DiffAD: A Unified Diffusion Modeling Approach for Autonomous Driving

Tao Wang¹, Cong Zhang¹, Xingguang Qu², Kun Li¹, Weiwei Liu¹, Chang Huang¹
¹Carizon ²Beihang University

Abstract

End-to-end autonomous driving (E2E-AD) has rapidly emerged as a promising approach toward achieving full autonomy. However, existing E2E-AD systems typically adopt a traditional multi-task framework, addressing perception, prediction, and planning tasks through separate task-specific heads. Despite being trained in a fully differentiable manner, they still encounter issues with task coordination and the system complexity remains high. In this work, we introduce DiffAD—a novel diffusion probabilistic model that redefines autonomous driving as a conditional image generation task. By rasterizing heterogeneous targets onto a unified bird's-eye view (BEV) and modeling their latent distribution, DiffAD unifies various driving objectives and jointly optimizes all driving tasks in a single framework, significantly reduces system complexity and harmonizes task coordination. The reverse process iteratively refine the generated BEV image, resulting in more robust and realistic driving behaviors. Closed-loop evaluations in Carla demonstrate the superiority of the proposed method, achieving a new state-of-the-art Success Rate and *Driving Score. The code will be made publicly available.*

1. Introduction

Achieving full autonomy in driving requires not only a deep understanding of complex scenes but also effective interaction with dynamic environments and comprehensive learning of driving behaviors. Traditional autonomous driving systems are built upon a modular architecture, where perception, prediction, and planning are developed independently and then integrated into the onboard system. While this design offers interpretability and facilitates debugging, the separate optimization objectives across modules often lead to information loss and error accumulation.

Recent end-to-end autonomous driving (E2E-AD) approaches (e.g., [3, 16, 20]) have attempted to overcome these limitations by enabling joint, fully differentiable training of all components, as illustrated in Fig. 1(a). However, several critical issues remain:

1. Sub-optimal Optimization: Methods like UniAD [16]

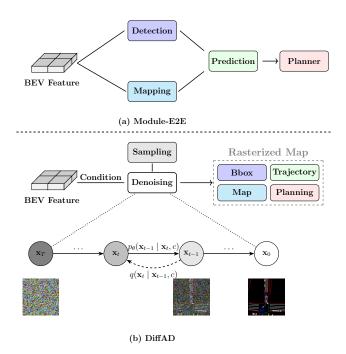


Figure 1. Paradigm overview. (a) Module-E2E adopts sequential pipelines where multi-task heads is optimized in a differentiable manner. (b) DiffAD (ours) integrates all components into a single denoising head and treat E2E-AD as a conditional image generation task, resulting a fully end-to-end joint optimization of all driving tasks.

and VAD [20] still rely on sequential pipelines, where the planning stage depends on the outputs of preceding modules. This dependency can amplify errors throughout the system.

- 2. **Inefficient Query Modeling**: Current query-based methods (e.g., [16, 20]) deploy thousands of learnable queries to capture potential traffic elements. This approach leads to an inefficient allocation of computational resources, with a disproportionate focus on upstream auxiliary tasks rather than the core planning module. For instance, in VAD, the perception task consumes 34.6% of the total runtime, while the planning module accounts for only 5.7%.
- 3. Complexity in Coordination: With each task head

optimized independently using distinct objective functions—and with targets varying in shape and semantic meaning—the overall system becomes fragmented and difficult to train cohesively [5].

To address these limitations, we propose a novel paradigm, DiffAD, which unifies the optimization of all driving tasks within a single model, as depicted in Fig. 1(b). Specifically, we rasterize heterogeneous targets from perception, prediction, and planning onto a unified bird's-eye view (BEV) space, thereby recasting the autonomous driving problem as one of conditional image generation. A denoising diffusion probabilistic model is employed to learn the distribution of the BEV image conditioned on surrounding views. This approach not only enables the simultaneous optimization of all tasks—thereby mitigating error propagation—but also replaces inefficient vector-based query methods with a computationally efficient generative modeling strategy in latent space using a shared decoding head. Moreover, by focusing solely on noise prediction without the need for multiple loss functions or complex bipartite matching, our method significantly simplifies the overall training procedure.

In summary, DiffAD overcomes the limitations of existing query-based, sequential methods by unifying tasks into a single, end-to-end framework that enhances coordination, reduces error propagation, and allocates computational resources more efficiently towards safe and effective planning. The main contributions of this paper are summarized as follows:

- We introduce an end-to-end paradigm for autonomous driving that leverages a unified, fully rasterized BEV representation to integrate diverse driving tasks into a single model.
- We reformulate driving tasks as a conditional image generation problem and present **DiffAD**, a diffusion model that learns the latent distribution of BEV images conditioned on surrounding views. Additionally, we propose a data-driven method to extract vectorized planning trajectories from the generated BEV images.
- We demonstrate that DiffAD achieves state-of-the-art performance in end-to-end planning, significantly outperforming previous methods in closed-loop evaluations.

