Improving Ego Vehicle Performance in Long-Tail Scenarios Using Reinforcement Learning

Long-Giang Vu
Masters Student, Computer Science and Engineering Department
University of California, San Diego
La Jolla, CA 92093
lgv001@ucsd.edu

Abstract

Motion planning is a critical component of autonomous driving systems. Early methods relied on rule-based algorithms with limited ability to represent complex environments, while recent data-driven approaches imitate expert demonstrations to produce more realistic and robust trajectories. However, these models often struggle in rare or poorly represented scenarios and have difficulty enforcing hard constraints due to their latent representations. This project applies GRPO, a reinforcement-learning algorithm, to address these limitations by enforcing hard constraints and improving trajectory selection in uncommon driving situations. The implementation is available at: `https://github.com/ginlov/wayformer`.

1 Introduction

1.1 Motion Planning for Autonomous Vehicles

Motion planning is a fundamental pillar of autonomous driving, serving as the bridge between perception and control. The primary objective is to generate safe, kinematically feasible, and socially compliant trajectories in dynamic environments. Traditional approaches relied on sampling-based methods or rule-based lattice planners, which offer strong safety guarantees but struggle to scale to the stochastic nature of dense urban traffic.

The field has recently gravitated toward data-driven Imitation Learning (IL). Transformer-based architectures, most notably Wayformer Nayakanti et al. [2023], have established state-of-the-art benchmarks by leveraging attention mechanisms to encode heterogeneous scene contexts, including road topology, traffic signal states, and agent interactions. By minimizing displacement errors against ground-truth data, these models effectively approximate the underlying expert policy from large-scale demonstrations.

However, IL-based planners face two significant limitations:

- **Long-Tail Distribution Shift:** Pure imitation learners often generalize poorly to "long-tail" scenarios—rare, high-stakes events (e.g., cut-ins, aggressive merges) that are underrepresented in the training data Codevilla et al. [2019].
- **Lack of Hard Constraints:** Because IL optimizes for likelihood rather than explicit safety rules, it lacks a mechanism to strictly enforce hard constraints. Consequently, even high-confidence predictions can occasionally violate collision boundaries or traffic rules, requiring post-hoc trajectory selection or safety filtering Lu et al. [2023].

1.2 Reinforcement Learning for AV Planning

To address the shortcomings of imitation, Reinforcement Learning (RL) allows for the direct optimization of non-differentiable safety metrics. By treating the motion planner as a policy, RL can fine-tune the model to maximize a reward function that explicitly penalizes collisions and rule violations.

Standard RL algorithms like Proximal Policy Optimization (PPO), however, are computationally prohibitive for large Transformer-based backbones such as Wayformer. They typically require a separate Value (Critic) network that matches the size of the Policy network, nearly doubling the memory footprint and training overhead.

This project proposes the application of Group Relative Policy Optimization (GRPO) Shao et al. [2024], a novel algorithm originally developed for reasoning in Large Language Models (LLMs), to the domain of autonomous driving. GRPO eliminates the need for a critic network by leveraging the "best-of-N" nature of multimodal trajectory prediction. By sampling a group of trajectory outputs and computing their relative advantages against a hard-constraint reward function, we can efficiently fine-tune Wayformer's classification head. This approach aims to combine the realistic priors of imitation learning with the rigorous safety enforcement of reinforcement learning, specifically targeting performance improvements in long-tail driving scenarios.

2 Methodology

2.1 Wayformer

The Wayformer architecture Nayakanti et al. [2023] serves as the baseline framework for this study. Designed for multimodal motion forecasting, Wayformer distinguishes itself through its capacity to encode a rich, heterogeneous driving scene—comprising agent history, road graph geometry, and traffic light states—into a compact, unified set of feature embeddings via a Transformer Encoder.

The standard Wayformer architecture follows a two-stage process:

1. **Scene Encoding:** A Self-Attention Encoder processes and fuses multimodal input tokens, generating a comprehensive scene context embedding Z_{scene}.
2. **Trajectory Decoding:** A Cross-Attention Decoder takes a set of K learned trajectory proposal queries and cross-attends to the Z_{scene} embedding. This process generates K high-dimensional trajectory embeddings, $Y = \{Y_1, \ldots, Y_K\}$, where $Y_k \in \mathbb{R}^{D_{emb}}$.

In the original formulation, the final prediction of the trajectory, $\mathbf{T}_k = \{(x_t, y_t)\}_{t=1}^{T_{pred}}$, for each mode k is obtained via two simple linear heads applied to the final embedding Y_k:

- **Classification Head:** A linear layer that predicts the unnormalized log-likelihood (mode probability) p_k.
- **Regression Head (Original):** A linear layer that predicts the time-series coordinates or offsets $\mathbf{T}_k$. This linear projection simplifies the model but struggles to capture complex, non-linear dependencies between consecutive predicted states and may accumulate temporal errors over the prediction horizon T_{pred}.

2.2 Temporal Gaussian Head for Trajectory Refinement

To improve the temporal coherence and long-term prediction accuracy of the baseline, I propose replacing the simple linear regression head with a Temporal Gaussian Decoder (TGD).

The TGD explicitly models the dependencies between future time steps, ensuring a more dynamically realistic trajectory output. This is achieved by introducing a set of learned queries to directly attend to the aggregated trajectory feature Y_k.

