

ConsistencyPlanner: Real-time Planning with Fast-Sampling Consistency Models

Qichao Zhang^{1,2}, Xing Fang¹, Jiaqi Fang³, Zhenwen Cai³, Jie Ling³, Qiankun Yu³, and Dongbin Zhao^{1,2}✉

¹State Key Laboratory of Multimodal Artificial Intelligence Systems,

Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

²School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China

³Guangzhou Zaofu Intelligent Technology Co., Ltd.

✉ Corresponding Author

Abstract—Closed-loop planning in complex, real-world driving scenarios presents a critical challenge for autonomous driving systems. While traditional rule-based methods are interpretable, their predefined heuristics lack the adaptability for dynamic traffic environments. Learning-based approaches have shown considerable promise. Conversely, learning-based approaches, despite their promise, struggle to balance the modeling diverse and multimodal driving behaviors and real-time planning, often leading to indecisive or unsafe actions. To address this limitation, we propose *ConsistencyPlanner*, a real-time planning framework with fast-sampling consistency models. Our approach is built upon two key technical contributions. *Efficient Multimodal Sampling*: We employ fast-sampling consistency models to generate a diverse set of plausible future trajectories. This enables efficient, real-time exploration of multimodal actions, overcoming the computational bottlenecks of previous iterative generative methods. *Heterogeneous Feature Fusion*: We introduce an attention-enhanced decoder that dynamically integrates heterogeneous input features—including scene feature and action token—into a cohesive representation for robust planning. Extensive evaluation in the Waymax simulator demonstrates superior performance in safety metrics compared to existing methods, with particularly strong results in challenging dynamic scenarios.

Index Terms—Autonomous Driving, Consistency Models, Diffusion Models, Imitation Learning

I. INTRODUCTION

The pursuit of safe and efficient autonomous driving has precipitated a paradigm shift from open-loop prediction [1] to closed-loop planning [2], [3]. This transition has been further accelerated by the emergence of closed-loop planning benchmarks tailored for real-world driving scenarios, such as Waymax [4], which have refocused the field from isolated prediction tasks to the more complex, interactive challenges inherent in closed-loop control. Within this context, current planning approaches can be broadly categorized into rule-based and learning-based methods. Rule-based systems rely on carefully engineered heuristics to govern vehicle behavior, offering strong controllability and interpretability in decision-making, as exemplified by platforms like Apollo [5]. However, their dependence on predefined logic often hampers adaptability in novel or dynamically complex urban environments, where unexpected situations may fall outside the scope of handcrafted rules. In contrast, learning-based methods, including models such as GameFormer [6], learn driving

policies directly from large-scale human demonstrations. By leveraging increasingly extensive datasets and powerful neural architectures like Transformers [7], these methods demonstrate enhanced generalization to diverse and unstructured real-world conditions.

While learning-based approaches have demonstrated considerable promise in autonomous driving [8], [9], they still face critical challenges in real-world deployment. A key limitation is their inability to capture the inherent multimodality and uncertainty of complex driving scenarios [1], [10]. Although behavior cloning techniques [11] can approximate expert demonstrations, they offer no theoretical guarantee of accurately representing the diversity of human decision-making, even when supported by large-scale transformer architectures [7]. Recent advances have explored generative models to improve multimodal expressiveness. For example, LatentDriver [12] explores the feasibility of deriving decisions from an autoregressive world model through the formulation of multiple probabilistic hypotheses on Waymax. Diffusion Planner [13] employs the diffusion model as the policy and achieve superior performances on the NuPlan Benchmark [14]. While these models demonstrate stronger capability in generating diverse trajectories, their substantial computational overhead during inference poses a serious barrier to real-time deployment. The resulting latency is incompatible with the low-latency requirements of practical autonomous driving systems.

Building upon recent advances in generative modeling, we identify consistency models [15] as a particularly promising solution to the dual challenges of multi-modal behavior representation and real-time inference in autonomous driving. These models combine the distribution modeling capabilities of diffusion processes [16], [17] with efficient single-step sampling, making them uniquely suited for capturing the diverse decision space of human drivers while meeting stringent latency requirements. We present *ConsistencyPlanner*, the framework to successfully adapt fast-sampling consistency models for closed-loop autonomous planning with real-world driving dataset. Specifically, we propose a new network architecture built upon the diffusion transformer [18] to fuse different input features. Evaluation results on the large-scale real-world autonomous planning benchmark Waymax [4]

demonstrate that *ConsistencyPlanner* achieves superior closed-loop performance among learning-based baselines.

In this paper, we demonstrate that, through appropriate architectural design, the potential of consistency models can be fully leveraged to enhance closed-loop planning performance in autonomous driving. The main contributions of this work are summarized as follows:

- To balance multimodal planning with real-time inference, we integrate consistency models into our closed-loop planning framework. Driven by real-world scenarios, these models provide fast sampling capabilities that accurately capture the multimodal characteristic of driving behaviors.
- We incorporate an attention mechanism into the consistency model decoder to achieve thorough fusion of input conditions and action information.
- Experimental results on the Waymax simulator validate the effectiveness of our method and demonstrate superior performance, particularly on safety metrics.

