

DiffusionDrive: Truncated Diffusion Model for End-to-End Autonomous Driving

Bencheng Liao^{1,2,✉} Shaoyu Chen^{2,3} Haoran Yin³ Bo Jiang^{2,✉} Cheng Wang^{1,2,✉} Sixu Yan²
 Xinbang Zhang³ Xiangyu Li³ Ying Zhang³ Qian Zhang³ Xinggang Wang^{2,✉}

¹ Institute of Artificial Intelligence, Huazhong University of Science & Technology

² School of EIC, Huazhong University of Science & Technology

³ Horizon Robotics

Code & Model & Demo: hustvl/DiffusionDrive

Abstract

Recently, the diffusion model has emerged as a powerful generative technique for robotic policy learning, capable of modeling multi-mode action distributions. Leveraging its capability for end-to-end autonomous driving is a promising direction. However, the numerous denoising steps in the robotic diffusion policy and the more dynamic, open-world nature of traffic scenes pose substantial challenges for generating diverse driving actions at a real-time speed. To address these challenges, we propose a novel truncated diffusion policy that incorporates prior multi-mode anchors and truncates the diffusion schedule, enabling the model to learn denoising from anchored Gaussian distribution to the multi-mode driving action distribution. Additionally, we design an efficient cascade diffusion decoder for enhanced interaction with conditional scene context. The proposed model, DiffusionDrive, demonstrates 10× reduction in denoising steps compared to vanilla diffusion policy, delivering superior diversity and quality in just 2 steps. On the planning-oriented NAVSIM dataset, with aligned ResNet-34 backbone, DiffusionDrive achieves 88.1 PDMS without bells and whistles, setting a new record, while running at a real-time speed of 45 FPS on an NVIDIA 4090. Qualitative results on challenging scenarios further confirm that DiffusionDrive can robustly generate diverse plausible driving actions.

1. Introduction

End-to-end autonomous driving has gained significant attention in recent years due to advancements in perception models (detection [4, 17, 24, 42], tracking [54–56], online mapping [27, 28, 32], etc.), which directly learns the driving policy from the raw sensor inputs. This data-driven ap-

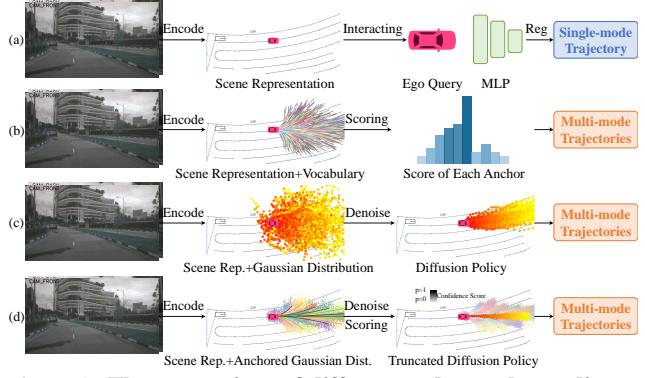


Figure 1. **The comparison of different end-to-end paradigms.**
 (a) Single mode regression [7, 16, 20]. (b) Sampling from vocabulary [3, 25]. (c) Vanilla diffusion policy [6, 19]. (d) The proposed truncated diffusion policy.

proach offers a scalable and robust alternative to traditional rule-based motion planning, which often struggles to generalize to complex real-world driving settings.

To effectively learn from data, mainstream end-to-end planners (e.g., Transfuser [7], UniAD [16], VAD [20]) typically regress a single-mode trajectory from an ego-query as shown in Fig. 1a. However, this paradigm does not account for the inherent uncertainty and multi-mode nature of driving behaviors. Recently, VADv2 [20] introduces a large fixed vocabulary of anchor trajectories (4096 anchors) to discretize the continuous action space and capture a broader range of driving behaviors, and then samples from these anchors based on predicted scores as shown in Fig. 1b. However, this large fixed-vocabulary paradigm is fundamentally constrained by the number and quality of anchor trajectories, often failing in out-of-vocabulary scenarios. Furthermore, managing a large number of anchors presents significant computational challenges for real-time applications. Rather than discretizing the action space, diffusion model [6] has proven to be a powerful genera-

✉ Intern of Horizon Robotics; ✉ Corresponding author: Xinggang Wang (xgwang@hust.edu.cn).

To distinguish the term “multimodal” used to describe input data, we use “multi-mode” in this paper to refer to diverse planning decisions.

tive decision-making policy in the robotics domain, which can directly sample multi-mode physically plausible actions from a Gaussian distribution via iterative denoising process.

This inspires us to replicate the success of the diffusion model in the robotics domain to end-to-end autonomous driving. We apply the vanilla robotic diffusion policy to the well-known single-mode-regression method, Transfuser [7], by proposing a variant, Transfuser_{DP}, which replaces the deterministic MLP regression head with a conditional diffusion model [34]. Though Transfuser_{DP} improves planning performance, two major issues arise: *1) The numerous 20 denoising steps in the vanilla DDIM diffusion policy introduce heavy computational consumption during inference as shown in Tab. 2, hindering the real-time application for autonomous driving.* *2) The trajectories sampled from different Gaussian noises severely overlap with each other, as illustrated in Fig. 2.* This underscores the non-trivial challenge of taming the diffusion models for the dynamic and open-world traffic scenes.

Unlike the vanilla diffusion policy, which samples actions from a random Gaussian noise conditioned on scene context, human drivers adhere to established driving patterns that they dynamically adjust in response to real-time traffic conditions. This insight motivates us to embed these prior driving patterns into the diffusion policy by partitioning the Gaussian distribution into multiple sub-Gaussian distributions centered around prior anchors, referred to as anchored Gaussian distribution. It is implemented by truncating the diffusion schedule to introduce a small portion of Gaussian noise around the prior anchors as shown in Fig. 3. Thanks to the multi-mode distributional expressivity of the diffusion model, the proposed truncated diffusion policy effectively covers the potential action space without requiring a large set of fixed anchors, as VADv2 does. With more reasonable initial noise samples from the anchored Gaussian distribution, we can truncate the denoising process, reducing the required steps from 20 to just 2—a substantial speedup that satisfies the real-time requirements of autonomous driving.

To enhance the interaction with conditional scene context, we propose an efficient transformer-based diffusion decoder that interacts not only with structured queries from the perception module but also with Bird’s Eye View (BEV) and perspective view (PV) features through a sparse deformable attention mechanism [62]. Additionally, we introduce a cascade mechanism to iteratively refine the trajectory reconstruction within the diffusion decoder at each denoising step.

With these innovations, we present **DiffusionDrive**, a diffusion model for real-time end-to-end autonomous driving. We benchmark our method on the planning-oriented NAVSIM dataset [10] using non-reactive simulation and closed-loop evaluations. Without bells and whis-

gles, DiffusionDrive achieves 88.1 PDMS on NAVSIM navtest split with the aligned ResNet-34 backbone, significantly outperforming previous state-of-the-art methods. Even compared to the NAVSIM challenge-winning solution Hydra-MDP- \mathcal{V}_{8192} -W-EP [25], which follows VADv2 with 8192 anchor trajectories and further incorporates post-processing and additional supervision, DiffusionDrive still outperforms it by 1.6 PDMS through directly learning from human demonstrations and inferring without post-processing, while running at real-time speed of 45 FPS on an NVIDIA 4090. We further validate the superiority of DiffusionDrive on popular nuScenes dataset [2] with open-loop evaluations, DiffusionDrive runs 1.8× faster than VAD and outperforms it [20] by 20.8% lower L2 error and 63.6% lower collision rate with the same ResNet-50 backbone, demonstrating state-of-the-art planning performance.

Our contributions can be summarized as follows:

- We firstly introduce the diffusion model to the field of end-to-end autonomous driving and propose a novel truncated diffusion policy to address the issues of mode collapse and heavy computational overhead found in direct adaptation of vanilla diffusion policy to the traffic scene.
- We design an efficient transformer-based diffusion decoder that interacts with the conditional information in a cascaded manner for better trajectory reconstruction.
- Without bells and whistles, DiffusionDrive significantly outperforms previous state-of-the-art methods, achieving a record-breaking 88.1 PDMS on the NAVSIM navtest split with the same backbone, while maintaining real-time performance at 45 FPS on an NVIDIA 4090.
- We qualitatively demonstrate that DiffusionDrive can generate more diverse and plausible trajectories, exhibiting high-quality multi-mode driving actions in various challenging scenarios.

