DIFFUSION-BASED PLANNING FOR AUTONOMOUS DRIVING WITH FLEXIBLE GUIDANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Achieving human-like driving behaviors in complex open-world environments is a critical challenge in autonomous driving. Contemporary learning-based planning approaches such as imitation learning methods often struggle to balance competing objectives and lack of safety assurance, due to limited adaptability and inadequacy in learning complex multi-modal behaviors commonly exhibited in human planning, not to mention their strong reliance on the fallback strategy with predefined rules. We propose a novel transformer-based *Diffusion Planner* for closed-loop planning, which can effectively model multi-modal driving behavior and ensure trajectory quality without any rule-based refinement. Our model supports joint modeling of both prediction and planning tasks under the same architecture, enabling cooperative behaviors between vehicles. Moreover, by learning the gradient of the trajectory score function and employing a flexible classifier guidance mechanism, *Diffusion Planner* effectively achieves safe and adaptable planning behaviors. Evaluations on the large-scale real-world autonomous planning benchmark nuPlan and our newly collected 200-hour delivery-vehicle driving dataset demonstrate that *Diffusion Planner* achieves state-of-the-art closed-loop performance with robust transferability in diverse driving styles.

1 INTRODUCTION

Autonomous driving as a cornerstone technology, is poised to usher transportation into a safer and more efficient era of mobility (Tampuu et al., 2020). The key challenge is achieving human-like driving behaviors in complex open-world environment, while ensuring safety, efficiency, and comfort (Muhammad et al., 2020). Rule-based planning methods have demonstrated initial success in industrial applications (Fan et al., 2018), by defining driving behaviors and establishing boundaries derived from human knowledge. However, their reliance on predefined rules limits adaptability to new traffic situations (Hawke et al., 2020), and modifying rules demands extensive engineering effort. In contrast, learning-based planning methods acquire driving skills by cloning human driving behaviors from collected datasets (Caesar et al., 2021), a process made simpler through straightforward imitation learning losses. Additionally, the capabilities of these models can potentially be enhanced by scaling up training resources (Chen et al., 2023).

Though promising, current learning-based planning methods still face several limitations. Firstly, human drivers often exhibit multi-modal behaviors in planning scenarios(Nayakanti et al., 2023). Existing methods that rely on behavior cloning lack a guarantee of fitting such complex data distributions, even when utilizing large transformer-based model architecture or sampling multiple trajectories (Cheng et al., 2023). Secondly, when encountering out-of-distribution (OOD) scenarios, directly using model output may result in low-quality planning outcomes, forcing many methods to fall back on rule-based approaches for trajectory refinement optimization or filtering (Vitelli et al., 2022; Huang et al., 2023), inevitably facing the same inherent limitations associated with rule-based methods. Thirdly, imitation learning alone is inadequate to capture the vast diversity of driving behaviors required for autonomous driving. For example, penalizing unsafe planning via auxiliary loss, as employed in existing methods (Bansal et al., 2018; Cheng et al., 2024), often results in multi-objective conflicts and poor safety performance due to the lack of learning signals that can teach the agent to recover from mistakes (Zheng et al., 2024; Chen et al., 2021). Additionally, well-trained models may be difficult to adapt behaviors to meet specific needs.

In this study, we discover that diffusion model (Ho et al., 2020) possesses huge potential to address the aforementioned issues. Its ability to model complex data distributions (Chi et al., 2023) allows for effective capturing of multi-modal human driving behavior. Additionally, the high-quality generation capability of the diffusion model also provides opportunities for improving the output trajectory quality through appropriate structural design, removing the reliance on rule-based refinement. The best part of diffusion lies in its flexible guidance mechanism (Dhariwal & Nichol, 2021), which allows adaptation to various planning behavioral needs without additional training. Inspired by these observations, we introduce a novel learning-based approach, *Diffusion Planner*, which pioneers the use of diffusion models (Ho et al., 2020) for enhancing closed-loop planning performance without any rule-based refinement. *Diffusion Planner* is realized by learning the gradient of vehicles' trajectory score function (Song & Ermon, 2019) to model the multi-modal data distribution, and further enables personalized planning behavior adaptation through a classifier guidance mechanism. Specifically, we propose a new network architecture built upon the diffusion transformer (Peebles & Xie, 2023). The diffusion loss is employed to jointly train both prediction and planning tasks within the same architecture, enabling cooperative behaviors between vehicles without the need for additional loss functions. Moreover, the versatility of classifier guidance is further demonstrated by its ability to modify the planning behavior of the trained model, such as enhancing safety and comfort, or controlling the vehicle's speed. The differentiable classifier score can be computed in parallel and is flexible for combination, without requiring additional training. Evaluation results on the large-scale real-world autonomous planning benchmark nuPlan (Caesar et al., 2021) demonstrate that *Diffusion Planner* achieves state-of-the-art closed-loop performance among learning-based baselines, comparable to or even surpassing rule-based methods. Additionally, we collected 200 hours of long-term delivery-vehicle driving data in various city-driving scenarios that further validate the transferability and robustness of the model in diverse driving styles.

In summary, our contributions are:

- To the best of our knowledge, we are the first to fully harness the power of diffusion models with a specifically designed architecture for high-performance motion planning, without reliant on rule-based refinement.
- We achieve state-of-the-art performance on the real-world nuPlan dataset, generating more robust and smoother trajectories compared to the baselines.
- We demonstrate that our model can achieve personalized and flexible driving behavior at runtime by utilizing a guidance mechanism, which is a desirable feature for real-world applications.
- We have collected and evaluated a new 200-hour delivery-vehicle dataset, which is compatible with the nuPlan framework, and we will open-source it.

2 Related Work

Rule-based Planner. Rule-based methods rely on predefined rules to dictate the driving behavior of autonomous vehicles, offering a highly controllable and interpretable decision-making process (Treiber et al., 2000a; Fan et al., 2018; Dauner et al., 2023a). While they have been widely validated in real-world scenarios (Leonard et al., 2008; Urmson et al., 2008), these frameworks are limited in their ability to handle novel complex situations that fall beyond the predefined rules.

Learning-based Planner. Learning-based planning focuses on leveraging methods such as behavior cloning in imitation learning to directly model human driving behaviors, which has emerged as a popular solution in autonomous driving, particularly in recent end-to-end training pipelines (Hu et al., 2023; Tampuu et al., 2020; Chen et al., 2023). Behavior cloning method was initially implemented using CNN (Bojarski et al., 2016; Kendall et al., 2019; Hawke et al., 2020) or RNN (Bansal et al., 2018) networks and has since been extended to Transformer due to its strong performance and efficiency in fitting complex data distributions (Scheel et al., 2021; Chitta et al., 2022). However, these methods lack theoretical guarantees for modeling multi-modal driving behavior, which can lead to serious error accumulation in closed-loop planning. As a result, most existing approaches still heavily rely on rules to refine (Vitelli et al., 2022; Huang et al., 2023) or select (Cheng et al., 2024) the generated trajectories, which in some sense, has failed their initial purpose of using learning to replace pre-defined rules. While learning-based methods could offer more human-like driving behavior, their uncontrollable outputs lack safety guarantees and are hard to adjust based on user

needs. Existing methods add extra training losses (Bansal et al., 2018; Cheng et al., 2024), but struggle to strike a balance among competing learning objectives. Additionally, these methods also lack flexibility, making post-training behavior adjustments difficult. In practice, it is desirable for a trained planning model to achieve flexible alignment to various safety and personalized driving preferences during inference, which is still lacking in the current literature. In this work, we develop a novel diffusion planner to tackle the above limitations, which enables the generation of high quality planning trajectories without the need for rule-based refinement, and flexible post-training adaptation to various driving styles through the diffusion guidance mechanism.

Diffusion-based Methods Used in Related Domain. Diffusion models have been recently explored in decision-making fields (Janner et al., 2022; Chi et al., 2023), however, their use in autonomous planning has not yet been fully explored. Some existing works employ diffusion models for motion prediction (Jiang et al., 2023) and traffic simulation (Zhong et al., 2023b;a; Chang et al., 2025), but their focus is on open-loop performance or diversity in simulation rather than quality or drivability, as the outputs are not directly used for control. There are also studies targeting planning (Hu et al., 2024; Yang et al., 2024; Sun et al., 2023), but these approaches often apply diffusion loss to existing frameworks or stack parameters without specific design considerations, making them heavily reliant on post-processing for reasonable performance. In this paper, we demonstrate that with appropriate structural design, the potential of diffusion models can be fully harnessed to enhance closed-loop planning performance in autonomous driving.

3 Preliminaries

3.1 Autonomous Driving and Closed-loop Planning

The primary objective of autonomous driving is to allow vehicles to navigate complex environments with minimal human intervention, where a critical challenge is closed-loop planning (Caesar et al., 2021). Unlike open-loop planning (Caesar et al., 2019) or motion prediction (Ngiam et al., 2021; Zhou et al., 2023), which only involves decision making that adapts to static conditions, closed-loop planning requires a seamless integration of real-time perception, prediction, and control. Vehicles must continuously assess their surroundings, predict the behavior of other neighboring vehicles, and implement precise maneuvers. The dynamic nature of real-world driving scenarios, combined with uncertainty in sensor data and environmental factors, makes closed-loop planning a formidable task.

3.2 Diffusion Model and Guidance Schemes

Diffusion Model. Diffusion Probabilistic Models (Sohl-Dickstein et al., 2015; Ho et al., 2020) are a class of generative models that generate outputs by reversing a Markov chain process known as the forward diffusion process. The transition distribution of the forward process satisfies:

$$q_{t0}(\boldsymbol{x}^{(t)}|\boldsymbol{x}^{(0)}) = \mathcal{N}(\boldsymbol{x}^{(t)} \mid \alpha_t \boldsymbol{x}^{(0)}, \sigma_t^2 \mathbf{I}), t \in [0, 1], \tag{1}$$

which gradually adds Gaussian noise to generate a series of noised data from $\boldsymbol{x}^{(0)}$ to $\boldsymbol{x}^{(t)}$ with $t \in [0, 1]$. $\sigma_t > 0$ is a variance term that controls the introduced noise and $\alpha_t > 0$ is typically defined as $\alpha_t = \sqrt{1 - \sigma_t^2}$, ensuring $\boldsymbol{x}^{(t)} \to \mathcal{N}(0, \mathbf{I})$, as $t \to 1$. The reversed denoising process of Eq. (1) can be equivalently expressed as a diffusion ODE (Song et al., 2021):

$$\text{(Diffusion ODE)} \quad \mathrm{d}\boldsymbol{x}^{(t)} = \left[f(t)\boldsymbol{x}^{(t)} - \frac{1}{2} g^2(t) \nabla_{\boldsymbol{x}^{(t)}} \log q_t(\boldsymbol{x}^{(t)}) \right] \mathrm{d}t, \tag{2}$$

where $f(t) = \frac{\mathrm{d} \log \alpha_t}{\mathrm{d}t}, g^2(t) = \frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t} - 2\frac{\mathrm{d} \log \alpha_t}{\mathrm{d}t}\sigma_t^2$ are determined by the fixed noise schedules α_t, σ_t, and q_t is the marginal distribution of $\boldsymbol{x}^{(t)}$. Diffusion model utilizes a neural network $\boldsymbol{s}_\theta(\boldsymbol{x}^{(t)}, t)$ to fit the probability score $\nabla_{\boldsymbol{x}^{(t)}} \log q_t(\boldsymbol{x}^{(t)})$. By learning the score function, diffusion models enjoy the strong expressiveness of modeling arbitrary complex distributions (Chi et al., 2023), making it highly versatile and adaptable for challenging tasks such as autonomous driving.