2. Related Work

We cover previous works on End-to-End autonomous driving, Driving VLM and Diffusion Model for the downstream tasks of end-to-end driving.

End-to-End Autonomous Driving. Traditionally, autonomous driving systems are composed of separate modules for detection [25, 30, 35, 46], mapping [24, 27, 31], prediction [9, 12, 32], and planning. While this modular

design facilitates task-specific optimizations, it often leads to information loss and error accumulation when integrating these components, resulting in suboptimal planning decisions. End-to-end (E2E) approaches seek to address these limitations by unifying all tasks within a fully differentiable framework, thereby enabling planning-oriented optimization. For instance, methods such as UniAD [16] and VAD [20] utilize query-based architectures to transfer information from perception to planning. Paradrive [47] employs a parallel learning pipeline, directly optimizing all tasks from dense BEV features, while SparseAD [51] adopts a sparse query-based strategy to bypass the inefficiencies of dense feature construction. However, these approaches rely on multi-head instance query modeling, which can introduce coordination challenges across tasks and lead to an inefficient allocation of computational resources toward auxiliary tasks rather than core planning. In contrast, our method, DiffAD, redefines autonomous driving as a conditional image generation task in a unified bird's-eye view space. By jointly optimizing perception, prediction, and planning within a single diffusion framework, DiffAD streamlines the overall optimization process, mitigates error propagation, and prioritizes safe and coherent planning outcomes.

Vision-Language Models (VLMs) for Driving. Recent works [6, 38, 44, 49] have explored applying large Vision-Language Models (VLMs) to autonomous driving. These models leverage the reasoning capabilities of large language models (LLMs) to provide natural language explanations for driving decisions, enhancing interpretability and generalization. However, deploying such models on edge devices for real-time inference remains challenging, and LLMs are prone to generating inaccurate or misleading outputs (hallucinations), which could compromise safety in autonomous driving.

Diffusion Model for Autonomous Driving. Denoising diffusion models [14, 42] have recently emerged as a powerful class of generative models, achieving state-of-the-art results in diverse applications such as image generation [7, 11, 37, 52], video generation [1, 15], and image editing [22]. In autonomous driving, several works have begun exploring their potential. For example, DiffBEV [53] leverages conditional diffusion models to generate a refined BEV representation with reduced noise, while Poly-Diffuse [4] employs guided diffusion to reconstruct polygonal shapes for mapping, and MotionDiffuser [21] utilizes diffusion-based representations to predict multi-agent trajectories. Similarly, DiffusionDrive [28] adopts a truncated diffusion process to capture multi-modal trajectory distributions. However, these methods primarily focus on refining specific components-improving BEV perception or trajectory prediction—and are built upon existing sequential multi-task frameworks. In contrast, our work is the first to formulate holistic end-to-end autonomous driving as a conditional image generation problem. By jointly learning perception, prediction, and planning within a unified diffusion framework, our approach significantly simplifies system architecture and enhances task coordination. This fundamental shift in problem formulation not only streamlines the overall optimization process but also leverages the inherent denoising capabilities of diffusion models to produce more robust and realistic driving decisions.

3. Preliminary: Diffusion Models

Diffusion models, also known as score-based generative models [14, 39, 42], progressively inject noise into the data during the forward (diffusion) process and generate data from noise through the reverse (denoising) process. This section provides key preliminary knowledge about denoising diffusion probabilistic models (DDPM), laying the groundwork for DiffAD.

Forward Process. DDPM considers a diffusion process that transforms data $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ into Gaussian noise:

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

for t = 1, ..., T, where \mathbf{x}_t represents the latent variable at time t. The noise schedule β_t can either be constant [14, 39] or learned via reparameterization [33]. This forward process can be expressed in closed form for any t:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha}_t)\mathbf{I}), \tag{2}$$

where $\bar{\alpha}_0 = 1$, $\bar{\alpha}_t := \prod_{s=1}^t \alpha_s$, and $\alpha_t := 1 - \beta_t$. The latent variable \mathbf{x}_t is then a linear combination of \mathbf{x}_0 and noise:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$
 (3)

Reverse Process. Diffusion models are trained to learn the reverse process $p_{\theta}(x_{t-1}|x_t) = \mathcal{N}(\mu_{\theta}(x_t), \Sigma_{\theta}(x_t))$, where neural networks predict the parameters of p_{θ} . The DDPM uses a noise prediction (denoising) network $\epsilon_{\theta}(\mathbf{x}_t,t)$ to connect the process with denoising score matching and Langevin dynamics [41, 45]. The sampling step of the reverse process is derived as:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left[\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_{\theta}(\mathbf{x}_t, t) \right] + \sigma_t \mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$
(4)

where σ_t^2 is set to either β_t or $\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$. The final training objective is a reweighted variational lower bound:

$$L_{\text{simple}}(\theta) := \mathbb{E}_{\mathbf{x}_0, t, \epsilon} \left[\| \epsilon - \epsilon_{\theta} (\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t) \|^2 \right]. \tag{5}$$

4. DiffAD

Overview. As illustrated in Fig. 2, DiffAD consists of three main components: a Latent Diffusion Model, a BEV Feature Generator, and a Trajectory Extraction Network (TEN).