2. Related Work

End-to-end autonomous driving. UniAD [16], as a pioneering work, demonstrates the potential of end-to-end autonomous driving by integrating multiple perception tasks to enhance planning performance. VAD [20] further explores the use of compact vectorized scene representations to improve efficiency. Subsequently, a series of works [5, 7, 12, 23, 26, 43, 45, 58] have adopted the single-trajectory planning paradigm to enhance planning performance further. More recently, VADv2 [3] shifts the paradigm towards multi-mode planning by scoring and sampling from a large fixed vocabulary of anchor trajectories. Hydra-MDP [25] improves the scoring mechanism of VADv2 by introducing extra supervision from a rule-based scorer. SparseDrive [39] explores an alternative BEV-free solution. Unlike existing multi-mode planning approaches, we propose a novel paradigm that leverages powerful generative

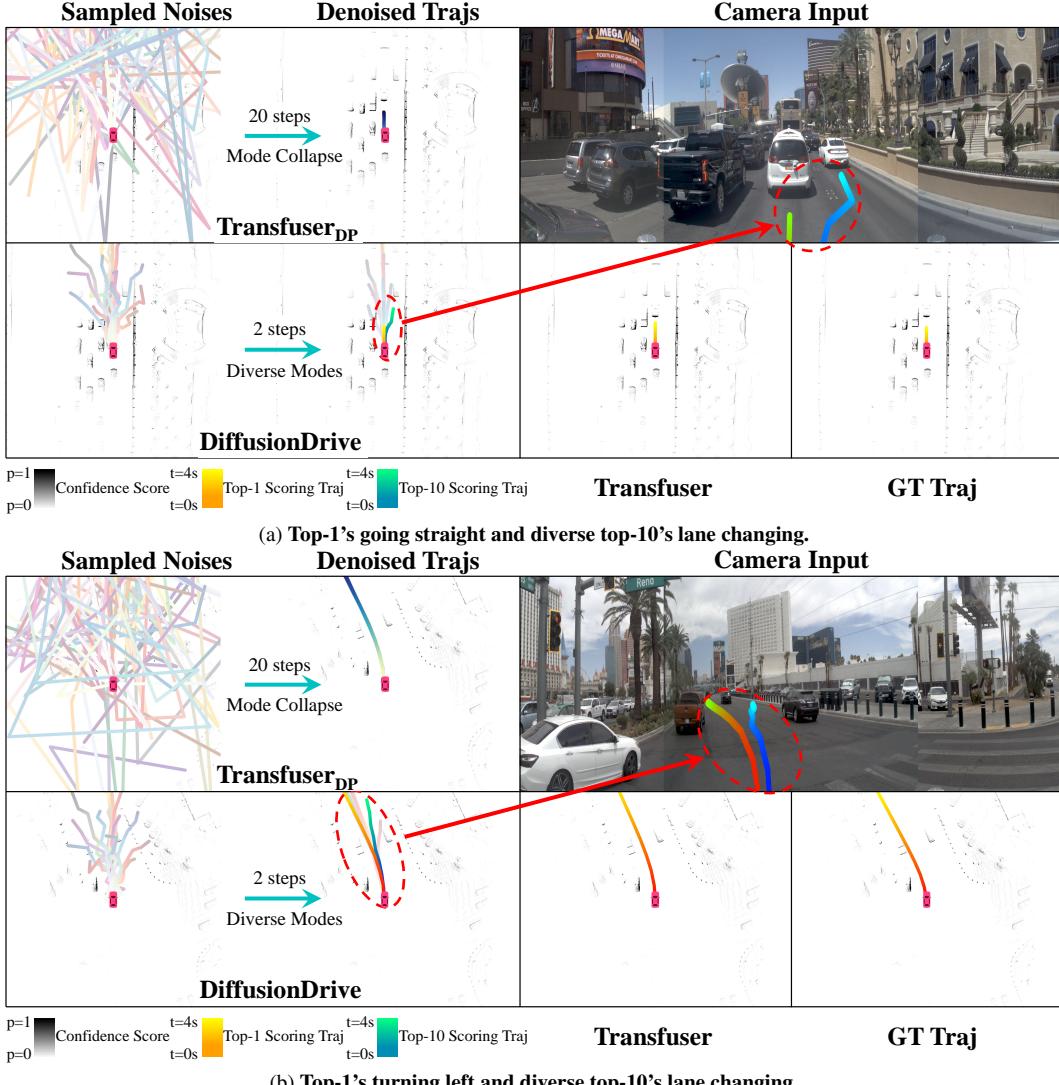


Figure 2. Qualitative comparison of Transfuser, Transfuser_{DP} and DiffusionDrive on challenging scenes of NAVSIM `navtest` split. With the same inputs from front cameras and LiDAR, DiffusionDrive achieves the highest planning quality of top-1 scoring trajectory as illustrated in Tab. 2. We render the highlighted diverse trajectories predicted by DiffusionDrive in the front view. (a) and (b) shows that the top-1 scoring trajectory of DiffusionDrive closely matches the ground truth for both going straight and turning left. Additionally, DiffusionDrive’s top-10 scoring trajectory demonstrates high-quality lane changing—an ability not observed in multi-mode Transfuser_{DP} and impossible for Transfuser.

diffusion models for end-to-end autonomous driving.

Diffusion model for traffic simulation. Driving diffusion policy has been explored in the traffic simulation by leveraging only abstract perception groundtruth [8, 18, 21, 44]. MotionDiffuser [21] and CTG [60] are pioneering applications of diffusion models for multi-agent motion prediction, using a conditional diffusion model to sample target trajectories from Gaussian noise. CTG++ [59] further incorporates a large language model (LLM) for language-driven guidance, improving usability and enabling realistic traffic simulations. Diffusion-ES [48] replaces reward-gradient-guided denoising with evolutionary search. Moving beyond diffusion models limited to traffic simulation with percep-

tion groundtruth, our approach unlocks the potential of diffusion models for real-time, end-to-end autonomous driving through our proposed truncated diffusion policy and efficient diffusion decoder.

Diffusion model for robotic policy learning. Diffusion policy [6] demonstrates the great potential in robotic policy learning, effectively capturing multi-mode action distributions and high-dimensional action spaces. Diffuser [19] proposes an unconditional diffusion model for trajectory sampling, incorporating techniques such as classifier-free guidance and image inpainting to achieve guided sampling. Subsequently, numerous works have applied diffusion models to various robotic tasks, including stationary manipula-

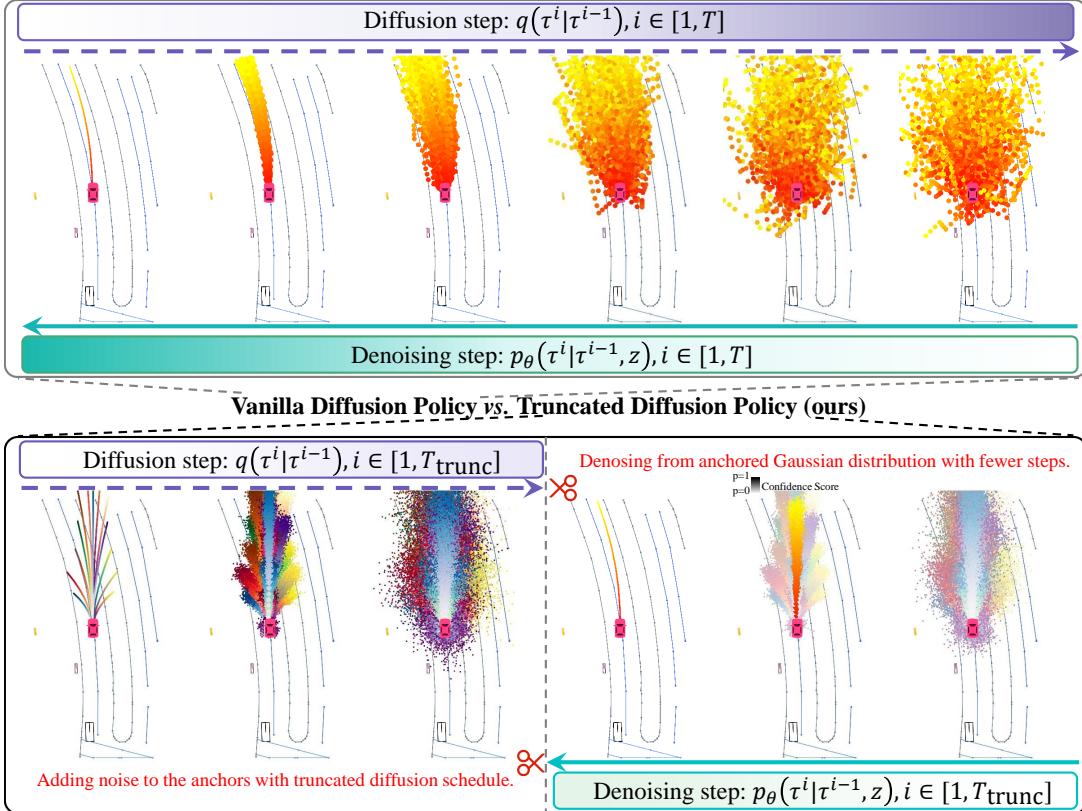


Figure 3. **Illustration of truncated diffusion policy by comparing with vanilla diffusion policy.** We truncate the diffusion process and only add a small portion of Gaussian noise to diffuse the anchor trajectories. Then, we train the diffusion model to reconstruct the ground-truth trajectory from the anchored Gaussian distribution with conditional scene context. During the inference, we also truncate the denoising process by starting from the better samples in the anchored Gaussian distribution than the pure Gaussian noise.

tion [1, 53], mobile manipulation [47], autonomous navigation [37, 51], quadruped locomotion [38], and dexterous manipulation [46]. However, directly applying vanilla diffusion policy to end-to-end autonomous driving poses unique challenges, as it requires real-time efficiency and the generation of plausible multi-mode trajectories in dynamic and open-world traffic scenes. In this work, we propose a novel truncated diffusion policy to address these challenges, introducing concepts that have not yet been explored in the robotics field.