II. RELATED WORK

A. Diffusion Models and Consistency Models

Diffusion models have emerged as the predominant paradigm for generative modeling, which operate through an iterative denoising process that progressively refines Gaussian noise into high-quality samples. The foundational DDPM framework [19] establish this paradigm, with subsequent improvements in sampling efficiency [20] and quality control via classifier-free guidance [21]. The field has undergone a paradigm shift with the advent of consistency models [15]. These models distill the iterative diffusion process into a single generation step via learned consistency mappings, thereby preserving the sample quality of traditional diffusion while enabling orders-of-magnitude faster inference. This breakthrough is particularly transformative for real-time applications. Subsequent advancements in training methodologies [22] have further solidified their efficacy, leading to successful deployment in diverse domains such as video [23], 3D motion generation [24], and robotic tasks [17], [25], where computational efficiency is paramount.

B. Diffusion Model for Autonomous Driving

Diffusion models [19] have emerged as a powerful class of generative frameworks that synthesize data through a learned denoising process. Their ability to model complex, multimodal data distributions has made them particularly attractive for robot tasks [26]. In recent years, these models have gained increasing traction in the field of autonomous driving [27], where tasks such as trajectory forecasting and traffic scene generation inherently involve uncertainty and multimodality. For instance, Jiang et al. [28] apply diffusion-based techniques to forecast future trajectories of surrounding agents, including vehicles and pedestrians, effectively capturing the diverse range of plausible future behaviors. Similarly, works such as [29], [30] extend diffusion models to traffic simulation,

demonstrating their capability to produce realistic and controllable environments for downstream testing and training. Beyond perception and simulation, diffusion models are also being explored for decision-making and planning. Yang et al. [31] propose a diffusion-based framework that generates a set of candidate trajectories, which are subsequently scored by pre-trained evaluators [2] to select the final motion plan. Notably, Zheng et al. [13] introduce a fully closed-loop planning system based on diffusion models, showing competitive or superior performance compared to traditional rule-based approaches, even without post refinement. Despite their promising capabilities, a major limitation of diffusion models is their reliance on iterative sampling procedures involving hundreds of denoising steps. This requirement significantly limits their suitability for real-time deployment in safety-critical systems like autonomous vehicles, where low-latency planning is essential.

III. METHOD

A. Preliminary

Consistency models provide an efficient alternative to traditional diffusion models by learning a time-consistent mapping from noisy inputs to clean data. Unlike diffusion models that require iterative denoising, consistency models enable one-step generation by enforcing consistency across different time steps of the forward diffusion process. Consider the stochastic differential equation (SDE) that defines the forward process:

$$dx^t = \mu(x^t, t)dt + \sigma(t)dw^t, \quad (1)$$

where $x^0 \sim p_{data}(x)$ is a clean sample and $x^T \sim \mathcal{N}(\mathbf{0}, T^2\mathbf{I})$ denotes the final noisy data.

A consistency model learns a function $F_\theta(x^t, t)$ satisfying the *consistency condition*: $F_\theta(x^t, t) \equiv x^0, \forall t \in [0, T]$, ensuring that all time steps along a diffusion path map to the same clean sample. To train F_θ , the model minimizes the consistency loss:

$$\mathcal{L}_{CT}(\theta) = \mathbb{E}_{x^0, t, t'} \left[d(F_\theta(x^t, t), F_\theta(x^{t'}, t')) \right], \quad (2)$$

where x^t and $x^{t'}$ are generated from the same x^0 at different time steps, d represents the loss function. At inference, one-step sampling is performed by evaluating: $\hat{x}^0 \sim F_\theta(x^T, T)$, offering a fast and practical generation strategy without iterative denoising.

B. ConsistencyPlanner

1) *Overview*: As illustrated in Figure 1, the ConsistencyPlanner framework consists of two key components: Scene feature encoder, and Consistency Decoder. For scene feature encoding, we first extract raw driving information from the WOMO dataset, including road features, route features, ego vehicle state, and surrounding vehicle states. However, such a comprehensive scene representation inherently contains redundancies. To mitigate this issue, we perform scene tokenization in the ego vehicle’s coordinate system, thereby obtaining a more condensed set of scene features. These tokenized

features are subsequently processed by a BERT encoder [32] to generate compact and unified representations for the decoder.

The consistency decoder leverages consistency models to enable multimodal representation learning. Drawing inspiration from Diffusion Planner [13], we highlight the pivotal role of route features in autonomous driving policy formulation. Specifically, route features are integrated with consistency timesteps to construct fused conditional inputs for the decoder. This unified framework facilitates expert-level motion planning generation through iterative denoising.