Training Process:

- Rasterization and Latent Space Encoding: DiffAD
 begins by rasterizing the perception, prediction, and
 planning targets into a BEV image. An off-the-shelf
 VAE Encoder is then used to compress the BEV image
 into a latent space for dimensionality reduction.
- Feature Extraction and Transformation: Surrounding view images are fed into a feature extractor, which transforms the resulting perspective-view features into unified BEV features.
- Diffusion Model for Noise Prediction: Gaussian noise
 is added to the latent BEV image to obtain noisy latent.
 A diffusion model is trained to predict the noise from
 the noisy latent representation, conditioned on the BEV
 features.
- 4. **Trajectory Extraction:** A query-based TEN is trained to recover the vectorized trajectory of the ego agent from the latent BEV image.

Inference Process:

- Conditional Denoising: DiffAD first generates a denoised latent BEV image from pure Gaussian noise, conditioned on the BEV features.
- Planning Extraction: The TEN then extracts the planned trajectory of the ego agent from the latent BEV image.
- 3. **Decoding BEV:** By decoding the latent BEV image back into pixel space, we can obtain the predicted BEV image for interpretation and debugging.

4.1. Rasterized BEV Representation

Perceiving surrounding traffic agents and map elements is essential for understanding the driving scene, while predicting agents' trajectories is crucial for making safe driving decisions. DiffAD utilizes a rasterized BEV representation to unify the heterogeneous targets of driving tasks—such as bounding boxes, lane elements, agent trajectories, and ego vehicle planning.

Specifically, we rasterize the bounding boxes and map elements onto an RGB canvas for the perception task, denoted as $m_{perc} \in \mathbb{R}^{3 \times H \times W}$, where different semantic elements are represented using distinct colors. For the trajectory prediction task, agents' trajectories are drawn onto second RGB canvas, $m_{pre} \in \mathbb{R}^{3 \times H \times W}$. Finally, the ego vehicle's future trajectory is rasterized on third RGB canvas for the planning task, $m_{plan} \in \mathbb{R}^{3 \times H \times W}$. The color of the trajectories is interpolated over time to represent the temporal relationship between points.

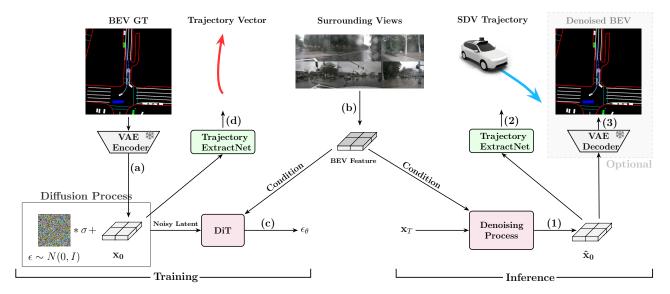


Figure 2. Pipeline of DiffAD. Training Process: (a) DiffAD rasterizes perception, prediction, and planning targets onto a BEV image, which is encoded into a latent space \mathbf{x}_0 using a VAE. (b) Surrounding images are transformed into BEV feature. (c) A diffusion model predicts noise ϵ_{θ} from the noisy latent BEV image, and (d) a trajectory extraction network (TEN) learns to recover the ego trajectory from the latent BEV image. Inference Process: (1) DiffAD generates a denoised latent BEV image $\hat{\mathbf{x}}_0$ from pure Gaussian noise \mathbf{x}_T , conditioned on BEV feature, (2) extracts the ego trajectory via TEN, and (3) decodes the latent BEV image for interpretation.

This unified BEV representation allows the diffusion model to simultaneously learn the tasks of perception, prediction, and planning. Moreover, it enables reasoning about the physical relationships and social interactions between the ego vehicle and its surroundings, leading to scene-level consistency results across all tasks.

4.2. Denoising Diffusion Learning

Following the Latent Diffusion Model (LDM) framework [36], we utilize a VQ-VAE to compress the rasterized BEV images into a low-dimensional latent space. The latent representations of perception, prediction, and planning are then concatenated along the channel dimension to construct the latent BEV image $z_{bev} \in \mathbb{R}^{c' \times h' \times w'}$.

$$[z_{perc}, z_{pre}, z_{plan}] = \operatorname{encoder}([m_{perc}, m_{pre}, m_{plan}]),$$

$$z_{bev} = \operatorname{concat}([z_{perc}, z_{pre}, z_{plan}])$$
(6)

Next, noise ϵ is added via the diffusion process to produce a noisy latent image $\{z_t\}_{t=0}^T$ at each timestep t, where $z_0=z$. The noisy latent image is divided into tokens and passed through multiple layers of DiT [34], with an MLP layer used at the end to predict the noise ϵ_{θ} .

