

An Improved Transformer-GAN Framework for Multimodal Vehicle Trajectory Prediction

Dexin Zhou and Xu Zhao*

Abstract—Reliable vehicle trajectory planning is essential for safe autonomous driving. Despite significant progress enabled by deep learning, two key challenges remain. First, real-world driving scenes involve heterogeneous multimodal information, including agent history trajectories, high-definition map structures, and diverse traffic rules, making effective feature fusion highly significant. Second, complex inter-agent interactions create highly multimodal future behaviors, where a vehicle may execute multiple feasible maneuvers such as yielding, overtaking, or lane changing. To address these challenges, we propose a Transformer-based generative adversarial framework for multimodal vehicle trajectory planning. Our method integrates agent trajectories and map polylines through a unified encoder, and introduces learnable noise input into a cross-attention decoder to generate diverse and realistic future trajectories. A dual-branch discriminator—combining a data-driven MLP scorer with a rule-based evaluator—further enforces physical plausibility and compliance with traffic rules.

I. INTRODUCTION

Autonomous driving decision-making and motion planning play a critical role in ensuring safe, efficient, and comfortable vehicle behavior in complex traffic environments [1]. The reliability and safety of planning decisions are of paramount importance, as planning failures may directly lead to traffic accidents or system breakdowns. Early solutions primarily relied on rule-based planning methods. These approaches encode traffic regulations, vehicle kinematics, and safety constraints into mathematical formulations, and compute feasible trajectories through analytical models or optimization procedures. However, such methods often require extensive manual tuning and struggle to generalize across diverse and unstructured traffic scenarios.

With the increasing complexity of driving environments, learning-based methods have been introduced. In particular, transformer-based architectures have gained significant attention due to their powerful attention mechanisms, which enable effective modeling of long-range dependencies and interactions across agents and map elements. To further approximate realistic driving behaviors, recent works have explored game-theoretic formulations for trajectory planning. Models such as GameFormer[2] explicitly model vehicle interactions through multi-layer decoding and iterative reasoning, treating driving as a strategic game among multiple agents, which can produce more socially compliant and realistic trajectories.

Despite these advances, existing approaches often exhibit limited robustness when encountering out-of-distribution scenarios or rare driving patterns. To address these limitations, we introduce a generative-model-based planner that explicitly models the multi-modality of future trajectories. In particular, planners based on Generative Adversarial Networks (GANs)[3] leverage the strong generative capacity of adversarial training to inject stochasticity and diversity into trajectory generation. Compared with traditional models such as IDM, GAN-based planners offer greater lateral and behavioral variability, naturally capturing maneuvers such as lane changes, overtaking, and turning. Moreover, the adversarial training paradigm aligns well with game-theoretic planning principles. Through the mutual competition and co-evolution of the generator and discriminator, the model is encouraged to produce trajectories that are not only diverse but also indistinguishable from expert driving behavior.

Building upon these advances in [4], [5], [6], we further investigate the use of generative models for motion planning under a learning framework. In this work, future trajectories are generated using a classical encoder-decoder architecture, where heterogeneous inputs such as agent histories and map information are first encoded into a compact scene representation and then decoded into multi-modal future trajectories. To better capture the intrinsic uncertainty and diversity of driving behaviors, we adopt a Game-theoretic training paradigm.

The main contributions of this paper can be summarized as follows:

- **Game-theoretic training for planning:** We introduce adversarial training into the trajectory planning process, shifting the optimization objective from rigidly imitating expert trajectories to generating trajectories that are sufficiently realistic to fool a discriminator, thereby better capturing human-like driving behaviors.
- **Rule-aware discriminative scoring:** A rule-based scoring module is incorporated into the discriminator to enhance interpretability and enforce physically and semantically meaningful constraints, bridging data-driven learning with domain knowledge.
- **State-of-the-art performance on nuPlan[7]:** Closed-loop simulation experiments on the nuPlan benchmark demonstrate that the proposed method achieves state-of-the-art performance across multiple planning metrics, validating its effectiveness and robustness in complex urban driving scenarios.