2) *Scene Feature Encoder*: At each timestep, we perform tokenization of the raw observations within the ego vehicle’s coordinate frame, considering a Field of View (FOV) with predefined dimensions $w_f \times h_f$ as specified in [33]. This processing yields two distinct element categories: static elements (road maps and navigation trajectories) and dynamic elements (vehicular agents). For static elements, we employ the Ramer-Douglas-Peucker algorithm [34], [35] to approximate dense coordinate sequences as parameterized rectangular regions $[x, y, w, h, \psi, \text{id}]$, where road features $S_{rd} \in \mathbb{R}^{N_{rd} \times 6}$ use a fixed 0.5m width while route features $S_{rt} \in \mathbb{R}^{N_{rt} \times 6}$ adopt the ego vehicle’s width. Dynamic elements consist of neighboring agents represented as $S_{neighbor} \in \mathbb{R}^{N_{neighbor} \times 6}$ (with velocity replacing the id field) and the ego vehicle state $S_{ego} \in \mathbb{R}^{1 \times 6}$. The complete observation representation $O \in \mathbb{R}^{(N_{rd} + N_{rt} + N_{neighbor} + 1) \times 6}$ integrates all features within the perceptual bounds, augmented with type indicators for distinction.

To integrate object-centered vectorized representations from the scene, we employ a BERT-based encoder [32]. The encoder processes the observation tensor O_t along with its associated class tokens, generating a compact scene embedding $\text{scene}_{\text{emb}} \in \mathbb{R}^d$ that encapsulates both the static and dynamic state representations.

3) *Consistency Decoder*: Route information plays a crucial role in guiding the driving policy. Following the EasyChaufeur [33], we extract route features $S_{\text{route}} \in \mathbb{R}^{20 \times D_{\text{pos}}}$ by downsampling the logged trajectories, where D_{pos} encodes the (x, y, ψ) coordinates. These features are distinct from the encoded representation S_{rt} . The route features are further processed using stacked MLP-Mixer layers [36], which perform bidirectional mixing across both the vector and feature dimensions:

$$S_{\text{route}} = S_{\text{route}} + \text{MLP}(S_{\text{route}}^T)^T, \quad S_{\text{route}} = S_{\text{route}} + \text{MLP}(S_{\text{route}}) \quad (3)$$

The consistency time step encoding module generates t_{emb} , and the route information encoding module produces S_{route} . These features, now in the same latent space dimension, are combined through addition to form the fused condition $\text{cond} = t_{\text{emb}} + S_{\text{route}}$. For the noisy action input a_t , it is encoded by an action information encoding module to obtain the latent action tokens a_{emb} . Finally, the scene encoding features $\text{scene}_{\text{emb}}$, obtained from the scene encoder, are incorporated. All components are integrated into the Transformer block for different feature processing. We employ Adaptive Layer

Normalization with Zero Initialization (AdaLN-Zero) [18] to achieve conditional feature fusion. The conditioned action embedding a_{emb} is then combined with scene features through multi-head cross-attention:

$$a_{\text{emb}} = \text{CrossAttn}(\text{query} = a_{\text{emb}}, \text{key} = \text{value} = \text{scene}_{\text{emb}}) \quad (4)$$

The final executable action is generated through conditioned feature projection, where the fusion condition modulates both scaling and shifting operations:

$$\beta, \gamma = \text{MLP}(\text{cond}) \quad a_{\text{final}} = \text{FFN}(\beta \text{LayerNorm}(a_{\text{emb}}) + \gamma) \quad (5)$$

4) *Training Loss*: We present a novel integration of consistency models [15] into closed-loop autonomous driving systems. The proposed consistency decoder architecture is formally defined as:

$$\pi_{\theta}(a|s) \triangleq f_{\theta}(a^t, t, s) = c_{\text{skip}}(t)a^t + c_{\text{out}}(t)F_{\theta}(a^t, t|s) \quad (6)$$

where t denotes the discrete timestep related to ordinary differential equations, $a^t \sim \mathcal{N}(\mathbf{0}, t^2 \mathbf{I})$ represents the noisy action sampled from a Gaussian distribution with zero mean and covariance matrix $t^2 \mathbf{I}$, and the corresponding state s in this chapter includes: S_{route} and $\text{scene}_{\text{emb}}$. $F_{\theta}(a^t, t|s)$ is the neural network model to be trained, which outputs a conditional action with the same dimensionality as a^t based on the current input state information s . $c_{\text{skip}}(t)$ and $c_{\text{out}}(t)$ are defined differentiable functions.

The specific loss function for the consistency decoder is formulated as:

$$L(\theta) = \mathbb{E}_{(s,a) \sim \mathcal{D}, z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} [d(f_{\theta}(a + t_{m+1}z, t_{m+1}|s), a)] \quad (7)$$

where the loss function d employs the L_1 norm, and t_{m+1} is obtained by discretizing the time interval $[\epsilon, T]$ using the Karra method [37]. Here, ϵ is a small constant close to zero to avoid numerical instability, and T represents the total timesteps of the consistency models. Finally, we use $a_{\text{final}} = \hat{a}_{\epsilon} \sim \pi_{\theta}(a|s)$ as the executed action.

IV. EXPERIMENTAL RESULTS

A. Experimental Dataset

All experiments are carried out using Waymax [4], a recently introduced simulation platform based on the Waymo Open Motion Dataset (WOMD v1.1.0) [38]. Each scenario spans 8 seconds and is sampled at 10 Hz, resulting in a total of 80 frames per scenario. Agent control also operates at 10 Hz, supporting up to 128 agents simultaneously within a single scene. The training dataset consists of 487,002 scenarios, while 44,096 scenarios are reserved for validation. Simulations proceed for the full 8-second duration without early termination, ensuring consistent evaluation across all runs.

