modal Foundation Models”. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. 2024, pp. 6687–6694. DOI: [10.1109/ICRA57147.2024.10611590](https://doi.org/10.1109/ICRA57147.2024.10611590).
- [188] Wenshuo Wang, Letian Wang, Chengyuan Zhang, Changliu Liu, Lijun Sun, et al. “Social interactions for autonomous driving: A review and perspectives”. In: *Foundations and Trends® in Robotics* 10.3-4 (2022), pp. 198–376.
- [189] Zhendong Wang, Yifan Jiang, Huangjie Zheng, Peihao Wang, Pengcheng He, Zhangyang Wang, Weizhu Chen, and Mingyuan Zhou. “Patch Diffusion: Faster and More Data-Efficient Training of Diffusion Models”. In: *arXiv preprint arXiv:2304.12526* (2023).
- [190] Zhendong Wang, Yifan Jiang, Huangjie Zheng, Peihao Wang, Pengcheng He, Zhangyang Wang, Weizhu Chen, Mingyuan Zhou, et al. “Patch diffusion: Faster and more data-efficient training of diffusion models”. In: *Advances in neural information processing systems* 36 (2023), pp. 72137–72154.
- [191] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. “Chain-of-thought prompting elicits reasoning in large language models”. In: *Advances in neural information processing systems* 35 (2022), pp. 24824–24837.
- [192] Ying Wei, Yu Zhang, Junzhou Huang, and Qiang Yang. “Transfer Learning via Learning to Transfer”. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, July 2018, pp. 5085–5094.

- [193] Benjamin Wilson et al. “Argoverse 2: Next Generation Datasets for Self-Driving Perception and Forecasting”. In: *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*. Ed. by J. Vanschoren and S. Yeung. Vol. 1. 2021. URL: https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/4734ba6f3de83d861c3176a6273cac6d-Paper-round2.pdf.
- [194] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. “PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes”. In: *Robotics: Science and Systems (RSS)*. 2018.
- [195] Siyu Xing, Jie Cao, Huaibo Huang, Xiao-Yu Zhang, and Ran He. *Exploring Straighter Trajectories of Flow Matching with Diffusion Guidance*. 2023. arXiv: [2311.16507](https://arxiv.org/abs/2311.16507) [cs.CV]. URL: <https://arxiv.org/abs/2311.16507>.
- [196] Chenfeng Xu, Tian Li, Chen Tang, Lingfeng Sun, Kurt Keutzer, Masayoshi Tomizuka, Alireza Fathi, and Wei Zhan. “PreTraM: Self-supervised Pre-training via Connecting Trajectory and Map”. In: *Computer Vision – ECCV 2022*. Ed. by Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner. Cham: Springer Nature Switzerland, 2022, pp. 34–50. ISBN: 978-3-031-19842-7.
- [197] Chenfeng Xu, Huan Ling, Sanja Fidler, and Or Litany. *3DiffTecton: 3D Object Detection with Geometry-Aware Diffusion Features*. 2023. arXiv: [2311.04391](https://arxiv.org/abs/2311.04391) [cs.CV].
- [198] Huazhe Xu, Yang Gao, Fisher Yu, and Trevor Darrell. *End-to-end Learning of Driving Models from Large-scale Video Datasets*. 2017. arXiv: [1612.01079](https://arxiv.org/abs/1612.01079) [cs.CV].
- [199] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee. K. Wong, Zhenguo Li, and Hengshuang Zhao. *DriveGPT4: Interpretable End-to-end Autonomous Driving via Large Language Model*. 2024. arXiv: [2310.01412](https://arxiv.org/abs/2310.01412) [cs.CV].
- [200] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee. K. Wong, Zhenguo Li, and Hengshuang Zhao. *DriveGPT4: Interpretable End-to-end Autonomous Driving via Large Language Model*. 2024. arXiv: [2310.01412](https://arxiv.org/abs/2310.01412) [cs.CV]. URL: <https://arxiv.org/abs/2310.01412>.
- [201] Senqiao Yang et al. “Lidar-llm: Exploring the potential of large language models for 3d lidar understanding”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 39. 9. 2025, pp. 9247–9255.
- [202] Zhenjie Yang, Xiaosong Jia, Hongyang Li, and Junchi Yan. “LLM4Drive: A Survey of Large Language Models for Autonomous Driving”. In: *arXiv e-prints* (2023), arXiv–2311.
- [203] Yuan Yao, Qianyu Chen, Ao Zhang, Wei Ji, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. “PEVL: Position-enhanced Pre-training and Prompt Tuning for Vision-language Models”. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 11104–11117. DOI: [10.18653/v1/2022.emnlp-main.763](https://doi.org/10.18653/v1/2022.emnlp-main.763).

- [204] Zhao-Heng Yin, Chenran Li, Liting Sun, Masayoshi Tomizuka, and Wei Zhan. “Iterative Imitation Policy Improvement for Interactive Autonomous Driving”. In: *ArXiv abs/2109.01288* (2021).
- [205] He Yisheng, Wang Yao, Fan Haoqiang, Chen Qifeng, and Sun Jian. “FS6D: Few-Shot 6D Pose Estimation of Novel Objects”. In: *CVPR* (2022).
- [206] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. “Bdd100k: A diverse driving dataset for heterogeneous multitask learning”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 2636–2645.
- [207] Jianhao Yuan, Shuyang Sun, Daniel Omeiza, Bo Zhao, Paul Newman, Lars Kunze, and Matthew Gadd. “Rag-driver: Generalisable driving explanations with retrieval-augmented in-context learning in multi-modal large language model”. In: *arXiv preprint arXiv:2402.10828* (2024).
- [208] Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. *Investigating the Catastrophic Forgetting in Multimodal Large Language Models*. 2023. arXiv: [2309.10313 \[cs.CL\]](https://arxiv.org/abs/2309.10313). URL: <https://arxiv.org/abs/2309.10313>.
- [209] Huijie Zhang, Jinfan Zhou, Yifu Lu, Minzhe Guo, Peng Wang, Liyue Shen, and Qing Qu. “The Emergence of Reproducibility and Consistency in Diffusion Models”. In: *Proceedings of the 41st International Conference on Machine Learning*. Ed. by Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp. Vol. 235. Proceedings of Machine Learning Research. PMLR, July 2024, pp. 60558–60590. URL: <https://proceedings.mlr.press/v235/zhang24cn.html>.
- [210] Pengze Zhang, Hubery Yin, Chen Li, and Xiaohua Xie. “Formulating Discrete Probability Flow Through Optimal Transport”. In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023. URL: <https://openreview.net/forum?id=I9GNrInbdf>.
- [211] Songyan Zhang, Wenhui Huang, Zihui Gao, Hao Chen, and Chen Lv. “WiseAD: Knowledge Augmented End-to-End Autonomous Driving with Vision-Language Model”. In: *arXiv preprint arXiv:2412.09951* (2024).
- [212] Yikang Zhang, Shuo Zhang, Zhichao Liang, Hanran Luna Li, Haiyan Wu, and Quanying Liu. “Dynamical Driving Interactions between Human and Mentalizing-designed Autonomous Vehicle”. In: *2022 IEEE International Conference on Development and Learning (ICDL)*. IEEE. 2022, pp. 178–183.
- [213] Yiyuan Zhang, Kaixiong Gong, Kaipeng Zhang, Hongsheng Li, Yu Qiao, Wanli Ouyang, and Xiangyu Yue. “Meta-Transformer: A Unified Framework for Multimodal Learning”. In: *arXiv preprint arXiv:2307.10802* (2023).
- [214] Xiaocong Zhao, Jian Sun, and Meng Wang. “Measuring sociality in driving interaction”. In: *IEEE Transactions on Intelligent Transportation Systems* (2024).

