2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2025)
October 19-25, 2025. Hangzhou, China

HiTail: Hierarchical Neural Planner for Adaptive and Flexible Long-Tail Trajectory Planning

Shenghong Zhang[1], Xiangyu Zhou[1] and Xiao Li[1]

***Abstract*— A planner for autonomous vehicles must be capable of operating in diverse and complex real-world environments. However, learning-based planners often struggle with limited generalization due to the long-tail distribution in datasets. Moreover, the black-box nature of neural networks limits their interpretability and complicates the integration of explicit rules. In this work, we propose a hierarchical neural trajectory planner that takes the bird's-eye view (BEV) rasters as input. The planner operates in two hierarchical phases: first, spatial proposals are sampled from a policy generated from interpretable learned reward maps, and second, learnable temporal velocity profiles are assigned to the spatial proposals using clothoid curves. We conduct training and closed-loop simulation on the nuPlan dataset. The results demonstrate that our proposed planner outperforms other learning-based methods, exhibiting superior adaptability in *long-tail scenarios*. Additionally, we explore the flexibility of our planner in integrating manually defined rule sets. Project website: https://iunone.github.io/HiTail**

I. INTRODUCTION

Autonomous driving in urban environments presents significant challenges due to the complexity of diverse interactions and the constraints imposed by traffic regulations. A common consensus is that manually designed cost functions or simple lane-following mechanisms with hard collision avoidance constraints are inadequate for robust motion planning. In contrast, data-driven approaches, such as imitation learning-based planners that utilize large-scale expert demonstrations, have shown considerable potential [1]. However, an intriguing result from the CVPR 2023 Autonomous Driving nuPlan Challenge [2] contradicted this consensus: the rule-based PDM planner [3] outperformed its learning-based counterparts. In this paper, we focus on addressing the following questions: "*(1) Can a learning-based planner exhibit superior adaptability in long-tail scenarios, where rule-based failures occur? (2) Can we employs hierarchical planning to a learning-based planner to mitigate shortcut learning while ensuring geometrically explainable spatiotemporal decision-making?*"

The authors of [4] argue that the nuPlan benchmark fails to incorporate sufficiently challenging driving scenarios, limiting its ability to evaluate planner performance in complex real-world conditions. Moreover, [5] highlights that large datasets may overlook critical rare situations, thereby misrepresenting a model's capacity to handle the "long-tail" problem. In this paper, we define long-tail scenarios as failure cases where rule-based methods struggle. We observe that the winning PDM method performs well in situations requiring minimal lateral movement, such as following a lane by adjusting speed via the intelligent driving model (IDM) [6] while maintaining relative position with other agents or using proportional-derivative control to avoid collision. However, in real-world scenarios, human drivers frequently need to take detours, overtake, change lanes, or adjust trajectories to avoid collisions and optimize movement. We refer to these situations requiring significant longitudinal movement as long-tail scenarios in the context of driving behavior.

On the other hand, Cheng et al. [7], [8] highlight the challenges of shortcut learning and causal confusion in learning-based methods, which significantly impair the generalization capabilities of these models. A promising approach to addressing these challenges is the decomposition of the causal factors underlying driving behavior. Directly mapping environmental information to waypoints often results in high-dimensional problems that require vast amounts of data for effective learning. An ideal dataset should capture a wide range of diverse conditions to ensure robustness. Furthermore, the implicit modeling of complex dependencies complicates the stable and reliable training of such models. Some approaches decompose the task into a high-level goal-generation module and a low-level goal-conditioned motion policy [9], [10], allowing the motion policy to align with expert-intended behavior. However, this strategy overlooks an essential step in human driving logic — the process of "applying acceleration or braking after determining the direction" — which presents challenges in mitigating biases introduced by inconsistent expert demonstrations under similar conditions. Other methods explore the use of large language models to break down high-level instructions into more granular decision-making processes, which are subsequently mapped into control signal spaces [11], [12]. However, these methods still suffer from a lack of explicit geometric information and fail to adequately align with spatial geometry.