Conditional Denoising. DiffAD utilizes multi-view images and driving commands [16, 20] as conditions to guide the denoising process. For the conditional guidance mechanism, we adopt Adaptive Layer Normalization (AdaLN)

with zero-initialization [34] due to its effectiveness and efficiency. Specifically, we employ BEVFormer [25] to convert multi-view images into a BEV feature map $x_{bev} \in \mathbb{R}^{c_{bev} \times h' \times w'}$, then the BEV feature x_{bev} is tokenized and combined with the timestep embedding and driving command embedding x_{cmd} as input to AdaLN.

$$cond = t_{emb} + x_{bev} + x_{cmd},$$

$$z_t = AdaLN(z_t, cond)$$
(7)

Temporal-Consistency. Planning is fundamentally a sequential decision-making task, where the agent must make decisions based on its current status and the dynamics of the environment. To capture temporal information, we adopt ConvLSTM to fuse historical BEV features. However, fusing BEV features alone is insufficient to ensure consistent planning over time. To address this problem, We introduce a Action-Guidance mechanisms, where we take an assumption that the current decision depends not only on the current observation but also on the last action. Thus, the joint distribution can be modeled as $\prod_{t=1}^{T} q(a_t \mid s_t, a_{t-1})$, where s_t represents the agent's state at time t, a_t represents the action taken at time t. For the implementation, we condition the current output on the previous latent BEV image z_{hev}^{t-1} [10] to regularize the current decision. This approach, however, could lead to a shortcut, where the network overrelies on the previous latent BEV image and neglects the current observation. To mitigate this issue, we introduce a dropout regularization on the previous latent BEV image tokens with a probability of 0.5. The final conditional guidance is formulated as follows:

cond =
$$t_{emb} + x_{bev} + x_{cmd} + \sigma_{p=0.5}(z_{bev}^{t-1}),$$
 (8)

where $\sigma_{p=0.5}$ represents the random dropout operation.

4.3. Trajectory Extraction Network

To obtain a vectorized trajectory for ego vehicle control, we need to recover the trajectory from the latent space. A straightforward approach would be to decode the latent BEV image back into pixel space and apply a rule-based post-processing method. However, to improve generalization and robustness, we opt for a data-driven approach.

Specifically, we design a query-based transformer network[2] to extract the trajectory from the latent BEV image. First, the latent BEV image $z_{bev} \in \mathbb{R}^{12 \times h' \times w'}$ is split into a sequence of tokens $X \in \mathbb{R}^{L \times D}$ through an embedding layer f_{emb} . A learnable query $Q_{\rm ego} \in \mathbb{R}^{T \times D}$ interacts with the tokenized sequence via a series of transformer layers. Finally, a single MLP decodes the learned query into the predicted trajectory $\hat{\tau} \in \mathbb{R}^{T \times 2}$. The process is summarized as follows:

$$X = f_{emb}(z_{bev}),$$

$$Q'_{ego} = \text{Transformer}(Q = Q_{ego}, K = X, V = X), \quad (9)$$

$$\hat{\tau} = \text{MLP}(Q'_{ego}).$$

4.4. End-to-End Learning

DiffAD is fully end-to-end trainable, based on the rasterized BEV representation and the diffusion model. Unlike traditional Module-E2E approaches, which involve multiple loss functions for different driving tasks, our system simplifies the optimization by using a unified loss function: the noise regression loss for denoising, and a trajectory extraction loss for the vectorized trajectory.

$$\mathcal{L} = \mathcal{L}_{\text{denoising}} + \mathcal{L}_{\text{extraction}} \tag{10}$$

Denoising Loss. We use the standard mean squared error (MSE) loss to optimize the diffusion model, ensuring it can accurately recover the noise from the noisy latent BEV image.

$$\mathcal{L}_{\text{denoising}} = \frac{1}{N} \sum \|\epsilon_{\theta} - \epsilon\|^2$$
 (11)

Trajectory Extraction Loss. The trajectory extraction loss is also based on MSE, applied between the predicted trajectory $\hat{\tau}$ and the ground truth ego trajectory τ . This loss ensures that the network can accurately recover the vectorized trajectory from the latent BEV image.

$$\mathcal{L}_{\text{extraction}} = \frac{1}{N} \sum \|\hat{\tau} - \tau\|^2$$
 (12)

5. Experiments

5.1. Datasets

Open-loop evaluations have been reported as insufficient for E2E models [19, 26]. To address this, we use the Bench2Drive dataset for training and closed-loop evaluation in the CARLA simulator[8]. Bench2Drive offers three data subsets: mini (10 clips for debugging), base (1,000 clips), and full (10,000 clips for large-scale studies). Following the methodology of [19], we use the base subset for training.

5.2. Metrics

- Success Rate (SR)[19]: This metric calculates the proportion of routes successfully completed without traffic violations within the allotted time.
- **Driving Score (DS)[19]:** This metric considers both route completion and penalties for infractions.
- **FID:** We use the Frechet Inception Distance (FID) [13] to assess scaling performance, which is a standard metric for evaluating generative models of images.