Diffusion model for image generation. Diffusion models have been extensively adopted for image generation tasks [33, 36, 49, 50, 61]. DDIM [35] enhances DDPM [14] by enabling efficient sampling with significantly fewer steps based on non-Markovian diffusion processes. Flow matching [30, 31] further optimizes the generative process by directly modeling continuous probability flows. TDPM [57] proposes truncated denoising, which initiates the generation process from an implicit intermediate distribution to accelerate sampling. In contrast to these approaches, our method introduces an explicit driving prior within the diffusion policy, effectively guiding the diffusion process toward more accurate and efficient generation tailored specifically

for end-to-end autonomous driving.

3. Method

3.1. Preliminary

Task formulation. End-to-end autonomous driving takes raw sensor data as input and predicts the future trajectory of the ego-vehicle. The trajectory is represented as a sequence of waypoints $\tau = \{(x_t, y_t)\}_{t=1}^{T_f}$, where T_f denotes the planning horizon, and (x_t, y_t) is the location of each waypoint at time t in the current ego-vehicle coordinate system.

Conditional diffusion model. The conditional diffusion model poses a forward diffusion process as gradually adding noise to the data sample, which can be defined as:

$$q(\tau^i | \tau^0) = \mathcal{N}\left(\tau^i; \sqrt{\bar{\alpha}^i} \tau^0, (1 - \bar{\alpha}^i) \mathbf{I}\right), \quad (1)$$

where τ^0 is the clean data sample, and τ^i is the data sample with noise at time i (Note: we use superscript i to denote diffusion timestep). The constant $\bar{\alpha}^i = \prod_{s=1}^i \alpha^s = \prod_{s=1}^i (1 - \beta^s)$ and β^s is the noise schedule. We train the reverse process model $f_\theta(\tau^i, z, i)$ to predict τ^0 from τ^i with the guidance of conditional information z , where θ is the

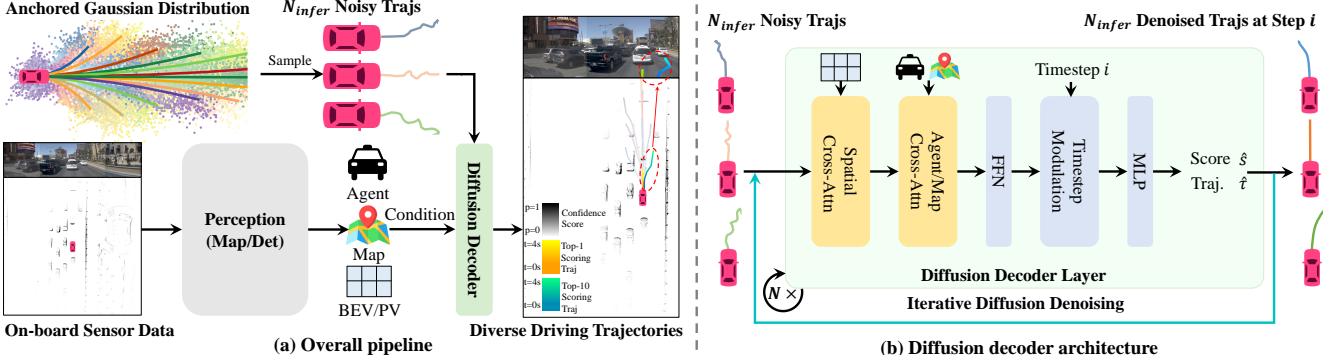


Figure 4. **Overall architecture of DiffusionDrive.** (a) DiffusionDrive can integrate various existing perception modules and sensor inputs. (b) The designed diffusion decoder takes the sampled noisy trajectories from anchored Gaussian distribution as input and progressively denoises them with enhanced interactions with the conditional scene context in a cascade manner to generate the final predictions.

trainable model parameter. During inference, the trained diffusion model f_θ progressively refines from the random noise τ^T sampled in Gaussian distribution to the predicted clean data sample τ^0 with the guidance of conditional information z , which is defined as:

$$p_\theta(\tau^0 | z) = \int p(\tau^T) \prod_{i=1}^T p_\theta(\tau^{i-1} | \tau^i, z) d\tau^{1:T}. \quad (2)$$

3.2. Investigation

Turn Transfuser [7] into conditional diffusion model. We begin from the representative deterministic end-to-end planner Transfuser [7] and turn it into a generative model Transfuser_{DP} by simply replacing the regression MLP layers with the conditional diffusion model UNet following vanilla diffusion policy [6]. During the evaluation, we sample a random noise and progressively refine it with 20 steps. Tab. 2 shows that Transfuser_{DP} achieves better planning quality than deterministic Transfuser.

Mode collapse. To further investigate the multi-mode property of the vanilla diffusion policy in driving, we sampled 20 random noises from Gaussian distribution and denoised them using 20 steps. As shown in Fig. 2, the different random noises converge to similar trajectories after the denoising process. To quantitatively analyze the phenomenon of mode collapse, we define a mode diversity score \mathcal{D} based on the mean Intersection over Union (mIoU) between each denoised trajectory and the union of all denoised trajectories:

$$\mathcal{D} = 1 - \frac{1}{N} \sum_{i=1}^N \frac{\text{Area}(\tau_i \cap \bigcup_{j=1}^N \tau_j)}{\text{Area}(\tau_i \cup \bigcup_{j=1}^N \tau_j)}, \quad (3)$$

where τ_i represents the i -th denoised trajectory, N is the total number of sampled trajectories and $\bigcup_{j=1}^N \tau_j$ is the union of all denoised trajectories. A higher mIoU indicates less diversity of the denoised trajectories. The quantitative mode diversity results in Tab. 2 further validate the observations presented in Fig. 2.

Heavy denoising overhead. The DDIM [35] diffusion policy requires 20 denoising steps to transform random noise into a feasible trajectory, which introduces significant computational overhead, reducing the FPS from 60 to 7, as shown in Tab. 2, and making it impractical for real-time online driving applications.

3.3. Truncated Diffusion

Human driving follows fixed patterns, unlike the random noise denoising in vanilla diffusion policy. Motivated by this, we propose a truncated diffusion policy that begins the denoising process from an anchored Gaussian distribution instead of a standard Gaussian distribution. To enable the model to learn to denoise from the anchored Gaussian distribution to the desired driving policy, we further truncate the diffusion schedule during training, adding only a small amount of Gaussian noise to the anchors.

Training. We first construct the diffusion process by adding Gaussian noise to anchors $\{\mathbf{a}_k\}_{k=1}^{N_{\text{anchor}}}$ clustered by K-Means on the training set, where $\mathbf{a}_k = \{(x_t, y_t)\}_{t=1}^{T_f}$. We truncate the diffusion noise schedule to diffuse the anchors to the anchored Gaussian distribution:

$$\tau_k^i = \sqrt{\bar{\alpha}^i} \mathbf{a}_k + \sqrt{1 - \bar{\alpha}^i} \epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}), \quad (4)$$

where $i \in [1, T_{\text{trunc}}]$ and $T_{\text{trunc}} \ll T$ is the truncated diffusion steps.