- [215] Tianyi Zheng, Cong Geng, Peng-Tao Jiang, Ben Wan, Hao Zhang, Jinwei Chen, Jia Wang, and Bo Li. “Non-uniform Timestep Sampling: Towards Faster Diffusion Model Training”. In: *Proceedings of the 32nd ACM International Conference on Multimedia*. MM '24. Melbourne VIC, Australia: Association for Computing Machinery, 2024, pp. 7036–7045. ISBN: 9798400706868. DOI: [10.1145/3664647.3680912](https://doi.org/10.1145/3664647.3680912). URL: <https://doi.org/10.1145/3664647.3680912>.
- [216] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. “Random erasing data augmentation”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 07. 2020, pp. 13001–13008.
- [217] Ziyuan Zhong, Davis Rempe, Yuxiao Chen, Boris Ivanovic, Yulong Cao, Danfei Xu, Marco Pavone, and Baishakhi Ray. “Language-Guided Traffic Simulation via Scene-Level Diffusion”. In: *Proceedings of The 7th Conference on Robot Learning*. Ed. by Jie Tan, Marc Toussaint, and Kourosh Darvish. Vol. 229. Proceedings of Machine Learning Research. PMLR, Nov. 2023, pp. 144–177. URL: <https://proceedings.mlr.press/v229/zhong23a.html>.
- [218] Yin Zhou and Oncel Tuzel. “Voxelnet: End-to-end learning for point cloud based 3d object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4490–4499.
- [219] Yunsong Zhou et al. “Embodied understanding of driving scenarios”. In: *arXiv preprint arXiv:2403.04593* (2024).
- [220] Julius Ziegler et al. “Making bertha drive—an autonomous journey on a historic route”. In: *IEEE Intelligent transportation systems magazine* 6.2 (2014), pp. 8–20.

Appendix A

Appendix for WOMD-Reasoning: A Large-Scale Dataset for Interaction Reasoning in Driving

In this appendix, we first show the building process of WOMD-Reasoning in detail in A.1. Specifically, we show the construction of the interpreted language of the motion data with an example (A.1.1), and we revisit the detailed methods utilizing GPT to build the language dataset as well as providing the full set of prompts (A.1.2). Some examples of the vision modal of WOMD-Reasoning provided by simulations are presented in A.2.

Then, for Motion-LLaVA, whose language output enhances trajectory prediction as shown in Section 3.6, we provide further implementation details. Specifically, in A.3, we describe how agent information from driving scenarios is incorporated into the input of Motion-LLaVA. The training hyperparameters used for the downstream trajectory prediction module are presented in A.4. The prompt used to obtain the GPT Score for evaluation is provided in A.5. We further provide a qualitative comparison in A.6 of the benchmarked multi-modal models listed in Table 3.4, and introduce in A.7 additional multi-modal models not included in the table. Examples of aggregated context (A.8) and factual description context (A.9) used in Chain-of-Thought interaction reasoning are also provided for reference. An ablation study on the fine-tuning strategy used in Motion-LLaVA is included in A.10, along with an analysis of the data efficiency of WOMD-Reasoning in helping Motion-LLaVA learn driving-related reasoning in A.11.

A.1 Details in Building WOMD-Reasoning

In this appendix, we show the details of how we build WOMD-Reasoning dataset, including detailed explanations on each step we take, the example of the translated motion scenarios, and the GPT prompts. The full version of codes and prompts used herein can be found in the supplemental materials.

A.1.1 Automatic Translation of Driving Scenarios into Language Descriptions

ChatGPT cannot effectively process raw data from motion datasets, as it is mainly trained on textual information rather than motion data. Therefore, we first translate the motion information into textual descriptions by a rule-based program. Accurate motion translation is critical to the quality of the dataset, as it serves as the only input observable to GPT. Ensuring thorough and well-structured translations is essential for GPT to perform accurate interaction analysis and reasoning.

To ensure enough information for analyzing interactions involving the ego agent, we translate the motion data into an ego-centric description of the traffic scenario. Specifically, before describing the ego agent status, we first narrate the map environment around the ego agent, including information about related intersections, lanes, stop signs, crosswalks, speed bumps, etc. The above information provides a comprehensive spatial guidance that facilitates describing the motion of either ego or other agents accurately. Then, we describe the status of the ego agent in the scenario, including its motion status (e.g., the velocity, acceleration, etc.) and the positional status (e.g., the lane it occupies and its position related to the intersection center). Furthermore, the traffic light and signal information related to it are summarized in the description. Besides the ego agent, the depiction of other agents, including other vehicles, bicycles, and pedestrians, is also crucial for GPT to analyze the interactions. Therefore, we then describe every other agent in the scenario in the same way as we describe the ego agent. We additionally include the positional relation between the other agent and the ego one to support the identification of every agent’s position.

Utilizing the methods above, we can translate motion datasets into languages. Due to the limitations of computational resources of GPT, we only choose to translate those scenarios that have `objects_of_interest` labels in WOMD, which indicates confirmed significant interactions happening in the scenario. We build two subsets of WOMD-Reasoning with the same setting: the training set is built on the training set of WOMD, while the validation set is built on the interactive validation set of WOMD. In total, we translate 63k scenarios into language. An example of the translation can be found in Table A.1.

A.1.2 GPT-based Interaction Analysis

With the language translation of WOMD as input, we further design a set of prompts to utilize ChatGPT-4 [130] to build the Q&A dataset, taking advantage of its abilities in analyzing interactions. The prompts come with 4 main parts. **(1) System Prompt**, which lets the GPT know its role and the format of its input, i.e. the language translation. **(2) Responsibility Prompt**, which describes the responsibilities of the GPT, including the questions it should ask and answer, as well as the output format it should use. This will be talked about in detail in the next paragraph. **(3) Rules Prompt**, which contains global rules to guide the output and the analysis. **(4) In-context Prompt**, which provides a few

Descriptions on map environment:

The ego agent is heading towards intersection. The intersection center is 18.0 meters in front of the ego agent, and is 17.0 meters on the left of the ego agent. The intersection is a 4 way intersection.

Descriptions on ego agent:

The ego agent is on the 1 lane from the right, out of 3 lanes. Its current speed is 8 m/s. It is accelerating. Traffic Light for the ego agent is red. The ego agent is approaching a crosswalk 3 meters ahead.

Descriptions on surrounding (other) agents:

Surrounding agent # 0 is a vehicle. It is 23 meters in front of the ego agent, and is 17 meters on the left of the ego agent. It is heading right of the ego agent. Its current speed is 6 m/s. It is accelerating. It is in the intersection. It is 29 meters away from the ego agent. It is approaching a crosswalk 3 meters ahead.

Surrounding agent # 2 is a vehicle. It is 11 meters on the right of the ego agent, and is 0 meters behind the ego agent. It is heading left of the ego agent. It is not moving. It is on the same side of the intersection as the ego agent. It is 34 meters away from the intersection center.

...

Descriptions on agents after 3 seconds:

The following is the description of the ego and surrounding agents after 3.0 seconds:

Surrounding agent # 0 will be 8 meters in front of the ego agent, and will be 5 meters on the left of the ego agent. It will be heading right of the ego agent. Its speed will be 7 m/s. It will be departing from the intersection. Looking from the agent's current angle, it will be on the same side of the intersection. It will be 18 meters away from the intersection center.

The ego agent will be 13 meters in front of the current place, and will be 2 meters on the right of the current place. It will be heading in the same direction as the current moment. Its speed will be 3 m/s. It will be departing from the intersection. Looking from the agent's current angle, it will be on the same side of the intersection. It will be 20 meters away from the intersection center.

Surrounding agent # 2 will be 13 meters on the right of the ego agent, and will be 9 meters behind the ego agent. It will be heading left of the ego agent. It will not be moving. Looking from the agent's current angle, it will be on the same side of the intersection. It will be 34 meters away from the intersection center.

...

Table A.1: An example of the language translation of motion data.

System prompt for GPT:

You are an AI assistant to analyze real-world driving scenes.

You are provided with the detailed information (formatted as: [start of the input] the detailed information [end of the input]) of the real-world driving scenario, which primarily includes information about the ego vehicle (denoted as 'Ego Agent'), other vehicles, cyclists or pedestrians (denoted as 'Surrounding agent #n'), intersections and road lanes. Note that the input includes the current situation as well as the situation in a future moment for your reference.