B. Evaluation Metrics

All experiments are conducted under a closed-loop evaluation protocol, following the setup proposed in [4]. The evaluation metrics include **Off-road Rate (OR)** and **Collision Rate (CR)**, both adopted directly from the original definitions in [4].

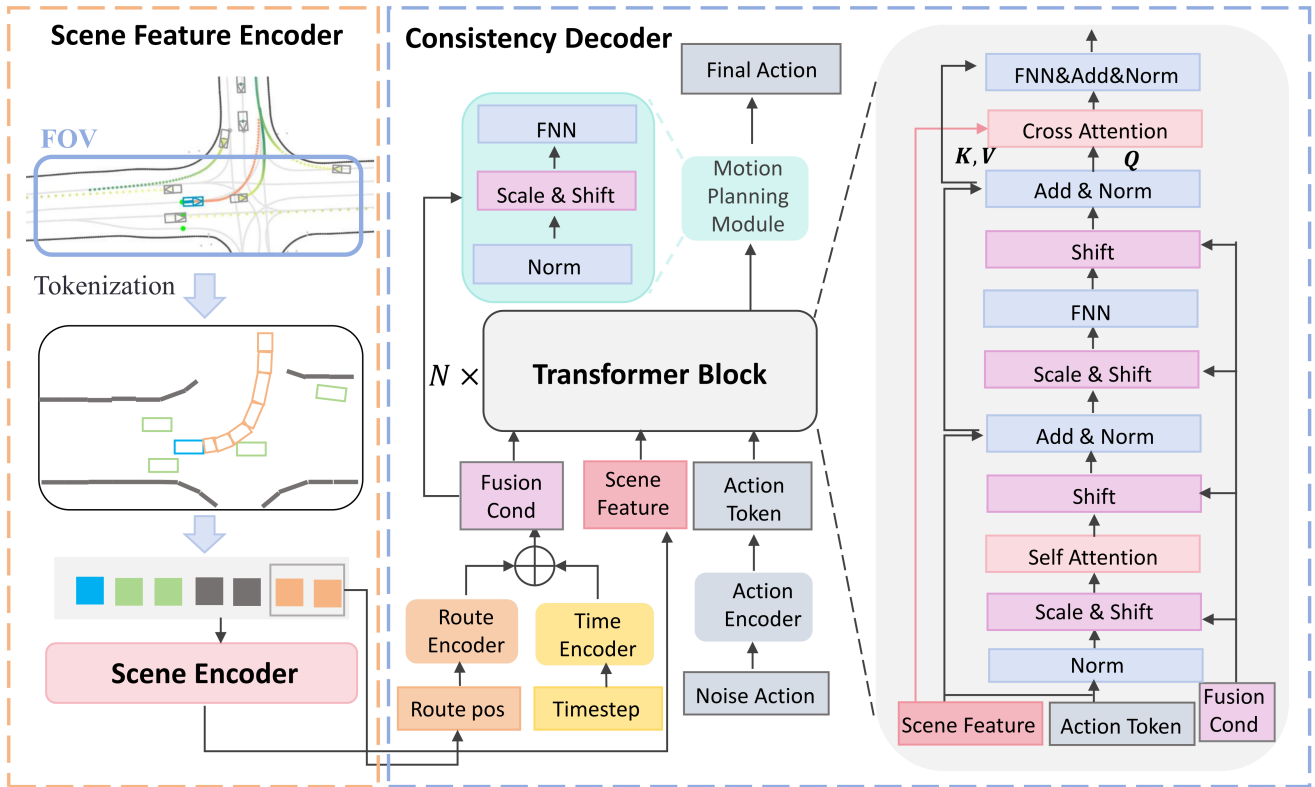


Fig. 1: Framework of the ConsistencyPlanner method with the scene feature encoder and consistency decoder. We incorporate an transformer block into the consistency decoder to achieve thorough fusion of input conditions and action information.

The **Progress Ratio (PR)** is measured in a manner consistent with prior work; however, in this study, the maximum value is capped at 100% due to the unavailability of future drivable area annotations in the WOMB dataset.

C. Implementation Details

For each frame of the dynamic driving scenario, our method discretizes the road information into $N_{rd} = 40$ road feature rectangles and extracts $N_{rt} = 20$ route information rectangles. The system considers $N_{vehicle} = 128$ dynamic entities comprising the autonomous vehicle and its neighboring dynamic obstacles, resulting in $N_{neighbor} = 127$ interaction targets.

In the network architecture design, the consistency model decoder employs $N_{block} = 3$ stacked layers with $h = 4$ attention heads, and the latent feature dimension is set to $D = 256$. For the time step encoding module, we use a frequency embedding space of dimension $d_{freq} = 256$ with a maximum time period $T_{max} = 10000$. The navigation information encoding module processes vector-dimensional features with 32 encoding dimensions and route features with 64 dimensions.