Classifier Guidance. Classifier guidance (Dhariwal & Nichol, 2021) is a technique used to generate preferred data by guiding the sampling process with a classifier $\mathcal{E}_\phi(\boldsymbol{x}^{(t)}, t)$. The gradient of the classifier score is used to modify the original diffusion score:

$$\tilde{\boldsymbol{s}}_\theta(\boldsymbol{x}^{(t)}, t) = \boldsymbol{s}_\theta(\boldsymbol{x}^{(t)}, t) - \nabla_{\boldsymbol{x}^{(t)}} \mathcal{E}_\phi(\boldsymbol{x}^{(t)}, t) \tag{3}$$

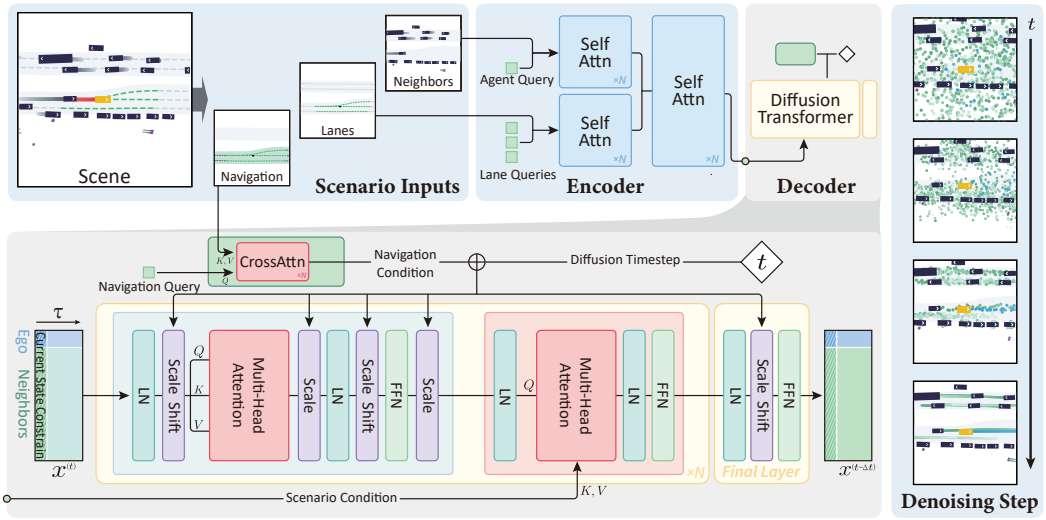

Figure 1: Model architecture of *Diffusion Planner*.

In autonomous driving, this approach offers greater flexibility compared to rule-based refinement because it directly improves the model's inherent ability, rather than overly relying on sub-optimal post-processing that requires significant human effort and targeted data collection.

4 METHODOLOGY

In this section, we redefine the planning task as a future trajectory generation task, which jointly generates the ego vehicle's planning and the prediction of neighboring vehicles. We then introduce the *Diffusion Planner*, a novel approach that leverages the expressive and flexible diffusion model for enhanced autonomous planning. Lastly, we demonstrate how the guidance mechanism in diffusion models can be utilized to align planning behavior with safe or human-preferred driving styles.

4.1 TASK REDEFINITION

Autonomous driving requires considering the close interaction between the ego and neighboring vehicles, resulting in a cooperative relationship between planning and motion prediction tasks (Ngiam et al., 2021). Supervising the future trajectories of neighboring vehicles has been shown to be helpful to enhance the ability of closed-loop planning models to handle complex interaction scenarios (Hu et al., 2023). For real-world deployment, motion prediction can also enhance safety by providing more controllable measures, facilitating the implementation of the system (Fan et al., 2018). Consequently, the trajectories of neighboring vehicles have become crucial privileged information for model training. However, the common approaches that use a dedicated sub module (Huang et al., 2023) or additional loss design (Cheng et al., 2023; Huang et al., 2023) to capture privileged information limit their modeling power during training and also lead to a more complex framework.

In this work, we address this issue by collectively considering the status of key participants in the driving scenario and jointly modeling the motion prediction and closed-loop planning tasks as a *future trajectory generation* task. Specifically, given conditions $\boldsymbol{C}$, which include current vehicle states, historical data, lane information, and navigation information, our goal is to generate future trajectories for all key participants simultaneously, enabling the modeling of cooperative behaviors among them. However, this joint modeling of complex distributions is challenging to solve with a simple behavior cloning approach. Benefiting from the strong expressive power of diffusion models, we adopt a diffusion model for this task and formulate the target as:

$$\boldsymbol{x}^{(0)} = \begin{bmatrix} x^{(0)}_{\text{ego}} \\ x^{(0)}_{\text{neighbor}_1} \\ \vdots \\ x^{(0)}_{\text{neighbor}_M} \end{bmatrix} = \begin{bmatrix} x^{1}_{\text{ego}} & x^{2}_{\text{ego}} & \cdots & x^{\tau}_{\text{ego}} \\ x^{1}_{\text{neighbor}_1} & x^{2}_{\text{neighbor}_1} & \cdots & x^{\tau}_{\text{neighbor}_1} \\ \vdots & \vdots & \ddots & \vdots \\ x^{1}_{\text{neighbor}_M} & x^{2}_{\text{neighbor}_M} & \cdots & x^{\tau}_{\text{neighbor}_M} \end{bmatrix}, \tag{4}$$

where we use superscripts with parentheses to represent the timeline of diffusion denoising, and regular superscripts to indicate the timeline of the future trajectory, which contains τ time steps. For each state x, we only consider the coordinates and the sine and cosine of the heading angle, which are sufficient for the downstream LQR controller. We select the nearest M neighboring vehicles to predict their possible future trajectories. By parameterizing our *Diffusion Planner* with θ, the training target can be expressed as:

$$\mathcal{L}_\theta = \mathbb{E}_{\boldsymbol{x}^{(0)},t\sim\mathbb{U}(0,1),\boldsymbol{x}^{(t)}\sim q_{t0}(\boldsymbol{x}^{(t)}|\boldsymbol{x}^{(0)})}\left[||\mu_\theta(\boldsymbol{x}^{(t)},t,\boldsymbol{C})-\boldsymbol{x}^{(0)}||^2\right], \tag{5}$$

where the goal is to recover the data distribution from noisy data (Ramesh et al., 2022). We can get the score function as $\boldsymbol{s}_\theta = (\alpha_t\mu_\theta - \boldsymbol{x}^{(t)})/\sigma_t^2$ and apply it during the diffusion denoising process, as shown in Section 3.2.

4.2 Diffusion Planner

Diffusion Planner is a model based on the DiT architecture (Peebles & Xie, 2023), with a core design focusing on the fusion mechanism between noised future vehicle trajectories $\boldsymbol{x}$ and conditional information $\boldsymbol{C}$. Figure 1 provides an overview of the complete architecture. A detailed description of these interaction and fusion modules is provided as follows.

Vehicle Information Integration. In the first step, the future vehicle trajectory $\boldsymbol{x}$ is concatenated with the current state of each vehicle, represented as $x^0 = [x^0_{\text{ego}}, x^0_{\text{neighbor1}}, \ldots, x^0_{\text{neighbor}_\text{M}}]^T$. This concatenation acts as a constraint to guide the model, simplifying the planning task by providing a clear starting point. Notably, velocity and acceleration information for the ego vehicle is excluded, which has been shown to enhance closed-loop performance, as highlighted in previous works (Cheng et al., 2023; Li et al., 2024). Integration of the information from different vehicles during model execution is achieved through multi-head self-attention mechanisms.

Historical Status and Lane Information Fusion. The historical status of neighboring vehicles and lane information is represented using vectors (Gao et al., 2020). Specifically, each neighboring vehicle is represented as $\boldsymbol{S}_{\text{neighbor}} \in \mathbb{R}^{L\times D_{\text{neighbor}}}$, and lanes as $\boldsymbol{S}_{\text{lane}} \in \mathbb{R}^{P\times D_{\text{lane}}}$, where L refers to the number of past timestamps, and P indicates the number of points per polyline. D_{neighbor} contains data such as vehicle coordinates, heading, velocity, size, and category, while D_{lane} provides lane details such as coordinates, traffic light status, and speed limits. We build modality-specific perceiver resampler (Alayrac et al., 2022) to extract representations from initial information-sparse vector data. This is achieved by learning a predefined number of latent input queries $\boldsymbol{Q}$, which cross-attend to latent vector data $\boldsymbol{S}_{\text{latent}}$ through several perceiver resampler blocks. The $\boldsymbol{S}_{\text{latent}}$ vectors are derived from $\boldsymbol{S}_{\text{neighbor}}$ and $\boldsymbol{S}_{\text{lane}}$ using MLPs. The forward process of each perceiver resampler block can be formulated as:

$$\boldsymbol{Q} = \boldsymbol{Q} + \text{MHA}(q=\boldsymbol{Q}, kv=\text{concat}(\boldsymbol{Q}, \boldsymbol{S}_{\text{latent}})), \boldsymbol{Q} = \boldsymbol{Q} + \text{FFN}(\boldsymbol{Q}). \tag{6}$$

Here, MHA donates the multi-head attention operator, and FFN is a feed-forward network. For each neighboring vehicle or lane $\boldsymbol{S_i}$, we will obtain a $\boldsymbol{Q}_i \in \mathbb{R}^{N\times D_{\text{latent}}}$, where N is the predefined number of latent queries. We then concatenate all $\boldsymbol{Q}_i$ and feed them into a vanilla transformer encoder to further aggregate the information and obtain the encoder representation $\boldsymbol{Q}_f$. Finally we fuse the $\boldsymbol{Q}_f$ with x and the process is:

$$\boldsymbol{x} = \boldsymbol{x} + \text{MHCA}(\boldsymbol{x}, \boldsymbol{Q}_f), \boldsymbol{x} = \boldsymbol{x} + \text{FFN}(\boldsymbol{x}), \tag{7}$$

where MHCA donate multi-head cross-attention.