Dexin Zhou, and Xu Zhao are with the School of Automation and Intelligent Sensing, Shanghai Jiao Tong University, Shanghai 200240, China. This work has been funded in part by the NSFC grants 62176156.

*Corresponding Author. Email: zhaoxu@sjtu.edu.cn

II. RELATED WORK

A. Rule-Based models

Autonomous driving decision-making and motion planning play a critical role in ensuring safe, efficient, and comfortable vehicle behavior in complex traffic environments[1]. A fundamental challenge in this task lies in effectively modeling and reasoning over highly multi-modal and heterogeneous information, including agent trajectories, high-definition maps, and dynamic interactions among traffic participants. Moreover, the reliability and safety of planning decisions are of paramount importance, as planning failures may directly lead to traffic accidents or system breakdowns.

Early solutions to autonomous driving planning primarily relied on rule-based approaches, which emphasize interpretability and safety guarantees. Among them, search-based method converts the continuous state space into a discrete graph and calculates feasible trajectories based on it. Representative algorithms include Dijkstra[8], A* search[9] and dynamic programming (DP)[10]. Numerous advanced variants have been proposed to improve efficiency and feasibility, such as Hybrid A*[11], bidirectional A* and semi-optimal A*[12]. In addition, selection-based methods determine optimal future states by evaluating candidate actions under predefined cost functions, with greedy strategies[13] and Markov Decision Process (MDP) based methods[14], [15] being typical examples. Based on these theories, a classical car-following model IDM[16], simulates the longitudinal motion behavior of the vehicle on the road by modeling the vehicle kinematics. Pdm[17] improves this model to provide more candidate trajectories by increasing lateral offset.

B. Learning-Based Models

Imitation Learning (IL) refers to a learning paradigm in which an agent acquires a policy by mimicking expert demonstrations, where expert trajectories typically provide high-quality decision-making and control information [18].

With the rapid advancement of deep learning in recent years, learning-based planning methods have emerged as a promising alternative to traditional rule-based approaches[19]. In particular, the development of Transformer[20] architectures has attracted significant attention due to their strong representation capacity and scalability, originally demonstrated in natural language processing. Transformer-based models have since been successfully applied to autonomous driving, including VectorNet[21], Scene-Transformer[22] and Wayformer[23] proposed by Waymo, which leverage attention mechanisms to capture spatial and temporal dependencies in traffic scenes. Furthermore, generative modeling has recently gained popularity in trajectory prediction and planning, owing to its powerful ability to model uncertainty and multi-modality. Representative examples include Social GAN[24], which employs GAN[3] for pedestrian trajectory prediction. [25] adopts diffusion models for vehicle trajectory planning.

III. METHOD

In this section, we introduce a Transformer-based GAN framework to address this challenge. The generator employs attention mechanisms to fuse multimodal information and incorporates stochastic noise to enhance trajectory diversity. The decoder further integrates a rule-based scoring module to complement neural scoring, thereby improving the model's ability to interpret and adhere to traffic rules.

A. Problem Definition

Our problem is defined as, in a scene, using high-definition map information and all the agent's historical trajectories as conditions to generate the ego vehicle's future trajectory based on random noise. We assume that the historical trajectory sequence of each agent is denoted as $X_i = \{X_i^1, X_i^2, \dots, X_i^{his}\}$, $X_i^t \in \mathbb{R}^2$, where $X_i^t = (x_i^t, y_i^t)$. Here, $i = 0, 1, \dots, neighbors$, 0 denotes the ego vehicle, $neighbors$ denotes the number of selected neighbor vehicles in the scene. The predicted future trajectory sequence of the ego vehicle is denoted as $Y_j = \{Y_j^1, Y_j^2, \dots, Y_j^{fut}\}$, $Y_j^t \in \mathbb{R}^2$, where $Y_j^t = (x_j^t, y_j^t)$. Here, $j = 1, \dots, M$, M denotes the number of modalities. The ground truth trajectory sequence of the ego vehicle is denoted as $\hat{Y} = \{\hat{Y}^1, \hat{Y}^2, \dots, \hat{Y}^{fut}\}$, $\hat{Y}^t \in \mathbb{R}^2$, where $\hat{Y}^t = (\hat{x}^t, \hat{y}^t)$. Here, his represents the number of historical time steps used, and fut represents the number of future time steps to be predicted.