To address these challenges, we propose a hierarchical approach to trajectory planning, decomposing the task into two distinct phases: spatial path planning and speed planning for a given spatial path, both with interpretable intermediate representations. In the first phase, the spatial planner generates geometrically interpretable intermediate representations in the form of reward maps, from which policy proposals are sampled. In the second phase, the neural temporal planner

*This work was supported in part by the National Key R&D Program of China (Grant No. 2024YFB4707400). This research is also funded by NSFC grant #52405029.

Shenghong Zhang, Xiangyu Zhou, and Xiao Li are with the School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China {zsh000, xiangyuzhou, sjtu_lixiao}@sjtu.edu

takes each spatial proposal as input and generates temporal velocity profiles, which are then integrated with the spatial path as clothoid curves. Finally, a simple scoring mechanism is employed to select the optimal trajectory based on the official nuPlan metrics.

To summarize our contributions, we:

- propose a neural trajectory planner that hierarchically learns *interpretable* spatial decision and temporal velocity control profiles;
- demonstrate that the proposed planner can *flexibly* integrate manually defined rules, enabling adaptation to different driving styles;
- evaluate on the nuPlan closed-loop simulation environment and show improved *adaptability* in **long-tail** scenarios.

II. RELATED WORK

Hierarchical trajectory planning. Hierarchical decomposition in trajectory planning is commonly realized through distinct stratification philosophies. While recent vision-language model-based approaches [12] operate at an abstract semantic level by decomposing tasks into meta-actions and language commands, our framework adopts a geometrically interpretable decomposition by segmenting waypoint generation into two core components: optimal spatial path search and adaptive velocity profile adjustment. For spatial path search, classical methods such as A*, dynamic programming and sampling-based approaches generate candidate trajectories by optimizing various cost functions within lattice-based [13], polynomial spline [14], or graph-based path spaces [15]. For adaptive velocity profile adjustment, traditional techniques optimize velocity profiles, such as trapezoidal speed profiles, using manually designed cost frameworks [16] or directly based on some task-specific empirical distribution cost functions [17]. Among the most closely related works, [15] integrates an A*-based route graph search with traversal time as a cost metric and employs a neural network-based displacement prediction in 1D space using Gaussian mixture models. However, their lane-centric route search approach inherently limits applicability in scenarios that require temporary lane deviations—an essential capability for handling real-world long-tail situations. In contrast, our work adopts a hierarchical decomposition strategy but with a different focus. Specifically, we aim to learn the lateral and longitudinal speed-time series distributions conditioned on spatial plans, addressing challenges such as shortcut learning and mode collapse. This approach enhances the robustness and adaptability of our trajectory planning framework, enabling it to better navigate diverse and dynamic real-world environments.

Interpretable Autonomous Driving. Learning-based autonomous driving systems have shown considerable promise in real-world applications. However, given the critical need for transparency and reliability in safety-critical environments, there is increasing research interest in enhancing the interpretability of these "black-box" models. One category of end-to-end methods focuses on producing intermediate perception outputs, such as semantic maps [18], bounding boxes [19], and bird's-eye-view (BEV) masks [20]. Additionally, some approaches learn semantically related representations, such as cost volumes [21] and risk maps [22], optimized for the driving task using max-margin or max-entropy frameworks. More recently, advancements in visual language models have been applied to generate natural language descriptions of the driving environment, contextual risks, and surrounding objects [23]. These models infer vehicle maneuvers and provide action reasoning, thus enhancing interpretability. However, most of these works focus solely on providing interpretable outputs without leveraging interpretability to manipulate the distribution of waypoints in planning outputs directly. This limitation significantly undermines the practical value of interpretability in real-world applications. In contrast, our approach addresses this gap by employing a spatiotemporal geometric decomposition that enables the model to assign higher rewards to optimal positions within a grid-based semantic space while simultaneously learning to allocate appropriate temporal velocities along the planned path. This mechanism not only facilitates flexible control over the planning output but also allows for the intuitive integration of artificial rules by directly processing the intermediate geometrically interpretable representations.