Unless you are told to guess, only provide information you are certain according to the input. The input is complete. Any agents or other things not mentioned in the input does NOT exist, and you should NOT guess what would happen if they exist.

Table A.2: System prompt for GPT.

human-written examples of input-output pairs to perform in-context learning. An example of each part of the prompts can be found in Table A.2, A.3, A.4, and A.5 respectively.

The questions in the responsibility part of the prompt decide what would be present in the generated language Q&A dataset. To provide a comprehensive language dataset, and to present interaction analysis as accurately as possible, we organize the questions into a chain to enable a well-designed Chain-of-Thought [191, 26]: First, we ask questions about the map environment, the ego and other agents' motion. These questions would let the GPT provide Q&As useful for fine-tuning scene description agents while enhancing the GPT's understanding of the input information at the same time. Then, we ask questions about the interactions that occur in the observable future period. To include traffic rule-induced and intention-induced interactions, we ask the GPT to think about the interaction between each pair of agents by answering a sequence of questions: **Q1:** Are the two agents vehicles, and are they close to each other with traffic rules or signals governing their movements? If the answer is yes, we request that GPT analyze their yielding relations according to the traffic rules. Otherwise, we ask **Q2:** Are the two agents vehicle and pedestrian respectively with intention conflicts? If so, the vehicle generally should yield to the pedestrian for safety. If both questions do not help find the interaction relations, we come to **Q3:** Does the scene show patterns of certain intention-induced interaction? We provide descriptions of a few common interactions like overtaking, following, etc. GPT can choose the best fit from these patterns to try to cover more interaction types. Based on this pipeline, we can provide interaction analysis with rich information on traffic rule-induced and human intention-induced interactions. After that, we can finally wrap up the interactions between the ego and all other agents to provide the intention of the ego agent.

Responsibility Prompt for GPT:

You are responsible for the following responsibilities:

First, print "[Env QA]". Generate Q&A on all the scenario description inputs about the scenario environment. The Q&A must be in the following format: "[Q][Question content] [A][Answer content]" The Q&A should cover ALL information about the CURRENT moment in the input description. No information in the provided future moment should be used. Your questions should include all the questions in the following categories except the answer is not mentioned in the input: (1) About the intersection: its existence? Its type? (2) About the lanes: How many lanes on ego's side? (3) About stop signs: its existence? where is it? which direction is it for?
...

Second, print "[Ego QA]". Generate Q&A on all the scenario description inputs about the ego agent. The Q&A must be in the following format: "[Q][Question content] [A][Answer content]" The Q&A should cover ALL information about the CURRENT moment in the input description. No information in the provided future moment should be used. Your questions should include all the questions in the following categories: (1) About its motion: its speed? its motion status? (2) About its position: its position? its lane? (3) About its direction: its facing direction? its turning direction?
...

Third, print "[Sur QA]" and generate Q&A on all the scenario description inputs about surrounding agents. The Q&A must be in the following format: "[Q][Question content] [A][Answer content]" The Q&A should cover ALL information about the CURRENT moment in the input description. No information in the provided future moment should be used. For each surrounding agent, your questions should include all the questions in the following categories except the answer is not mentioned in the input: (1) About itself: its type? (2) About its motion: its speed? its motion status? (3) About its position: its position to the ego agent? its position to the intersection? its lane?
...

Fourth, start a new paragraph and print "[Int QA]". For each surrounding agent, start a new paragraph and provide one Q&A for the interactions between it and the ego agent. The Q&A must be in the following format: "[Q][Question content] [A][Answer content]" Here [Question content] should have the meaning "What interactions will happen between surrounding agent #n and the ego agent?" without any additional information like "considering what" or "given what", but you must use diverse ways to put it. (i.e. ask in different ways for different surrounding agents) When providing [Answer content], you should use the descriptions for both the current moment and the future moment, and analyze based on them. Pay attention to the special situations where you should consider that there is an interaction:
...

Fifth, for the ego agent, start a new paragraph, print "[Intention]", and provide one Q&A for the intention of the ego agent. The Q&A must be in the following format: "[Q][Question content] [A][Answer content]" Here [Question content] should have the meaning "What will be the intention of the ego agent?", but you must use diverse ways to put it. When providing [Answer content], and think about following questions in the time period from now to the provided future moment based on given description: (1) What is the ego agent's intention if no other agents exist? (2) If applicable, what actions does the ego agent take to respond to traffic lights, stop signs, crosswalks, speed bumps and traffic rules? If some traffic controls are not applicable, do not output on that. (3) In each interaction involving the ego agent, what actions do the ego and surrounding agent take to respond to each other? If one agent does not have interaction with the ego agent, exclude it in your answer.
...

Remember to use diverse ways to put the [Answer content]. Do not include any numbers regarding speeds or positions in your [Answer content].

Table A.3: Responsibility prompt for GPT.

Global rules prompt for GPT:

Here are some global rules that your need to follow:

1. Please provide the summary and analysis of the driving scenario according to your responsibilities.
2. Never use bullet points. Be concise and compact.
3. Give natural language, no parenthesis.
4. Give the results in the order of the responsibilities. Always start a new paragraph for each responsibility, but one responsibility can take several paragraphs. Do not combine multiple responsibilities together. Finish one responsibility before starting another.
5. Do not show the intersection number, lane number or the ego agent's number. The surrounding agents' numbers need to be shown.
6. Output in full. Never omit anything regardless of any reason. You are allowed to give very long outputs.
7. Remember that all the agents and traffic controls in the scene have been provided, so if any agents or traffic controls are not mentioned in the scene, they do not exist. You don't need to consider them or give any output about them. Therefore, do not output words like "not mentioned" because this is inaccurate.
8. The traffic light info is for the current moment only. The future traffic light info is unknown.
9. Any action caused by traffic controls like lights, stop signs or crosswalks may also constitute yielding interactions. Traffic controls are only an indication of right-of-way, but any actions that are caused by the traffic controls are still considered interactions. For example, if a car is waiting for the red light or a stop sign while someone other cross the intersection with green light, the car is still considered yielding to the other car.
10. When analyzing the future, talk about the period between the current and future moment provided, not the two moments.

Table A.4: Rule prompt for GPT.

Context prompt for GPT:

Here is one given example that you can refer to to help you better understand your responsibility Fourth and Fifth: For the input case of:

[start of input]

The ego agent is in intersection. The intersection center is 10.0 meters in front of the ego agent, and is 1.0 meters on the left of the ego agent. The intersection is a 4-way intersection.

It is going straight. It is exiting the intersection. Its current speed is 6 m/s. It is accelerating. The ego agent is approaching a stop sign 1 meters ahead. There are 4 stop signs in the intersection. The ego agent is approaching a crosswalk 1 meters ahead.

Surrounding agent # 0 is a vehicle. It is 4 meters on the left of the ego agent, and is 2 meters behind the ego agent. It is heading the opposite direction as the ego agent. Its current speed is 9 m/s. It is accelerating. It is departing from the intersection. It is on the same side of the intersection as the ego agent. It is 12 meters away from the intersection center.

...

[end of input]

you may give the following analysis:

[Int QA]

[Q] What kind of interaction can be expected between the ego agent and surrounding agent #0 given their positions and directions?

[A] Surrounding agent #0 will have no interaction with the ego agent as their intended paths have no conflicts.

...

[End Int QA]

[Intention]

[Q] What will the ego agent aim to do in the upcoming moments?

[A] The ego agent intends to continue exiting the intersection. There is a stop sign on its side but it has already stopped and started afterwards. Surrounding agent #1 will yield to it so the ego agent does not need to respond. Surrounding agent #3 will follow it but the ego agent does not need to respond as well. Therefore, the ego agent will continue in the intersection and finish its left turn.

[End Intention]

Here is another given example that you can refer to to help you better understand your responsibility Fourth and Fifth:

...