B. Generator

Our model follows a classical encoder-decoder architecture as illustrated in Fig. 1.

Encoder The encoder receives two types of inputs. The historical trajectory information of agents is denoted as $A \in \mathbb{R}^{(1+neighbors) \times his \times d_t}$, where d_t represents state attributes such as position, velocity, and heading. The map information is represented as polylines $M \in \mathbb{R}^{(1+neighbors) \times N_m \times N_w \times d_p}$, where N_m denotes the number of map elements surrounding each agent (e.g., lanes, traffic signals), N_w represents the number of waypoints for each map element, and d_p is the dimension of each map element.

These two types of information are processed by separate encoders. The agent's historical trajectories are encoded using an LSTM network, producing encoded features $A_E \in \mathbb{R}^{(1+neighbors) \times D}$, where D is the hidden feature dimension. The map polylines are encoded using an MLP network, followed by max-pooling to aggregate information, resulting in encoded map features $M_E \in \mathbb{R}^{(1+neighbors) \times \hat{N}_m \times D}$, where $\hat{N}_m$ denotes the number of aggregated map elements.

Next, for each agent, we feed the two features into a Transformer encoder to capture the contextual interactions among agents and surrounding map elements. This process yields the scene context representation $S_c \in \mathbb{R}^{N \times (N + \hat{N}_m) \times D}$. Compared with traditional encoding methods, the attention mechanism enables the model to adaptively focus on the most relevant entities, providing a richer and more discriminative scene embedding for downstream trajectory generation.