Navigation Information Fusion. Navigation information is crucial for autonomous driving planning, as it provides essential guidance on the intended route, enabling the vehicle to make informed decisions. In the nuPlan benchmark (Caesar et al., 2021), navigation information is represented as a set of lanes in a route. We handle navigation $\boldsymbol{S}_{\text{navi}} \in \mathbb{R}^{D\times P\times D_{\text{lane}}}$ with learned predefined number of navigation query $\boldsymbol{Q}_n$ to extract essential guidance. The process can be formulated as :

$$\boldsymbol{Q}_n = \boldsymbol{Q}_n + \text{MHCA}(\boldsymbol{Q}_n, \boldsymbol{S}_{\text{navi}}), \boldsymbol{Q}_n = \boldsymbol{Q}_n + \text{FFN}(\boldsymbol{Q}_n). \tag{8}$$

$\boldsymbol{Q}_n$ is then combined with the diffusion timestep condition $\boldsymbol{C}_t$ and applied through an adaptive layer norm block to guide trajectory generation across all tokens. The fusion process can be written as:

$$\boldsymbol{x} = \boldsymbol{x} * \text{MLP}_1(\text{concat}(\boldsymbol{Q}_n, \boldsymbol{C}_t)) + \text{MLP}_2(\text{concat}(\boldsymbol{Q}_n, \boldsymbol{C}_t)). \tag{9}$$

4.3 Planning Behavior Alignment via Classifier Guidance

Enforcing versatile and controllable driving behavior is crucial for real-world autonomous driving. For example, vehicles must ensure safety and comfort while adjusting speeds to align with user preferences. Thanks to its close relationship to Energy-Based Models (Lu et al., 2023), diffusion model can conveniently inject such preferences via classifier guidance. It can steer the model outputs via gradient surgery during inference, offering significant potential for customized adaptation.

Specifically, given the original driving behavior $q_0(\boldsymbol{x}^{(0)})$, we aim to encode additional guidance to reinforce some preferred behavior upon the existing behavior q_0. This operation can be formulated as generating a target behavior as a joint distribution with the following form:

$$p_0(\boldsymbol{x}^{(0)}) \propto q_0(\boldsymbol{x}^{(0)}) e^{-\mathcal{E}(\boldsymbol{x}^{(0)})}, \tag{10}$$

where $\mathcal{E}(\boldsymbol{x}^{(0)})$ can be some form of energy function that encodes safety or preferred behavior. The gradient of intermediate energy $\mathcal{E}_t(\boldsymbol{x}^{(t)})$ is used to modify the original probability score, to encourage the model to generate trajectories within the target distribution:

$$\begin{aligned} \nabla_{\boldsymbol{x}^{(t)}} \log p_t(\boldsymbol{x}^{(t)}) &= \nabla_{\boldsymbol{x}^{(t)}} \log q_t(\boldsymbol{x}^{(t)}) - \nabla_{\boldsymbol{x}^{(t)}} \mathcal{E}_t(\boldsymbol{x}^{(t)}) \\ &= \nabla_{\boldsymbol{x}^{(t)}} \log q_t(\boldsymbol{x}^{(t)}) - \nabla_{\boldsymbol{x}^{(t)}} \log \mathbb{E}_{q_{0t}(\boldsymbol{x}^{(0)}|\boldsymbol{x}^{(t)})} \left[e^{-\mathcal{E}(\boldsymbol{x}^{(0)})} \right]. \end{aligned} \tag{11}$$

However, the calculation of the latter term is intractable, necessitating a time-dependent classifier to obtain an unbiased estimation of the gradient of intermediate energy $\mathcal{E}_t(\boldsymbol{x}^{(t)})$ using contrastive loss (Lu et al., 2023). The training of the classifier will inevitably introduce additional training difficulty and approximation errors, making it unsuitable for safety-critical applications, especially in complex scenarios like autonomous driving. Instead, diffusion posterior sampling (Chung et al., 2022) offers a training free method that only uses the trained diffusion model μ_θ in Eq. (5) to approximate the guidance energy, bypassing the notorious difficult classifier training:

$$\begin{aligned} \nabla_{\boldsymbol{x}^{(t)}} \log p_t(\boldsymbol{x}^{(t)}) &\approx \nabla_{\boldsymbol{x}^{(t)}} \log q_t(\boldsymbol{x}^{(t)}) - \nabla_{\boldsymbol{x}^{(t)}} \mathcal{E} \left(\mathbb{E}_{q_{0t}(\boldsymbol{x}^{(0)}|\boldsymbol{x}^{(t)})}[\boldsymbol{x}^{(0)}] \right) \\ &= \nabla_{\boldsymbol{x}^{(t)}} \log q_t(\boldsymbol{x}^{(t)}) - \nabla_{\boldsymbol{x}^{(t)}} \mathcal{E} \left(\mu_\theta(\boldsymbol{x}^{(t)}, t, \boldsymbol{C}) \right). \end{aligned} \tag{12}$$

Although this method is easy to implement, it can only provide accurate guidance when t approaches zero since the energy gradient becomes approximately equal to the gradient in Eq. (11) (Lu et al., 2023) as $t \to 0$. As shown in Figure 2, applying guidance too early can lead to inaccurate estimations, resulting in abnormal trajectories. Therefore, we apply classifier guidance only during the final few denoising steps to ensure accurate energy estimation (Clark et al., 2023; Xu et al., 2024). One restriction of this method is that Eq. (12) needs to use a pre-defined differentiable energy function $\mathcal{E}(\cdot)$ to calculate the guidance energy. Fortunately, in autonomous driving scenarios, many trajectory evaluation protocols can be defined using differentiable functions. Next, we briefly describe some applicable energy functions that can be used to customize the planning behavior of the model, more details are shown in Appendix C.3.

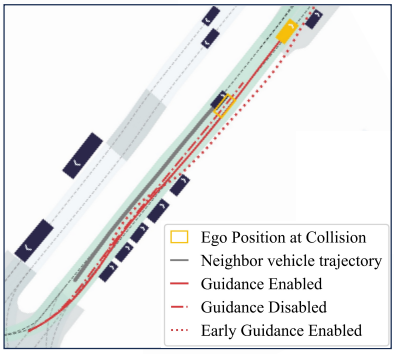

Figure 2: Collision avoidance guidance: Closed-loop tests start from the same point, with driven trajectories plotted.

- **Collision avoidance**: The signed distance between the ego vehicle and neighboring vehicles is computed at each timestamp. When bounding boxes overlap, the minimum separation distance can be used as the energy function. Otherwise, the nearest-point distance is calculated.
- **Target speed maintenance**: The speed difference is used as the energy, calculated by comparing the planned average speed with the set target speed.
- **Comfort**: The longitudinal and lateral accelerations, and the yaw rate are calculated. Comfort thresholds are set using human driving data from the dataset (Caesar et al., 2021), and the extent of any deviations beyond these limits is measured.
- **Staying within drivable area**: We construct the differentiable cost map by using Euclidean Signed Distance Field with parallel computation (Cheng et al., 2024), which can compute the distance that the ego vehicle goes outside the lane at each time step.

4.4 Practical Implementation for Closed-Loop Planning

Data augmentation can help alleviate the out-of-distribution issue and is widely used in planning. Before training, we add random perturbations to the current state (Cheng et al., 2023). Then, interpolation is applied to create a physically feasible transition, enabling the model to resist perturbations and regress to the ground-truth trajectory (Bansal et al., 2018). After that, we transform the data from the global coordinate system into an ego-centric formulation through coordinate transformation. Considering the significant difference between the longitudinal and lateral distances traveled by the vehicle, z-score normalization is used to ensure the mean of the data distribution is close to zero, thereby further stabilizing the training process. During inference, DPM-Solver (Lu et al., 2022) is employed to achieve faster sampling, while low-temperature sampling (Ajay et al., 2022) enhances determinism in the planning process. We can complete trajectory planning for the next 8 seconds at 10 Hz, along with predictions for neighboring vehicles, within 0.08 seconds. Please see Appendix C for implementation details.

5 Experiments

Evaluation Setups. We conduct extensive evaluations on the large-scale real-world autonomous planning benchmark, nuPlan (Caesar et al., 2021), to compare *Diffusion Planner* with other state-of-the-art planning methods. For testing, the Test14 benchmark (Cheng et al., 2023) is used, where 20 random scenarios are selected for each of the 14 scenario types specified in the nuPlan challenge. The final score is the average across all scenarios, ranging from 0 to 100, with a higher score indicating better algorithm performance. To further validate the algorithm's performance across various driving scenarios and with vehicles exhibiting different driving behaviors, 200 hours of real-world data were collected from a delivery-vehicle. Unlike nuPlan, the delivery-vehicle displays more conservative planning behavior and can operate in bike lanes, which involve dense human-vehicle interactions and different traffic regulations. The original data are integrated into the nuPlan framework, and the same metrics are used to evaluate the model in closed-loop simulations, as detailed in Appendix D. All experimental results are tested in closed-loop non-reactive mode, a test mode commonly used by most baselines for comparison.

Baselines. The baselines are categorized into three groups (Dauner et al., 2023b): *Rule-based*, *Learning-based*, and *Hybrid*, which added extra refinement to the output of the learning-based model. We compare the *Diffusion Planner* against the following baselines, and more implementation details of baselines are shown in Appendix C.4.

- *IDM* (Treiber et al., 2000b): A classic rule-based method implemented by nuPlan.
- *PDM* (Dauner et al., 2023b): The first-place winner of the nuPlan challenge offers a rule-based version that follows the centerline (*PDM-Closed*), a learning-based version conditioned on the reference line (*PDM-Open*), and a hybrid approach that combines both (*PDM-Hybrid*).
- *UrbanDriver* (Scheel et al., 2021): A learning-based method using policy gradient optimization and implemented by nuPlan.
- *GameFormer* (Huang et al., 2023): Modeling ego and neighboring vehicle interactions using game theory (*GameFormer w/o refine.*), followed by rule-based refinement.
- *PlanTF* (Cheng et al., 2023): A state-of-the-art learning-based method built on a transformer architecture, exploring various designs suitable for closed-loop planning.
- *PLUTO* (Cheng et al., 2024): Building on *PDM-Open*, a complex model with contrastive learning enhances environmental understanding (*PLUTO w/o refine.*), followed by post-processing.