The architectural modification is applied as follows for each trajectory embedding $Y_k \in \mathbb{R}^{D_{emb}}$:

1. **Learned Temporal Queries:** I define a set of T_{pred} learned **Time Queries** $Q_{\text{time}} \in \mathbb{R}^{T_{\text{pred}} \times D_{\text{query}}}$. Each query $q_t \in Q_{\text{time}}$ is responsible for predicting the state at future time step t.

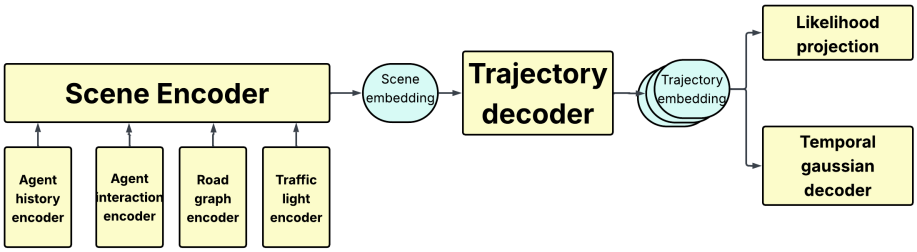


Figure 1: Conceptual Diagram of the Modified Wayformer Architecture. The Encoder and Cross-Attention Decoder are preserved. The key modification is replacing the linear Regression Head with the Temporal Gaussian Decoder (TGD) for enhanced temporal coherence.

2. **Temporal Decoding (Cross-Attention):** The Time Queries Q_{time} are fed as the query (Q) input to a stack of Transformer Decoder layers. The trajectory embedding Y_k serves as both the key (K) and value (V) input, effectively performing cross attention:

$$\text{Attention}(Q_{\text{time}}, Y_k, Y_k)$$

   This mechanism allows each future time step query q_t to attend to the global features of the proposed trajectory, Y_k, and generate a time-step-specific feature vector $H_{k,t} \in \mathbb{R}^{D_{\text{token}}}$.

3. **Temporal Decoding (Self-Attention):** The resulting sequence of time-step features $H_k = \{H_{k,1}, \ldots, H_{k,T_{\text{pred}}}\}$ then passes through a layer of self-attention along the temporal dimension. This crucial step enables the model to globally reason about the entire path, ensuring dynamic consistency and correcting accumulated errors.
4. **Final Projection:** The output of the TGD is projected via a small multi-layer perceptron (MLP) into the physical state space (e.g., $\Delta x, \Delta y$), yielding the final trajectory $\mathbf{T}_k$.

The resulting architecture provides a highly accurate and temporally consistent set of base trajectories $\mathbf{T}_k$, which serve as the action space for the subsequent reinforcement learning phase. The focus of the GRPO fine-tuning will be on refining the associated mode probability p_k generated by the Classification Head.

2.3 Group Relative Policy Optimization (GRPO)

I utilize the pre-trained Wayformer with the Temporal Gaussian Decoder (TGD) as the base policy π_{ref}, fine-tuning the parameters θ of the Classification Head to yield the optimized policy π_θ. Group Relative Policy Optimization (GRPO) Shao et al. [2024] is a critic-free policy gradient method that estimates the advantage function by normalizing rewards within a group of sampled trajectories, thereby eliminating the need for a separate Value network and reducing memory overhead.

2.3.1 Action Sampling and Advantage Calculation

In the context of multimodal motion planning, the process is adapted to leverage the K trajectory proposals as the group of sampled actions:

1. **Group Sampling:** For a given driving state q, the policy π_θ generates a group of G multimodal trajectory outputs, $\mathbf{O} = \{o_1, \ldots, o_G\}$, where $G = K$ is the number of proposals from the Wayformer TGD. Each output o_i consists of a trajectory $\mathbf{T}_i$ and its predicted log-likelihood $\log(\pi_\theta(o_i|q))$.
2. **Reward Assignment:** Each sampled trajectory o_i is evaluated against a pre-defined rule-based reward function $R(o_i)$, yielding a scalar reward r_i.
3. **Group Advantage:** The relative quality of each sampled trajectory o_i within its group is quantified by the Group Relative Advantage A_i. This advantage is calculated by standardizing the reward r_i relative to the group rewards $\mathbf{R} = \{r_1, \ldots, r_G\}$, using the mean $\mu_{\mathbf{R}}$ and standard deviation $\sigma_{\mathbf{R}}$:

$$A_i = \frac{r_i - \mu_{\mathbf{R}}}{\sigma_{\mathbf{R}} + \delta}$$

where δ is a small constant (e.g., 10^{-6}) for numerical stability.

2.3.2 Policy Optimization Objective

The GRPO objective function is used to update the policy parameters θ to favor trajectories with a positive Group Relative Advantage. The objective includes a term for the advantage-weighted log-likelihood ratio, akin to PPO, and a Kullback-Leibler (KL) divergence penalty to stabilize training by limiting the deviation from the reference policy π_{ref}:

$$\mathcal{L}_{\text{GRPO}}(\theta) = \sum_{i=1}^{G} \left\{ \min \left[\frac{\pi_\theta(o_i|q)}{\pi_{\text{old}}(o_i|q)} A_i, clip \left(\frac{\pi_\theta(o_i|q)}{\pi_{\text{old}}(o_i|q)}, 1-\epsilon, 1+\epsilon \right) A_i \right] \right\} - \beta D_{\text{KL}}[\pi_\theta(\cdot|q) \| \pi_{\text{ref}}(\cdot|q)]$$

Here, π_{old} is the policy before the current update step, ϵ is the clipping hyperparameter, and the KL coefficient β controls the trade-off between maximizing the safety-based reward and preserving the prior knowledge learned during Imitation Learning.