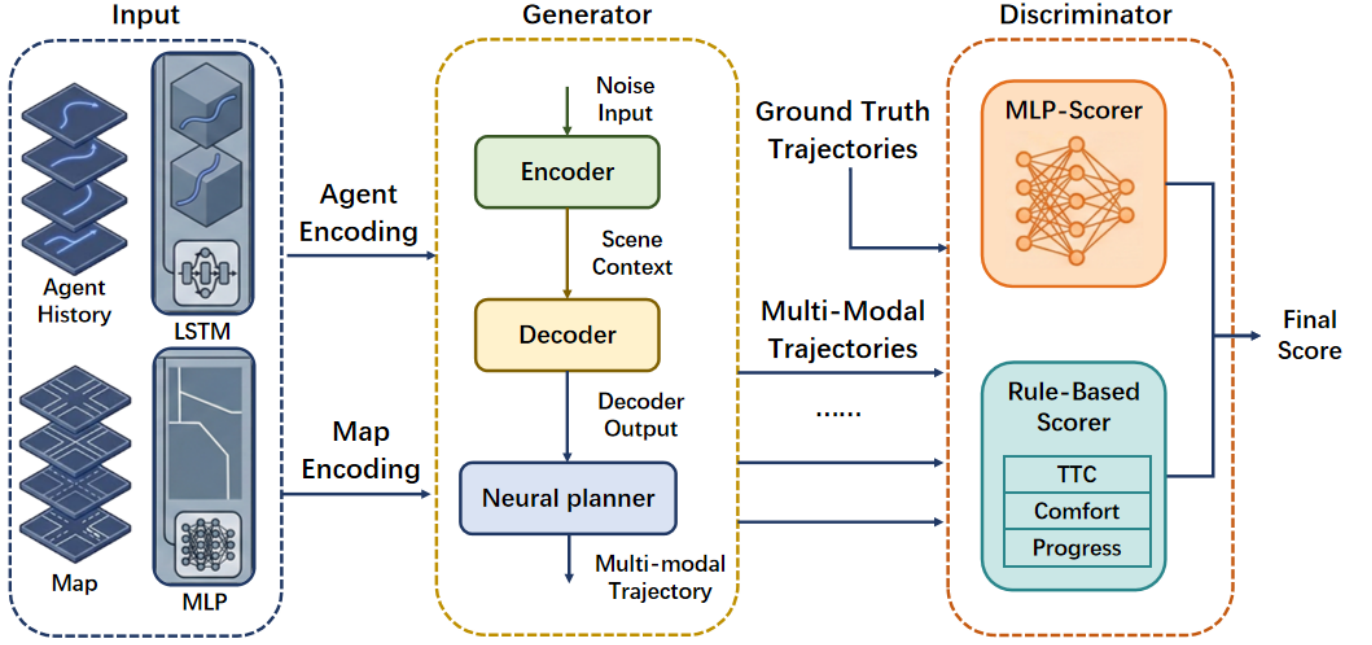


Fig. 1: Overview of our encoder-decoder architecture. It mainly consists of three parts: encoding of input representation, generator based on transformer model, and discriminator composed of two complementary scoring modules

Decoder To generate a realistic set of diverse future trajectories that cover multiple plausible behaviors, we introduce learnable noise. Specifically, we define a learnable noise matrix $Z \in \mathbb{R}^{M \times D_z}$, where M is the number of predefined modalities and D_z is the dimensionality of the latent space. This matrix is randomly initialized at the start of training and optimized jointly with the rest of the model parameters via gradient descent. Each row $z_m \in \mathbb{R}^{D_z}$ represents a prototype of a distinct driving intent or trajectory pattern, enabling the decoder to condition on different latent modes and generate diverse trajectory hypotheses.

We employ a multi-head cross-attention Transformer module. The encoded agent histories A_E are first combined with the multimodal latent variables Z to form a set of future-trajectory candidate queries $A_E + Z$. These candidate queries are then concatenated with the scene context encoding S_c to construct the final query input to the decoder, while the scene context encoding S_c itself serves as both the key and value for retrieving contextual information from the environment. Through multi-head cross-attention, each query attends to different subspaces of the scene representation, enabling the decoder to selectively focus on the most relevant agents, map elements, and interaction cues for each latent mode. The decoder output is a latent representation $Z_D \in \mathbb{R}^{(1+neighbors) \times M \times D}$. This representation is subsequently passed through an MLP network to generate the final predicted future trajectories $T \in \mathbb{R}^{(1+neighbors) \times M \times fut \times 2}$, where each modality corresponds to a distinct trajectory hypothesis produced under different latent driving intentions.

C. Discriminator

The discriminator consists of two complementary scoring modules as illustrated in 2.

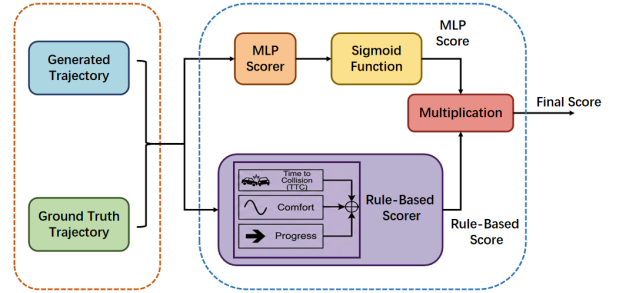


Fig. 2: Overview of our discriminator. This discriminator consists of two scoring modules. The MLP scoring module normalizes the sigmoid function to a range of 0-1, while the rule-based scoring module is a weighted sum of scores from different discrimination criteria.

MLP Scorer The first component of the discriminator is a data-driven score implemented using an MLP. Given an input trajectory, the MLP computes a scalar confidence value, which is subsequently passed through a sigmoid function to map it into the range $[0, 1]$. This output can be interpreted as the probability that the trajectory is judged to be realistic and consistent with expert behavior. The data-driven score captures implicit decision-making patterns, interactive dynamics, and high-level trajectory semantics that are difficult to encode

using explicit rules.