All training procedures were conducted on 8 NVIDIA V100 GPUs using the Adam optimizer with the following hyperparameters: 20 training epochs, initial learning rate of 2×10^{-4} with OneCycleLR scheduling policy. During training,

the sub-sequence $\{t_m | m \in [T]\}$ differs from inference and follows the Karras boundary schedule [37]:

$$t_m = \left(\epsilon^{1/\rho} + \frac{m-1}{T-1} (t_T^{1/\rho} - \epsilon^{1/\rho}) \right)^\rho,$$

where t_T denotes the terminal timestep, ϵ is a small constant for numerical stability, and ρ controls the curvature of the schedule. The consistency model operates with $t_T = 80$ total time steps, $\epsilon = 0.002$ as the minimum noise level, $T = 40$ and $\rho = 7$.

D. Performance Comparison

The comparative analysis with other methods on Waymax [4] is presented in Table I. Non-ego agents are controlled by the Intelligent Driver Model (IDM) [39], as implemented in previous works [4], [33]. The asterisk * on PR indicates that this metric cannot be fairly compared under some methods due to the absence of drivable future under ‘Route’. This discrepancy is highlighted in gray. The baselines evaluated include: (1) **Wayformer** [1], a Transformer-based model for joint trajectory prediction of ego and surrounding agents, and **Wayformer-BC**, which uses the same encoder with an MLP decoder for ego action prediction; (2) **EasyChauffeur-PPO** [33], which adopts the same encoder as ConsistencyPlanner, employs the reinforcement learning algorithm PPO to directly optimize control actions in a closed-loop setting,

TABLE I: Comparison with state-of-the-art methods. Non-ego agents are controlled by IDM [39]. ‘LT’ and ‘DF’ under **Route** indicate Logged Trajectory and Drivable Futures (unavailable), respectively.

Methods	Venue	Route	Action Space	OR↓	CR↓	PR*↑
Wayformer-BC [4]	NeurIPS 2023	LT+DF	Bicycle(Discrete)	1.11	4.59	129.84
DQN [4]		LT+DF		3.74	6.50	177.91
Waymax-BC [4]	Arxiv 2024	LT+DF	Bicycle	13.59±12.71	11.20±5.34	137.11±33.78
EasyChauffeur-PPO [33]		LT		3.95	4.72	98.26
Wayformer [1]	ICRA 2023	LT+DF	Waypoints	7.89	10.68	123.58
Waymax-BC [4]	NeurIPS 2023	LT+DF		4.14±2.04	5.83±1.09	79.58±24.98
PlanT [†] [40]	CoRL 2023	LT		2.29±0.11	3.08 ± 0.02	95.38±1.03
LatentDriver [12]	ICRA 2025	LT		2.33±0.13	3.17±0.04	99.57±0.1
ConsistencyPlanner (Ours)		LT		2.09	2.77	93.72

The dagger symbol (†) on PlanT [40] denotes results reproduced in our experiments using the officially released checkpoints.

enabling policy learning through interaction with the environment. (3) **PlanT** [40], which adopts the same encoder as ConsistencyPlanner but replaces the decoder with an MLP without consistency modeling; and (4) **LatentDriver** [12], which models the latent future dynamics as a mixture distribution with sampling-based decoding. The mean and standard deviation are reported under three random seeds.

The experimental results demonstrate that the Waymax closed-loop simulation platform supports performance comparisons of advanced baseline algorithms in both continuous and discrete action spaces. However, since the platform does not provide open-source configuration scripts for discrete action settings, this section focuses on evaluating the closed-loop motion planning performance of the proposed ConsistencyPlanner model under a continuous action space. As shown in Table I, the ConsistencyPlanner consistently outperforms all baseline methods in terms of OR and CR, despite achieving a slightly lower PR. Notably, it attains the lowest CR among all evaluated methods, and its OR is only marginally higher than that of the Wayformer-BC, which operates in a discrete action space.

E. Ablation Study

This subsection further analyzes the potential impact of model architecture choices and generative model categories on closed-loop decision-making and planning performance through ablation experiments.

1) *Impact of Consistency Decoder Architecture:* The consistency decoder must effectively integrate multiple input features—including scene encodings, action features, consistency timesteps, and navigation paths—to generate precise and context-aware driving decisions. Our framework primarily employs Transformer blocks, utilizing self-attention and cross-attention mechanisms to achieve hierarchical feature fusion. For comparison, we also evaluate a simpler Multi-Layer Perceptron (MLP)-based architecture, where features are concatenated into a latent vector and processed through fully connected layers.

As shown in Table II, the Transformer-based ConsistencyPlanner outperforms the MLP variant in closed-loop motion planning, achieving higher overlap rate (OR) and collision rate

(CR) metrics, albeit with a marginally lower progress rate (PR). These results suggest that the Transformer’s attention mechanisms enhance feature fusion efficacy, particularly in improving safety-critical metrics.

TABLE II: Impact of consistency decoder architecture on closed-loop motion planning performance

Architecture	Action Space	OR↓	CR↓	PR↑
MLP	Waypoints	2.39	2.82	94.99
Transformer Block	Waypoints	2.09	2.77	93.72

TABLE III: Impact of generative model selection on closed-loop motion planning performance

Method	OR↓	CR↓	PR↑	Inference (ms) ↓
ConsistencyPlanner	2.09	2.77	93.72	15
DiffusionPlanner	2.07	2.76	93.63	122

2) *Impact of Generative Models:* ConsistencyPlanner employs fast-sampling consistency models as its core framework and conducts a systematic comparison with an enhanced diffusion model accelerated via the DPM-Solver algorithm. The performance comparison between these two generative models in closed-loop settings is summarized in Table III.