Main Results. Evaluation results on the nuPlan benchmark are presented in Table 1. With the guidance of the collision avoidance classifier in Section 4.3, *Diffusion Planner w/ Collision Guidance* achieves the highest score among learning-based methods, surpassing rule-based approaches, and performing comparably to hybrid methods. A case demonstrating the performance of collision avoidance guidance is shown in Figure 2. Notably, compared to the transformer-based *PlanTF*, *Diffusion Planner* leverages the power of diffusion to achieve strong performance, ranking just below *PLUTO w/o refine*. However, *PLUTO w/o refine* benefits from using pre-searched reference paths as input, which significantly reduces the demand on the model's planning capability. *GameFormer*,

Table 1: Closed-loop planning results on nuPlan dataset. : The highest scores of baselines in various types. *: Using pre-searched reference lines as model input provides prior knowledge, reducing the difficulty of planning compared to standard learning-based methods.

Type	Planner	Score	Collisions	TTC	Drivable	Comfort	Progress
Rule -based	IDM	70.39	86.78	77.01	87.35	91.19	95.40
	PDM-Closed	90.05	96.93	91.57	98.85	93.87	98.85
Hybrid	PDM-Hybrid	90.10	97.31	91.19	98.85	93.49	98.85
	GameFormer	83.88	97.32	93.49	94.63	93.49	96.55
	PLUTO	92.23	97.31	95.38	98.85	93.08	99.62
Learning-based	PDM-Open*	52.81	74.71	71.26	87.35	98.85	88.88
	UrbanDriver	51.83	69.73	64.37	82.76	98.85	95.02
	GameFormer w/o refine.	11.36	43.68	39.09	33.33	97.32	65.90
	PlanTF	85.62	94.64	90.04	96.93	92.72	98.85
	PLUTO w/o refine.*	89.90	97.68	94.21	97.68	95.37	99.61
	Diffusion Planner (Ours)	89.22	95.59	91.57	98.85	95.79	100.0
	w/ Collision Guidance	91.05	97.70	94.25	98.08	95.02	100.0

Table 2: Closed-loop planning results on delivery-vehicle driving dataset.

Type	Planner	Score	Collisions	TTC	Drivable	Comfort	Progress
Rule -based	IDM	75.38	86.00	79.43	99.43	89.14	95.43
	PDM-Closed	80.95	86.51	80.00	100.0	97.21	97.47
Hybrid	GameFor mer	51.35	82.50	72.50	65.00	98.00	90.00
	PDM-Hybrid	80.72	86.50	77.00	100.0	92.50	99.00
Learning-based	PDM-Open*	64.84	75.75	70.50	93.50	98.50	95.00
	GameFormer w/o refine.	22.41	62.00	57.50	33.00	98.50	77.00
	PlanTF	90.89	95.00	90.50	99.50	96.00	99.50
	Diffusion Planner (ours)	92.08	96.00	91.00	100.0	94.00	100.0
	w/ Collision Guidance	93.75	97.75	92.00	100.0	93.50	100.0

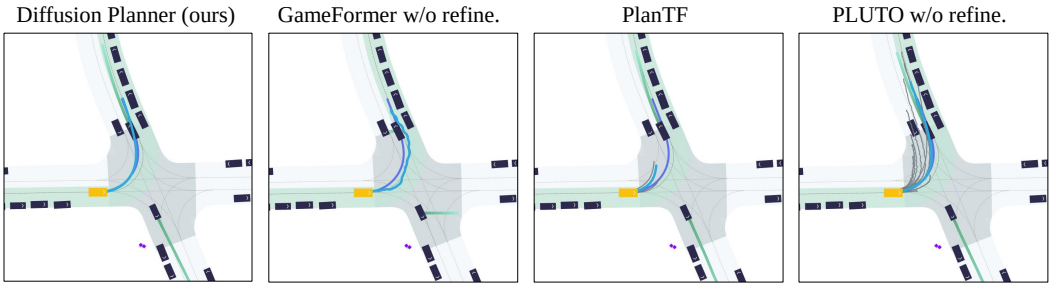

Figure 3: Future trajectory generation visualization. A frame from a challenging narrow road turning scenario in the closed-loop test, including the future planning of the ego vehicle (*PlanTF* and *PLUTO w/o refine.* showing multiple candidate trajectories), predictions for neighboring vehicles, and the ground truth ego trajectory.

which models the interactions between the ego vehicle and neighboring vehicles using game theory, exhibits limited model capabilities, making it overly reliant on rule-based refinements. We further present the planning results on delivery-vehicle driving dataset as shown in Table 2. *PDM* and *GameFormer* include certain designs specifically tailored to the nuPlan benchmark, which limits their ability to transfer to delivery-vehicle driving tasks, resulting in a drop in performance. In contrast, *Diffusion Planner* demonstrates strong transferability across different driving behaviors.

Qualitative Results. To further demonstrate the capabilities of learning-based models, we plot the trajectory generation results of representative baselines (without refinement) as shown in Figure 3. *Diffusion Planner* shows high-quality trajectory generation, with accurate predictions for neighboring vehicles and smooth ego planning trajectories that reasonably account for the speed of the vehicle ahead, demonstrating the advantages of joint modeling of both prediction and planning tasks. More closed-loop planning results are shown in Appendix A. In contrast, *GameFormer w/o*

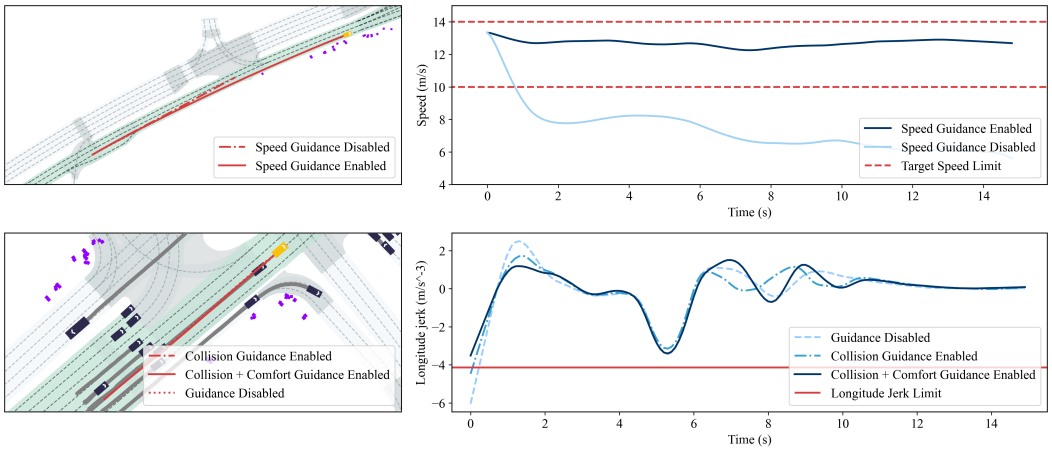

Figure 4: Target speed and comfort guidance: For target speed guidance, the speed changes before and after guidance are plotted. For comfort guidance, the longitudinal jerk changes are compared before and after applying comfort guidance on top of collision avoidance guidance.

refine produces less smooth trajectories and inaccurate predictions for neighboring vehicles, which explains why it heavily relies on refinement. Although *PlanTF* and *PLUTO w/o refine.* sample multiple trajectories at once, most of them are of low quality.

5.1 Empirical Studies of Diffusion Planner Properties

Multi-modal Planning Behavior. We selected an intersection scenario and performed multiple inferences without low temperature sampling from the same initial position to obtain different possible outputs, in order to evaluate the model's ability to fit multi-modal driving behaviors. As shown in Figure 5, without navigation information, the vehicle can exhibit three distinct driving behaviors—left turn, right turn, and straight ahead—with clear differentiation. When navigation information is provided, the model accurately follows it to make a left turn, demonstrating the diffusion model's ability to fit driving behaviors with varying distributions and its capacity for switching between them.

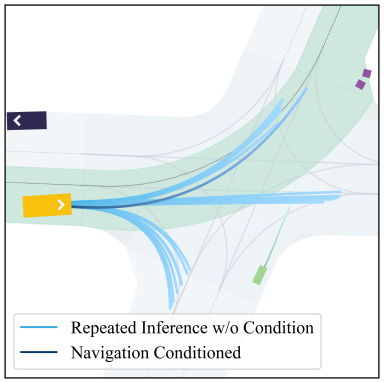

Figure 5: Multi-modal planning behavior of *Diffusion Planner*.

Flexible guidance mechanism. Based on the trained *Diffusion Planner* model, different types of classifier guidance, as described in Section 4.3, are added. We consider three types of guidance: collision avoidance, target speed maintenance, and comfort. We primarily showcase the feasibility of the guidance mechanism and leave the design of detailed guidance energy functions as a user choice for practical implementation. 1) For collision avoidance, as shown in Table 1 and Figure 2, the model's performance in avoiding collisions significantly improved, with the score increasing from 89.22 to 91.05. 2) For target speed setting, we masked all lane speed limit information to prevent it from influencing the model's planning, ensuring that speed adjustments are made solely through guidance. As shown in Figure 4, the model exhibited a lower speed without guidance. By setting the speed between 10m/s and 14m/s, the model is able to closely match the desired speed range while maintaining smooth speed transitions. 3) For the comfort guidance, it is incorporated on top of collision avoidance guidance, which does not account for comfort and thus leading to possible undesirable behaviors due to gradient corrections. This combination effectively addresses the issue, as is shown in Figure 4.

5.2 Ablation Studies

Design Choices for training. We demonstrate the effectiveness of individual components of our method: data processing, the handling approach of ego current state, and the number of predicted

Table 3: Ablation of each modules during the training process on nuPlan Test14 Benchmark.

Type	Planner	Score
Base	Diffusion Planner	89.22
Data	w/o z-score norm	85.02
	w/o interpolation	83.78
	w/o augmentation	76.53
Ego state	w/ SDE	82.90
	w/ ego state	78.65
	w/o current state	81.11

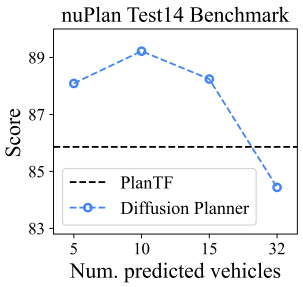

Figure 6: Ablation on the number of predicted vehicles M.