Rule-based Scorer The second component evaluates each trajectory using a set of interpretable, physically and semantically grounded criteria. These include collision avoidance, kinematic feasibility, comfort (e.g., acceleration and jerk limits), lane adherence, and goal-reaching success. Each criterion produces a penalty factor in the range $[0, 1]$, where values close to zero indicate severe violations and values close to one indicate compliance. The score is calculated by multiplying the weighted sum of individual factors by the penalty term. The outputs of the two scoring modules are combined multiplicatively to form the final discriminator score.

Training details Generative Adversarial Networks (GANs) train a generator G to produce samples that mimic the true data distribution by playing a minimax game with a discriminator D . To stabilize training and to preserve physical plausibility, we adopt a two-stage training protocol.

Supervised pretraining We first warm up the generator with a supervised regression objective. Given a training batch of size B , the generator produces multimodal predictions $T \in \mathbb{R}^{B \times fut \times 2}$, and the ground-truth future trajectories are denoted by $Y \in \mathbb{R}^{B \times fut \times 2}$. To encourage the generator to cover multiple modes, we use a best-of- M L2 loss (also called min-over-modes):

$$\mathcal{L}_{L2} = \frac{1}{B} \sum_{b=1}^B \min_{m \in \{1, \dots, M\}} \sum_{t=1}^{fut} \|T_{b,m}^t - Y_{b,i}^t\|_2^2.$$

The generator parameters are updated by minimizing $\mathcal{L}_{L2}$, which provides a stable initialization that leverages direct supervision from expert trajectories.

Adversarial fine-tuning After pretraining, we introduce the discriminator and perform adversarial training. The discriminator produces a final score

$$S_f = S_{MLP} \times S_{rule},$$

where $S_{MLP} \in (0, 1)$ is the data-driven (sigmoid-normalized) score and $S_{rule} \in (0, 1)$ is the rule-based score. Using the standard GAN objective (logarithmic loss), the adversarial min-max problem is

$$\min_G \max_D \mathbb{E}_{Y \sim p_{data}} [\log S_f(Y)] + \mathbb{E}_{Z \sim p_Z} [\log (1 - S_f(G(Z)))],$$

where Z denotes the multimodal latent inputs to the generator. In practice, we replace or augment this objective with stabilized variants (e.g., non-saturating loss or WGAN-style losses) as needed.

This two-stage scheme leverages supervised learning to produce reasonable initial trajectories and then uses adversarial refinement to encourage realism and diversity. The multiplicative fusion $S_f = S_{MLP} \times S_{rule}$ ensures that the discriminator enforces both data-driven realism and strict rule compliance, while the differentiability of S_{MLP} permits effective gradient flow back to the generator during adversarial updates.

IV. EXPERIMENT

A. Experimental Settings

Dataset NuPlan is the world’s first large-scale autonomous driving planning benchmark. The existing benchmarks for motion prediction of autonomous vehicles (Argoverse[26], Lyft[27], Waymo[28]) focus on short-term motion prediction of other agents rather than long-term planning of autonomous vehicles. NuPlan’s benchmark overcomes these limitations by providing a training framework for developing machine learning based planners, a lightweight closed-loop simulator, motion planning specific metrics, and an interactive tool for visualizing results. In our work, we adopt the closed-loop simulation setting to ensure realistic long-term evaluation. Based on the official scenario taxonomy, we select highway (2,000 scenarios), roundabout (2,000 scenarios), and intersection (3,000 scenarios) as our primary test scenarios, covering a broad range of driving complexities.