[end of input]

you may give the following analysis:

[Int QA]

[Q] What kind of interaction can be expected between the ego agent and surrounding agent #0?

[A] The ego agent will yield to surrounding agent #0 because they are both approaching the intersection and the traffic light is red for the ego agent, indicating that the ego agent must stop.

[End Int QA]

[Intention]

[Q] What will the ego agent aim to do in the upcoming moments?

[A] The ego agent intends to enter the intersection, but it has to stop at the red traffic light and wait for it to change to green before proceeding through the intersection. It will also yield to surrounding agents #0 who is already in the intersection. Therefore, the ego agent will wait until the lights turn green and the surrounding agent #0 passes before entering the intersection. [End Intention]

...

Table A.5: In-context prompt for GPT.

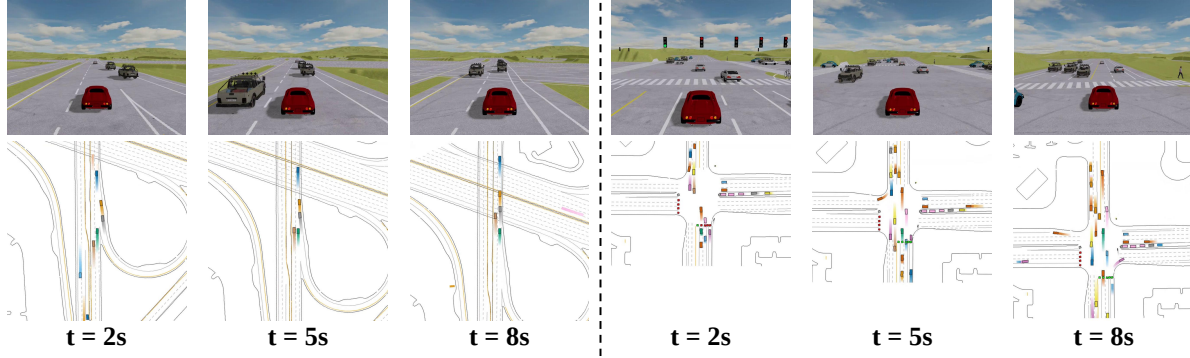


Figure A.1: More visual examples of WOMD-Reasoning through simulations.

A.2 Examples of Simulated Vision Modality

Figure A.1 provides some whole examples of the simulated scenarios generated by MetaDrive [102] simulator using ScenarioNet [101] traffic scenario simulation platform.

A.3 Processing Agent IDs for Motion-LLaVA

Careful processing of agent IDs is essential for enabling Motion-LLaVA to accurately distinguish between different agents. In our approach, these IDs are used purely as textual identifiers within the prompts, allowing the LLM to differentiate agents through natural language understanding. The prompt format is:

Ego agent: <motion>\n Agent #0: <motion>\n...\n Now, please answer: {Question}

, where <motion> denotes the encoded motion data from each agent’s perspective. The IDs are assigned by range: #0–#99 for vehicles, #100–#199 for bicycles, and #200–#299 for pedestrians. We remap WOMD agent IDs to these local ranges to prevent downstream models from overfitting to specific agents’ behaviors. Importantly, these agent IDs do not encode any additional information beyond their role as identifiers.

A.4 Hyperparameters for Multipath++

In the experiment using languages from Motion-LLaVA outputs to assist vehicle trajectory predictions, we use a batch size of 128 and a learning rate of $1e - 4$ for Multipath++. As language is not available for every agent, we use "No captions available for this agent." for those agents without language inputs. The $t5 - v1_1 - xxl$ model is used as the language encoder.

We would like to request your feedback on the performance of an AI assistant in response to the user question displayed above. The user asks the question on observing a driving scenario. The correct answer is included for your reference.

Scoring rules:

- Your scores for each answer should be between 0 (worst) and 10 (best).
- If the answer is mostly incorrect but includes important relevant information, such as identifying correct agents or actions, give a score of less than 3.
- Give intermediate scores for partially correct answers or when the numbers are close. Only give a score of 10 for flawless answers.
- Think critically and carefully. Most importantly, check the answer for factual correctness, but reward partial credit where applicable.

Grading process For each of the xxx questions, provide a one line assessment of the student’s answer in the format:

““ n. <short one sentence assessment>. Score: <score 0-10>. ““

A few examples of what a good assessment might look like:

1. Acknowledged question unrelated to driving and attempted to answer. Score: 10.
2. Doesn’t answer the question but has all information necessary. Score: 3.
3. Incorrectly stated there are no vehicles around even though there is one. Score: 0.
4. Unable to answer question given information available. Score: 10.
5. Give 12 m/s or m or percentage for the correct answer of 13 (within 10% deviation). Score: 9.
6. Give 15 m/s or m or percentage for the correct answer of 13 (within 20% deviation). Score: 7.
7. Give 18 m/s or m or percentage for the correct answer of 13 (within 40% deviation). Score: 5.
8. Give 2 m/s or m or percentage for the correct answer of 13. Score: 0.

Table A.6: Prompt for GPT Score.

A.5 Prompts for evaluating GPT Score

We employ the OpenAI GPT-4-turbo model [130] to evaluate the performance of all models benchmarked on the WOMD-Reasoning. The GPT Score prompt is presented in Table A.6. Notably, we design our GPT Score prompt using a similar template to the one from [20].

A.6 Qualitative Comparison of Benchmarked Multi-modal Models on Answering Questions in WOMD-Reasoning

In this appendix, we present examples of answers to questions in WOMD-Reasoning generated by the benchmarked multi-modal models listed in Table 3.4. The selected scenario corresponds to Figure 3.7, which already showcases answers from Motion-LLaVA. Additionally, we provide results for both the Non-Fine-tuned (i.e., off-the-shelf) and Fine-tuned LLaVA models, both using BEV plots as vision input.

For the Non-Fine-tuned LLaVA, we perform prompt engineering to make the answers coherent. However, the low GPT scores indicate that it remains challenging for LLaVA to directly comprehend the scenario:

- **Q:** Can you describe the type of intersection present in the current driving scenario?
A: The type of intersection present in the current driving scenario is a four-way intersection.
GPT Score: 0. Incorrectly identifies the type of intersection.
- **Q:** What is the ego agent’s current motion status?
A: The ego agent is currently stopped at the intersection.
GPT Score: 0. Incorrectly describes the motion status of the ego agent.
- **Q:** Where is surrounding agent #5 in relation to the ego agent?
A: Surrounding agent #5 is located to the right of the ego agent.
GPT Score: 0. Incorrect location of surrounding agent #5 in relation to the ego agent.
- **Q:** What kind of interaction will occur between the ego agent and surrounding agent #5?
A: The ego agent will pass by the surrounding agent #5 at the intersection.
GPT Score: 0. Incorrect prediction of the interaction between the ego agent and surrounding agent #5.

Quantitatively, the Non-Fine-tuned LLaVA achieves a remarkably low GPT Score of 2.31 across all questions in the validation set, demonstrating difficulty in understanding the driving scenarios based solely on the prior knowledge of the Vision-Language Model (VLM). For comparison, we evaluate the performance of the Fine-tuned LLaVA with vision input under the same scenario. The fine-tuned model achieves much more reasonable results, as illustrated below:

- **Q:** Can you describe the type of intersection present in the current driving scenario?
A: The intersection is a 3 way intersection.
GPT Score: 10. The student’s answer matches the correct answer exactly.

- **Q:** What is the ego agent’s current motion status?
A: The ego agent is accelerating.
GPT Score: 10. The student’s answer matches the correct answer exactly.
- **Q:** Where is surrounding agent #5 in relation to the ego agent?
A: It is 14 meters on the right and 10 meters in front of the ego agent.
GPT Score: 0. The student’s answer incorrectly describes the position of surrounding agent #5.
- **Q:** What kind of interaction will occur between the ego agent and surrounding agent #5?
A: Surrounding agent #5 will follow the ego agent as they are both heading towards the intersection and are on the same side.
GPT Score: 5. The student’s answer is partially correct but does not specify that agent #5 is behind the ego agent.