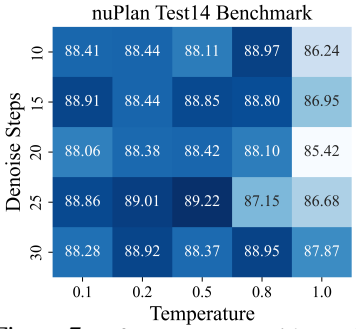

Figure 7: Inference param grid search.

vehicles. 1) We ablate the model's performance without using z-score normalization (*w/o z-score norm*), as well as without data augmentation (*w/o augmentation*), or by only perturbing the current state without applying interpolation to future trajectories (*w/o interpolation*). The results are summarized in Table 3. For the *w/o z-score norm* variant, even with ego-centric transformation, the data range remains large, making it difficult for the model to fit the distribution. The *w/o augmentation* variant faces out-of-distribution issues, leading to poor performance. Results also show that future trajectory interpolation is essential compared to perturbing only the current state. 2) We analyze the impact of the ego vehicle's current state on the model. Retaining velocity, acceleration, and yaw rate (*w/ ego state*) may lead to learning shortcuts, resulting in decreased planning capability. While a state dropout encoder (Cheng et al., 2023) (*w/ SDE*) mitigates this, directly discarding the information is more effective. Additionally, the *w/o current state* shows that adding current state information to the decoder improves planning capability. 3) We also ablate the choice of the number of M. Figure 6 shows that including too many neighboring vehicle tokens in the decoder introduces noise, affecting the performance of the ego vehicle. However, most choices still outperform *PlanTF*.

Design Choices for Inference. We sweep two hyperparameters: the number of denoise steps and the magnitude of low-temperature sampling, as shown in Figure 7. Low temperature helps improve the stability of the output trajectories, thus avoiding abrupt trajectory changes that could pose safety risks. Additionally, the model leverages DPM-Solver to achieve efficient denoising and remains robust across different step counts. We report the detailed parameter selection in Table 6.

6 Conclusion

We propose *Diffusion Planner*, a learning-based approach that fully exploits the expressive power and flexible guidance mechanism of diffusion models for high-quality autonomous planning. A transformer-based architecture is introduced to jointly model the multi-modal data distribution in motion prediction and planning tasks through a diffusion objective. Classifier guidance is employed to align planning behavior with safe or user preferred driving styles. *Diffusion Planner* achieves state-of-the-art closed-loop performance without relying on any rule-based refinement on the nuPlan benchmark and a newly collected 200-hour delivery-vehicle driving dataset, demonstrating strong adaptability across diverse driving styles. *Diffusion Planner* provides a very versatile planning framework for autonomous driving. It is also possible to extend it into an end-to-end framework with raw sensor inputs by incorporating a well-designed encoder structure, which we will leave it as a meaningful direction for future investigation. Due to space limit, more discussion on limitations and future direction can be found in Appendix E.

References

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 *The Eleventh International Conference on Learning Representations*, 2022.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language

model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*, 2018.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021.

Wei-Jer Chang, Francesco Pittaluga, Masayoshi Tomizuka, Wei Zhan, and Manmohan Chandraker. Safe-sim: Safety-critical closed-loop traffic simulation with diffusion-controllable adversaries. In *European Conference on Computer Vision*, pp. 242–258. Springer, 2025.

Dian Chen, Vladlen Koltun, and Philipp Krähenbühl. Learning to drive from a world on rails. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15590–15599, 2021.

Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *arXiv preprint arXiv:2306.16927*, 2023.

Jie Cheng, Yingbing Chen, Xiaodong Mei, Bowen Yang, Bo Li, and Ming Liu. Rethinking imitation-based planners for autonomous driving, 2023.

Jie Cheng, Yingbing Chen, and Qifeng Chen. Pluto: Pushing the limit of imitation learning-based planning for autonomous driving. *arXiv preprint arXiv:2404.14327*, 2024.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12878–12895, 2022.

Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards. *arXiv preprint arXiv:2309.17400*, 2023.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *Conference on Robot Learning (CoRL)*, 2023a.

Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *Conference on Robot Learning*, pp. 1268–1281. PMLR, 2023b.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Haoyang Fan, Fan Zhu, Changchun Liu, Liangliang Zhang, Li Zhuang, Dong Li, Weicheng Zhu, Jiangtao Hu, Hongye Li, and Qi Kong. Baidu apollo em motion planner, 2018.

Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11525–11533, 2020.

Jeffrey Hawke, Richard Shen, Corina Gurau, Siddharth Sharma, Daniele Reda, Nikolay Nikolov, Przemysław Mazur, Sean Micklethwaite, Nicolas Griffiths, Amar Shah, et al. Urban driving with conditional imitation learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 251–257. IEEE, 2020.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17853–17862, 2023.

Yihan Hu, Siqi Chai, Zhening Yang, Jingyu Qian, Kun Li, Wenxin Shao, Haichao Zhang, Wei Xu, and Qiang Liu. Solving motion planning tasks with a scalable generative model. *arXiv preprint arXiv:2407.02797*, 2024.

Zhiyu Huang, Haochen Liu, and Chen Lv. Gameformer: Game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3903–3913, 2023.

Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, pp. 9902–9915. PMLR, 2022.

Chiyu Jiang, Andre Cornman, Cheolho Park, Benjamin Sapp, Yin Zhou, Dragomir Anguelov, et al. Motiondiffuser: Controllable multi-agent motion prediction using diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9644–9653, 2023.

Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, and Amar Shah. Learning to drive in a day. In *2019 international conference on robotics and automation (ICRA)*, pp. 8248–8254. IEEE, 2019.

John J. Leonard, Jonathan P. How, Seth J. Teller, Mitch Berger, Stefan Campbell, Gaston A. Fiore, Luke Fletcher, Emilio Frazzoli, Albert S. Huang, Sertac Karaman, Olivier Koch, Yoshiaki Kuwata, David C. Moore, Edwin Olson, Steven C. Peters, Justin Teo, Robert Truax, Matthew R. Walter, David Barrett, Alexander K Epstein, Keoni Maheloni, Katy Moyer, Troy Jones, Ryan Buckley, Matthew E. Antone, Robert Galejs, Siddhartha Krishnamurthy, and Jonathan Williams. A perception-driven autonomous urban vehicle. *Journal of Field Robotics*, 25, 2008. URL https://api.semanticscholar.org/CorpusID:1906145.

Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and Jose M Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14864–14873, 2024.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

Cheng Lu, Huayu Chen, Jianfei Chen, Hang Su, Chongxuan Li, and Jun Zhu. Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. *arXiv preprint arXiv:2304.12824*, 2023.

Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14297–14306, 2023.

Khan Muhammad, Amin Ullah, Jaime Lloret, Javier Del Ser, and Victor Hugo C de Albuquerque. Deep learning for safe autonomous driving: Current challenges and future directions. *IEEE Transactions on Intelligent Transportation Systems*, 22(7):4316–4336, 2020.

Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2980–2987. IEEE, 2023.

Jiquan Ngiam, Benjamin Caine, Vijay Vasudevan, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, et al. Scene transformer: A unified architecture for predicting multiple agent trajectories. *arXiv preprint arXiv:2106.08417*, 2021.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

Oliver Scheel, Luca Bergamini, Maciej Wołczyk, Błażej Osiński, and Peter Ondruska. Urban driver: Learning to drive from real-world demonstrations using policy gradients, 2021.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Qiao Sun, Shiduo Zhang, Danjiao Ma, Jingzhe Shi, Derun Li, Simian Luo, Yu Wang, Ningyi Xu, Guangzhi Cao, and Hang Zhao. Large trajectory models are scalable motion predictors and planners. *arXiv preprint arXiv:2310.19620*, 2023.

Qiao Sun, Huimin Wang, Jiahao Zhan, Fan Nie, Xin Wen, Leimeng Xu, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. Generalizing motion planners with mixture of experts for autonomous driving. *arXiv preprint arXiv:2410.15774*, 2024.

Ardi Tampuu, Tambet Matiisen, Maksym Semikin, Dmytro Fishman, and Naveed Muhammad. A survey of end-to-end driving: Architectures and training methods. *IEEE Transactions on Neural Networks and Learning Systems*, 33(4):1364–1384, 2020.

Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2):1805–1824, August 2000a. ISSN 1095-3787. doi: 10.1103/physreve.62.1805. URL http://dx.doi.org/10.1103/PhysRevE.62.1805.

Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805, 2000b.

Chris Urmson, Joshua Anhalt, J. Andrew Bagnell, Christopher R. Baker, Robert Bittner, M. N. Clark, John M. Dolan, David Duggins, Tugrul Galatali, Christopher Geyer, Michele Gittleman, Sam Harbaugh, Martial Hebert, Thomas M. Howard, Sascha Kolski, Alonzo Kelly, Maxim Likhachev, Matthew McNaughton, Nick Miller, Kevin M. Peterson, Brian Pilnick, Ragunathan Raj Rajkumar, Paul E. Rybski, Bryan Salesky, Young-Woo Seo, Sanjiv Singh, Jarrod M. Snider, Anthony Stentz, William Whittaker, Ziv Wolkowicki, Jason Ziglar, Hong Bae, Thomas Brown, Daniel Demitrish, Bakhtiar Litkouhi, James N. Nickolaou, Varsha Sadekar, Wende Zhang, Joshua Struble, Michael Taylor, Michael Darms, and Dave Ferguson. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25, 2008. URL https://api.semanticscholar.org/CorpusID:11849332.

Matt Vitelli, Yan Chang, Yawei Ye, Ana Ferreira, Maciej Wołczyk, Błażej Osiński, Moritz Niendorf, Hugo Grimmett, Qiangui Huang, Ashesh Jain, et al. Safetynet: Safe planning for real-world self-driving vehicles using machine-learned policies. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 897–904. IEEE, 2022.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36, 2024.

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. *arXiv preprint arXiv:2402.06559*, 2024.

Yinan Zheng, Jianxiong Li, Dongjie Yu, Yujie Yang, Shengbo Eben Li, Xianyuan Zhan, and Jingjing Liu. Safe offline reinforcement learning with feasibility-guided diffusion model. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=j5JvZCaDM0.

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 *Conference on Robot Learning*, pp. 144–177. PMLR, 2023a.