III. BACKGROUND

A. Max Entropy Inverse Reinforcement Learning

In this section, we provide a concise overview of the IRL framework [24], which plays a vital role in the training process for the learned reward map. A discrete-state, discrete-action Markov Decision Process (MDP) can be defined as $\mathcal{M} = \{S, A, T, r\}$, where S represents the set of states on a 2D grid, and $A = \{\text{left}, \text{forward}, \text{right}, \text{backward}, \text{end}\}$ is the set of actions. The transition function $T : S \times A \to S$ deterministically maps each state-action pair to a subsequent state, while the reward function $r : S \to \mathbb{R}$ assigns a scalar reward to each state. Following the maximum entropy principle, the probability of a state-action trajectory $\tau = \{(s_1, a_1), (s_2, a_2), \ldots, (s_K, a_K)\}$ is proportional to the exponential of its total reward.

$$P(\tau) = \frac{1}{Z} \exp \sum_{i=1}^{K} r(s_i) \tag{1}$$

where, Z is the normalizing constant. MaxEnt IRL seeks to learn a reward function $r_\theta(s)$, parameterized by θ, based on features of each state s. The goal is to learn the reward function that maximizes the likelihood of demonstrations $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_N\}$.

$$\arg\max_\theta \mathcal{L}_\theta = \arg\max_\theta \sum_{i=1}^{N} \log \left(\frac{1}{Z_\theta} \exp \left(r_\theta(\tau) \right) \right) \tag{2}$$

The gradient of the above objective function can be approximated as follows:

$$\frac{d\mathcal{L}_\theta}{d\theta} = \sum_{\tau=1}^{N}(D_e(\tau) - D_\theta(\tau))\frac{dr_\theta(\tau)}{d\theta} \tag{3}$$

where D_e represents the observed feature counts from demonstrations, and D_θ denotes the learner's expected feature counts, which can be expressed through state visitation frequencies (SVFs), a kind of cumulative probability distribution of each position in the grid world.

B. Clothoid Curves for Trajectory Planning

Clothoid curves, also known as Euler or Cornu spirals, are widely used in trajectory planning due to their ability to ensure smooth transitions between straight and curved segments. This results in continuous curvature, enhancing both stability and ride comfort [25]. These properties make Clothoids particularly suitable for automotive and robotics applications where smooth, feasible paths are crucial.

The curvature κ of a Clothoid increases linearly with the arc length ξ from the curve's starting point, following the relation $\kappa(\xi) = \dot{\kappa}\xi + \kappa_0$, where $\dot{\kappa}$ is the curvature rate of change, and κ_0 is the initial curvature. This linear change allows for gradual transitions between different path segments. The parametric equations defining a Clothoid $C(\xi) = (x(\xi), y(\xi))$ are:

$$\begin{aligned} x(\xi) &= x_0 + \int_0^\xi \cos\left(\frac{\dot{\kappa}}{2}u^2 + \kappa_0 u + t_0\right) du \\ y(\xi) &= y_0 + \int_0^\xi \sin\left(\frac{\dot{\kappa}}{2}u^2 + \kappa_0 u + t_0\right) du \end{aligned} \tag{4}$$

where, (x_0, y_0) is the starting point, and t_0 denotes the initial tangent angle. When $\dot{\kappa} = 0$, the curve reduces to a circle arc, and when $\kappa = 0$, it becomes a straight segment.

Given a pair of starting and final points, starting tangent and curvature, [26] provides a fast and compact numerical solution to solve the forward problem to get the parameters $\dot{\kappa}$. Thus, we can uniquely determine the spatial position of the continuous curve. Then discretized trajectory points can be obtained by attaching $\{\xi_t\}$ at different timestamps.

IV. Hierarchical Neural Planner

In this section, we introduce our proposed planner with its three key components: (1) a UNet-based reward generator that learns interpretable intermediate representations for planning and outputs spatial proposals; (2) a motion predictor that recurrently learns and predicts dynamic occupancy grids for subsequent planning tasks; and (3) a neural throttle decoder that produces temporal velocity profiles, which are combined with the corresponding spatial proposals to form discrete trajectory points. Finally, a simple scorer calculates the official metrics provided by nuPlan to select the optimal trajectory for tracking within the closed-loop simulator. The overall architecture is depicted in Figure1.