In terms of experimental setup, the diffusion model leverages the DPM-Solver with 10 sampling steps, significantly improving inference efficiency while preserving generation quality. In contrast, the consistency model enables rapid prediction through single-step inference. The results highlight the consistency model’s remarkable advantage in computational efficiency, reducing inference time to approximately 10% of that required by the diffusion-based approach—clearly demonstrating the benefits of single-step sampling. With respect to safety metrics, the diffusion model exhibits slightly superior performance in indicators such as collision rate, largely attributable to its iterative refinement process, which facilitates a more thorough exploration of multimodal decision spaces. Nevertheless, the consistency model’s exceptional inference speed makes it more suitable for real-time autonomous driving scenarios. As a result, the consistency model achieves

an effective trade-off between computational efficiency and safety, offering strong support for high-performance closed-loop planning.

F. Visualization Results

We evaluate the performance of four methods across distinct driving scenarios in Fig. 2. In the turning-right scenario, the other three methods collide with an oncoming left-turning vehicle due to their delayed response, whereas ConsistencyPlanner successfully negotiates the interaction through decisive maneuver planning. A similar pattern emerges in the turning-left scenario: while PlanT, EasyChauffeur, and Latent Driver fail to avoid contact with a straight-moving vehicle, ConsistencyPlanner maintains safe passage. The U-turn scenarios proves particularly challenging. Other methods exhibit temporary off-road deviations near the target point due to the aggressive roadway curvature, yet only ConsistencyPlanner successfully completes the maneuver while remaining within drivable boundaries. The quantitative results clearly demonstrate that our approach reliably maintains lane discipline and robustly prevents off-road excursions.

V. CONCLUSION

We present ConsistencyPlanner, a novel driving strategy designed for closed-loop planning in real-world autonomous vehicle scenarios. Our approach introduces a consistency model framework that effectively balances accurate multimodal behavior modeling with real-time decision-making capabilities. The proposed architecture integrates an attention mechanism during the ego vehicle’s action decoding phase, enabling comprehensive fusion of conditional inputs with dynamic action information to produce precise planning decisions. This design results in future motion trajectories that are safe and comfortable, demonstrating the model’s strong potential for real-world deployment.

VI. LIMITATION

While the proposed ConsistencyPlanner achieves notable improvements in both off-road rate and collision rate metrics, its performance on the progress rate remains suboptimal—occasionally falling behind the baseline PlanT algorithm. Our in-depth qualitative analysis reveals that this limitation primarily arises from the model’s conservative motion tendencies, particularly at intersections, where it favors safety and collision avoidance over aggressive trajectory progression. To mitigate this, future work will explore the integration of classifier-free guidance, enabling the incorporation of progress-related objectives—such as the progress rate metric—directly into the sampling process through gradient-based optimization. This approach offers a principled way to balance safety with route completion, enabling finer control over the trade-offs between conservative and efficient planning behaviors.

VII. ACKNOWLEDGMENTS

This work is supported by the Beijing Natural Science Foundation under Grant 4242052 and the National Key Research and Development Program of China under Grant 2022YFA1004000.