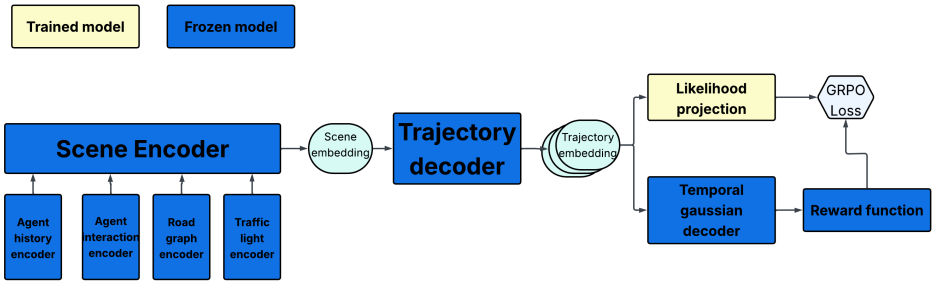


Figure 2: Mechanism of Group Relative Policy Optimization in the Context of Motion Planning. The classification head outputs K probabilities. Rewards are calculated for all K trajectories (the group), and the advantage is derived from relative group performance, eliminating the need for a Critic network.

2.4 Reward Function Design

The reward function $R(o_i)$ is critical for encoding the desired safety and feasibility principles for the ego vehicle, specifically addressing the failure modes observed in long-tail scenarios. The total reward is a weighted combination of two components: one that maintains proximity to expert demonstrations and one that strictly enforces safety constraints.

$$R(o_i) = w_{\text{IL}} \cdot R_{\text{IL}}(o_i) + w_{\text{Coll}} \cdot R_{\text{Coll}}(o_i)$$

2.4.1 Imitation Learning (IL) Component (Naive Approach)

This component ensures the agent maintains fluent, human-like driving behavior in non-critical situations by rewarding similarity to the ground-truth expert trajectory $\mathbf{T}_{\text{GT}}$.

The reward is defined as the negative mean squared error (MSE) between the predicted trajectory $\mathbf{T}_i$ and the closest ground-truth expert trajectory $\mathbf{T}_{\text{GT}}$:

$$R_{\text{IL}}(o_i) = -\frac{1}{T_{\text{pred}}} \sum_{t=1}^{T_{\text{pred}}} \|\mathbf{T}_{i,t} - \mathbf{T}_{\text{GT},t}\|_2^2$$

Fundamentally, this reward formulation utilizes the L2 metric as a direct instructional signal for the classification head. By penalizing displacement errors, the objective forces the model to learn a ranking logic that prioritizes trajectory candidates geometrically aligned with the ground truth.

2.4.2 Collision Reasoning Component (Safety Focus)

This component serves as the primary safety incentive, explicitly penalizing trajectories that violate hard collision constraints and thereby guiding the classification head toward selecting the safest candidate in the generated set.

I define a collision indicator I_{coll} over the prediction horizon T_{pred} as the proportion of future timesteps at which the ego trajectory intersects an object or boundary. For each timestep $\mathbf{T}_i$,

$$I_{\text{coll}}(\mathbf{T}_i) = \begin{cases} 1, & \text{if } \mathbf{T}_i \text{ intersects any object or boundary,} \\ 0, & \text{otherwise.} \end{cases}$$

The collision-based reward for a predicted trajectory is then defined as the negative proportion of collision timesteps:

$$R_{\text{Coll}}(o_i) = -\frac{1}{T_{\text{pred}}} \sum_{i=1}^{T_{\text{pred}}} I_{\text{coll}}(\mathbf{T}_i),$$

which assigns a larger penalty as a trajectory incurs more collisions within the prediction horizon.

By maximizing this reward during GRPO, the classification head is encouraged to prioritize trajectories with minimal collision likelihood, even when doing so requires deviating from expert demonstrations in ambiguous or adversarial long-tail scenarios.

3 Implementation Details

3.1 Dataset and Input Representation

I evaluate our method on a curated subset of the Waymo Open Motion Dataset (WOMD) Ettinger et al. [2021], comprising 15,000 training tracks, 2,000 validation tracks, and 2,000 testing tracks.

Detailed specifications of the selected tracks are available in the project repository. For this study, I utilize 1 second of history (10 frames) to predict 4 seconds of future trajectories (40 frames), sampled at 10 Hz.

The input is represented as a set of heterogeneous modalities:

- **Agent History:** The past states (position, velocity, yaw, size) of the ego vehicle and surrounding agents (vehicles, pedestrians, cyclists).
- **Map Geometry:** Road graph elements including lane centerlines, boundaries, and crosswalks, represented as polylines.
- **Traffic Lights:** The dynamic state of traffic signals associated with lane segments (e.g., red, yellow, green, unknown).

3.2 Wayformer Architecture

Our implementation follows the late Fusion variant of Wayformer, utilizing Latent Query Attention for efficient scene encoding.

3.2.1 Input Modalities and Encoders

To handle the heterogeneity of the driving environment, I decompose the input into four distinct modalities. In our Late Fusion strategy, each modality is processed independently by a dedicated transformer encoder (consisting of linear embedding followed by self-attention layers) before any fusion occurs. This ensures that the specific dependencies within each domain—temporal, social, or spatial—are captured effectively.