A. Future Dynamic Occupancy Predictor

Dynamic occupancy grids capture spatial and temporal information over a specified time horizon, with each cell encoding the precise timestamp of an agent's presence at a given location. This representation enables the modeling of agent movements over time. Stationary agents are excluded from the map to ensure causal consistency and focus on relevant dynamics. The grid values are normalized to the range $[0, 1]$, reflecting the relative duration of occupancy within the time window. Representing motion prediction in raster form preserves the multimodality of forecasts in a more compact and interpretable format.

The input data comprises rasterized Bird's Eye View (BEV) imagery spanning a 2-second historical window and a high-definition (HD) map. BEV images are encoded into 3-channel tensors, representing the position, x-velocity, and y-velocity of surrounding vehicles and other traffic participants, such as pedestrians and bicyclists. To optimize memory usage and computational efficiency, temporal information is stacked along the channel dimension, avoiding the need for 3D convolutions. Semantic segmentation maps encode various elements of the traffic environment into distinct binary channels, capturing positional details for features like lanes, intersections, stop lines, crosswalks, and parking areas. Additionally, traffic light information is encoded using a string line map across three channels, enabling reasoning about traffic flow and interactions, particularly at intersections.

We propose a future dynamic occupancy grid predictor structured within the encoder-decoder paradigm. The encoder backbone is adapted from the detection network described in [27], comprising four convolutional blocks that sequentially downsample the input size to 1/8 and then resize it to 1/4 to capture multi-scale traffic scenario context features. These extracted scenario context features then serve as the initial hidden state for a decoder based on Convolutional Gated Recurrent Units (ConvGRU) [28]. In the first time step, the decoder takes the historical dynamic occupancy grid as input. It relies on the predicted grid from the previous step for subsequent steps. At each step, the hidden states generated by the ConvGRU are upsampled and refined through deconvolutional layers, ultimately producing the dynamic occupancy grid for the current time step. The recurrent nature of this design ensures temporal consistency in forecasting future occupancy maps.

Since the dynamic occupancy grid values are normalized within $[0, 1]$, binary cross-entropy (BCE) loss is adopted instead of regression loss. BCE loss effectively quantifies the divergence between predicted probabilities and actual occupancy states, treating the grid values as probabilities. This approach enhances the model's capacity to distinguish between occupied and unoccupied states, facilitating an adaptive learning process. The loss is defined as:

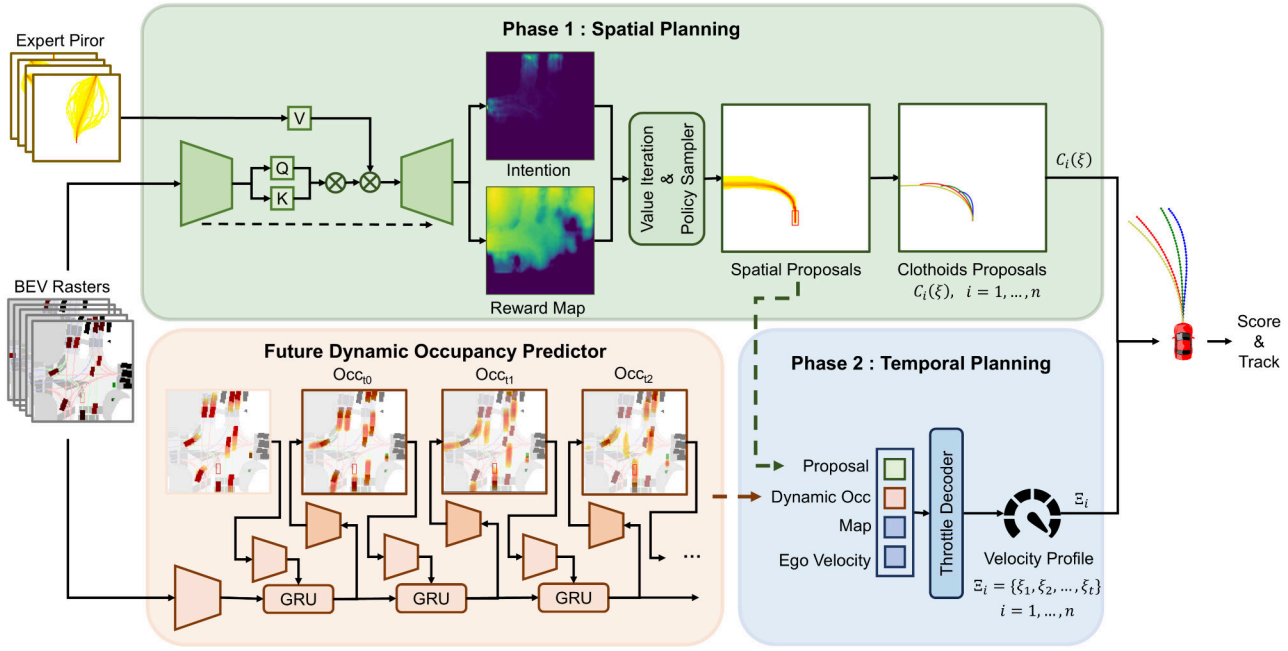

Fig. 1: Overview of the proposed planner. The planner comprises of three main components: (1) a UNet-based reward generator that learns interpretable intermediate representations for planning and outputs spatial proposals; (2) a motion predictor that recurrently learns and predicts future dynamic occupancy grids for subsequent planning tasks; and (3) a neural throttle decoder that produces temporal velocity profiles, which are combined with the corresponding spatial proposals to form discrete trajectory points. Finally, a simple scorer calculates metrics provided by nuPlan to select the best trajectory for tracking within the closed-loop simulator.

$$\mathcal{L}_{occ} = \sum_{t=1}^{T} \sum_{(x,y)\in I} o_{x,y}^{t} \log \hat{o}_{x,y}^{t} + (1 - o_{x,y}^{t}) \log(1 - \hat{o}_{x,y}^{t}). \tag{5}$$

where $o_{x,y}^{t}$ denotes the dynamic occupancy value at position (x, y) within the region of interest I at time step t.

B. Spatial Planning

Inspired by [29], the proposed approach generates rewards from BEV inputs and applies value iteration to compute an optimal spatial policy. An expert trajectory prior grid is introduced to enhance the model's learning capability. This grid encodes the distribution of empirical states with identical endpoints into a single raster. Each channel aggregates trajectories sharing a common starting point but diverging in their endpoints, thereby providing a structured prior for planning.

The reward generation task is formulated as an image-to-image translation problem, addressed using a customized U-Net architecture [30]. Following [31], features extracted from the current BEV input and map are treated as query and key in a self-attention mechanism, facilitating the weighting of features at each spatial position. This weighted feature map is subsequently multiplied with the expert prior raster to aid in reward prediction.

The value iteration method, detailed in Algorithm 1, incorporates modifications to restrict goals to the grid boundaries, mitigating ambiguities arising from varying trajectory durations in the SVFs from expert demonstrations. Specifically, Algorithm 1 involves solving for the policy, given the reward function. $\pi(a|s)$ represents the probability of taking action a, given state s, based on the advantage of each action. At each iteration, the policy is updated by normalizing these advantages. The key steps are summarized as follows:

Algorithm 1 Value Iteration

Inputs: reward map $\mathcal{R}$; intention map $\mathcal{I}$
$V_{\mathcal{R}} \leftarrow -\infty$
for n=N $\ldots$ 1 **do**
  $V_{\mathcal{I}} \leftarrow \mathcal{I}$
  $Q(s,a) = r(s) + \gamma \cdot V(s') \quad \forall s, a, s' = \mathcal{T}(s,a)$
  $V(s) = \text{logsumexp}_a Q(s,a) \quad \forall s$
  $A^{(n)} = Q(s,a) - V(s)$
end for
$\pi_{\tau}^{(n)}(a|s) = \text{softmax}_a(A^{(n)}/\tau)$

During the training phase, the policy is propagated across N steps for all grid states, starting from the initial state distribution and summing over time to get its corresponding SVF $D_\theta(\tau)$, following [29]. For inference, temperature coefficients τ are introduced to adjust policy randomness. Actions are sampled from the policy to leverage better the robustness provided by the maximum entropy framework.