During training, the diffusion decoder f_θ takes as input N_{anchor} noisy trajectories $\{\tau_k^i\}_{k=1}^{N_{\text{anchor}}}$ and predicts classification scores $\{\hat{s}_k\}_{k=1}^{N_{\text{anchor}}}$ and denoised trajectories $\{\hat{\tau}_k\}_{k=1}^{N_{\text{anchor}}}$:

$$\{\hat{s}_k, \hat{\tau}_k\}_{k=1}^{N_{\text{anchor}}} = f_\theta(\{\tau_k^i\}_{k=1}^{N_{\text{anchor}}}, z), \quad (5)$$

where z represents the conditional information. We assign the noisy trajectory around the closest anchor to the ground truth trajectory τ_{gt} as positive sample ($y_k = 1$) and others as negative samples ($y_k = 0$). The training objective combines

| Method | Input | Img. | Backbone | Anchor | NC \uparrow | DAC \uparrow | TTC \uparrow | Comf. \uparrow | EP \uparrow | PDMS \uparrow |
|--|--------|----------------|----------|-------------|---------------|----------------|----------------|------------------|---------------|-----------------|
| UniAD [16] | Camera | ResNet-34 [13] | 0 | 97.8 | 91.9 | 92.9 | 100 | 78.8 | | 83.4 |
| PARA-Drive [45] | Camera | ResNet-34 [13] | 0 | 97.9 | 92.4 | 93.0 | 99.8 | 79.3 | | 84.0 |
| LTF [7] | Camera | ResNet-34 [13] | 0 | 97.4 | 92.8 | 92.4 | 100 | 79.0 | | 83.8 |
| Transfuser [7] | C & L | ResNet-34 [13] | 0 | 97.7 | 92.8 | 92.8 | 100 | 79.2 | | 84.0 |
| DRAM [52] | C & L | ResNet-34 [13] | 0 | 98.0 | 93.1 | 94.8 | 100 | <u>80.1</u> | | 85.5 |
| VAdv2- \mathcal{V}_{8192} [3] | C & L | ResNet-34 [13] | 8192 | 97.2 | 89.1 | 91.6 | 100 | 76.0 | | 80.9 |
| Hydra-MDP- \mathcal{V}_{8192} [25] | C & L | ResNet-34 [13] | 8192 | 97.9 | 91.7 | 92.9 | 100 | 77.6 | | 83.0 |
| Hydra-MDP- \mathcal{V}_{8192} -W-EP [25] | C & L | ResNet-34 [13] | 8192 | 98.3 | <u>96.0</u> | 94.6 | 100 | 78.7 | | 86.5 |
| DiffusionDrive (Ours) | C & L | ResNet-34 [13] | 20 | <u>98.2</u> | 96.2 | <u>94.7</u> | 100 | 82.2 | | 88.1 |

Table 1. **Comparison on planning-oriented NAVSIM navtest split with closed-loop metrics.** “C & L” denotes the use of both camera and LiDAR as sensor inputs. “ \mathcal{V}_{8192} ” denotes 8192 anchors. “Hydra-MDP- \mathcal{V}_{8192} -W-EP” is a variant of Hydra-MDP [25], which is further trained to fit the EP evaluation metric with additional supervision from the rule-based evaluator and uses weighted confidence post-processing. DiffusionDrive simply learns from human demonstrations and infers without post-processing. The **best** and the second best results are denoted by **bold** and underline.

| Method | Plan Module Time | | | | | Arch. | Step Time \downarrow | Steps \downarrow | Total \downarrow | $\mathcal{D} \uparrow$ | Para. \downarrow | FPS \uparrow | |
|--------------------------|------------------|----------------|----------------|------------------|---------------|------------------|------------------------|--------------------|--------------------|------------------------|--------------------|----------------|-----------|
| | NC \uparrow | DAC \uparrow | TTC \uparrow | Comf. \uparrow | EP \uparrow | | | | | | | | |
| Transfuser | 97.7 | 92.8 | 92.8 | 100 | 79.2 | 84.0 | MLP | 0.2ms | 1 | 0.2ms | 0% | 56M | 60 |
| Transfuser _{DP} | 97.5 | 93.7 | 92.7 | 100 | 79.4 | 84.6 <u>+0.6</u> | UNet | 6.5ms | 20 | 130.0ms | 11% | 101M | 7 |
| Transfuser _{TD} | <u>97.9</u> | <u>94.2</u> | <u>93.9</u> | 100 | 80.2 | <u>85.7+1.7</u> | UNet | 6.9ms | 2 | 13.8ms | <u>70%</u> | 102M | 27 |
| DiffusionDrive | 98.2 | 96.2 | 94.7 | 100 | 82.2 | 88.1+4.1 | Dec. | <u>3.8ms</u> | <u>2</u> | <u>7.6ms</u> | 74% | 60M | 45 |

Table 2. **Roadmap from Transfuser to DiffusionDrive on NAVSIM navtest split.** “Transfuser_{DP}” denotes Transfuser with vanilla DDIM diffusion policy [6]. “Transfuser_{TD}” denotes Transfuser with truncated diffusion policy. “Step Time” denotes the runtime of each denoising step. “FPS” and runtime are measured on an NVIDIA 4090 GPU. “ \mathcal{D} ” denotes the mode diversity score defined in Eq. (3).

trajectory reconstruction and classification:

$$\mathcal{L} = \sum_{k=1}^{N_{\text{anchor}}} [y_k \mathcal{L}_{\text{rec}}(\hat{\tau}_k, \tau_{\text{gt}}) + \lambda \text{BCE}(\hat{s}_k, y_k)], \quad (6)$$

where λ balances the simple L1 reconstruction loss \mathcal{L}_{rec} and binary cross-entropy (BCE) classification loss.

Inference. We use a truncated denoising process that starts with noisy trajectories sampled from the anchored Gaussian distribution and progressively denoises them to final predictions. At each denoising timestep, the estimated trajectories from the previous step are passed to the diffusion decoder f_θ , which predicts classification scores $\{\hat{s}_k\}_{k=1}^{N_{\text{infer}}}$ and coordinates $\{\hat{\tau}_k\}_{k=1}^{N_{\text{infer}}}$. After obtaining the current timestep’s predictions, we apply the DDIM [35] update rule to sample trajectories for the next timestep.

Inference flexibility. A key advantage of our approach lies in its inference flexibility. While the model is trained with N_{anchor} trajectories, the inference process can accommodate an arbitrary number of trajectory samples N_{infer} , where N_{infer} can be dynamically adjusted based on computational resources or application requirements.

3.4. Architecture

The overall architecture of our proposed method, DiffusionDrive, is illustrated in Fig. 4. DiffusionDrive can integrate

various existing perception modules used in previous end-to-end planners [7, 16, 20, 39] and take different sensor inputs. The designed diffusion decoder is tailored for the complex and challenging driving application, which has enhanced interactions with the conditional scene context.

Diffusion decoder. Given the set of sampled noisy trajectories $\{\hat{\tau}_k\}_{k=1}^{N_{\text{infer}}}$ from the anchored Gaussian distribution, we begin by applying deformable spatial cross-attention [29, 42, 62] to interact with Bird’s Eye View (BEV) or Perspective View (PV) features based on the trajectory coordinates. Subsequently, cross-attention is performed between the trajectory features and the agent/map queries derived from the perception module, followed by a feed-forward network (FFN). To encode the diffusion timestep information, we utilize a Timestep Modulation layer, which is followed by a Multi-Layer Perceptron (MLP) that predicts the confidence score and the offset relative to the initial noisy trajectory coordinates. The output from this diffusion decoder layer serves as the input for the subsequent cascade diffusion decoder layer. DiffusionDrive further reuses the cascade diffusion decoder to iteratively denoise the trajectory during inference, with parameters shared across the different denoising timesteps. The final trajectory with the highest confidence score is selected as the output.

| ID | UNet | Ego Query | Spatial | Agent/Map | Cascade | Param. \downarrow | Planning Metric | | | | | |
|----|---------|-------------|------------|------------|---------|---------------------|-----------------|----------------|----------------|------------------|---------------|-----------------|
| | Decoder | Interaction | Cross-attn | Cross-attn | Decoder | | NC \uparrow | DAC \uparrow | TTC \uparrow | Comf. \uparrow | EP \uparrow | PDMS \uparrow |
| 1 | ✓ | ✓ | ✗ | ✗ | ✗ | 102M | 97.9 | 94.2 | 93.9 | 100 | 80.2 | 85.7 |
| 2 | ✗ | ✓ | ✗ | ✗ | ✗ | 57M | 88.7 | 83.2 | 80.0 | 84.8 | 43.3 | 55.1 |
| 3 | ✗ | ✓ | ✓ | ✗ | ✗ | 58M | 98.2 | 95.4 | 94.4 | 100 | 81.3 | 87.1 |
| 4 | ✗ | ✓ | ✗ | ✓ | ✗ | 58M | 97.9 | 93.5 | 93.8 | 100 | 79.8 | 85.1 |
| 5 | ✗ | ✓ | ✓ | ✓ | ✗ | 59M | 98.0 | 95.8 | 94.4 | 100 | 81.7 | 87.4 |
| 6 | ✗ | ✓ | ✓ | ✓ | ✓ | 60M | 98.2 | 96.2 | 94.7 | 100 | 82.2 | 88.1 |

Table 3. **Ablation for design choices.** “Cascade Decoder” indicates that we stack 2 cascade diffusion decoder layers. ID-1 refers to Transfuser_{TD} in Tab. 2, utilizing conditional UNet and interaction with the ego-query, which Transfuser uses to directly regress the single-mode trajectory.

| Steps | Param. | NC | DAC | TTC | Comf. | EP | PDMS |
|-------|--------|------|------|------|-------|------|------|
| 1 | 60M | 98.3 | 96.0 | 94.7 | 100 | 82.1 | 87.9 |
| 2 | 60M | 98.2 | 96.2 | 94.7 | 100 | 82.2 | 88.1 |
| 3 | 60M | 98.2 | 96.3 | 94.7 | 100 | 92.2 | 88.1 |

Table 4. Denoising step number.

| Stages | Param. | NC | DAC | TTC | Comf. | EP | PDMS |
|--------|--------|------|------|------|-------|------|------|
| 1 | 59M | 98.0 | 95.8 | 94.4 | 100 | 81.7 | 87.4 |
| 2 | 60M | 98.2 | 96.2 | 94.7 | 100 | 82.2 | 88.1 |
| 4 | 65M | 98.4 | 96.2 | 94.9 | 100 | 82.4 | 88.2 |

Table 5. Cascade stages.

| N_{infer} | Param. | NC | DAC | TTC | Comf. | EP | PDMS |
|--------------------|--------|------|------|------|-------|------|------|
| 10 | 60M | 97.9 | 93.5 | 93.1 | 100 | 80.0 | 84.9 |
| 20 | 60M | 98.2 | 96.2 | 94.7 | 100 | 82.2 | 88.1 |
| 40 | 60M | 98.5 | 96.2 | 94.8 | 100 | 82.5 | 88.2 |

Table 6. Number of sampled noises N_{infer} .