The results indicate that fine-tuning significantly improves LLaVA’s ability to interpret driving scenarios. This improvement can be attributed to the inclusion of our WOMD-Reasoning dataset. However, despite these enhancements, notable under-performance persists in questions related to **Other Agent Status** and **Interaction & Intention**, especially when compared to the results in Figure 3.7, which are generated using Motion-LLaVA. We attribute this limitation to potential information loss during the conversion of vectorized motion data into a BEV plot with a scalar bar.

A.7 Additional Baseline Language Models on WOMD-Reasoning

Our main results in Table 3.4 primarily evaluate different variants of LLaVA [113] to demonstrate the effectiveness of WOMD-Reasoning. For clarity, we explain the categories of questions assessed. The questions are grouped into two main types: factual questions and interaction reasoning questions.

- **Factual questions:**
 - **Environment-related:** Questions about the scenario environment, such as the existence and type of intersections, the number of lanes on the ego vehicle’s side, and the presence or location of crosswalks and stop signs.
 - **Ego agent-related:** Questions regarding the ego vehicle’s properties, including speed, motion status, lane position, and direction.
 - **Surrounding agents-related:** Questions about other agents in the scenario, covering their type, speed, motion status, and position relative to the ego vehicle and the intersection.

• **Interaction reasoning questions:**

- Questions about the interactions that will occur between each surrounding agent and the ego agent.
- Questions about the overall intention of the ego agent in the given scenario.

We further evaluate additional language models (LMs) to confirm the effectiveness of WOMD-Reasoning. Specifically, we assess the answer quality of several baseline models, including LLaMA-Adapter v2.1 [49], ViTA-1.5 [44], and Qwen2.5-VL [7] (all 7B versions), both before and after fine-tuning on WOMD-Reasoning. The results are summarized in Table A.7.

We observe that without fine-tuning, all models can hardly answer driving-related questions, supporting the motivation of building WOMD-Reasoning. We then fine-tune LLaMA-Adapter on WOMD-Reasoning, which also significantly benefit from the fine-tuning. Besides, we find that fine-tuned our Motion-LLaVA works better than fine-tuned LLaVA or LLaMA-adapter, proving its well-designed structure in utilizing WOMD-Reasoning information.

Table A.7: Additional language model baselines evaluated on WOMD-Reasoning. We report the answer quality of several vanilla and WOMD-Reasoning-fine-tuned models, including LLaMA-Adapter, ViTA, and Qwen2.5-VL (all 7B versions), to further validate the effectiveness of WOMD-Reasoning.

Model	Fine-tuned on WOMD-Reasoning	ROUGE (↑)	BLEU (↑)	METEOR (↑)	CIDEr (↑)	SPICE (↑)	GPT Score (↑)
LLaVA	✗	0.512	0.211	0.275	1.36	0.455	2.31
LLaMA-Adapter v2.1	✗	0.413	0.174	0.235	0.91	0.372	1.62
ViTA-1.5	✗	0.278	0.121	0.190	0.40	0.227	1.77
Qwen2.5-VL	✗	0.384	0.156	0.247	0.62	0.379	2.36
LLaVA	✓	0.779	0.581	0.439	5.51	0.735	6.88
LLaMA-Adapter v2.1	✓	0.722	0.470	0.375	4.72	0.691	5.14
Motion-LLaVA (Ours)	✓	0.792	0.616	0.449	5.69	0.744	7.02

A.8 Example Aggregated Context for Chain-of-Thought Reasoning

We present an example of an **aggregated** context, derived from **the answers generated by Motion-LLaVA** for factual questions, which is used by Motion-LLaVA to infer answers to reasoning-based questions. This context corresponds to the **CoT** configuration in Table A.8:

The ego agent is turning left. Its current speed is 5 m/s. It is accelerating.

Surrounding agent #7 is a vehicle. It is 26 meters on the left of the ego agent and 8 meters in front. It is heading right of the ego agent. Its current speed is 3 m/s. It is decelerating.

The **blue** and **orange** sections contain answers to questions appearing in WOMD-Reasoning: **blue** addresses the Road Environment and Ego Agent Status, while **orange** covers Other Agent Status.

To generate this context, we prompt the Motion-LLaVA model with relevant questions and then concatenate the answers to create a complete context.

A.9 Example Factual Description Context for Chain-of-Thought Reasoning

We provide an example of a factual description context, generated using our motion data translation program as part of constructing WOMD-Reasoning, as described in Appendix A.1.1. This context is used by Motion-LLaVA to infer answers to reasoning-based questions and corresponds to the **CoT w/ Facts Contexts** configuration in Table A.8. Notably, these contexts exclude future information compared to the translated motion data, thereby avoiding any information leakage:

The ego agent is in intersection. The intersection center is 6.0 meters in front of the ego agent, and is 0.0 meters on the left of the ego agent. The intersection is a 4 way intersection. It is turning left. It is exiting the intersection. Its current speed is 3 m/s. It is accelerating. Traffic Light for the ego agent is green. The ego agent is approaching a crosswalk 3 meters ahead.

Surrounding agent #7 is a vehicle. It is 26 meters on the left of the ego agent, and is 8 meters in front of the ego agent. It is heading right of the ego agent. Its current speed is 4 m/s. It is decelerating. It is heading towards the intersection. It is on the left of the intersection. Traffic Light for it is red 14.5 meters ahead. It is 26 meters away from the intersection center. It is approaching a crosswalk 4 meters ahead.

Similar to the aggregated context in Appendix A.8, the **blue** and **orange** sections contain answers to questions appearing in WOMD-Reasoning: **blue** addresses the Road Environment and Ego Agent Status, while **orange** covers Other Agent Status.

A.10 Ablations on Motion-LLaVA

Chain-of-Thought Inference. To address the hallucinated response problem [46, 176, 159] in reasoning interactions, we use a Chain-of-Thought (CoT) approach to provide Motion-LLaVA with structured context for more accurate reasoning. Table A.8 compares model performance with and without CoT on interaction and intention analysis. We see that with CoT, the model performs better in GPT Scores. We further show that if our motion data translation program A.1.1 is used during inference to provide contexts, i.e., with factual

Table A.8: Effects of CoT in answering reasoning-related questions.

Reasoning Inference Method	ROUGE (↑)	BLEU (↑)	METEOR (↑)	CIDEr (↑)	SPICE (↑)	GPT Score (↑)
Direct	0.616	0.474	0.365	2.54	0.578	6.59
CoT	0.614	0.474	0.366	2.52	0.571	6.76
CoT w/ Facts Contexts	0.664	0.542	0.403	3.10	0.631	7.64

Table A.9: Benefits of using pre-trained motion vector encoder for fine-tuning multi-modal models on WOMD-Reasoning for factual Q&As.

Motion Vector Encoder Pre-training	ROUGE (↑)	BLEU (↑)	METEOR (↑)	CIDEr (↑)	SPICE (↑)	Acc _{1.0} (↑)	Median Error(↓)	GPT Score (↑)
None	0.814	0.684	0.491	5.95	0.774	30.0%	2.00	6.79
Motion Caption	0.824	0.711	0.498	6.05	0.775	42.6%	2.24	6.97
AutoEncoder	0.837	0.727	0.516	6.35	0.803	47.3%	1.41	7.07
Motion Prediction (Ours)	0.840	0.736	0.516	6.35	0.794	56.2%	1.00	7.09

description context (an example can be found in Appendix A.9) and CoT, the performance can be significantly improved, proving the potential of the CoT inference strategy.

Influence of Pre-training for the Motion Vector Encoder. To establish a pre-training strategy for the motion vector encoder, we examine factual question answering across Motion-LLaVA with four pre-training configurations: no pre-training, autoencoder-style pre-training, motion-caption pre-training [20], and pre-training with a motion prediction task. All configurations shared the same encoder structure. In the motion-caption pre-training, we pre-train all but LLM weights in Motion-LLaVA, by penalizing errors in captioning the motion. For autoencoder-style pre-training and pre-training with a motion prediction task, we retain only the encoder after pre-training, to function as the motion vector encoder for Motion-LLaVA.