The sampled policy yields a sequence of spatial grid indices, $\mathcal{S} = \{s_1, s_2, ..., s_k\}$. We employ clothoid curves

to map these discrete grid points to continuous space and improve planning stability. To accelerate real-time clothoid fitting, we precompute a family of clothoid curves $\mathcal{C}$ that emanate from a fixed initial pose and terminate at distinct endpoints aligned with the centroids of discretized grid cells. The initial curvatures of these curves are uniformly sampled within a specified range. To address the limitations of clothoids in representing complex trajectories, we partition the grid index sequence $\mathcal{S}$ into n segmentations, $\mathcal{S}_i = \{s_1, s_2, ..., s_{k//i}\}$, where $i = 1, 2, ..., n$. Each segment starts from the same initial point but varies in length. For each segment $\mathcal{S}_i$, the L^2 distance is computed between the segment's grid center points and the predefined clothoid anchors. The optimal clothoid proposal is selected by finding the closest match from a subset of the clothoid set $\mathcal{C}_i$, which consists of curves ending at $s_{k//i}$.

$$C_i(\xi) = \underset{\bar{C} \in \mathcal{C}_i}{\operatorname{argmin}} \sum_{s \in \mathcal{S}_i, s^C \in \bar{C}} \|s^C - s\|_2^2. \tag{6}$$

, where s^C represents points uniformly sampled along the clothoid curve $\bar{C}$. This process ensures smooth trajectory generation while maintaining planning stability.

C. Temporal Planning

Temporal planning focuses on generating temporal control plans for specific spatial proposals while considering future dynamic scenarios. The ego vehicle's velocity profile is modeled using a uniformly accelerated motion pattern, with a neural throttle module predicting acceleration over several discrete time windows. To mitigate noise in the original lateral acceleration data, only longitudinal acceleration from expert demonstrations is used, with mean filtering applied to stabilize the learning target over a specified time window. Additionally, a simple data augmentation strategy is introduced to mitigate discrepancies between spatial proposals derived from policy sampling and expert demonstration proposals. This strategy applies random small offsets to spatial proposals derived from expert demonstrations, particularly those involving significant lateral movement. This approach enhances the model's robustness to the dataset's sparse or incomplete input data.

The temporal planning module employs a standard transformer decoder architecture. Spatial proposal rasters serve as queries, while dynamic occupancy grid rasters, map rasters, and ego vehicle velocity provide keys and values. Each raster input is embedded independently using a dedicated convolutional neural network, followed by positional encoding to retain spatial relationships. During training, dynamic occupancy grids are sampled randomly, either from ground truth data or the outputs of a trained future dynamic occupancy predictor. This strategy balances the learning process by incorporating both ideal and predicted inputs.

V. Experiments and Results

A. Setup

NuPlan dataset. [2] is a large-scale dataset for the closed-loop validation of autonomous driving planners. It contains approximately 1200 hours of real-world vehicle driving data collected from Boston, Pittsburgh, Las Vegas, and Singapore. We use nuPlan's closed-loop simulator for validation. Each simulation scenario involves a 15-second rollout at 10 Hz, utilizing an LQR controller with bicycle kinematic constraints for two-stage trajectory tracking. We examine two scenario selection schemes: (1) Test-Longtail. We introduce a new benchmark consisting of 115 carefully curated driving scenarios, each selected to induce significant lateral maneuvers such as lane changes and collision-avoidance steering, ensuring a comprehensive evaluation of performance in long-tail situations. As shown in Figure 2, the scenarios are classified into three main categories: (i) Done Navigation Task (approximately 25%), focusing on tasks such as early lane changes when approaching turns and overtaking, and (ii) Environmental Constraints (approximately 75%), which address obstacle-related challenges. The latter is further divided into dynamic obstacles (30%, including pedestrian crossings and vehicle cut-ins) and static obstacles (45%, such as illegal parking and vehicle ahead temporarily stops). (2) Val14. A widely used benchmark consisting of 100 scenarios for each of the 14 scenario types defined in the nuPlan leaderboard provides a comprehensive evaluation across standardized conditions.