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 *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3560–3566. IEEE, 2023b.

Zikang Zhou, Jianping Wang, Yung-Hui Li, and Yu-Kai Huang. Query-centric trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17863–17873, 2023.

A VISUALIZATION OF CLOSED-LOOP PLANNING RESULTS

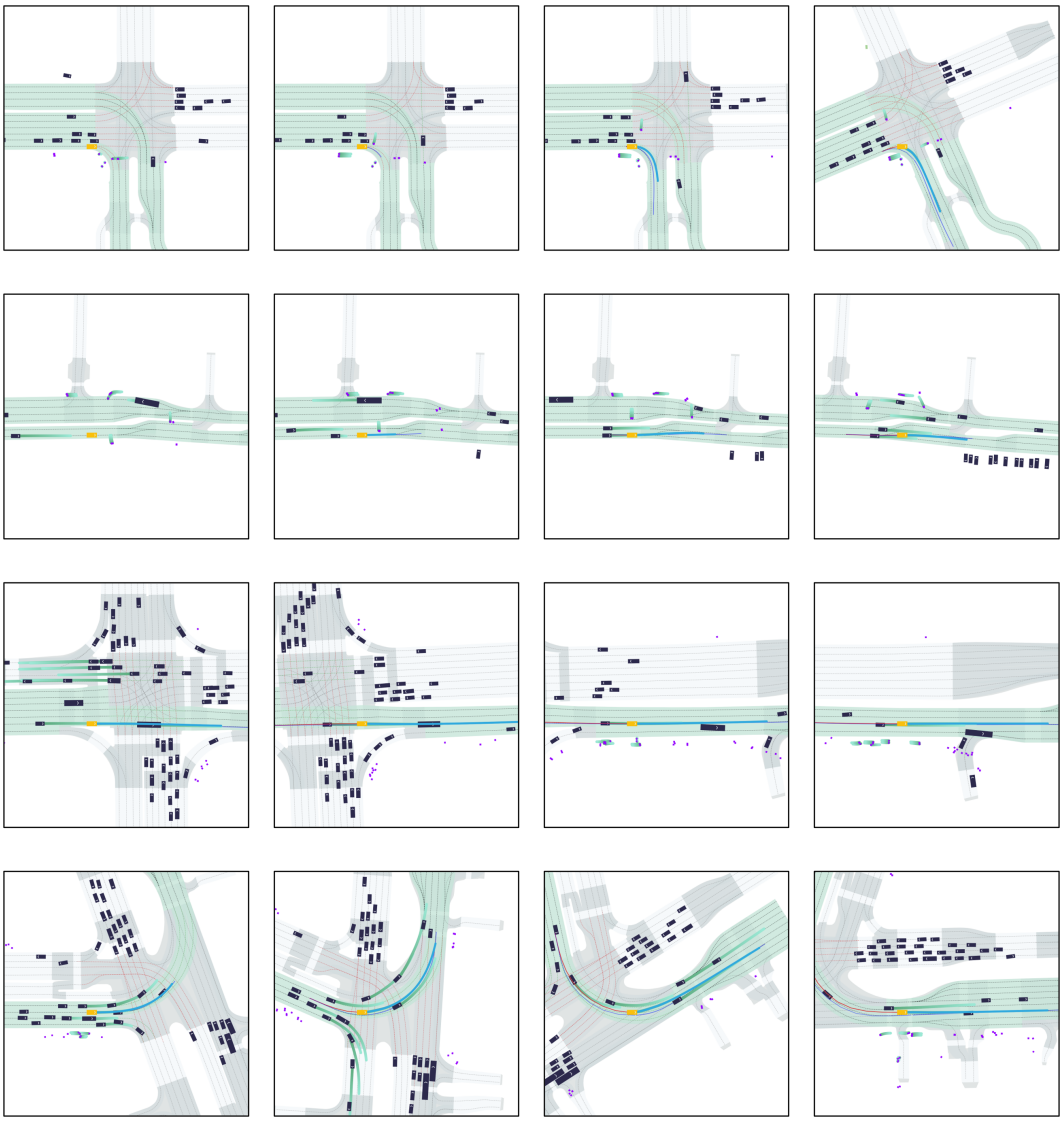

Figure 8: Closed-loop planning results: each row represents a scenario at 0, 5, 10, and 15 seconds intervals. Each frame includes the future planning of the ego vehicle, predictions for neighboring vehicles, the ground truth ego trajectory, and the driving history of the ego vehicle.

B ADDITIONAL RESULTS

We report the Val14 and Test14-Hard closed-loop non-reactive and reactive results, including reactive results for Test14, for all baselines, as shown in Table 4. Additionally, using an existing refinement module (Sun et al., 2024), which applies offsets to the model outputs and scores all trajectories (Dauner et al., 2023b), without any parameter tuning, we incorporate this as post-processing for the diffusion planner (Diffusion Planner w/ refine.). To further demonstrate the advantages of our model, we compared it with two recent works using diffusion models for motion planning. Diffusion-es (Yang et al., 2024) enhances a diffusion model by incorporating an LLM as a trajectory filter. STR-16M (Sun et al., 2023) uses a diffusion model as a decoder. STR2-CPKS-800M (Sun et al., 2024) builds on the former with 800M parameters and includes a PDM-like refinement module. We compared the model's performance in non-reactive mode and recorded the inference time, as shown in Table 5.

Table 4: More closed-loop planning results on nuPlan dataset. : The highest scores of baselines in various types. *: Using pre-searched reference lines as model input provides prior knowledge, reducing the difficulty of planning compared to standard learning-based methods. NR: non-reactive mode. R: reactive mode.

Type	Planner	Val14		Test14-hard		Test14
		NR	R	NR	R	R
Expert	Log-replay	93.53	80.32	85.96	68.80	75.86
Rule-based & Hybrid	IDM	75.60	77.33	56.15	62.26	74.42
	PDM-Closed	92.84	92.12	65.08	75.19	91.63
	PDM-Hybrid	92.77	92.11	65.99	76.07	91.28
	GameFormer	79.94	79.78	68.70	67.05	82.05
	PLUTO	92.88	76.88	80.08	76.88	90.29
	Diffusion Planner w/ refine. (Ours)	94.26	92.90	78.87	82.00	91.75
Learning-based	PDM-Open*	53.53	54.24	33.51	35.83	57.23
	UrbanDriver	68.57	64.11	50.40	49.95	67.15
	GameFormer w/o refine.	13.32	8.69	7.08	6.69	9.31
	PlanTF	84.27	76.95	69.70	61.61	79.58
	PLUTO w/o refine.*	88.89	78.11	70.03	59.74	78.62
	Diffusion Planner (Ours)	89.76	82.56	75.67	68.56	83.36

As shown in Table 4, we achieve state-of-the-art performance across more benchmarks compared to all learning-based baselines. Our method, with the addition of post-processing, achieved scores that other methods could not reach, even surpassing human performance. This is due to our model's ability to output high-quality trajectories, which are further enhanced by post-processing. It should also be noted that, many high-scoring methods experience significant score drops when refinement is removed, and their performance falls short of ours. This reflects their over-reliance on refinement. From Table 5, we observe that current diffusion-based methods also experience significant performance degradation when detached from LLMs or rule-based refinement. Another important point is that these methods, due to their reliance on LLMs or a large number of model parameters, have higher computational costs, making them difficult to deploy in real-world applications.

Table 5: Closed-loop non-reactive planning results on the nuPlan dataset among diffusion-based planners.

Planner	Test14	Test14-hard	Val14	Inference Time (s)
Diffusion-es w/o LLM	-	-	50	-
Diffusion-es w/ LLM	-	-	92	0.5
STR-16M	-	27.59	45.06	-
STR2-CPKS-800M w/o refine.	14.77	10.99	8.80	>11
Diffusion Planner (ours)	89.22	75.67	89.76	0.08

C Experimental Details

This section outlines the experimental details to reproduce the main results in our papers.

C.1 Training Details

Datasets. We use the training data from the nuPlan[1] dataset and randomly sample 1 million scenarios for our training set. To balance the number of scenarios per type, we set a maximum limit for each scenario type, as shown in Figure 10. For each scenario, we consider the lane and navigation information within a $100m$ radius around the ego vehicle at the current time, including the neighboring vehicles' history from the past two seconds. Each type of data is padded to a unified dimension for model input, and attention masking is used to effectively eliminate irrelevant information.

[1] https://www.nuplan.org/

Data augmentation. The current state of the ego vehicle is first perturbed slightly in terms of its x, y coordinates, orientation angle θ, speed v, acceleration a, and angular acceleration ω.

$$\Delta x^0 \sim \mathbb{U}\left([-\Delta x, -\Delta y, -\Delta\theta, -\Delta v, -\Delta a, -\Delta\omega], [\Delta x, \Delta y, \Delta\theta, \Delta v, \Delta a, \Delta\omega]\right),$$

where $[\Delta x, \Delta y, \Delta\theta, \Delta v, \Delta a, \Delta\omega] = [0.0, 0.75, 0.35, 1.0, 0.2, 0.85]$. For the augmented state $\tilde{x}^0_{ego} = x^0_{ego} + \Delta x^0$, we ensure that the speed v always remains greater than 0 to prevent the vehicle from learning to move in reverse. Additionally, numerical constraints are applied to ω, restricting it to a range of ± 0.85 to prevent overly aggressive steering. After that, a quintic polynomial interpolation is applied between current state $\tilde{x}^0_{ego}$ and $x^{\tau_{2s}}_{ego}$ to generate a new trajectory that adheres to the dynamic constraints, replacing the ground truth trajectory.

Normalization. Following previous works (Huang et al., 2023; Cheng et al., 2023; 2024), we apply an ego-centric transformation to process the original dataset. The global coordinates are converted into the ego vehicle's local coordinate system, using the vehicle's heading and position. Afterward, we observe that the ego vehicle's longitudinal progress is significantly larger than its lateral progress. To improve training stability, we apply z-score normalization to all x-axis coordinates, while the y-axis is scaled to the same magnitude to avoid distortion:

$$\tilde{x} = \frac{x - \mu}{\sigma}, \quad \tilde{y} = \frac{y}{\sigma},$$

where $\mu = 10$, $\sigma = 20$. The same approach is applied other scenario inputs.

Training was conducted using 8 NVIDIA A100 80GB GPUs, with a batch size of 2048 over 500 epochs, with a 5-epoch warmup phase. We use AdamW optimizer with a learning rate of $1e^{-3}$. We report the detailed setup in Table 6.