5.3. Baselines

- UniAD [16]: A classic module-based E2E approach that employs a query-based architecture to explicitly link perception, prediction, and planning tasks.
- VAD [20]: Another module-based E2E method, which enhances computational efficiency by utilizing Transformer Queries with a vectorized scene representation.
- AD-MLP [50]: A baseline model that predicts future trajectories by simply feeding the ego vehicle's historical states into an MLP.
- TCP [48]: A simple yet effective baseline, using only the front cameras and ego state to predict both trajectories and control commands.
- ThinkTwice [18]: A method that promotes a coarse-tofine framework, refining planning routes iteratively and leveraging expert feature distillation.
- DriveAdapter [17]: The top-performing method on the Bench2Drive leaderboard, which fully utilizes expert feature distillation to enhance performance by decoupling perception and planning.

5.4. Implementation Details

Training. We use an off-the-shelf pre-trained variational autoencoder (VAE) model [23] from Stable Diffusion[36]. The VAE encoder has a downsample factor of 8. Across all experiments in this section, our diffusion models operate in latent space. We retain diffusion hyperparameters from DiT [34]. To facilitate the learning process, we start with single image learning in the first stage for perception parts, i.e., detection and mapping, while prediction and planning BEV images are padding with zero. Then train the model jointly with all perception, prediction and planning parts in

Table 1. Multi-Abilit	and Overall Results of E2E-AD Methods in Bench2Drive. * denotes expert feature disti	Illation.

Method	Multi-Ability (%) ↑						Ove	Overall	
	Merging	Overtaking	Emergency Brake	Give Way	Traffic Sign	Mean	Driving Score ↑	Success Rate ↑	
AD-MLP	0.00	0.00	0.00	0.00	4.35	0.87	18.05	0.00	
UniAD-Tiny	8.89	9.33	20.00	20.00	15.43	14.73	40.73	13.18	
UniAD-Base	14.10	17.78	21.67	10.00	14.21	15.55	45.81	16.36	
VAD	8.11	24.44	18.64	20.00	19.15	18.07	42.35	15.00	
TCP*	16.18	20.00	20.00	10.00	6.99	14.63	40.70	15.00	
TCP-ctrl*	10.29	4.44	10.00	10.00	6.45	8.23	30.47	7.27	
TCP-traj*	8.89	24.29	51.67	40.00	46.28	34.22	59.90	30.00	
ThinkTwice*	27.38	18.42	35.82	50.00	54.23	37.17	62.44	31.23	
DriveAdapter*	28.82	26.38	48.76	50.00	56.43	42.08	64.22	33.08	
DiffAD (ours)	30.00	35.55	46.66	40.00	46.32	38.79	67.92	38.64	

temporal setting.

Inference. We utilize the DDIM-10 sampler [40] for inference and employ official evaluation tools [19] to compute closed-loop metrics. For vehicle control, we adopt the officially provided PID controller.

5.5. Main Results.

The overall results in Tab. 1 shows that DiffAD significantly outperforms baseline methods, including UniAD and VAD, and exceeds the performance of distillation-based approaches such as ThinkTwice and DriveAdapter. In multiability evaluations, DiffAD demonstrates notable advantages over UniAD and VAD in interactive scenarios like merging and emergency braking. This improvement is attributed to its integrated learning framework, which enables explicit interactions among task objectives, resulting in more coherent and effective planning. Due to the relatively small size of the training dataset, DiffAD exhibits slightly lower performance in Traffic Sign compared to methods utilizing expert feature distillation. Incorporating expert features, which provide valuable driving knowledge, could help mitigate potential overfitting. Consequently, models leveraging expert feature distillation (e.g., TCP, ThinkTwice, and DriveAdapter) generally outperform those without it (e.g., VAD and UniAD).

We conducted a failure case analysis of DiffAD, as shown in Fig. 3. The analysis reveals that a significant portion of route failures was caused by collisions with traffic agents, indicating the challenges of interacting in CARLA v2. Additionally, a small number of failures were attributed to timeouts, typically caused by the planning module's occasional inability to resume motion after stopping, this issue can be effectively mitigated by utilizing expert distillation or incorporating more traffic light interactions data. A

small percentage of failures occurred when the agent ran a red light, likely due to the low-quality rendering of traffic lights in CARLA or challenging lighting conditions, making them difficult to detect.

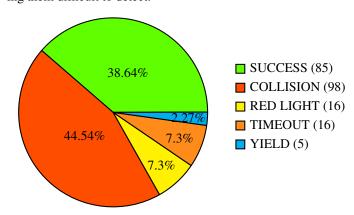


Figure 3. Status Distribution.

5.6. Ablation Study

The impact of denoising steps on performance DiffAD's iterative denoising follows a coarse-to-fine refinement process, progressively improving perception and planning. As shown in Tab. 2, increasing denoising steps from 3 to 10 significantly reduces FID (-53.5%) while improving Driving Score (+2.18) and Success Rate (+3.64), demonstrating that multi-step refinement helps resolve trajectory ambiguities. However, extending steps beyond 10 (e.g., to 20) leads to performance saturation, suggesting an optimal balance between computational overhead and planning precision.