The performances of models with these pre-training settings are presented in Table A.9. Generally, pre-training motion vector encoders can improve the response quality. Specifically, pre-training with motion prediction tasks outperforms the other two approaches. We attribute this to the close alignment between WOMD-Reasoning’s question categories and the information needed in motion prediction tasks. As a result, pre-training with the motion prediction task achieves the highest GPT Score and overall accuracy, especially in numerical precision reflected by $Acc_{1.0}$.

Motion-LLaVA Fine-tuning Strategy. The original LLaVA [113] training pipeline consists of two stages: a projector alignment stage (also known as pre-training) to align vision and language features, followed by a fine-tuning stage to enhance instruction-following capabilities. Throughout this process, the vision encoder remains frozen. Since our Motion-LLaVA builds on the architecture of the original LLaVA [113], it seems reasonable to adopt a similar training approach for the WOMD-Reasoning task.

Table A.10: Refinement of Motion-LLaVA training strategy.

Multi-modal Fine-tuning Strategy	Projector Alignment	Encoder Freezing	ROUGE (↑)	BLEU (↑)	METEOR (↑)	CIDEr (↑)	SPICE (↑)	Acc _{1.0} (↑)	Median (↓)	GPT Score (↑)
LLaVA	✓	✓	0.821	0.678	0.484	5.82	0.771	2.3%	12.00	6.17
– Projector Alignment	✗	✓	0.821	0.697	0.492	5.96	0.782	2.3%	8.06	6.49
– Encoder Freezing (Ours)	✗	✗	0.840	0.736	0.516	6.35	0.794	56.2%	1.00	7.09

Table A.11: Performance of Motion-LLaVA with different WOMD-Reasoning dataset sizes on factual and interaction reasoning questions. Results confirm the data efficiency of WOMD-Reasoning in improving driving-related reasoning.

Dataset Size	Factual Questions						Interaction Reasoning Questions					
	ROUGE	BLEU	METEOR	CIDEr	SPICE	GPT	ROUGE	BLEU	METEOR	CIDEr	SPICE	GPT
0.5M	0.806	0.683	0.477	5.77	0.760	6.69	0.562	0.410	0.339	1.94	0.513	5.67
1M	0.818	0.701	0.491	5.90	0.773	6.74	0.589	0.447	0.354	2.23	0.547	6.36
3M	0.840	0.736	0.516	6.35	0.794	7.09	0.614	0.474	0.366	2.52	0.571	6.76

However, based on extensive experiments with fine-tuning strategies in Table A.10, we find that incorporating projector alignment stage or keeping the motion vector encoder frozen throughout, which are practices employed in LLaVA [113], resulted in degraded performance on WOMD-Reasoning. Consequently, our Motion-LLaVA adopts a single-stage fine-tuning approach, where the motion vector encoder is unfrozen.

A.11 WOMD-Reasoning Dataset Efficiency Analysis

To assess the data efficiency of WOMD-Reasoning, we conduct scaling studies to analyze how model performance improves as the used dataset’s size increases, from 0.5M to 1M and up to 3M Q&A pairs (full dataset).

In Table A.11, we find that with more WOMD-Reasoning data involved, the fine-tuned model first gains the ability to answer factual questions, which hits a good GPT score with only 0.5M data. As the dataset grows, the model gradually learns to answer interaction reasoning questions. These justify the size of WOMD-Reasoning, as well as its data efficiency in helping models to answer hierarchical driving-related questions.

Appendix B

Appendix for Immiscible Diffusion: Accelerating Diffusion Training with Noise Assignment

B.1 Additional Experiment Results

B.1.1 Quantitative Training Efficiency Improvements for Immiscible Consistency Model

Table B.1: Immiscible Diffusion boosts training efficiency for Consistency Model on CIFAR-10 dataset.

FID threshold	12.00	11.00	10.00
Training Steps (k) for Baseline Consistency Model	290	450	>800
Training Steps (k) for Immiscible Consistency Model	110	140	190

B.1.2 Quantitative FID Improvements for Immiscible DDIM with Different Inference Steps.

B.1.3 Ablation on the Distance Measurement Methods in Noise Assignment.

We use the L2 norm for our experiments. However, we note that the L2 norm may face more challenges in distance evaluation in high-dimensional spaces compared to the L1 norm. Therefore, we compare the performance of immiscible DDIMs using assignments based on

Table B.2: FID improvements of Immiscible DDIM with different inference steps

Inference Steps	1000	500	100	50	20
FID with baseline DDIM	3.82	3.91	5.2	6.63	10.03
FID with Immiscible DDIM	3.67	3.74	4.32	5.14	7.46
Δ FID	-0.15	-0.17	-0.88	-1.49	-2.57

the L1 and L2 norms. The results, as illustrated in Tab. B.3, show that using the L2 norm provides better performance than the L1 norm.

Table B.3: FID of using L1 or L2 norm for noise assignment in immiscible DDIM on CIFAR-10.

Training Steps (k)	70.2	93.6	117.0	140.4	163.8
DDIM	6.30	5.56	4.86	4.34	4.12
Immiscible DDIM using L2 Norm	5.28	4.56	4.13	3.81	3.70
Immiscible DDIM using L1 Norm	5.34	4.66	4.16	3.87	3.82

B.1.4 Ablation on the Batch Size on Immiscible DDIM.

The effectiveness of image-noise assignment can intuitively rely on the batch size. Therefore, we perform a comparison to see the effectiveness of Immiscible Diffusion on DDIM across a selected range of batch sizes, whose result is shown in Fig. B.1. We observe that while larger batch sizes consistently accelerate the training as expected, its training efficiency enhancement is not as large as that from Immiscible Diffusion. Immiscible Diffusions continuously improve the training efficiency and the performance in the whole selected range of batch sizes.

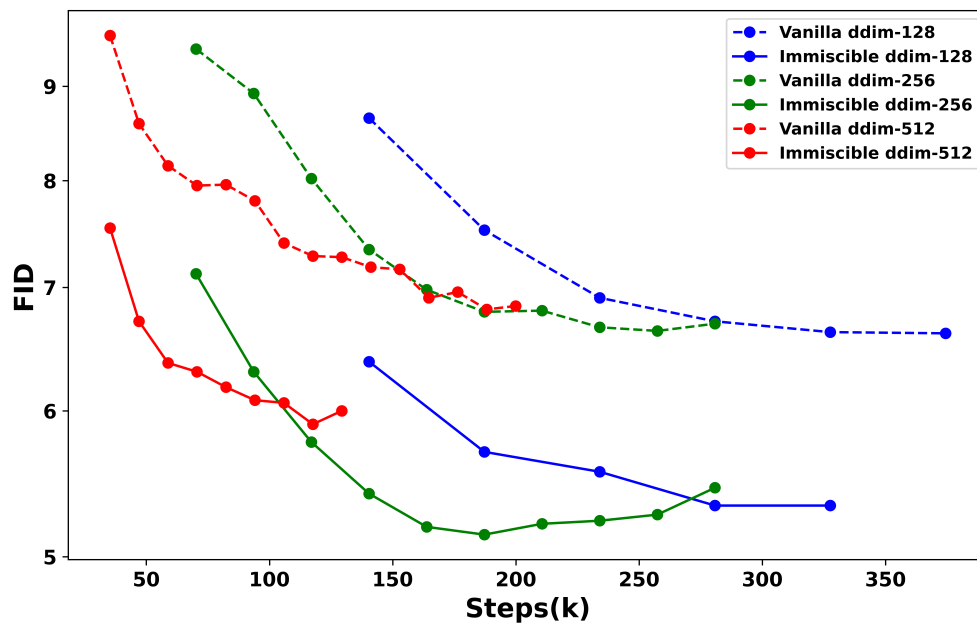


Figure B.1: Effectiveness of Immiscible DDIM in a selected range of batch sizes.