The four input branches are:

1. **Agent History:** Encodes the temporal sequence of the ego vehicle's past states (position, velocity, yaw, size) over the 1-second history horizon.

2. **Agent Interaction:** Encodes the states of surrounding vehicles relative to the ego vehicle. This branch explicitly models the social context and local traffic density.
3. **Road Graph:** Encodes the static map geometry, including lane centerlines, road boundaries, and crosswalks, represented as polyline vectors.
4. **Traffic Lights:** Encodes the dynamic state of traffic signals (color, active duration) associated with specific lane segments.

3.2.2 Late Fusion with Modality-Specific Latent Queries

We decompose the scene into four distinct modalities: Agent History, Agent Interaction, Road Graph, and Traffic Lights. For each modality m, we initialize a distinct set of learnable latent queries Q^m_{latent}. The encoding and fusion process proceeds as follows:

1. **Latent-Based Encoding:** For each modality, the specific latent queries Q^m_{latent} (where $|Q^m_{latent}| = 64$) are fed into the modality-specific Transformer Encoder. The encoder performs **Cross-Attention** where the latent queries attend to the raw input features of that modality (acting as Key and Value). This directly extracts a fixed-size feature embedding Z_m for each modality.
2. **Late Fusion (Concatenation):** The resulting fixed-size embeddings from all four modalities ($Z_{\text{hist}}, Z_{\text{inter}}, Z_{\text{map}}, Z_{\text{lights}}$) are concatenated to form the final unified scene embedding $Z_{scene} \in \mathbb{R}^{N_{total} \times D_{emb}}$.

In our implementation, since we utilize 64 latent queries for each of the 4 modalities, the total sequence length of the fused scene embedding is $N_{total} = 64 \times 4 = 256$. This design ensures that specific semantic information is extracted per-domain before being aggregated for trajectory decoding.

3.2.3 Decoder and Hyperparameters

The decoder consists of $L = 4$ transformer layers with $H = 8$ attention heads. I use $K = 6$ learned trajectory queries to generate multimodal proposals. The model is trained end-to-end using the AdamW optimizer with a learning rate of $3e^{-4}$ and a cosine annealing scheduler for 100 epochs.

3.3 GRPO Fine-tuning

Following the pre-training of the Wayformer baseline, I freeze the encoder and the regression head of the TGD. I fine-tune only the Classification Head and the final projection layer using GRPO.

- **Group Size:** I use the $G = K = 6$ trajectory proposals generated by the model as the group for advantage estimation.
- **Clipping:** I use a PPO-style clipping parameter $\epsilon = 0.2$.
- **KL Penalty:** The KL divergence coefficient is set to $\beta = 0.005$ to prevent the policy from deviating excessively from the learned kinematic priors.
- **Reward Weights:** I evaluated two distinct reward configurations to assess the impact of safety constraints: (1) a baseline setting using only the Imitation Learning reward (R_{IL} only), and (2) a safety-prioritized composite setting with weights $w_{\text{IL}} = 1$ and $w_{\text{Coll}} = 5$.
- **Training Config:** The fine-tuning is performed for 20 epochs with a batch size of 64 on a single NVIDIA GTX 4090 GPU.

4 Experimental Results

4.1 Wayformer performance

I analyze the training dynamics of the Wayformer baseline to identify the specific limitations of the Imitation Learning (IL) approach. Figure 3 illustrates the training and validation losses for both the regression and classification heads over 24k steps.

(a) Training regression loss

(b) Validation regression loss

(c) Training classification loss

(d) Validation classification loss

Figure 3: Visualizing the Wayformer loss values during training. (a) Training regression loss, (b) Validation regression loss, (c) Training classification loss, (d) Validation classification loss.

4.1.1 Regression vs. Classification Disparity

A distinct disparity is observed between the learning progress of the two heads:

- **Regression Convergence (Geometric Feasibility):** As shown in the regression plots, the model successfully learns to generate geometrically accurate trajectories. The training regression loss shows a steady, smooth decline, and the validation loss stabilizes below 1.0.
- **Classification Stagnation (Mode Ambiguity):** In contrast, the classification head exhibits significant difficulty during training. The training loss is highly volatile with high variance, and the validation loss plateaus around a value of 1.6 without significant descent. This suggests that the standard IL objective (maximizing the likelihood of the expert mode) is insufficient for distinguishing between the $K = 6$ proposals.

4.1.2 Low Confidence in Decision Making

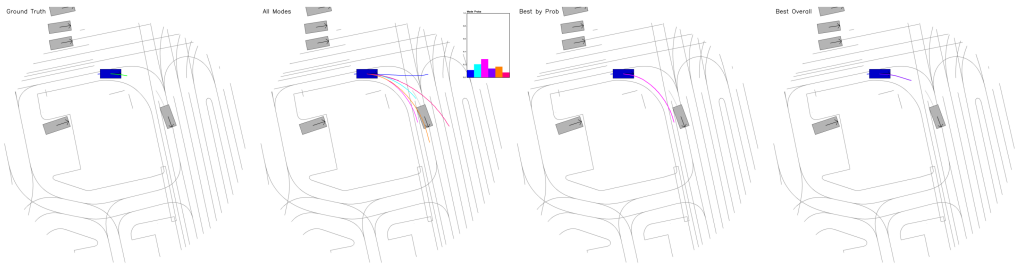
The stagnation of the classification loss at a relatively high value has critical implications for inference, specifically regarding decision confidence. A high classification loss correlates with high entropy in the output distribution.