The impact of tasks joint optimization We investigate the impact of jointly optimizing auxiliary tasks on planning

Table 2. The impact of denoising steps on performance

Steps	FID ↓	Driving Score ↑	Success Rate ↑
3	78.19	64.78	33.63
5	50.80	65.78	34.55
10	46.90	66.96	37.27
20	45.09	66.42	35.91

performance. As shown in Tab. 3, the full joint optimization of all three tasks achieves the best results, highlighting the importance of task joint optimization in enhancing planning performance.

Table 3. The impact of joint optimization of auxiliary tasks

Detection	Motion	Planning	Driving Score ↑	Success Rate ↑
	/	✓	30.11	5.9
\checkmark	/	\checkmark	59.10	29.09
\checkmark	\checkmark	\checkmark	66.96	37.27

The impact of Action Guidance dropout. Intuitively, relying too heavily on previous decisions can increase response latency in critical emergency scenarios. Conversely, making decisions without considering prior actions can lead to abrupt perception errors, resulting in unrealistic planning outcomes. To better understand this trade-off, we analyze the impact of different dropout rates in the Action Guidance module on planning performance. Tab. 4 presents the results for various dropout rates. A dropout rate of 0.95 achieves the best balance, indicating that retaining a small portion of previous action guidance is beneficial for robust planning.

Drop Rate	FID↓	Driving Score ↑	Success Rate ↑
0.5	47.37	66.28	35.91
0.75	47.22	66.47	35.45
0.95	47.08	67.92	38.64
1.0	46.90	66.96	37.27

Table 4. Impact of Action Guidance dropout on planning performance.

Efficiency of Unified Generative Modeling As shown in Tab. 5, despite having a larger parameter size (545.6M vs. 58.1M for VAD), DiffAD achieves competitive latency (258ms vs. 140ms) and real-time FPS (3.9). This efficiency is attributed to two key innovations:

• **Task-agnostic compression**: The VAE effectively compresses BEV image while preserving critical information,

Table 5. Comparison of parameters and FPS on GeForce RTX 4090

Model	Parameters(M)	Latency(ms)	FPS (↓)
UniAD-base	84.2	355	2.8
VAD-base	58.1	140	7.1
DiffAD-10steps	545.6	258	3.9

Table 6. Module runtime (ms) of DiffAD on GeForce RTX 4090.

Model	BEV	DiT (10 steps)	TEN	Total	FPS
FP32	41	214	3	258	3.9
TRT-FP16	1.6	40	1	42	23.8

significantly reduce the number of tokens for interactions and refinements in transformer layers.

• **Parallelized diffusion head**: Unlike sequential multitask pipelines, DiffAD employs a shared denoising network to optimize all driving tasks jointly, eliminating inefficiencies of cascaded inference.

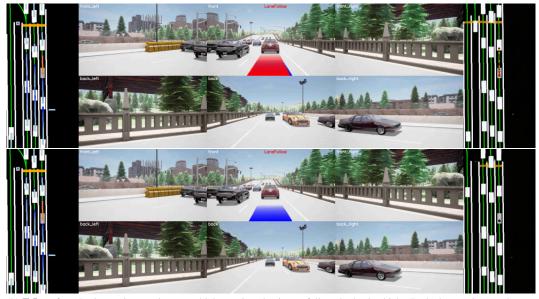
With TensorRT-FP16, DiffAD achieves 23.8 FPS (42ms inference, Tab. 6). Notably, 83% of its runtime is dominated by the diffusion process, which can be further optimized via distillation techniques [29, 43] with minimal performance trade-offs.

The Multi-modality of Generative Modeling In Fig. 4, we present qualitative results that showcase DiffAD's powerful generative capabilities and its ability to produce diverse planning outcomes. For each scenario, we generate two decisions by sampling different latent variables. To enhance clarity, we overlay planned trajectories (in red) and expert trajectories (in blue) onto the raw front-view image from surrounding cameras. The BEV ground truth (GT) is displayed on the left, while the predicted BEV is shown on the right. Notably, the generated BEV closely aligns with the ground truth, and the diverse planned trajectories is consistently safe and reasonable. This demonstrates DiffAD's ability to accurately perceive the environment and effectively learn interactive behaviors.

6. Conclusion and Future work

In this work, we present DiffAD, an end-to-end autonomous driving model built on a diffusion-based framework. Our key contribution lies in transforming heterogeneous targets of driving tasks into a unified rasterized representation, framing the E2E-AD as a conditional image generation task. This approach simplifies the problem and provides a clear pathway for leveraging various generative models, such as Diffusion models, GANs, VAEs, and auto-regressive models. We believe the strong performance of DiffAD high-

(a) Yielding to an Emergency Vehicle: In the top image, the ego vehicle changes lanes to the left and gradually merges with traffic to yield to an approaching emergency vehicle. In the bottom image, the ego vehicle cancels the lane change and returns to its original lane due to heavy traffic.



(b) **T-Junction:** In the top image, the ego vehicle moving slowing to follow the lead vehicle. In the bottom image, the ego vehicle comes to a stop, maintaining a safe distance as the lead vehicle slowly approaches the stop line.