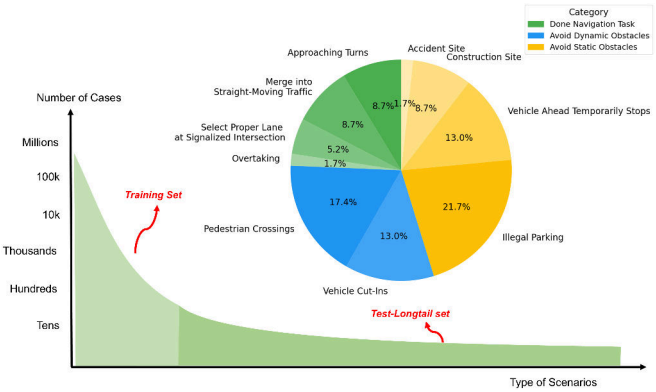

Fig. 2: Distribution of Scenarios in the Proposed Test-Longtail Set. The dataset is categorized into three primary driving behavior types: (1) Done Navigation Task (green), (2) Avoid Dynamic Obstacles (blue), and (3) Avoid Static Obstacles (yellow). This breakdown provides an overview of the relative frequency of different driving scenarios encountered during testing. All models are trained on a standard 1M data split, which contains a large proportion of simple lane-following scenarios.

Methods of evaluation. We employ the following evaluation metrics provided by nuPlan simulator in our method and comparison cases. *No-Collision Rate (NCR)* measures the frequency of collisions between the ego vehicle and other agents. *Progress Rate (PR)* quantifies the distance traveled by the ego vehicle along the expert track. *Area Compliance (AC)* assesses adherence to the permitted driving area. *Comfort* evaluates driving smoothness based on jerk, acceleration, and steering rate, ensuring they remain within a predefined tolerance range. *Non-Reactive Closed-Loop Score (NR-CLS)* represents an aggregate score combining all metrics in the NuPlan simulator. For detailed metric definitions and cal-

TABLE I: Performance Comparison

Benchmark	Method	NR-CLS ↑	No-Collision Rate↑	Progress Rate ↑	Area Compliance↑	Comfort ↑
Test-Longtail	**PDM** [3]	54.7	90.0	46.3	**97.6**	91.8
	Gameformer [32]	57.5	79.4	46.2	90.6	91.8
	PlanTF [7]	31.4	44.7	74.1	96.1	95.3
	Ours	**75.8**	**90.6**	**80.0**	96.3	**96.5**
Val14	**IDM** [6]	74.9	86.7	82.9	90.4	90.7
	PDM [3]	**92.8**	**98.1**	**92.1**	**99.6**	**95.1**
	Gameformer [32]	78.5	90.9	84.5	95.1	93.9
	RasterModel [2]	66.5	86.8	80.5	84.4	82.9
	PlanTF [7]	84.6	94.2	**90.7**	99.2	93.2
	Ours	**86.5**	**97.4**	88.9	**99.4**	**94.3**