Evaluation metric In autonomous driving trajectory and planning, open-loop metrics such as the Average Displacement Error (ADE) and Final Displacement Error (FDE) are widely used to quantify prediction accuracy. Given a predicted trajectory $\hat{Y} = \{\hat{Y}^1, \dots, \hat{Y}^{fut}\}$ and its ground truth counterpart $Y = \{Y^1, \dots, Y^{fut}\}$, ADE measures the mean Euclidean distance across all predicted future time steps, while FDE evaluates the displacement error at the final time step:

$$ADE = \frac{1}{fut} \sum_{t=1}^{fut} \|\hat{Y}^t - Y^t\|_2, \quad FDE = \|\hat{Y}^{fut} - Y^{fut}\|_2.$$

Moreover, for our nuPlan-based closed-loop simulation, we adopt three additional planning-oriented metrics: *Success Rate*, *Collision-Free Rate*, and *On-Route Rate*. A scenario is considered successful if the ego vehicle achieves sufficient progress relative to the ground truth while avoiding collisions and off-road violations. Safety is further evaluated using the Collision-Free Rate, where a collision is detected whenever the ego vehicle’s 2D bounding box overlaps with that of another object at any time step. Finally, compliance with the drivable area is measured using the On-Route Rate, which counts the percentage of scenarios in which the ego vehicle remains within the designated drivable region.

Implementation Details We conduct all experiments on NVIDIA RTX-2080Ti GPUs. In the first pretraining stage, the generator adopts a Transformer architecture with three encoder layers and one decoder layer, optimized using an L2 loss. The model is trained for 100 epochs with an initial learning rate of 1×10^{-4} and a batch size of 32, producing a pretrained generator. In the second stage, we perform adversarial training between the generator and discriminator following a minimax objective. To ensure stable optimization, the discriminator is updated three times for every update of the generator. During training, we utilize 31,964 randomly sampled scenarios, each containing 1 second of historical observations and the corresponding 8-second future trajectories. The dataset is split into training and validation

TABLE I: Comparison of different methods in three metrics across different scenarios.

Method	Scene	Success Rate $\uparrow$ (%)	Collision-Free Rate $\uparrow$ (%)	On-Route Rate $\uparrow$ (%)
IDM	Intersection	54.10	88.63	88.43
	Roundabout	39.55	77.37	82.45
	Highway	72.29	91.67	93.05
PDM-closed	Intersection	73.33	98.30	99.10
	Roundabout	62.60	93.40	98.85
	Highway	86.25	97.85	99.40
PDM-open	Intersection	19.00	86.18	83.63
	Roundabout	22.60	60.70	73.60
	Highway	22.80	91.67	92.80
PlanTF[29]	Intersection	42.30	93.92	93.17
	Roundabout	41.70	89.20	98.30
	Highway	48.50	95.10	92.40
PDM-hybrid	Intersection	73.40	98.25	99.07
	Roundabout	63.70	95.35	99.45
	Highway	86.50	97.90	99.45
Ours	Intersection	93.97	99.38	99.52
	Roundabout	84.42	98.70	97.40
	Highway	95.90	99.87	99.21

subsets with a 4:1 ratio, yielding 25,571 training scenarios and 6,393 validation scenarios.

B. Ablation Study

To evaluate the effectiveness of our discriminator design, we compare three variants: using only the MLP-based score, using only the rule-based score, and using both components jointly. All models predict six candidate trajectories, from which the one closest to the ground-truth trajectory is selected. We report performance using the PlannerADE and PlannerFDE metrics to assess ego-vehicle planning quality. Table II summarizes the 8-second closed-loop planning results. The results indicate that using only the MLP-based discriminator leads to poor performance, primarily due to the large discrepancy in model capacity between the generator and discriminator, which destabilizes adversarial training. In contrast, using only the rule-based scoring module yields more stable optimization; however, the strict handcrafted constraints limit trajectory flexibility and reduce overall quality. By integrating both data-driven and rule-based scores within our two-stage training paradigm, the final model achieves the best performance, effectively balancing training stability and planning adaptability.

TABLE II: Comparison of different discriminator on Planner metrics.

Discriminator	Planner-ADE	Planner-FDE
MLP-Only	9.5231	23.0922
Rule-Only	1.8714	4.1117
MLP + Rule	0.7755	1.6266

Additionally, we perform a comparative evaluation against other game-theoretic planning approaches, focusing on the original GameFormer model and its improved variant, GameFormer-UAM[30]. Table III presents the planning results for all three methods. The results demonstrate that the

adversarial training paradigm employed in our model leads to superior trajectory planning performance.