4. Experiment

4.1. Dataset

NAVSIM. The NAVSIM dataset [10] is a real-world planning-oriented dataset builds upon OpenScene [9], a compact redistribution of nuPlan [22], the largest publicly available annotated driving dataset. NAVSIM leverages eight cameras to achieve a full 360° FOV, along with a merged LiDAR point cloud derived from five sensors. Annotations are provided at a frequency of 2Hz and include both HD maps and object bounding boxes. The dataset is designed to emphasize challenging driving scenarios involving dynamic changes in driving intentions, while deliberately excluding trivial situations such as stationary scenes or constant-speed driving.

NAVSIM benchmarks planning performance using non-reactive simulations and closed-loop metrics for comprehensive evaluation. In this paper, we employ the proposed PDM score (PDMS) [10], which is a weighted combination of several sub-scores: no at-fault collisions (NC), drivable area compliance (DAC), time-to-collision (TTC), comfort (Comf.), and ego progress (EP).

4.2. Implementation Detail

We adopt the same perception modules and ResNet-34 backbone [13] as Transfuser for fair comparison. In the diffusion decoder layer, we employ spatial cross-attention to only interact with BEV features following Transfuser’s BEV-based setting. We only perform agent cross-attention, since the perception module of Transfuser does not include vectorized map construction. We stack 2 cascade diffusion decoder layers and apply truncated diffusion policy with 20 clustered anchors. The training diffusion schedule is truncated by 50/1000 to diffuse the anchors, while during inference, we use only 2 denoising steps and select the top-

1 scoring predicted trajectory for evaluation. The training and inference recipe directly follows Transfuser: We use three cropped and downscaled forward-facing camera images, concatenated as a 1024×256 image, and a rasterized BEV LiDAR as input; DiffusionDrive is trained on navtrain split from scratch for 100 epochs with AdamW optimizer on 8 NVIDIA 4090 GPUs with total batch size of 512, setting the learning rate to 6×10^{-4} . No test-time augmentation is applied and the final output for evaluation on navtest split is 8-waypoint trajectory over 4 seconds.

4.3. Quantitative Comparison

Tab. 1 compares DiffusionDrive with state-of-the-art methods on NAVSIM navtest split. With the same ResNet-34 backbone, DiffusionDrive achieves 88.1 PDMS score, demonstrating significant superior performance over the previous learning-based methods. Compared to VADv2, DiffusionDrive surpasses it by 7.2 PDMS while reducing the number of anchors from 8192 to 20, representing a 400× reduction. DiffusionDrive also outperforms Hydra-MDP, which follows VADv2’s sampling-from-vocabulary paradigm, with a 5.1 PDMS improvement. Even compared to the Hydra-MDP- \mathcal{V}_{8192} -W-EP, which is a variant of Hydra-MDP [25] by further training to fit the EP evaluation metric with additional supervision and using weighted confidence post-processing, DiffusionDrive still outperforms it by 3.5 EP and 1.6 overall PDMS, relying solely on a straightforward learning-from-human approach without any post-processing. Compared to the Transfuser baseline, where we only differ in the planning module, DiffusionDrive delivers a notable 4.1 PDMS improvement, outperforming it across all sub-scores.

| Method | Input | Img. Backbone | L2 (m) ↓ | | | | Collision Rate (%) ↓ | | | | FPS ↑ |
|-----------------------|--------|-----------------|-------------|-------------|-------------|-------------|----------------------|-------------|-------------|-------------|------------|
| | | | 1s | 2s | 3s | Avg. | 1s | 2s | 3s | Avg. | |
| ST-P3 [15] | Camera | EffNet-b4 [40] | 1.33 | 2.11 | 2.90 | 2.11 | 0.23 | 0.62 | 1.27 | 0.71 | 1.6 |
| UniAD [16] | Camera | ResNet-101 [13] | 0.45 | 0.70 | 1.04 | 0.73 | 0.62 | 0.58 | 0.63 | 0.61 | 1.8 |
| OccNet [41] | Camera | ResNet-50 [13] | 1.29 | 2.13 | 2.99 | 2.14 | 0.21 | 0.59 | 1.37 | 0.72 | 2.6 |
| VAD [20] | Camera | ResNet-50 [13] | 0.41 | 0.70 | 1.05 | 0.72 | 0.07 | 0.17 | 0.41 | 0.22 | 4.5 |
| SparseDrive [39] | Camera | ResNet-50 [13] | 0.29 | 0.58 | 0.96 | 0.61 | 0.01 | 0.05 | 0.18 | 0.08 | 9.0 |
| DiffusionDrive (Ours) | Camera | ResNet-50 [13] | 0.27 | 0.54 | 0.90 | 0.57 | 0.03 | 0.05 | 0.16 | 0.08 | 8.2 |

Table 7. Comparison on nuScenes dataset with open-loop metrics. FPS is measured on a single NVIDIA 4090 GPU following the recipe of SparseDrive [39]. Metric calculation follows ST-P3 [15].

4.4. Roadmap

In Tab. 2, converting Transfuser into the generative Transfuser_{DP} using vanilla diffusion policy improves the PDMS score by 0.6 and the mode diversity score \mathcal{D} by 11%. However, it also significantly increases the overhead of the planning module, requiring $20\times$ more denoising steps and $32\times$ the step time, resulting in a total $650\times$ increase in runtime overhead. With the proposed truncated diffusion policy, Transfuser_{TD} reduces the number of denoising steps from 20 to 2 while improving PDMS by 1.1 and mode diversity by 59%. By further incorporating the proposed diffusion decoder, the final model, DiffusionDrive, reaches 88.1 PDMS and 74% mode diversity score \mathcal{D} . Compared to the Transfuser_{DP}, DiffusionDrive shows improvements in 3.5 PDMS and 64% mode diversity, and a $10\times$ reduction in denoising steps, resulting in a $6\times$ speedup in FPS. This enables real-time, high-quality, multi-mode planning.

4.5. Ablation Study

Effect of designs in diffusion decoder. Tab. 3 shows the effectiveness of our design choices in the diffusion decoder. ID-1 is the Transfuser_{TD} in the Tab. 2. By comparing ID-6 and ID-1, we can see that the proposed diffusion decoder reduce the 39% parameters and significantly improves the planning quality by 2.4 PDMS. ID-2 shows severe performance degeneration due to the lack of rich and hierarchical interaction with the environment. By comparing ID-2 and ID-3, we show that spatial cross-attention is vital for accurate planning. ID-5 shows that the proposed cascade mechanism is effective and can further improve the performance.

Denoising step number. Tab. 4 shows that due to a reasonable start point, DiffusionDrive achieves good planning quality even with only 1 step, and further denoising steps offer quality improvement and inference flexibility for complex environments.

Cascade stages. Tab. 5 ablates the impact of cascade stage number. Increasing the stage number can improve the planning quality but saturate at the 4 stages and cost more parameters and inference time at each step.

Number of sampled noises N_{infer} . As stated in Sec. 3.3,

DiffusionDrive can generate varied trajectories by simply sampling a variable number of noises from anchored Gaussian distribution. Tab. 6 shows that 10 sampled noises can already achieve a decent planning quality. By sampling more noises, DiffusionDrive can cover potential planning action space and lead to improved planning quality.

4.6. Qualitative Comparison

Since the PDMS planning metric calculates based on the top-1 scoring trajectory and our proposed \mathcal{D} score evaluates mode diversity, these metrics alone cannot fully capture the quality of diverse trajectories. To further validate the quality of multi-mode trajectories, we visualize the planning results of Transfuser, Transfuser_{DP} and DiffusionDrive on challenging scenarios of NAVSIM navtest split in Fig. 2. The results indicate that the multi-mode trajectories generated by DiffusionDrive are not only diverse but also of high quality. In Fig. 2a, the top-1 scoring trajectory generated by DiffusionDrive closely resembles the ground-truth trajectory, while the highlighted top-10 scoring trajectory surprisingly tries to perform high-quality lane changing. In Fig. 2b, the highlighted top-10 scoring trajectory also performs a lane change, and a neighboring low-scoring trajectory further interacts with surrounding agents to effectively avoid collisions.