Instead of confidently assigning high probability to a single "best" trajectory, the model spreads probability mass across multiple modes, resulting in a low-confidence decision boundary. This is visually demonstrated in Figure 4b, where the probability histogram is nearly uniform across conflicting maneuvers. The planner is "uncertain" about whether to proceed straight or turn, assigning similar likelihoods to disparate modes.

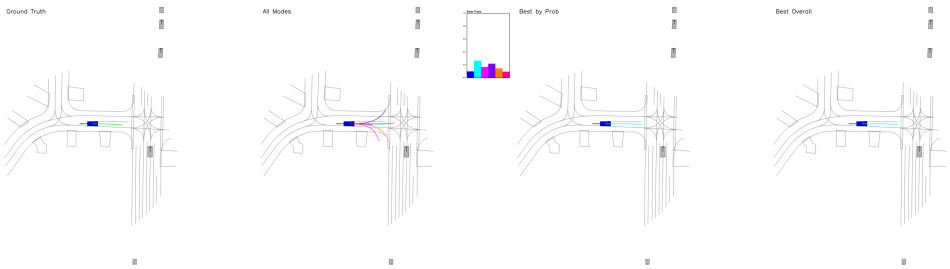
This inability to discriminate effectively purely from supervised data motivates the need for GRPO. By introducing explicit rewards for safety and collision avoidance, GRPO aims to sharpen this distribution, forcing the classification head toward distinct, high-confidence selections.

4.2 Limitations of Supervised Fine-tuning

To investigate whether the mode confusion observed in the baseline could be resolved through extended training, I performed a fine-tuning experiment. In this setup, I froze the weights of the



(a) **Mode Misclassification:** The model generates the correct sharp left turn (Best Overall, Purple) but incorrectly assigns higher probability to the wider turn (Best by Prob, Pink).



(b) **Decision Ambiguity:** The probability distribution (histogram) is high-entropy, with the model splitting confidence nearly evenly across straight and turning modes. This lack of decisiveness leads to unstable planning.

Figure 4: Qualitative visualization of baseline failure modes. The histograms (inset) show the probability distribution over the top modes.

Encoder, the Temporal Gaussian Decoder (TGD), allowing gradients to update *only* the classification head. I continued training using the original u=imitation learning (negative log-likelihood) loss.

The results, presented in Figure 5, reveal a critical limitation of the supervised approach:

- **Loss Stagnation:** Both the training and validation classification losses (Figures 5c and 5d) exhibit no convergence. The loss values fluctuate around 1.6 without any downward trend, indicating that the model has reached a performance plateau.
- **Inability to Disambiguate:** Since the regression loss is stable (as expected with frozen weights, see Figures 5a and 5b), the stagnation in classification suggests that the Imitation Learning signal alone is insufficient to further distinguish between the multimodal proposals. The expert demonstration represents only one realization of many valid futures; therefore, the model cannot confidently suppress alternative plausible modes based solely on likelihood.

This failure of supervised fine-tuning confirms that the Classification Head requires a fundamentally different training signal—one based on explicit quality and safety outcomes rather than imitation—justifying the application of GRPO.

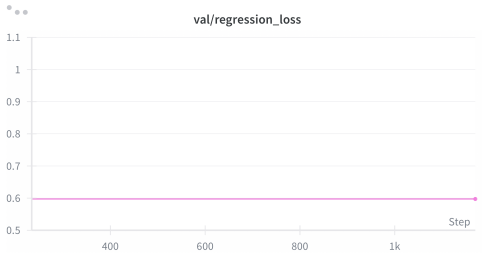
4.3 GRPO fine-tuning

4.3.1 Naive Reward Function Analysis

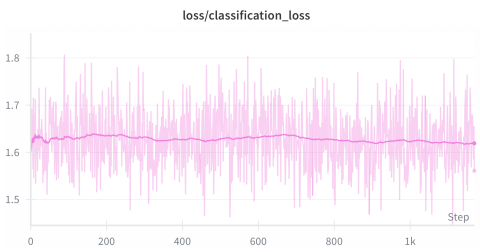
Quantitative Analysis: Reward Maximization I first evaluate the training dynamics of GRPO using the naive L2-based reward function. Since the regression head and encoder are frozen, any improvement in the reward metric must stem solely from the Classification Head assigning higher probabilities to better trajectory proposals.

As shown in Figure 6a, the training loss converges smoothly, indicating that the policy gradient objective is effectively optimizing the classification parameters. More importantly, Figure 6b demonstrates a steady increase in the validation reward. Here, the validation reward is computed as the

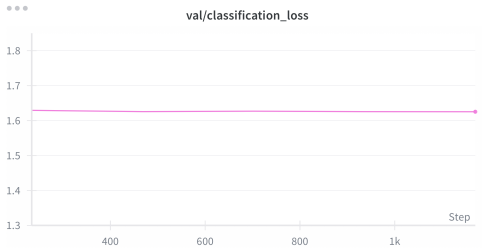
(a) Training regression loss



(b) Validation regression loss



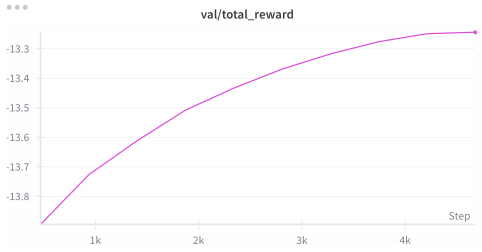
(c) Training classification loss



(d) Validation classification loss

Figure 5: Visualizing the Wayformer loss values during fine-tunining. (a) Training regression loss, (b) Validation regression loss, (c) Training classification loss, (d) Validation classification loss.