REFERENCES

- [1] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp, “Wayformer: Motion forecasting via simple & efficient attention networks,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2980–2987, IEEE, 2023.
- [2] D. Dauner, M. Hallgarten, A. Geiger, and K. Chitta, “Parting with misconceptions about learning-based vehicle motion planning,” in *Proceedings of the 7th Conference on Robot Learning*, vol. 229, pp. 1268–1281, PMLR, 2023.
- [3] Y. Zheng, Z. Xing, Q. Zhang, B. Jin, *et al.*, “Planagent: A multi-modal large language agent for closed-loop vehicle motion planning,” *IEEE Transactions on Cognitive and Developmental Systems*, 2026.
- [4] C. Gulino, J. Fu, W. Luo, G. Tucker, E. Bronstein, Y. Lu, J. Harb, X. Pan, Y. Wang, X. Chen, J. Co-Reyes, R. Agarwal, R. Roelofs, Y. Lu, N. Montali, P. Mouglin, Z. Yang, B. White, A. Faust, R. McAllister, D. Anguelov, and B. Sapp, “Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research,” in *Advances in Neural Information Processing Systems*, vol. 36, pp. 7730–7742, 2023.
- [5] H. Fan, F. Zhu, C. Liu, L. Zhang, L. Zhuang, D. Li, W. Zhu, J. Hu, H. Li, and Q. Kong, “Baidu apollo em motion planner,” *arXiv preprint arXiv:1807.08048*, 2018.
- [6] Z. Huang, H. Liu, and C. 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 (ICCV)*, pp. 3903–3913, 2023.
- [7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [8] P. Yang, B. Lu, Z. Xia, C. Han, Y. Gao, T. Zhang, K. Zhan, X. Lang, Y. Zheng, and Q. Zhang, “Worldrft: Latent world model planning with reinforcement fine-tuning for autonomous driving,” *Proceedings of the AAAI conference on artificial intelligence*, 2026.
- [9] Y. Zheng, P. Yang, Z. Xing, Q. Zhang, Y. Zheng, Y. Gao, P. Li, T. Zhang, Z. Xia, P. Jia, *et al.*, “World4drive: End-to-end autonomous driving via intention-aware physical latent world model,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.
- [10] D. Li, Q. Zhang, Z. Xia, Y. Zheng, K. Zhang, M. Yi, W. Jin, and D. Zhao, “Planning-inspired hierarchical trajectory prediction via lateral-longitudinal decomposition for autonomous driving,” *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 692–703, 2023.
- [11] P. Yang, Y. Zheng, Q. Zhang, K. Zhu, Z. Xing, Q. Lin, Y.-F. Liu, Z. Su, and D. Zhao, “Uncad: Towards safe end-to-end autonomous driving via online map uncertainty,” *2025 IEEE International Conference on Robotics and Automation*, 2025.
- [12] L. Xiao, J.-J. Liu, S. Yang, X. Li, X. Ye, W. Yang, and J. Wang, “Learning multiple probabilistic decisions from latent world model in autonomous driving,” *arXiv preprint arXiv:2409.15730*, 2024.
- [13] Y. Zheng, R. Liang, K. ZHENG, J. Zheng, L. Mao, J. Li, W. Gu, R. Ai, S. E. Li, X. Zhan, and J. Liu, “Diffusion-based planning for autonomous driving with flexible guidance,” in *The Thirteenth International Conference on Learning Representations*, 2025.
- [14] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari, “nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles,” *arXiv preprint arXiv:2106.11810*, 2021.
- [15] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, “Consistency models,” in *Proceedings of the 40th International Conference on Machine Learning*, vol. 202, pp. 32211–32252, PMLR, 2023.
- [16] Y. Chen, H. Li, and D. Zhao, “Boosting continuous control with consistency policy,” *arXiv preprint arXiv:2310.06343*, 2023.
- [17] A. Prasad, K. Lin, J. Wu, L. Zhou, and J. Bohg, “Consistency policy: Accelerated visuomotor policies via consistency distillation,” *arXiv preprint arXiv:2405.07503*, 2024.
- [18] W. Peebles and S. Xie, “Scalable diffusion models with transformers,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, 2023.