4.7. Quantitative Comparison on nuScenes dataset

The nuScenes dataset is previously popular benchmark for end-to-end planning. Since the major scenarios of nuScenes are simple and trivial situations, we only perform comparison in Tab. 7. We implement DiffusionDrive on top of SparseDrive [39] following its training and inference recipe using open-loop metrics proposed in ST-P3 [15]. We stack 2 cascade diffusion decoder layers and apply the truncated diffusion policy with 18 clustered anchors.

As shown in Tab. 7, DiffusionDrive reduces the average L2 error of SparseDrive by 0.04m, achieving the lowest L2 error and average collision rate. While DiffusionDrive is also efficient and runs $1.8\times$ faster than VAD with 20.8% lower L2 error and 63.6% lower collision rate.

5. Conclusion

In this work, we propose a novel generative driving decision-making model, DiffusionDrive, for end-to-end autonomous driving by incorporating the proposed truncated diffusion policy and efficient cascade diffusion decoder. DiffusionDrive can denoise a variable number of samples from an anchored Gaussian distribution to generate diverse planning trajectories at real-time speeds. Comprehensive experiments and qualitative comparisons validate the superiority of DiffusionDrive in planning quality, running efficiency, and mode diversity.

Acknowledgement

We would like to acknowledge Tianheng Cheng for helpful feedback on the draft.

References

- [1] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua B. Tenenbaum, Tommi S. Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision making? In *ICLR*, 2023. 4
- [2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liang, Qiang Xu, Anush Krishnan, Yu Pan, Giacarlo Baldan, and Oscar Beijbom. nuscenes: A multi-modal dataset for autonomous driving. In *CVPR*, 2020. 2, 1
- [3] Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*, 2024. 1, 2, 6
- [4] Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Chang Huang, and Wenyu Liu. Polardetr: Polar parametrization for vision-based surround-view 3d detection. *Image and Vision Computing*, 156:105438, 2025. 1
- [5] Zhili Chen, Maosheng Ye, Shuangjie Xu, Tongyi Cao, and Qifeng Chen. Ppad: Iterative interactions of prediction and planning for end-to-end autonomous driving. In *ECCV*, 2024. 2
- [6] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *RSS*, 2023. 1, 3, 5, 6
- [7] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *TPAMI*, 2022. 1, 2, 5, 6
- [8] Younwoo Choi, Ray Coden Mercurius, Soheil Mohamad Alizadeh Shabestary, and Amir Rasouli. Dice: Diverse diffusion model with scoring for trajectory prediction. In *IV*, 2024. 3
- [9] OpenScene Contributors. Openscene: The largest up-to-date 3d occupancy prediction benchmark in autonomous driving. <https://github.com/OpenDriveLab/OpenScene>, 2023. 7
- [10] Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *NeurIPS*, 2024. 2, 7, 1
- [11] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017. 1
- [12] Xunjiang Gu, Guanyu Song, Igor Gilitschenski, Marco Pavone, and Boris Ivanovic. Producing and leveraging online map uncertainty in trajectory prediction. In *CVPR*, 2024. 2
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6, 7, 8, 1
- [14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 4
- [15] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *ECCV*, 2022. 8
- [16] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhui Wang, et al. Planning-oriented autonomous driving. In *CVPR*, 2023. 1, 2, 6, 8
- [17] Junjie Huang, Guan Huang, Zheng Zhu, Yun Ye, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 1
- [18] Zhiyu Huang, Zixu Zhang, Ameya Vaidya, Yuxiao Chen, Chen Lv, and Jaime Fernández Fisac. Versatile scene-consistent traffic scenario generation as optimization with diffusion. *arXiv preprint arXiv:2404.02524*, 2024. 3
- [19] Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *ICLR*, 2022. 1, 3
- [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. In *ICCV*, 2023. 1, 2, 6, 8
- [21] Chiyu Jiang, Andre Cornman, Cheolho Park, Benjamin Sapp, Yin Zhou, Dragomir Anguelov, et al. Motiondiffuser: Controllable multi-agent motion prediction using diffusion. In *CVPR*, 2023. 3
- [22] Napat Karnchanachari, Dimitris Geromichalos, Kok Seang Tan, Nanxiang Li, Christopher Eriksen, Shakiba Yaghoubi, Noushin Mehdipour, Gianmarco Bernasconi, Whye Kit Fong, Yiluan Guo, et al. Towards learning-based planning: The nuplan benchmark for real-world autonomous driving. In *ICRA*, 2024. 7
- [23] Yingyan Li, Lue Fan, Jiawei He, Yuqi Wang, Yuntao Chen, Zhaoxiang Zhang, and Tieniu Tan. Enhancing end-to-end autonomous driving with latent world model. *arXiv preprint arXiv:2406.08481*, 2024. 2
- [24] Zhiqi Li, Wenhui 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 *ECCV*, 2022. 1
- [25] Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, et al. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024. 1, 2, 6, 7
- [26] Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahua Li, Jan Kautz, Tong Lu, and Jose M Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *CVPR*, 2024. 2
- [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 *ICLR*, 2023. 1
- [28] Bencheng Liao, Shaoyu Chen, Yunchi Zhang, Bo Jiang, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Maptrv2: An end-to-end framework for online vectorized hd map construction. *IJCV*, 2024. 1
- [29] Xuewu Lin, Tianwei Lin, Zixiang Pei, Lichao Huang, and Zhizhong Su. Sparse4d: Multi-view 3d object detection with sparse spatial-temporal fusion. *arXiv preprint arXiv:2211.10581*, 2022. 6
- [30] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. 4
- [31] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. 4
- [32] Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: End-to-end vectorized hd map learning. In *ICML*, 2023. 1
- [33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 4
- [34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 2
- [35] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 4, 5, 6
- [36] 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. 4
- [37] Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masked diffusion policies for navigation and exploration. In *ICRA*, 2024. 4
- [38] Maria Stamatopoulou, Jianwei Liu, and Dimitrios Kanoulas. Dippest: Diffusion-based path planner for synthesizing trajectories applied on quadruped robots. *arXiv preprint arXiv:2405.19232*, 2024. 4
- [39] Wencho Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. *arXiv preprint arXiv:2405.19620*, 2024. 2, 6, 8, 1
- [40] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 8
- [41] Wenwen Tong, Chonghao Sima, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, et al. Scene as occupancy. In *ICCV*, 2023. 8
- [42] 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 *CoRL*, 2022. 1, 6
- [43] Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *CVPR*, 2024. 2
- [44] Yixiao Wang, Chen Tang, Lingfeng Sun, Simone Rossi, Yichen Xie, Chensheng Peng, Thomas Hannagan, Stefano Sabatini, Nicola Poerio, Masayoshi Tomizuka, et al. Optimizing diffusion models for joint trajectory prediction and controllable generation. In *ECCV*, 2024. 3
- [45] Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *CVPR*, 2024. 2, 6
- [46] Zehang Weng, Haofei Lu, Danica Kragic, and Jens Lundell. Dexdiffuser: Generating dexterous grasps with diffusion models. *arXiv preprint arXiv:2402.02989*, 2024. 4
- [47] Sixu Yan, Zeyu Zhang, Muzhi Han, Zaijin Wang, Qi Xie, Zhitian Li, Zhehan Li, Hangxin Liu, Xinggang Wang, and Song-Chun Zhu. M2diffuser: Diffusion-based trajectory optimization for mobile manipulation in 3d scenes. *arXiv preprint arXiv:2410.11402*, 2024. 4
- [48] Brian Yang, Huangyuan Su, Nikolaos Gkanatsios, Tsung-Wei Ke, Ayush Jain, Jeff Schneider, and Katerina Fragkiadaki. Diffusion-es: Gradient-free planning with diffusion for autonomous driving and zero-shot instruction following. In *CVPR*, 2024. 3
- [49] Jingfeng Yao and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. *arXiv preprint arXiv:2501.01423*, 2025. 4
- [50] Jingfeng Yao, Cheng Wang, Wenyu Liu, and Xinggang Wang. Fasterdit: Towards faster diffusion transformers training without architecture modification. *arXiv preprint arXiv:2410.10356*, 2024. 4
- [51] Wenhao Yu, Jie Peng, Huanyu Yang, Junrui Zhang, Yifan Duan, Jianmin Ji, and Yanyong Zhang. Ldp: A local diffusion planner for efficient robot navigation and collision avoidance. *arXiv preprint arXiv:2407.01950*, 2024. 4
- [52] Chengran Yuan, Zhanqi Zhang, Jiawei Sun, Shuo Sun, Zefan Huang, Christina Dao Wen Lee, Dongen Li, Yuhang Han, Anthony Wong, Keng Peng Tee, et al. Drama: An efficient end-to-end motion planner for autonomous driving with mamba. *arXiv preprint arXiv:2408.03601*, 2024. 6
- [53] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy. In *RSS*, 2024. 4
- [54] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-to-end multiple-object tracking with transformer. In *ECCV*, 2022. 1

- [55] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *IJCV*, 2021.
- [56] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *ECCV*, 2022. [1](#)
- [57] Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Truncated diffusion probabilistic models and diffusion-based adversarial auto-encoders. In *The Eleventh International Conference on Learning Representations*, 2023. [4](#)
- [58] Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. In *ECCV*, 2024. [2](#)
- [59] 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 *CoRL*, 2023. [3](#)
- [60] Ziyuan Zhong, Davis Rempe, Danfei Xu, Yuxiao Chen, Sushant Veer, Tong Che, Baishakhi Ray, and Marco Pavone. Guided conditional diffusion for controllable traffic simulation. In *ICRA*, 2023. [3](#)
- [61] Lianghui Zhu, Zilong Huang, Bencheng Liao, Jun Hao Liew, Hanshu Yan, Jiashi Feng, and Xinggang Wang. Dig: Scalable and efficient diffusion models with gated linear attention. *arXiv preprint arXiv:2405.18428*, 2024. [4](#)
- [62] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. [2](#), [6](#)

DiffusionDrive: Truncated Diffusion Model for End-to-End Autonomous Driving

Supplementary Material

A. Further Implementation Detail

We provide additional implementation details for our method on the NAVSIM [10] and nuScenes [2] datasets.