Figure 4. Demonstration of the Model's Multi-Modal Decision-Making

lights the potential of generative models in advancing autonomous driving research and hope it inspires further exploration in the field.

Limitations and future work. Despite promising, the success rate on Carla v2 is still far from perfect. Effectively leveraging multi-modality generative predictions for

planning, as well as aligning the model outputs with human preferences, are worth for further exploration. Additionally, there is a significant gap between the traffic simulation in Carla and real-world conditions. To address this, we are working towards deploying the system onboard to evaluate its performance in real traffic scenarios.

References

- [1] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023. 2
- [2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-toend object detection with transformers. In *Computer Vision* – ECCV 2020: 16th European Conference, 2020. 5
- [3] Sergio Casas, Abbas Sadat, and Raquel Urtasun. Mp3: A unified model to map, perceive, predict and plan. In 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 1
- [4] Jiacheng Chen, Ruizhi Deng, and Yasutaka Furukawa. Polydiffuse: Polygonal shape reconstruction via guided set diffusion models. In Advances in Neural Information Processing Systems, 2023. 2
- [5] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 2
- [6] Tushar Choudhary, Vikrant Dewangan, Shivam Chandhok, Shubham Priyadarshan, Anushka Jain, Arun K. Singh, Siddharth Srivastava, Krishna Murthy Jatavallabhula, and K. Madhava Krishna. Talk2bev: Language-enhanced bird'seye view maps for autonomous driving. In 2024 IEEE International Conference on Robotics and Automation (ICRA), 2024. 2
- [7] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. In *Proceedings of the 35th Inter*national Conference on Neural Information Processing Systems, 2024. 2
- [8] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, 2017. 5
- [9] 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 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 2
- [10] Anirudh Goyal, Alex Lamb, Ying Zhang, Saizheng Zhang, Aaron Courville, and Yoshua Bengio. Professor forcing: a new algorithm for training recurrent networks. In *Proceedings of the 30th International Conference on Neural Infor*mation Processing Systems, 2016. 4
- [11] Alexandros Graikos, Nikolay Malkin, Nebojsa Jojic, and Dimitris Samaras. Diffusion models as plug-and-play priors. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2024. 2
- [12] Junru Gu, Chen Sun, and Hang Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021. 2
- [13] 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. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017. 5
- [14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Advances in Neural Information Processing Systems, 2020. 2, 3
- [15] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *Proceedings of the 36th International Con*ference on Neural Information Processing Systems, 2024. 2
- [16] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 1, 2, 4, 5
- [17] Xiaosong Jia, Yulu Gao, Li Chen, Junchi Yan, Patrick Langechuan Liu, and Hongyang Li. Driveadapter: Breaking the coupling barrier of perception and planning in end-to-end autonomous driving. In *ICCV*, 2023. 5
- [18] Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In CVPR, 2023. 5
- [19] Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. *arXiv* preprint arXiv:2406.03877, 2024. 5, 6
- [20] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. ICCV, 2023. 1, 2, 4, 5
- [21] Chiyu "Max" Jiang, Andre Cornman, Cheolho Park, Benjamin Sapp, Yin Zhou, and Dragomir Anguelov. Motiondiffuser: Controllable multi-agent motion prediction using diffusion. In 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023. 2
- [22] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023. 2
- [23] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Rep*resentations (ICLR), 2014. 5
- [24] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework. In 2022 International Conference on Robotics and Automation (ICRA), 2022.
- [25] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *Computer Vision ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX, 2022.* 2, 4
- [26] Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and Jose M. Alvarez. Is ego status all you need for openloop end-to-end autonomous driving? In *Proceedings of*

- the Conference on Computer Vision and Pattern Recognition (CVPR), 2024. 5
- [27] Bencheng Liao, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Chang Huang. Maptr: Structured modeling and learning for online vectorized hd map construction. In *International Conference on Learning Representations*, 2023. 2
- [28] Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, and Xinggang Wang. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving, 2024.
- [29] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. 7
- [30] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *Computer Vision ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII, 2022. 2*
- [31] Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: end-to-end vectorized hd map learning. In *Proceedings of the 40th International Conference on Machine Learning*, 2023. 2
- [32] Jiquan Ngiam, Benjamin Caine, Vijay Vasudevan, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, David Weiss, Ben Sapp, Zhifeng Chen, and Jonathon Shlens. Scene transformer: A unified architecture for predicting multiple agent trajectories. In *International Conference on Learning Representations (ICLR)*, 2022. 2
- [33] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings* of the 38th International Conference on Machine Learning, 2021. 3
- [34] William Peebles and Saining Xie. Scalable diffusion models with transformers. In 2023 IEEE/CVF International Conference on Computer Vision (ICCV), 2023. 4, 5
- [35] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Proceedings of the European Conference on Computer Vision*, 2020. 2
- [36] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 4, 5
- [37] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Lit, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Raphael Gontijo-Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In Proceedings of the 36th International Conference on Neural Information Processing Systems, 2024. 2
- [38] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual ques-