TABLE III: Comparison of different methods on Planner metrics.

Method	Planner-ADE	Planner-FDE
Gameformer	0.9356	1.9324
Gameformer-UAM	0.8653	1.6493
GAN	0.7755	1.6266

C. Multi-Expert Planning experiments

We use a multi expert planning framework in our project, which means that in different scenarios or traffic conditions, the generated trajectories will be scored and the one with the highest score will be selected. Due to each expert optimizing their decision-making strategy based on different scenarios, traffic conditions, or objectives, it is crucial for planners to select the optimal expert for decision-making in different contexts. Table I presents a comparison between the proposed multi-expert framework and single-expert planning. The results clearly demonstrate the advantages of the multi-expert approach, including improved trajectory diversity and enhanced adaptability to complex driving scenarios.

In order to further optimize the switching strategy of the planner, we conducted statistical analysis on the usage ratio of each expert in different scenarios. We have selected ten scenarios each, including highways, intersections, and roundabouts, to calculate the proportion of planner usage for each scenario, as shown in Fig. 3. Compared to Other planners, the GAN planner’s consistent presence (about 20%) across diverse scenarios highlights its balanced performance and practical utility.

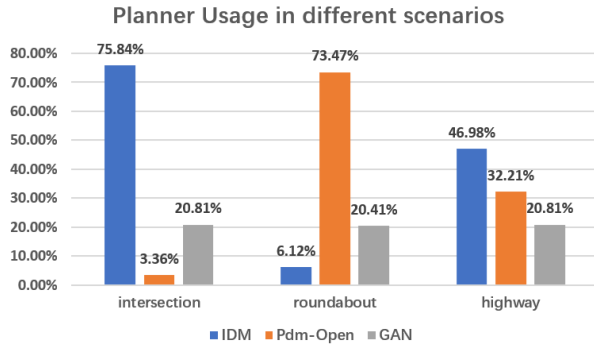


Fig. 3: Proportion of planner usage in multi expert hybrid planning frameworks in different scenarios

V. CONCLUSIONS

In this work, we proposed a transformer-enhanced generative adversarial architecture for vehicle trajectory planning, which combines data-driven scoring with rule-based evaluation. This design enables the generator to learn both the geometric distribution of feasible trajectories and the semantic constraints required for safe autonomous driving, resulting in improved stability and expressiveness during adversarial training. The results on Nuplan Dataset validate the effectiveness of combining transformer representations, GAN-based trajectory generation, and multi-expert decision mechanisms.