B.2 Qualitative Evaluations of Immiscible Diffusion

B.2.1 Generated images from immiscible and baseline Consistency Models trained on CIFAR-10 (Top) and CelebA (Down) with the same training steps.

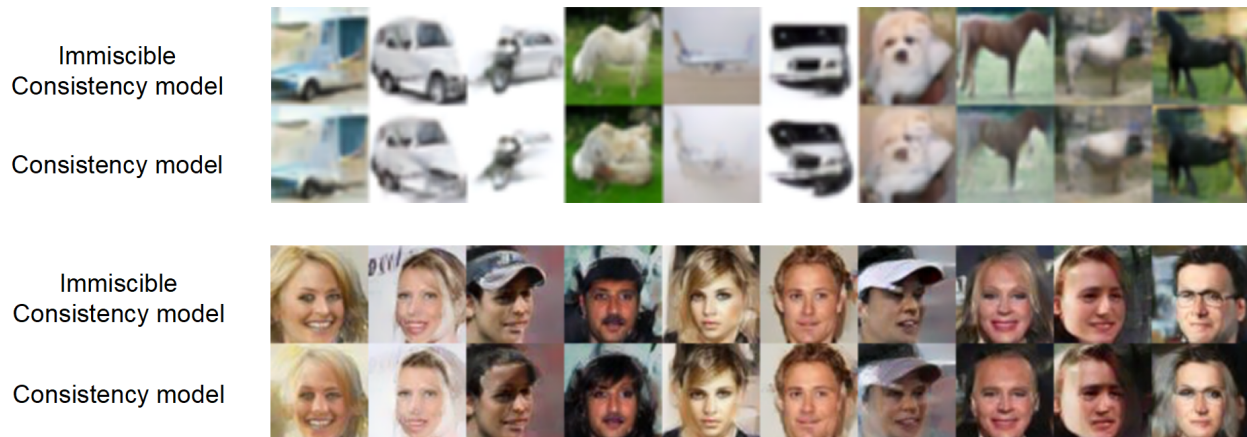


Figure B.2: **Qualitative comparison for Immiscible and baseline Consistency Model.** We show images generated with the two models trained for 100k steps respectively. Compared to baseline method, immiscible models capture more details and more features of objects.

B.2.2 Generated images from immiscible and baseline Consistency Models trained on CIFAR-10 Dataset for 100k steps without cherry-picking.



Figure B.3: Generated images from baseline Consistency Models trained on CIFAR-10 Dataset for 100k steps without cherry-picking.



Figure B.4: Generated images from immiscible Consistency Models trained on CIFAR-10 Dataset for 100k steps without cherry-picking.

B.2.3 Generated images from immiscible and baseline Consistency Models trained on CelebA Dataset for 100k steps without cherry-picking.



Figure B.5: Generated images from baseline Consistency Models trained on CelebA Dataset for 100k steps without cherry-picking.



Figure B.6: Generated images from Immiscible Consistency Models trained on CelebA Dataset for 100k steps without cherry-picking.

B.2.4 Generated images from immiscible and baseline stable diffusion models trained unconditionally on 10% ImageNet for 70k steps.

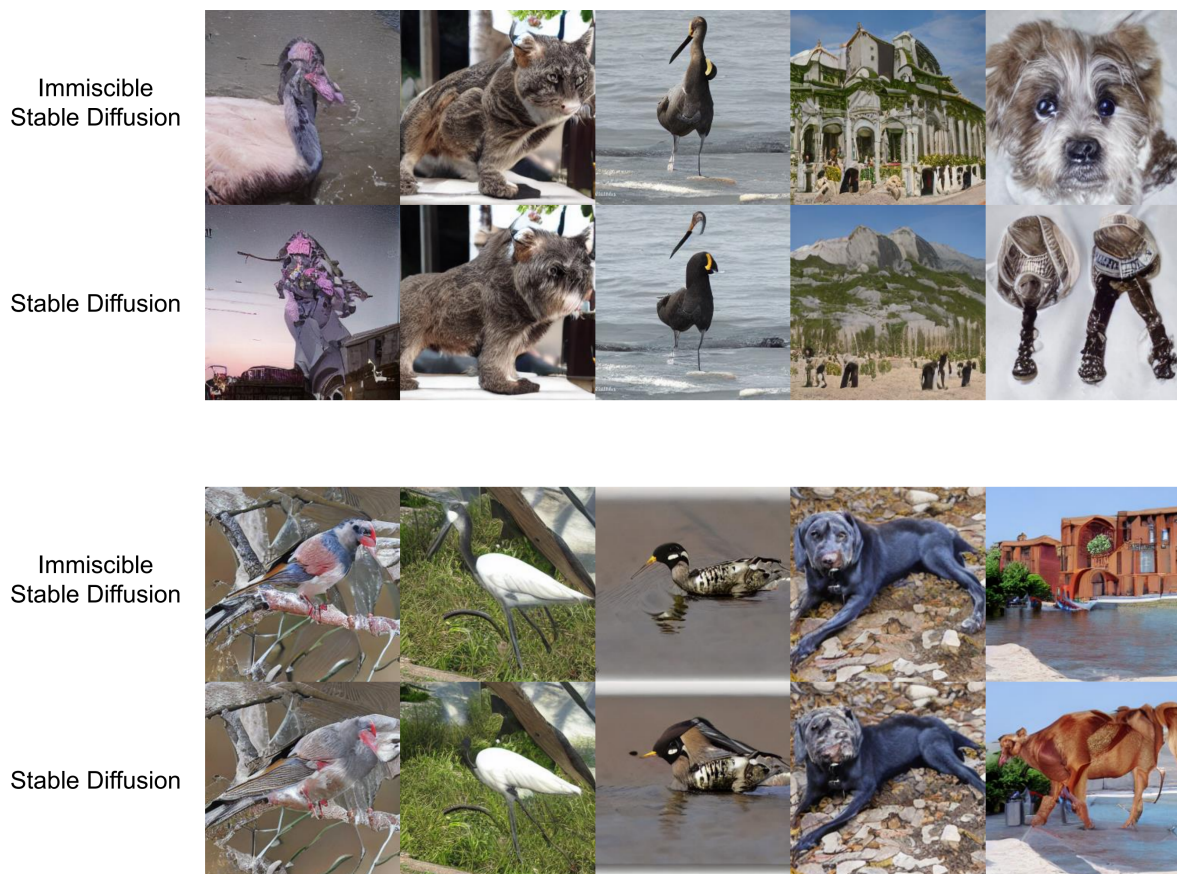


Figure B.7: Images generated by immiscible and baseline Stable Diffusion trained unconditionally on ImageNet for 70k steps. We see that the Immiscible Stable Diffusion presents more reasonable modal and catch more general features and details.

B.2.5 Generated images from immiscible and baseline stable diffusion models trained unconditionally on 10% ImageNet Dataset for 70k steps without cherry-picking.

Immiscible
Stable Diffusion

Stable Diffusion



Immiscible
Stable Diffusion

Stable Diffusion



Immiscible
Stable Diffusion

Stable Diffusion



Figure B.8: Generated images from immiscible and baseline stable diffusion models trained unconditionally on 10% ImageNet Dataset for 70k steps without cherry-picking

B.2.6 Generated images from immiscible and baseline stable diffusion models trained conditionally on ImageNet Dataset for 20k steps.

Immiscible
Stable Diffusion



Stable Diffusion



Figure B.9: Generated images from immiscible and baseline stable diffusion models trained conditionally on ImageNet Dataset for 20k steps.

Appendix C

Appendix for Improved Immiscible Diffusion: Accelerate Diffusion Training by Reducing Its Miscibility

C.1 Experiment Setup Details

Table C.1: Image Generation Experiment setting.