C.2 Inference Details

We utilize DPM-Solver++ as diffusion reverse process solver, adopting variance-preserving(VP) noise schedule where the noise is $\sigma_t = (1-t)\beta_{\min} + t\beta_{\max}$, and enabling *denoise to zero*. Low-temperature sampling is employed to further enhance the stability of the denoising process. We report the detailed setup in Table 6.

C.3 Classifier Guidance Details

We then specifically introduce the mathematical formulation of the different energy functions, as mentioned in Section 4.3.

Collision Avoidance. Based on the ego vehicle's planning and the neighboring vehicles' predictions from the decoder at diffusion timestamp t, we calculate the signed distance $\mathbf{D}$ between the ego vehicle and each neighboring vehicle at each timestamp τ. When the bounding boxes of the vehicles overlap, we use the minimum separation distance, otherwise, we use the distance between the nearest points. The energy function for collision avoidance is then defined as:

$$\begin{aligned}\mathcal{E}_{\text{collision}} =& \frac{1}{\omega_c} \cdot \frac{\sum_{M,\tau} \mathbb{1}_{\mathbf{D}^\tau_M > 0} \cdot \Psi\left(\omega_c \cdot \max\left(1 - \frac{\mathbf{D}^\tau_M}{r}, 0\right)\right)}{\sum_{M,\tau} \mathbb{1}_{\mathbf{D}^\tau_M > 0} + \text{eps}} \\ &+ \frac{1}{\omega_c} \cdot \frac{\sum_{M,\tau} \mathbb{1}_{\mathbf{D}^\tau_M < 0} \cdot \Psi\left(\omega_c \cdot \max\left(1 - \frac{\mathbf{D}^\tau_M}{r}, 0\right)\right)}{\sum_{M,\tau} \mathbb{1}_{\mathbf{D}^\tau_M < 0} + \text{eps}},\end{aligned} \tag{13}$$

where $\Psi(x) := e^x - x$, r represents the collision-sensitive distance, which controls the maximum distance at which gradients are produced, and eps is added to ensure numerical stability (Jiang et al., 2023).

Target Speed Maintenance. We calculate the energy function based on the difference between the average speed of the generated trajectory and the target speed range:

$$\mathcal{E}_{\text{target\_speed}} = \max\left(\overline{\frac{\mathrm{d}x^\tau_{\text{ego}}}{\mathrm{d}\tau}} - v_{\text{low}}, 0\right)^2 + \max\left(v_{\text{high}} - \overline{\frac{\mathrm{d}x^\tau_{\text{ego}}}{\mathrm{d}\tau}}, 0\right)^2. \tag{14}$$

Where v_{low} is the setting lower bound of speed, v_{low} is the setting higher bound of speed.

Comfort. Taking longitudinal jerk as an example, the difference between each point and the comfort threshold is calculated, ignoring cases where the comfort requirements are met:

$$\mathcal{E}_{\text{comfort}} = \mathbb{E}\left[\max\left(\left(j_{\max} - \left|\frac{\mathrm{d}^3 x^{\tau}_{\text{ego}}}{\mathrm{d}\tau^3}\right|\right)\Delta\tau^3, 0\right)^2\right]. \tag{15}$$

Where $j_{\max}$ is the maximum longitude jerk limit.

Staying within Drivable Area. We construct the differentiable cost map $\mathbf{M}$ by using Euclidean Signed Distance Field with parallel computation (Cheng et al., 2024), which can compute the distance the ego vehicle goes beyond the lane at each timestamp. Then the energy is defined as:

$$\mathcal{E}_{\text{drivable}} = \frac{1}{\omega_{\text{d}}} \cdot \frac{\sum_{\tau} \Psi\left(\omega_{\text{d}} \cdot \mathbf{M}\left(x^{\tau}_{\text{ego}}\right)\right)}{\sum_{\tau} \mathbb{1}_{\mathbf{M}(x^{\tau}_{\text{ego}})>0} + \text{eps}}. \tag{16}$$

As illustrated in Figure 2, we only employ classifier guidance when the diffusion timesteps $t < 0.1$.

Given the diverse options for energy function design, our choices were made primarily to validate whether the model could support various types of guidance and may not be optimal. However, through extensive empirical experiments, we can share some of our insights and experiences regarding energy function selection to assist future work in exploring more effective options:

- **Smooth and continuous gradients**: Guidance functions with smooth and continuous gradients facilitate the generation of stable trajectories.
- **Gradient sparsity**: It is preferable for the guidance function to generate gradients only in specific situations, such as when trajectory points approach potential collisions.
- **Indirect guidance for higher-order state derivatives**: For higher-order state derivatives, such as velocity, acceleration, or angular velocity, indirect guidance through position and heading is preferable. For instance, to control trajectory speed, we can guide trajectory length instead.
- **Consistent gradient magnitude**: The guidance function should ensure that the magnitude of gradients remains approximately consistent across different conditions. It can be achieved by averaging cost values over the number of points contributing to the cost.

Table 6: Hyperparameters of *Diffusion Planner*

Type	Parameter	Symbol	Value
Training	Num. neighboring vehicles	/	32
	Num. past timestamps	L	21
	Dim. neighboring vehicles	D_{neighbor}	11
	Num. lanes	/	70
	Num. points per polyline	P	20
	Dim. lanes vehicles	D_{lane}	12
	Num. navigation lanes	D	25
	Num. predicted neighboring vehicles	M	10
	Num. encoder/decoder block	/	3
	Dim. hidden layer	/	192
	Num. multi-head	/	6
Inference	Noise schedule	/	Linear
	Noise coefficient	$\beta_{\min}, \beta_{\max}$	0.1, 20.0
	Temperature	/	0.5
	Denoise step	/	25

C.4 BASELINES SETUP

nuPlan Datasets Evaluation. For *IDM* and *UrbanDriver*, we use the official nuPlan code[2], with the *UrbanDriver* checkpoint sourced from the *PDM* codebase[3], which also provides the checkpoints for *PDM-Hybrid* and *PDM-Open*. For *PlanTF* and *PLUTO*, we use the checkpoints from their respective official codebases[4][5]. In the case of *PLUTO w/o refine*, we skip the post-processing code and rerun the simulation without retraining. Following the guidelines from the official codebase[6], we train *GameFormer* and skip the refinement step to obtain *GameFormer w/o refine*.

Delivery-vehicle Datasets Evaluation. We adopt the same testing methods and models as those used on nuPlan, but by modifying various vehicle-related parameters to adapt the baselines to the delivery-vehicle training. Based on this, we retrain and test the models following the official training code. Due to the high transfer cost, however, we did not conduct tests for *PLUTO*.

D DETAILS ON DELIVERY VEHICLE EXPERIMENTS

We collected approximately 200 hours of real-world data using an autonomous logistics delivery vehicle. The task of the delivery vehicle is similar to that of a robotaxi in nuPlan, as it autonomously navigates a designated route. During operation, the vehicle must comply with traffic regulations, ensure safety, and complete the delivery as efficiently as possible. Compared to the vehicles in the nuPlan dataset, the delivery vehicle is smaller, as shown in Table 7, and operates at lower speeds. As a result, it is able to travel on both main roads and bike lanes. During deliveries, it frequently interacts with pedestrians and cyclists, and the driving rules differ from those for motor vehicles, as shown in 9. This dataset serves as a supplement to nuPlan, allowing for the evaluation of algorithm performance under diverse driving scenarios.

Table 7: Vehicle parameter details

Parameter (m)	Delivery Vehicle	nuPlan Vehicle
Width	1.03	2.30
Length	2.34	5.18
Height	1.65	1.78
Wheel base	1.20	3.09

Specifically, we transform the original data into the nuPlan data structure, allowing it to be stored as DB files compatible with the nuPlan API for seamless integration and usage. We use the same training pipeline from the nuPlan benchmark to train both the model and baselines. For some baselines that require crosswalk information, we replace it with stop line data. Additionally, the vehicle parameters are substituted with those of the delivery vehicle. The model's performance is evaluated using the nuPlan metrics.

E LIMITATIONS & DISCUSSIONS & FUTURE WORK

Here, we discuss our limitations, potential solutions and interesting future works.

- **Training Time**. In the current version, we simply stack transformer blocks in the encoder part. However, in more complex scenarios where additional neighboring vehicles and lane information need to be considered, the increase in token count results in a significant computational burden. Furthermore, the diffusion model itself already contributes to considerable training time.

  Solution and future work: This problem can be addressed through proper architectural design, incorporating more inductive bias to pre-process scenario inputs, or utilizing advanced attention

[2] https://github.com/motional/nuplan-devkit
[3] https://github.com/autonomousvision/tuplan_garage
[4] https://github.com/jchengai/planTF
[5] https://github.com/jchengai/pluto
[6] https://github.com/MCZhi/GameFormer-Planner

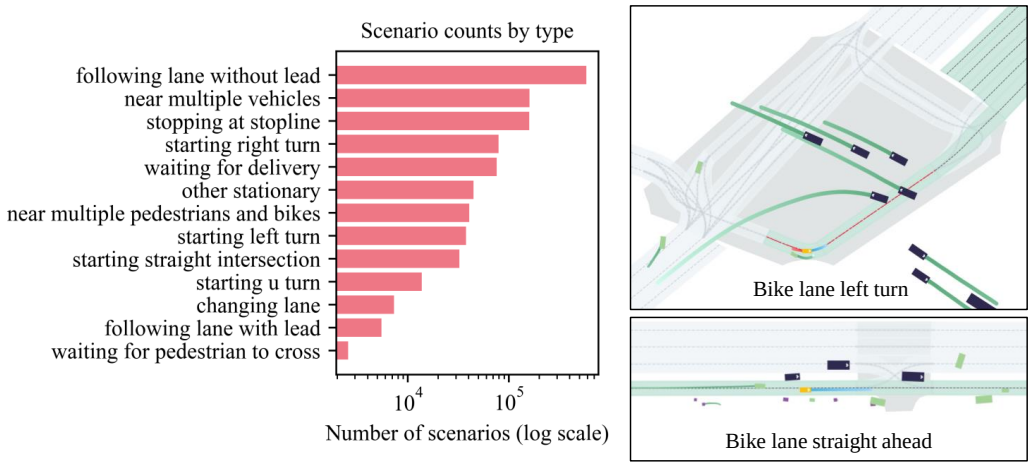

Figure 9: Scenario count by type in the delivery-vehicle driving dataset, with representative visualizations.

mechanism such as Flash Attention (Dao et al., 2022) to reduce the computational complexity. In this paper, we primarily focus on exploring the effectiveness of diffusion planning without introducing too much specific designs for the model architecture. We believe that a more training-efficient architecture is necessary for practical implementation.