REFERENCES

- [1] L. Chen, Y. Zhang, B. Tian, Y. Ai, D. Cao, and F.-Y. Wang, "Parallel driving os: A ubiquitous operating system for autonomous driving in cpss," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 4, pp. 886–895, 2022.
- [2] 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*, 2023, pp. 3903–3913.
- [3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.
- [4] L. Chen, Q. Zhou, Y. Cai, H. Wang, and Y. Li, "Cae-gan: A hybrid model for vehicle trajectory prediction," *IET Intelligent Transport Systems*, vol. 16, no. 12, pp. 1682–1696, 2022.
- [5] Z. Lv, X. Huang, and W. Cao, "An improved gan with transformers for pedestrian trajectory prediction models," *International Journal of Intelligent Systems*, vol. 37, no. 8, pp. 4417–4436, 2022.
- [6] P. Dendorfer, A. Osep, and L. Leal-Taixé, "Goal-gan: Multimodal trajectory prediction based on goal position estimation," in *Proceedings of the Asian Conference on Computer Vision*, 2020.
- [7] 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.
- [8] E. W. Dijkstra, "A note on two problems in connexion with graphs," in *Edsger Wybe Dijkstra: his life, work, and legacy*, 2022, pp. 287–290.
- [9] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 485–501, 2010.
- [10] W. Xu, J. Pan, J. Wei, and J. M. Dolan, "Motion planning under uncertainty for on-road autonomous driving," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2507–2512.
- [11] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Practical search techniques in path planning for autonomous driving," University of Michigan, Tech. Rep. 48105, 2008.
- [12] S. M. Bagheri, H. Taghaddos, A. Mousaei, F. Shahnava, and U. Hermann, "An a* algorithm for semi-optimization of crane location," *Journal of Manufacturing Systems*, vol. 121, p. 103447, 2021.
- [13] X. Li, Z. Sun, D. Cao, Z. He, and Q. Zhu, "Real-time trajectory planning for autonomous urban driving: Framework, algorithms, and verifications," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 2, pp. 740–753, 2015.
- [14] H. Bai, S. Cai, N. Ye, D. Hsu, and W. S. Lee, "Intention-aware online pomdp planning for autonomous driving in a crowd," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 454–460.
- [15] N. Ye, A. Somani, D. Hsu, and W. S. Lee, "Despot: online pomdp planning with regularization," *J. Artif. Int. Res.*, vol. 58, no. 1, p. 231–266, jan 2017.
- [16] S. Albeaik, A. Bayen, M. T. Chiri, X. Gong, A. Hayat, N. Kardous, A. Keimer, S. T. McQuade, B. Piccoli, and Y. You, "Limitations and improvements of the intelligent driver model (idm)," *SIAM Journal on Applied Dynamical Systems*, vol. 21, no. 3, pp. 1862–1892, 2022.
- [17] D. Dauner, M. Hallgarten, A. Geiger, and K. Chitta, "Parting with misconceptions about learning-based vehicle motion planning," in *Conference on Robot Learning*. PMLR, 2023, p. 1268–1281.
- [18] Z. Zhang, R. Tian, R. Sherony, J. Domeyer, and Z. Ding, "Attention-based interrelation modeling for explainable automated driving," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1564–1573, 2023.
- [19] K. Muhammad, A. Ullah, J. Lloret, J. Del Ser, and V. H. C. De Albuquerque, "Deep learning for safe autonomous driving: Current challenges and future directions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4316–4336, 2020.
- [20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [21] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 525–11 533.
- [22] J. Ngiam, B. Caine, V. Vasudevan, Z. Zhang, H.-T. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal *et al.*, "Scene transformer: A unified architecture for predicting multiple agent trajectories," *arXiv preprint arXiv:2106.08417*, 2021.
- [23] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp, "Wayformer: Motion forecasting via simple & efficient attention networks," *arXiv preprint arXiv:2207.05844*, 2022.
- [24] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2255–2264.
- [25] Y. Zheng, R. Liang, K. Zheng, J. Zheng, L. Mao, J. Li, W. Gu, R. Ai, S. E. Li, X. Zhan *et al.*, "Diffusion-based planning for autonomous driving with flexible guidance," *arXiv preprint arXiv:2501.15564*, 2025.
- [26] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, "Argoverse: 3d tracking and forecasting with rich maps," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8740–8749.
- [27] J. Houston, G. Zuidhof, L. Bergamini, Y. Ye, L. Chen, A. Jain, S. Omari, V. Iglovikov, and P. Ondruska, "One thousand and one hours: Self-driving motion prediction dataset," in *Proceedings of the 2020 Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Kober, F. Ramos, and C. Tomlin, Eds., vol. 155. PMLR, 16–18 Nov 2021, pp. 409–418.
- [28] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2446–2454.
- [29] J. Cheng, Y. Chen, X. Mei, B. Yang, B. Li, and M. Liu, "Rethinking imitation-based planners for autonomous driving," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 14 123–14 130.
- [30] Y. Chen, Y. Zhang, and X. Zhao, "Uncertainty-aware module for trajectory prediction and ego-planning of challenging agents in autonomous driving," in *2025 IEEE International Conference on Real-time Computing and Robotics (RCAR)*. IEEE, 2025, pp. 1167–1172.