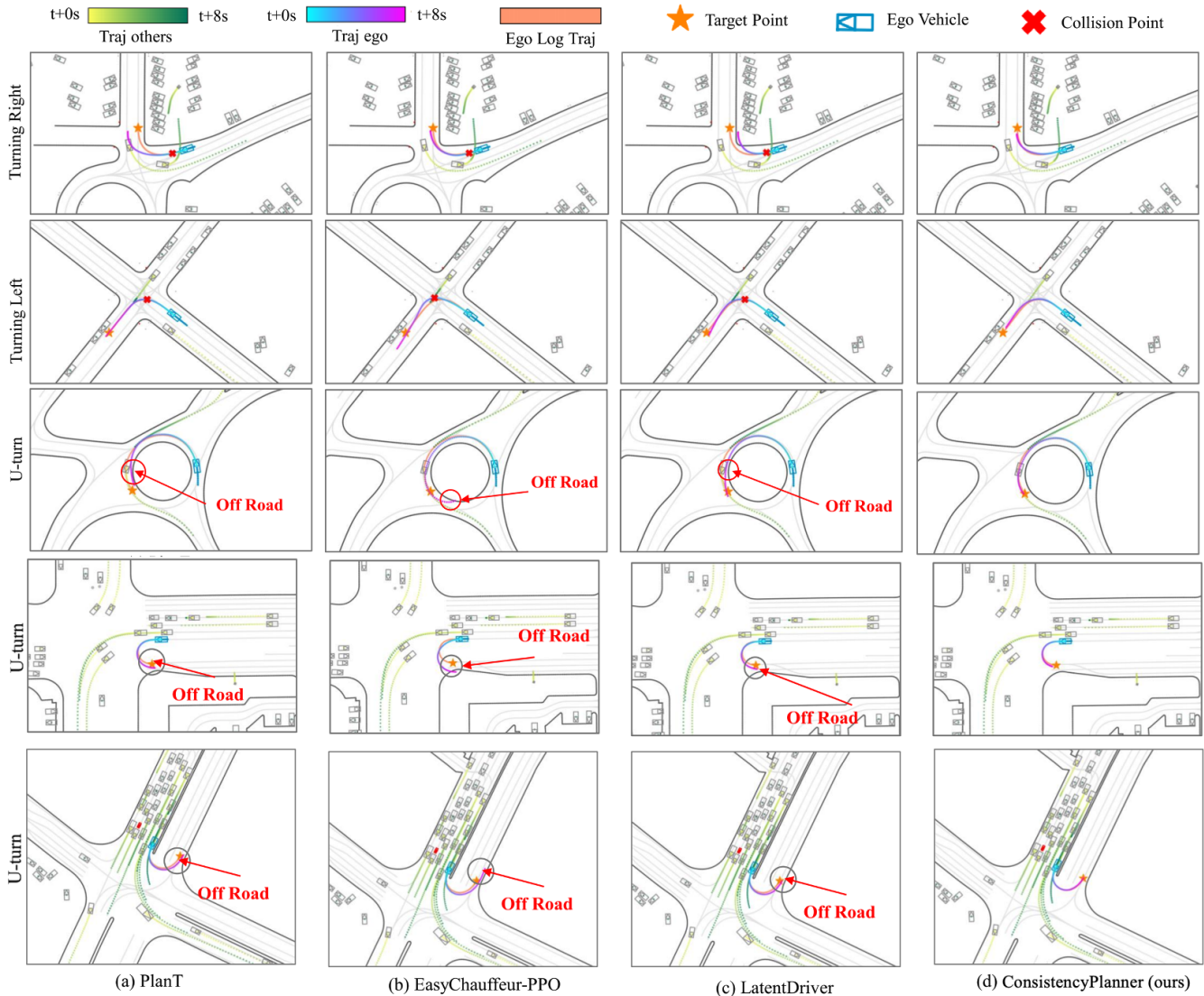


Fig. 2: Visualization results of ConsistencyPlanner against other three methods in complex driving scenarios.

- [19] J. Ho, A. Jain, and P. Abbeel, “Denosing diffusion probabilistic models,” in *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.
- [20] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-based generative modeling through stochastic differential equations,” in *The Ninth International Conference on Learning Representations*, 2021.
- [21] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” *arXiv preprint arXiv:2207.12598*, 2022.
- [22] Y. Song and P. Dhariwal, “Improved techniques for training consistency models,” in *The Twelfth International Conference on Learning Representations*, 2024.
- [23] X. Wang, S. Zhang, H. Zhang, Y. Liu, Y. Zhang, C. Gao, and N. Sang, “Videolcm: Video latent consistency model,” *arXiv preprint arXiv:2312.09109*, 2023.
- [24] W. Dai, L.-H. Chen, J. Wang, J. Liu, B. Dai, and Y. Tang, “Motionlcm: Real-time controllable motion generation via latent consistency model,” in *European Conference on Computer Vision (ECCV)*, pp. 390–408, Springer, 2024.
- [25] H. Li, Z. Jiang, Y. CHEN, and D. Zhao, “Generalizing consistency policy to visual RL with prioritized proximal experience regularization,” in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [26] H. Li, Y. Zhang, H. Wen, Y. Zhu, and D. Zhao, “Stabilizing diffusion model for robotic control with dynamic programming and transition feasibility,” *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 9, pp. 4585–4594, 2024.
- [27] Y. Zheng, Z. Xing, Q. Zhang, B. Jin, P. Li, Y. Zheng, Z. Xia, K. Zhan, X. Lang, Y. Chen, *et al.*, “Planagent: A multi-modal large language agent for closed-loop vehicle motion planning,” *arXiv preprint arXiv:2406.01587*, 2024.
- [28] C. Jiang, A. Cornman, C. Park, B. Sapp, Y. Zhou, D. Anguelov, *et al.*, “Motiondiffuser: Controllable multi-agent motion prediction using diffusion,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9644–9653, 2023.
- [29] Z. Zhong, D. Rempe, D. Xu, Y. Chen, S. Veer, T. Che, B. Ray, and M. Pavone, “Guided conditional diffusion for controllable traffic simulation,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3560–3566, IEEE, 2023.
- [30] K. Chitta, D. Dauener, and A. Geiger, “Sledge: Synthesizing driving environments with generative models and rule-based traffic,” in *European Conference on Computer Vision (ECCV)*, pp. 57–74, Springer, 2024.
- [31] B. Yang, H. Su, N. Gkanatsios, T.-W. Ke, A. Jain, J. Schneider, and

- K. Fragkiadaki, "Diffusion-es: Gradient-free planning with diffusion for autonomous and instruction-guided driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15342–15353, 2024.
- [32] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HIT*, pp. 4171–4186, 2019.
- [33] L. Xiao, J.-J. Liu, X. Ye, W. Yang, and J. Wang, "Easychauffeur: A baseline advancing simplicity and efficiency on waymax," *arXiv preprint arXiv:2408.16375*, 2024.
- [34] U. Ramer, "An iterative procedure for the polygonal approximation of plane curves," *Computer graphics and image processing*, vol. 1, no. 3, pp. 244–256, 1972.
- [35] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: the international journal for geographic information and geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.
- [36] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, M. Lucic, and A. Dosovitskiy, "Mlp-mixer: An all-mlp architecture for vision," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 24261–24272, 2021.
- [37] Y. Balaji, S. Nah, X. Huang, A. Vahdat, J. Song, K. Kreis, M. Aittala, T. Aila, S. Laine, B. Catanzaro, T. Karras, and M.-Y. Liu, "ediff-i: Text-to-image diffusion models with ensemble of expert denoisers," *arXiv preprint arXiv:2211.01324*, 2022.
- [38] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou, Z. Yang, A. Chouard, P. Sun, J. Ngiam, V. Vasudevan, A. McCauley, J. Shlens, and D. Anguelov, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9710–9719, October 2021.
- [39] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.
- [40] K. Renz, K. Chitta, O.-B. Mercea, A. Koepke, Z. Akata, and A. Geiger, "Plant: Explainable planning transformers via object-level representations," in *Conference on Robot Learning*, pp. 459–470, 2023.