NAVSIM Dataset. We initialize the ResNet-34 [13] backbone with ImageNet pre-trained weights, and the LiDAR range is 32m to the front, back, left, and right following Transfuser baseline [10]. We also perform auxiliary perception tasks following Transfuser baseline [10], which include 3D object detection, 2D BEV semantic segmentation. The object queries and BEV features are taken as input of the proposed diffusion decoder.

nuScenes Dataset. We follow the SparseDrive baseline [39] to perform two-stage training. The model is directly initialized with the stage-1 pre-trained weight, which is trained solely on perception tasks (3D object detection/tracking, vectorized HD map construction, and motion prediction) and provided by the official open-source implementation. We train the stage-2 model on the nuScenes dataset for 10 epochs, replacing the planning module of SparseDrive with our proposed diffusion decoder and truncated diffusion mechanism. Object queries, map queries, and PV features are taken as inputs to the diffusion decoder.

B. Further Ablation Study

| Train | Infer | NC↑ | DAC↑ | TTC↑ | Comf.↑ | EP↑ | PDMS↑ |
|----------------|-----------------|------|------|------|--------|------|-------|
| Anchored Dist. | Anchored. Dist. | 98.2 | 96.2 | 94.7 | 100 | 82.2 | 88.1 |
| | Extra. Traj. | 96.3 | 91.7 | 90.4 | 100 | 76.8 | 81.3 |
| Extra. Traj. | Extra. Traj. | 97.3 | 94.0 | 92.6 | 100 | 79.6 | 84.7 |

Table 8. **Comparison on driving priors.** “Anchored Dist.” denotes anchored Gaussian distribution. “Extra. Traj.” denotes extrapolated trajectory based on current status. Row-1 marked in blue denotes the DiffusioDrive baseline of the main paper.

Comparison on driving priors. In Tab. 8, we validate the superiority of prior anchors over the prior extrapolated trajectory based on the current status. Row-1 is DiffusionDrive baseline. Row-2 uses the DiffusionDrive baseline model to infer from an extrapolated trajectory instead of sampled N_{infer} trajectories. Row-3 represents DiffusionDrive trained with a single anchor, *i.e.*, the extrapolated trajectory, and infers by sampling around it. The results demonstrate the superiority of the proposed anchored Gaussian distribution over extrapolated prior, which fails to cover the potential action space and can not effectively handle challenging scenarios (*e.g.*, obstacle avoidance and turning) in real-world application (consistent with comparisons to ego-status-based planners in Tab. 1 of NAVSIM paper [10]).

| Method | Anchor Source | DS↑ | RC↑ | IS↑ |
|-------------------------|---------------|------------------------------------|------------------------------------|-----------------------------------|
| Transfuser [†] | - | 47.30 ± 5.72 | 93.38 ± 1.20 | 0.50 ± 0.06 |
| DiffusionDrive | NAVSIM | 64.27 ± 2.43 | 94.16 ± 1.46 | 0.69 ± 0.02 |

Table 9. **Generalization of anchor source.** We test DiffusionDrive on Carla Longest6 benchmark with clustered anchors from NAVSIM dataset. [†] denotes that the result is taken from Transfuser paper [7].

Generalization of anchor source. To further investigate the generalization of anchor source, we train DiffusionDrive on CARLA [11] with NAVSIM-clustered anchored Gaussian distribution (Row-2 in Tab. 9). Since the CARLA dataset is totally different from NAVSIM, the superior results validate the generalization capability of our anchored Gaussian distribution, which is designed to cover potential multi-mode driving action space instead of train/val information leakage.

C. Further Qualitative Comparison

In this section, we provide additional qualitative comparisons on challenging scenarios from the planning-oriented NAVSIM dataset navtest split [10].

Going straight. Fig. 5a and Fig. 5b show that the top-1 scoring trajectories of DiffusionDrive are similar to the ground truth trajectories, while the highlighted top-10 scoring trajectories can perform robust lane changes. Notably, Fig. 5c demonstrates that the diverse and highlighted top-10 trajectories can further recognize the traffic light, enabling reasonable lane changes and stopping at the stop line.

Turning left. Fig. 6 shows that the denoised diverse trajectories are dynamically adjusted based on the traffic conditions. The highlighted top-10 scoring trajectories are robust and reasonable, effectively performing lane changes.

Turning right. Fig. 7a and Fig. 7b show that the top-1 scoring trajectories of DiffusionDrive are going to perform car-following like the ground truth trajectories, while the highlighted top-10 scoring trajectories tend to overtake the leading vehicle. These results validate that DiffusionDrive can robustly generate diverse and plausible driving actions.

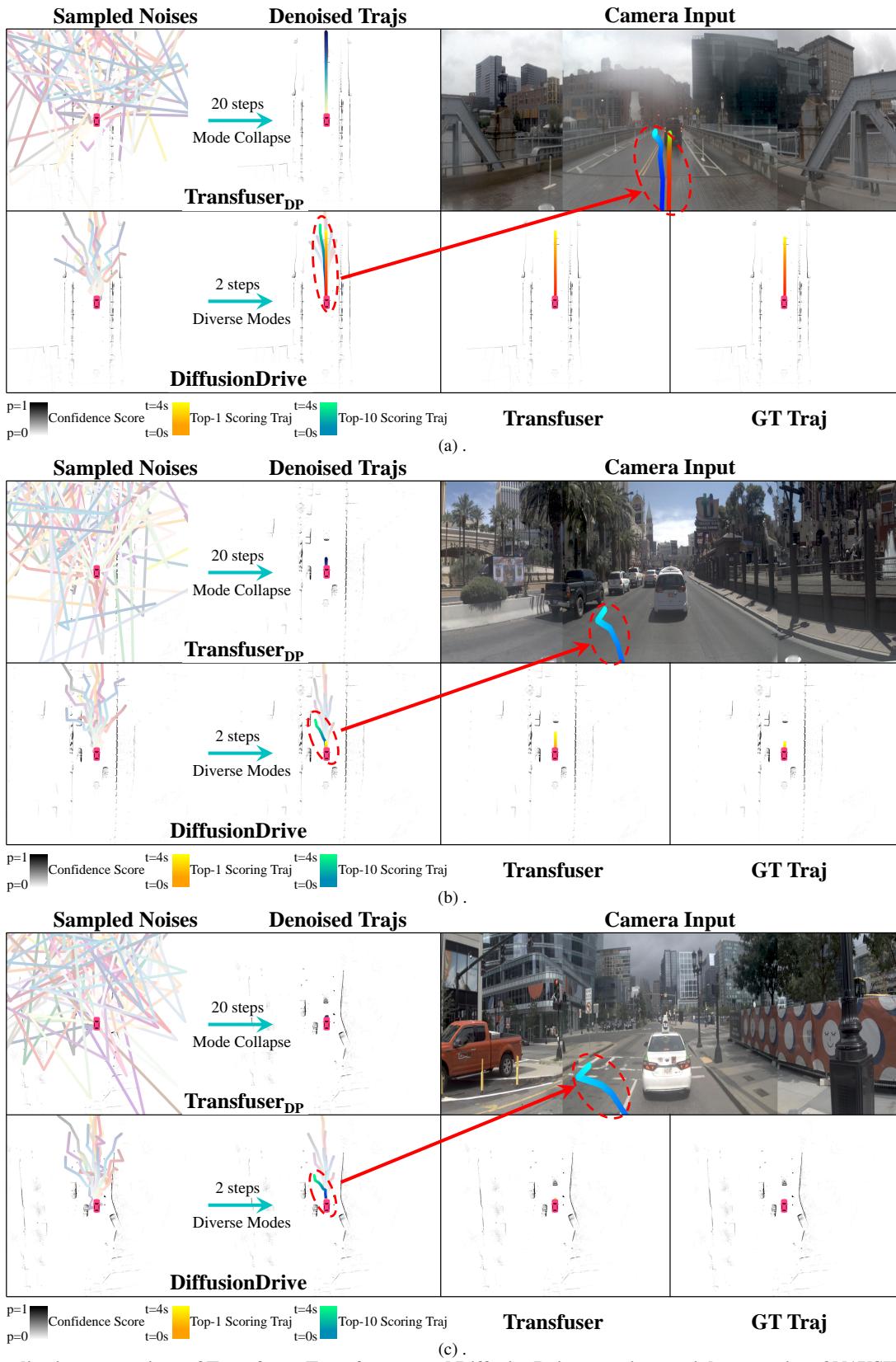


Figure 5. Qualitative comparison of Transfuser, Transfuser_{DP} and DiffusionDrive on going straight scenarios of NAVSIM `navtest` split.