- **Scenario Inputs**. Our method relies on vectorized map information and detection results of neighboring vehicles. Compared to mainstream end-to-end pipelines, this approach involves some information loss and requires a data processing module. However, unlike end-to-end methods, our focus is more on the planning stage, particularly on the ability for closed-loop planning.

  Solution and future work: We demonstrate the performance of the diffusion model for closed-loop planning without rule-based refinement. An interesting future direction would be to modify the encoder architecture and use images as inputs, enabling an end-to-end training pipeline.

- **Lateral Flexibility**. We find that learning-based methods struggle with flexibility, particularly when significant lateral movement is required. In contrast, rule-based methods perform better in this aspect due to the provision of a reference trajectory. Being consistent with findings from previous work (Li et al., 2024), we find this is mostly because that the dataset mainly consists of straight-driving scenarios, with few instances of lane changes or avoidance maneuvers. This makes it challenging for learning-based methods to generalize and acquire these skills. Additionally, since the model only outputs the planned trajectories instead of the controlling signal such as brake and throttle, there is a gap between the planned trajectory and the results from the downstream controller (Cheng et al., 2023). This discrepancy also leads to potential poor performance, or even out-of-distribution behavior, in scenarios that require more flexible actions.

  Solution and future work: We find that data augmentation can somewhat alleviate the issue of the vehicle being reluctant to make lateral movements, but it still performs poorly in cases requiring significant lane changes. This could be improved by incorporating more data involving large lateral progress, leveraging reinforcement learning with a reward mechanism, or designing a more effective diffusion guidance mechanism to help the model learn lane-changing behaviors. We believe this is an interesting observation and leave this direction for future work.

- **Sample Efficiency**. The high performance of Diffusion comes at the cost of requiring multiple model inferences, leading to reduced sample efficiency.

  Solution and future work: We addressed this issue to a large extent by using a high-order ODE solver, enabling trajectory planning for 8 seconds at 10 Hz in 0.08 seconds. Considering real-world application requirements, techniques such as consistency models (Song et al., 2023) or distillation-based sampling methods (Meng et al., 2023) could be employed for further acceleration.

Overall, although some design choices may appear simple and certain limitations exist, we have thoroughly demonstrated the capabilities of diffusion models for closed-loop planning in autonomous driving through extensive experiments. Moreover, we demonstrate the potential of the diffusion model to align with safety or human-preferred driving behaviors. It provides a high-performance, highly adaptable planner for autonomous driving systems.

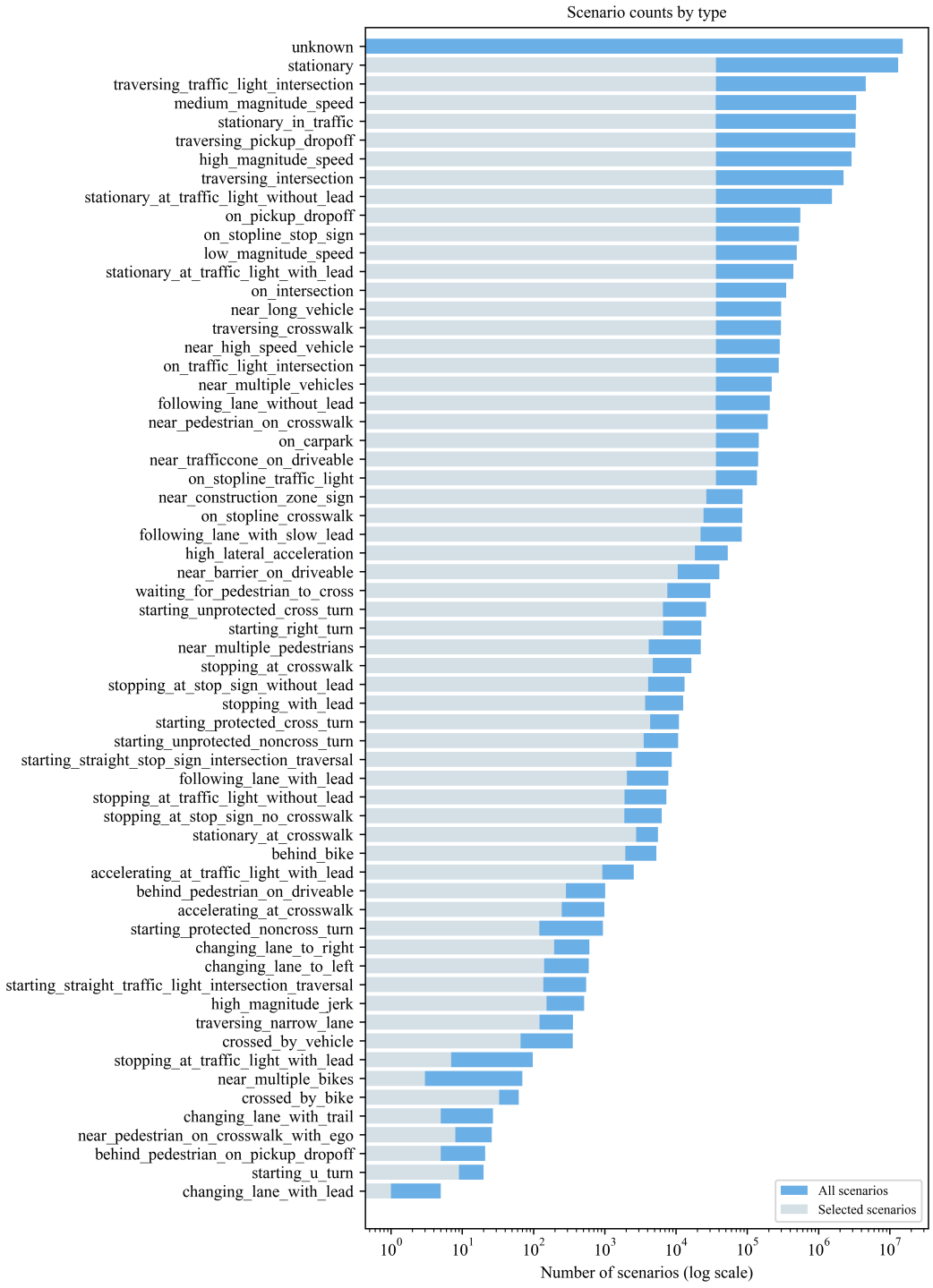

Figure 10: Scenario count by type in the nuPlan dataset.

F A CASE STUDY ON COLLISION AND DRIVABLE AREA GUIDANCE.

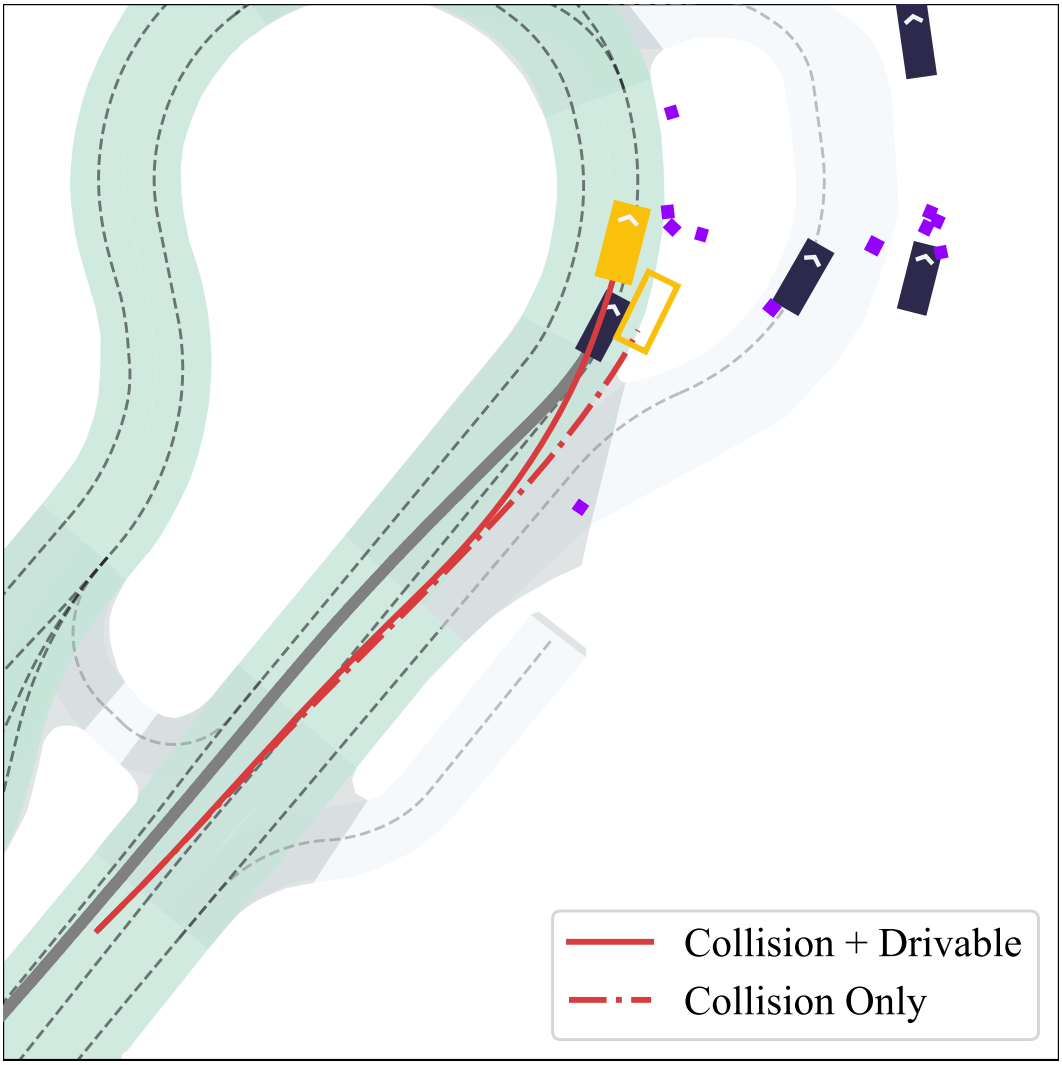

Figure 11: Under collision guidance alone, the ego vehicle veers off the road to avoid a rear-approaching vehicle. When drivable guidance is added, the vehicle stays on the road while maintaining safety.