(a) Training loss



(b) Validation reward value

Figure 6: Model's behavior during GRPO fine-tuning

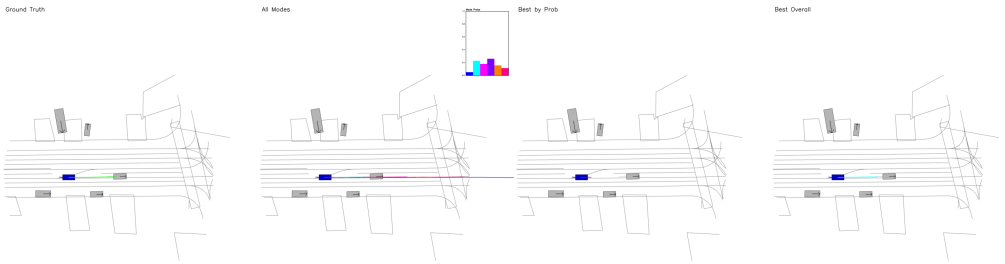
probability-weighted sum of the rewards for all K proposals:

$$R_{val} = \sum_{k=1}^{K} p_k \cdot R(\mathbf{T}_k)$$

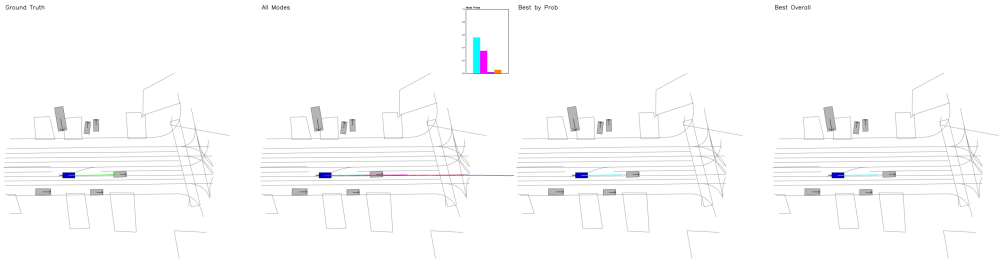
The upward trend confirms that GRPO successfully "pushes" probability mass from poor trajectories toward those with lower L2 displacement errors. The model effectively learns to rank its own generated hypotheses more accurately.

Qualitative Analysis: Trajectory Correction The quantitative improvement translates into tangible corrections in decision-making. Figure 7 (top vs. bottom) illustrates a scenario where the baseline Wayformer (top) suffers from a ranking error—it generates a valid ground-truth-aligned trajectory but assigns it a low probability, selecting a deviant path instead.

After GRPO fine-tuning (bottom), the model correctly re-evaluates the proposals. Without changing the geometry of the available trajectories, the refined classification head shifts the dominant probability mass to the trajectory that aligns with the expert path. This demonstrates GRPO's ability to fix "silent failures" where the planner has the correct solution in its latent space but fails to select it.