- tion answering. In Proceedings of the European Conference on Computer Vision (ECCV), 2024. 2
- [39] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, 2015. 3
- [40] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021. 6
- [41] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Proceedings* of the 33rd International Conference on Neural Information Processing Systems, 2019. 3
- [42] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 2, 3
- [43] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models, 2023. 7
- [44] Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, XianPeng Lang, and Hang Zhao. DriveVLM: The convergence of autonomous driving and large vision-language models. In 8th Annual Conference on Robot Learning, 2024. 2
- [45] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 2011. 3
- [46] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Proceedings of the 5th Conference on Robot Learning*, 2022. 2
- [47] Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for realtime autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024. 2
- [48] Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. In *NeurIPS*, 2022. 5
- [49] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee Kenneth Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automa*tion Letters, 2024. 2
- [50] Jiang-Tian Zhai, Ze Feng, Jihao Du, Yongqiang Mao, Jiang-Jiang Liu, Zichang Tan, Yifu Zhang, Xiaoqing Ye, and Jing-dong Wang. Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes. arXiv preprint arXiv:2305.10430, 2023. 5
- [51] Diankun Zhang, Guoan Wang, Runwen Zhu, Jianbo Zhao, Xiwu Chen, Siyu Zhang, Jiahao Gong, Qibin Zhou, Wenyuan Zhang, Ningzi Wang, Feiyang Tan, Hangning Zhou, Ziyao Xu, Haotian Yao, Chi Zhang, Xiaojun Liu, Xiaoguang Di, and Bin Li. Sparsead: Sparse query-centric

- paradigm for efficient end-to-end autonomous driving, 2024.
- [52] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 2
- [53] Jiayu Zou, Kun Tian, Zhu Zheng, Yun Ye, and Xingang Wang. Diffbev: Conditional diffusion model for bird's eye view perception. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. 2

DiffAD: A Unified Diffusion Modeling Approach for Autonomous Driving

Supplementary Material

7. Additional Experiments

Scaling up parameters. We investigate how increasing model parameters enhances capacity by training three DiffAD models with different DiT configurations (B, L, XL), as detailed in Tab. 7. Throughout training of temporal stage, we track FID-2K scores, as illustrated in Fig. 5, which show a consistent improvement in FID as the transformer depth and width increase. Since configurations L and XL achieve similar FID scores, we adopt DiT-L as the default setting for experiments.

Model	Layers N	${\it Hidden \ size} \ d$	Heads	Params(M)
DiT-B	12	768	12	131.8
DiT-L	24	1024	16	460.3
DiT-XL	28	1152	16	677.9

Table 7. Details of DiT models.

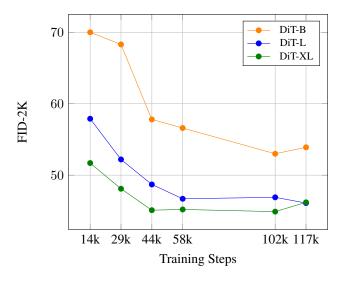


Figure 5. Comparison of different DiT configs.

Visualization of different denoising steps As demonstrated in earlier experiments, increasing the number of denoising steps improves both image quality and planning performance. However, excessively high NFE introduces additional computational costs without yielding significant planning benefits. To further investigate the impact of NFE, we visualize the generated BEV images at different denoising steps. As shown in Fig. 6, increasing NFE enhances the

clarity of traffic elements. When NFE=10, the system successfully captures sufficient environmental details, ensuring reliable planning performance. Further denoising primarily refines minor details of traffic elements without substantially affecting planning outcomes. This experiment highlights that placing excessive emphasis on perception tasks may not necessarily lead to meaningful improvements in planning performance, suggesting the need for a balanced allocation of computational resources between perception and planning.

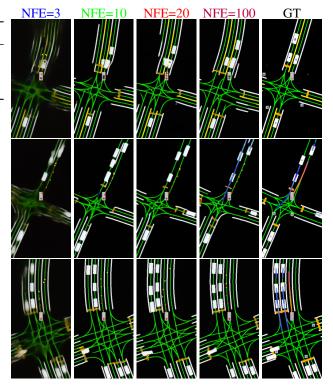


Figure 6. Effect of different NFE values on image quality

Open-loop metric vs Closed-loop metric As reported in Bench2Drive, open-loop evaluation ignores critical factors such as distribution shift and causal confusion, thus it fails to assess a model's ability to handle dynamic interactions. As shown in Tab. 8, DiffAD achieves the best closed-loop performance in terms of Driving Score and Success Rate, despite having a higher L2 error than UniAD and DriveAdapter. This highlights the importance of closed-loop evaluation in capturing the true effectiveness of autonomous driving models.

Table 8. Comparison of Open-loop and Closed-loop metrics on $\mbox{Bench2Drive}.$

Method	Avg. L2↓	Driving Score ↑	Success Rate ↑
UniAD-Base	0.73	45.81	16.36
TCP-traj*	1.70	59.90	30.00
DriveAdapter*	1.01	64.22	33.08
DiffAD	1.55	67.92	38.64