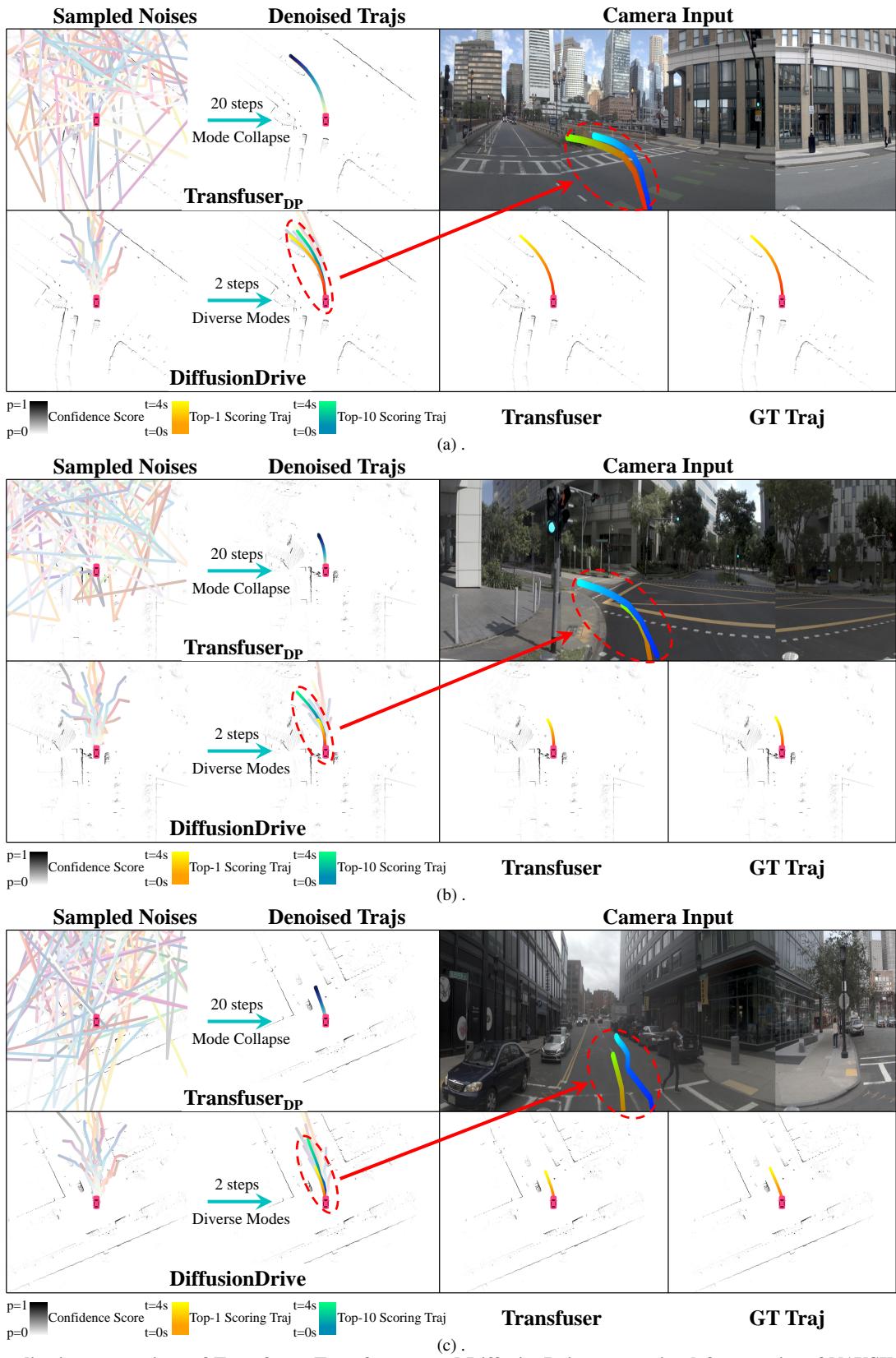


Figure 6. Qualitative comparison of Transfuser, Transfuser_{DP} and DiffusionDrive on turning left scenarios of NAVSIM `navtest` split.

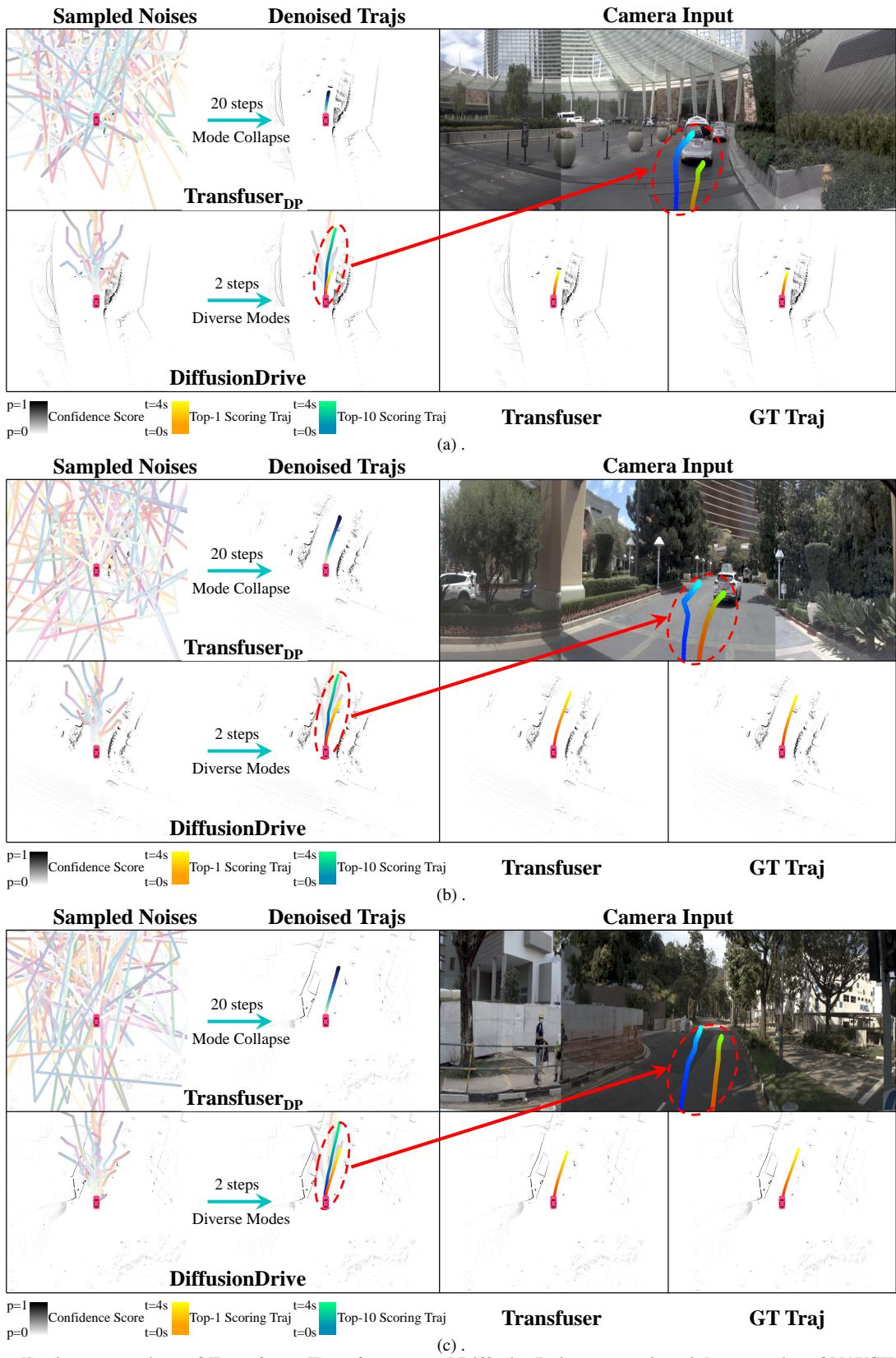


Figure 7. Qualitative comparison of Transfuser, Transfuser_{DP} and DiffusionDrive on turning right scenarios of NAVSIM `navtest` split.