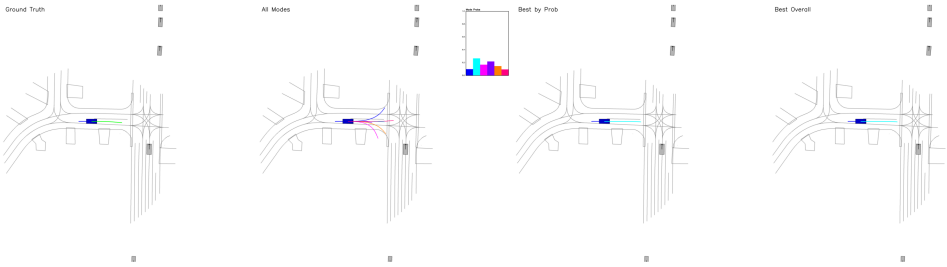


(a) Wayformer output

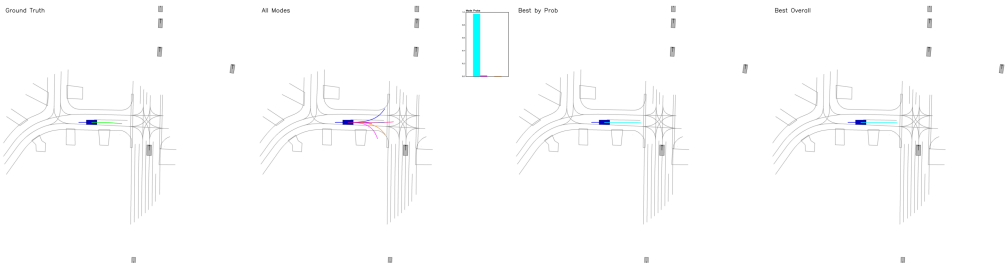


(b) Wayformer w/ GRPO output

Figure 7: GRPO corrects trajectory choice



(a) Wayformer output

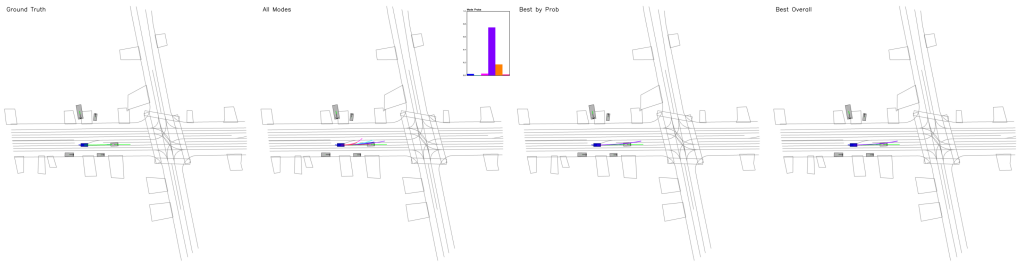


(b) Wayformer w/ GRPO output

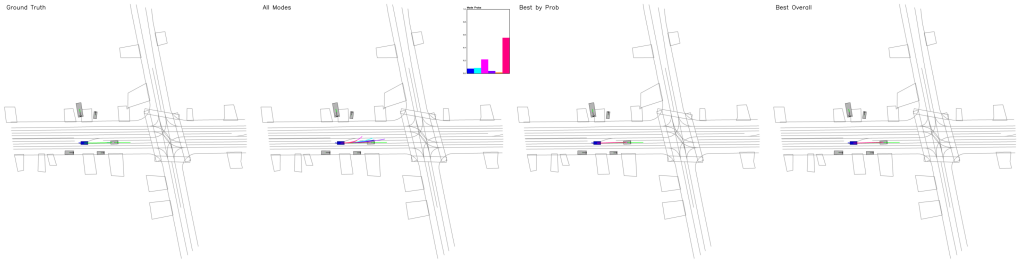
Figure 8: GRPO increase the confidence of trajectories distribution

Qualitative Analysis: Confidence Sharpening In addition to correcting distinct errors, GRPO significantly reduces decision ambiguity. Figure 8 compares the probability distributions of the baseline and the fine-tuned model in a high-uncertainty scenario.

The baseline output (top) exhibits high entropy, with probability mass spread nearly evenly across multiple diverging modes. This lack of decisiveness can lead to unstable planning behavior. In contrast, the GRPO-tuned model (bottom) produces a much sharper distribution. By optimizing for the expected reward, the model suppresses low-quality modes and concentrates confidence on the



(a) Wayformer output



(b) Wayformer w/ GRPO output

Figure 9: Synergistic Reward Alignment: The collision-free trajectory also minimizes displacement error, reinforcing the expert demonstration.

single most optimal trajectory. This "peaked" distribution indicates a higher degree of certainty and a more robust decision boundary.

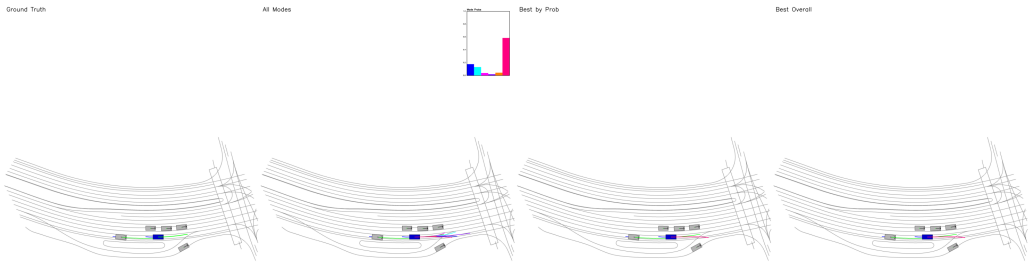
4.3.2 Collision Reasoning Reward Analysis

I further evaluate the model's performance when the reward function is augmented with the collision penalty term ($w_{Coll} = 5, w_{IL} = 1$). This configuration explicitly penalizes trajectories that intersect with obstacles, prioritizing safety over pure kinematic imitation.

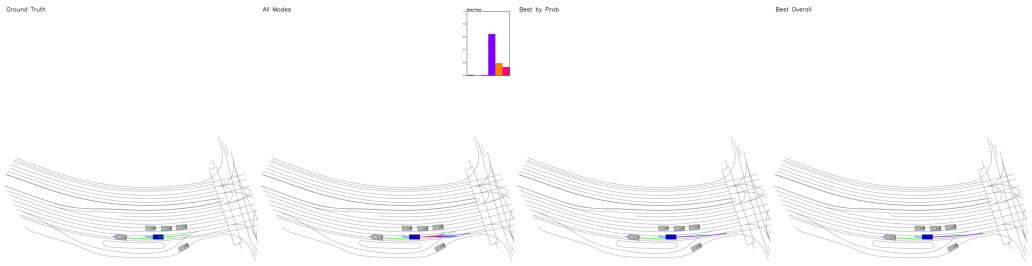
Scenario 1: Synergistic Improvement (Safety + Accuracy) Figure 9 illustrates a scenario where the collision reasoning reward leads to a strictly better outcome in both dimensions. In the baseline prediction (Figure 9a), the model selects a trajectory that likely clips a static obstacle or boundary, resulting in a high collision risk. After GRPO fine-tuning (Figure 9b), the model shifts its high-probability selection to a safer, obstruction-free path. Crucially, in this instance, the safe path also happens to align more closely with the ground truth, reducing the L2 displacement error. This demonstrates that for clear-cut maneuvering decisions, penalizing collisions reinforces the expert's logic, leading to synergistic improvements in both safety and imitation metrics.