Model	Consistency Model	DDIM	Flow Matching	Stable Diffusion Class-conditional	Stable Diffusion Conditional	Stable Diffusion Fine-tuning
Dataset	CIFAR-10	CIFAR-10	CIFAR-10	ImageNet	MS-COCO	ImageNet
Batch Size	512	256	256	2048	2048	512
Resolution	32×32	32×32	32×32	256×256	256×256	256×256
Devices	$4 \times A6000$	$1 \times A5000$	$1 \times A6000$	$8 \times A800$ or $4 \times A6000$	$4 \times A6000$	$4 \times A6000$

The training hyperparameter are listed in Table C.1. Unspecified hyper-parameters are taken the same as those in their baseline methods’ original papers. For evaluations, we compare the generated images by Immiscible Diffusion and the baseline using the quantitative evaluation metric FID [63]. Note that for Consistency Models, we use the single-step generation consistency training. For DDIM, we add no noise during the sampling and use linear scheduling to select sampling steps. For Stable Diffusion, we directly use the implementation from Diffusers of Huggingface team [149]. For fine-tuning, we use Stable Diffusion v1.4 [149] as the pre-trained model. Image in-painting and out-painting does not involve additional training, and details on robotics plannings will be listed in the Section 5.4.5.

C.2 Qualitative Comparisons of Image-to-image Tasks

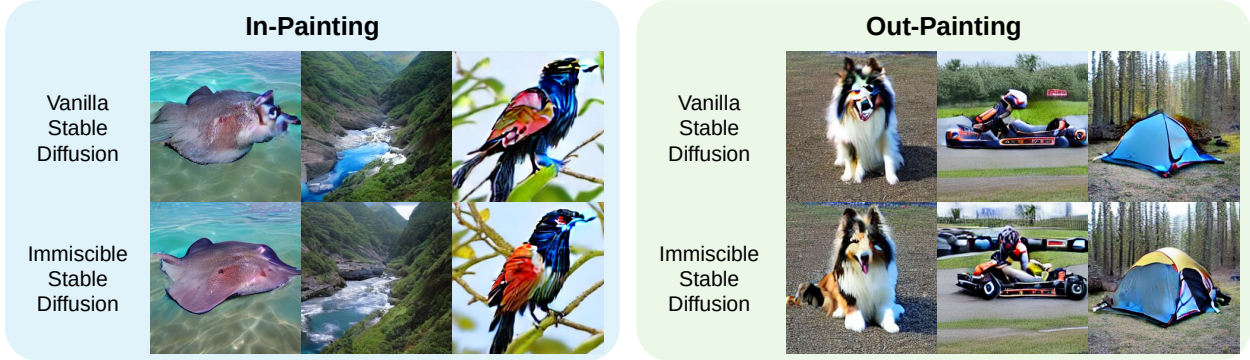


Figure C.1: **Qualitative Illustration of Immiscible Diffusion in Image-to-image Tasks.** We notice that immiscible diffusion can better preserve existed information (i.e. edge information in in-painting and center information in out-painting) so as to provide better overall completed images.

C.3 Choice of k in the KNN implementation

Model Dataset	DDIM CIFAR-10	Flow Matching CIFAR-10	Consistency Model CIFAR-10	Stable Diffusion ImageNet-1k
Diffusion Dimension	3,072	3,072	3,072	4,096
Best k	8	4	4	64
L_2 Dist. Δ (%)	-1.58%	-1.10%	-1.10%	-2.32%

Table C.2: Best k 's in KNN Immiscible Diffusion Implementation.

We investigate the k values for different method and dataset pair, as shown in Table C.2. We find that while the best k for the same dataset's diffusion dimension is generally the same, with small fluctuations observed, datasets with larger data dimension needs larger k to provide stronger immiscibility. It is noteworthy that for each experiment, we only try $k = 1, 2, 4, 8, 16, 32, 64, 128, \dots$ to save computational resources. Finer experiments with more k 's will provide more precise results.

C.4 tSNE of Noise Point Clusters for Different Methods

The tSNE of noise point clusters for different methods is shown in Fig. C.2.

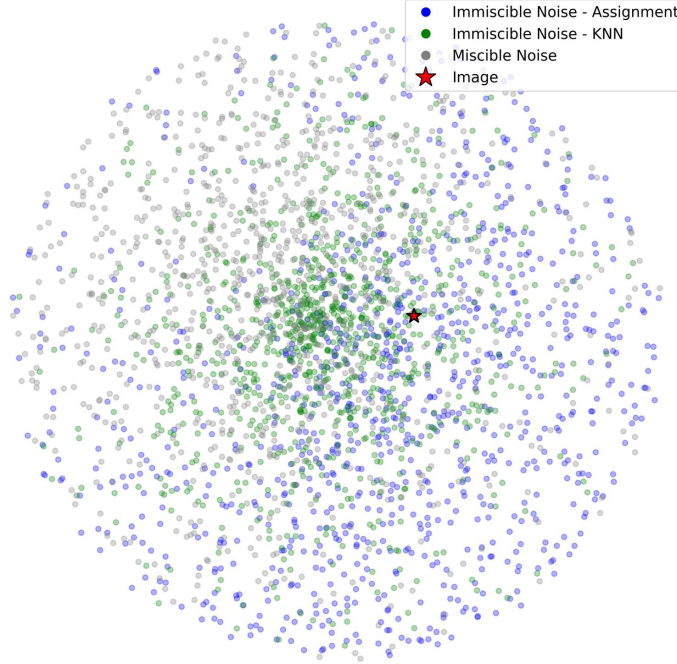


Figure C.2: tSNE of noise clusters belonging respectively to vanilla, assign and KNN DDIM.

Table C.3: Execution Time (ms) of Immiscible Diffusion on a single A5000 GPU.

Batch Size	128	256	512	1024
Linear Assignment [104]	5.4	6.7	8.8	22.8
KNN	0.2	0.2	0.3	0.7
t_{assign}/t_{knn}	27.0x	33.5x	29.3x	32.6x

C.5 Illustrations Comparing Immiscible Diffusion to Other Common Methods

The illustrations comparing immiscible diffusion to other common methods are shown in Fig. C.3

C.6 Execution Time of Immiscible Diffusion

We evaluate the execution time of two immiscible diffusion implementations: linear assignment and KNN. As shown in Table C.3, we find that the KNN implementation is much faster than the linear assignment implementation, demonstrating the efficiency of our proposed alternative implementation to achieve immiscibility.

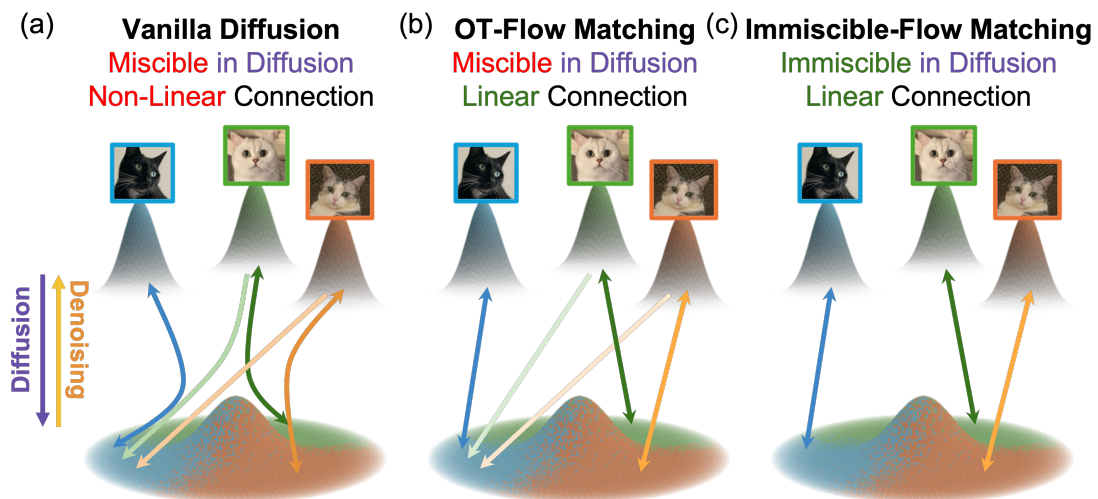


Figure C.3: Comparison between Diffusion Models, Flow Matching and Immiscible Flow Matching.