Scenario 2: The Safety-Imitation Trade-off Figure 10 presents a more complex case where the objectives of safety and imitation conflict. The baseline model (Figure 10a) predicts a trajectory that minimizes the L2 distance to the ground truth. However, closer inspection reveals a critical flaw: while spatially close to the expert's path, this trajectory has a heading vector directed dangerously toward an adjacent vehicle. It mimics the expert's position but fails to account for the dynamic risk.

The GRPO-tuned model (Figure 10b) successfully suppresses this dangerous mode, selecting a trajectory that explicitly diverts away from the neighboring car. This correction comes at a cost. The safe trajectory is spatially farther from the ground truth than the risky one, resulting in a higher L2 error. This demonstrates the necessity of the collision-reasoning reward: it forces the planner to prioritize a safe heading over geometric imitation, rejecting "accurate" but unsafe trajectories that would otherwise satisfy a standard supervised loss.



(a) Wayformer output



(b) Wayformer w/ GRPO output

Figure 10: Safety-Imitation Trade-off: The model deviates from the expert trajectory to satisfy hard safety constraints, prioritizing collision avoidance over L2 minimization.

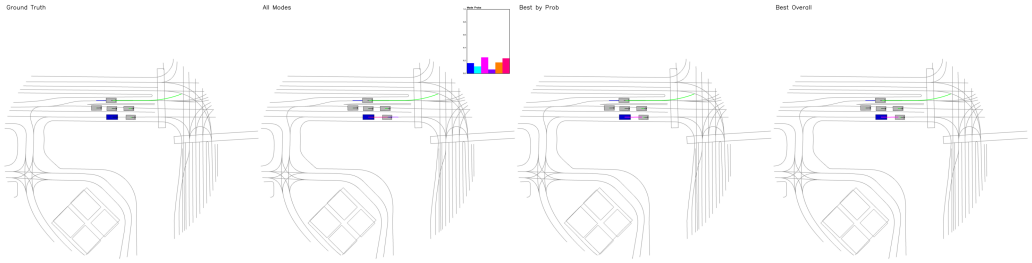


Figure 11: Qualitative visualization of the Wayformer model output in a scenario having red traffic light signal.

5 Discussion

Our experiments suggest that GRPO acts as a promising mechanism for refining the decision boundary of the Wayformer baseline. However, the current open-loop evaluation setup presents certain limitations regarding long-horizon consistency and rule adherence. Below, I discuss three potential avenues to address these challenges and move toward more robust, constraint-aware planning.

- **Formalizing Logical Constraints and Stationarity:** Data-driven planners frequently exhibit non-stationary behaviors in strictly static scenarios, such as "creeping" at red lights (Figure 11). This occurs because the model approximates the moments of a noisy expert distribution rather than learning discrete logical states. To address this, reinforcement learning can be hybridized with explicit constraint interpreters. Methodologies such as those proposed by Yang et al. [2021] map discrete, free-form safety constraints directly into the policy's state-space. Adapting this "constraint interpretation" layer would allow GRPO to enforce hard stationarity rules (e.g., $v = 0$ if $Light = Red$) that are difficult to enforce through scalar reward shaping alone.
- **Physics-Informed Knowledge Injection:** While Imitation Learning effectively captures the *what* of driving, it often fails to encode the *why*—the underlying physical and social dynamics. Reinforcement Learning offers a pathway to inject this unobserved domain

knowledge directly into the policy. This approach parallels physics-informed strategies Liao et al. [2024], where fundamental kinematic principles are integrated into the learning objective to handle missing or noisy observations. By similarly injecting symbolic priors—such as ride comfort metrics or interaction potential fields—into the GRPO reward structure, we can transfer engineering knowledge into the agent, ensuring physical consistency even in edge cases where expert data is sparse.

- **Mitigating Covariate Shift via Closed-Loop Simulation:** The current evaluation protocol relies on open-loop metrics, which are insufficient for capturing the compounding errors typical of sequential decision-making. A pivotal next step is the migration to high-fidelity, closed-loop simulation benchmarks like **nuPlan** Caesar et al. [2021]. Unlike static datasets, closed-loop environments expose the agent to the long-term consequences of its drift, effectively addressing the covariate shift problem. In this setting, a minor violation at a red light evolves into a critical safety failure (e.g., intersection encroachment), providing a naturally adversarial survival signal that fosters a more robust and reactive policy.

References

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.

Felipe Codevilla, Eder Santana, Antonio M López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9329–9338, 2019.

Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9710–9719, 2021.

Haicheng Liao, Chengyue Wang, Zhenning Li, Yongkang Li, Bonan Wang, Guofa Li, Chengzhong Xu, et al. Physics-informed trajectory prediction for autonomous driving under missing observation. *Available at SSRN*, 4809575, 2024.

Yiren Lu, Justin Fu, George Tucker, Xinlei Pan, Eli Bronstein, Rebecca Roelofs, Benjamin Sapp, Brandyn White, Aleksandra Faust, Shimon Whiteson, et al. Imitation is not enough: Robustifying imitation with reinforcement learning for challenging driving scenarios. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7553–7560. IEEE, 2023.

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)*, pages 2980–2987, 2023. doi: 10.1109/ICRA48891.2023.10160609.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Tsung-Yen Yang, Michael Y Hu, Yinlam Chow, Peter J Ramadge, and Karthik Narasimhan. Safe reinforcement learning with natural language constraints. *Advances in Neural Information Processing Systems*, 34:13794–13808, 2021.