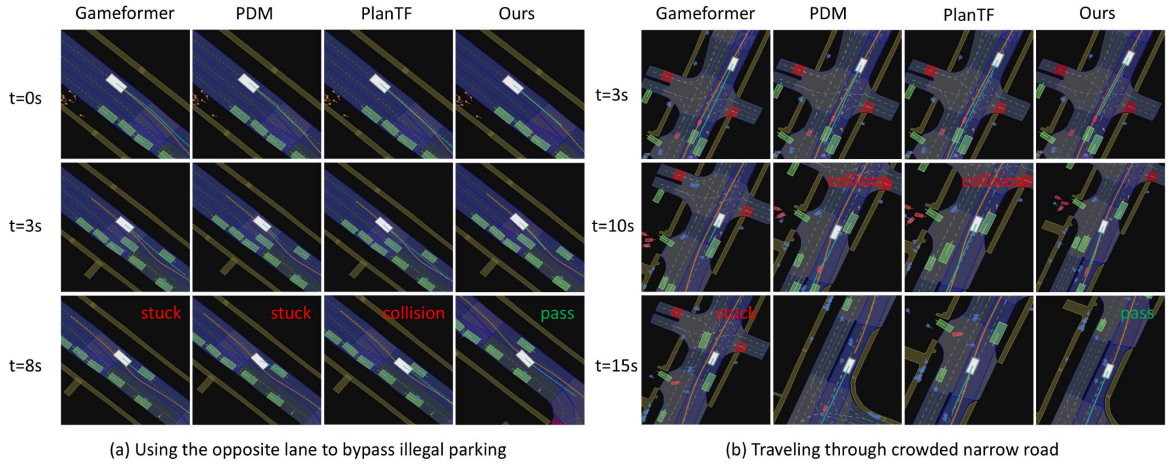

Fig. 3: Cases of Long-tail comparison. (a) and (b) illustrate two distinct scenarios. (a) On a one-way street, ego car should use the opposite lane to bypass the illegal parking ahead and avoid colliding with oncoming vehicles. (b) The ego vehicle must nudge dynamically to avoid a stationary vehicle and moving pedestrians. Only our HiTail planner effectively adjusts its path and success to pass, whereas other planners either stagnate or collide.

culation methods, refer to [2]. It is worth mentioning that these metrics involve trade-offs. For instance, a planner that remains stationary can achieve a perfect NCR but would perform poorly in terms of PR.

Comparison cases. Four state-of-the-art planners are used for performance comparison. *Ours* refers to the proposed method; *PDM* [3] is the winning solution of the 2023 nuPlan Planning Challenge, which is a purely rule-based approach that ensembles the IDM[6] with different hyper-parameters. *GameFormer* [32] is an interactive prediction and planning framework based on the level-k game theory, incorporating a post-optimization step to refine the final trajectory. *PlanTF* [7] is a purely imitation-based planner that addresses planning and simulation gap issues through a state dropout encoder and a reinforcement learning-based adapter. *RasterModel* is a learning-based baseline provided in [2], consisting of a ResNet combined with a fully connected network for direct trajectory prediction.

B. Results and discussion

HiTail planner showcases improved adaptability in long-tail scenarios, achieving at least a 70% improvement in progress rate while ensuring safety. Table I compares the closed-loop performance of our hierarchical planner with SOTA methods across different scenario selection schemes. On the Test-Longtail set, HiTail consistently achieves the highest overall score, excelling in progress rate without compromising safety, which underscores its capability in handling rare and complex cases. On the Val14 regular set, HiTail performs comparably to learning-based planners and surpasses them in key metrics but slightly lags behind rule-based methods, which are optimized for structured environments. The hierarchical structure of HiTail enhances its ability to navigate spatially complex and dynamic settings. In contrast, PDM often stagnates due to rigid rule dependencies, while other learning-based planners exhibit high collision rates, frequently caused by shortcut-related failures. These findings confirm that hierarchical planning mitigates shortcut dependencies, improving generalization in challenging conditions. Figure 3 illustrates two representative long-tail scenarios. In (a), the ego vehicle must use the opposite lane to bypass an obstruction while avoiding oncoming traffic. Gameformer and PDM, constrained by lane-following rules, become stuck, while PlanTF suffers from mode collapse, leading to a collision. Only HiTail successfully navigates the situation. In (b), the ego vehicle must dynamically nudge to avoid both a stationary vehicle and moving pedestrians. Due to pedestrian unpredictability, Gameformer adopts an

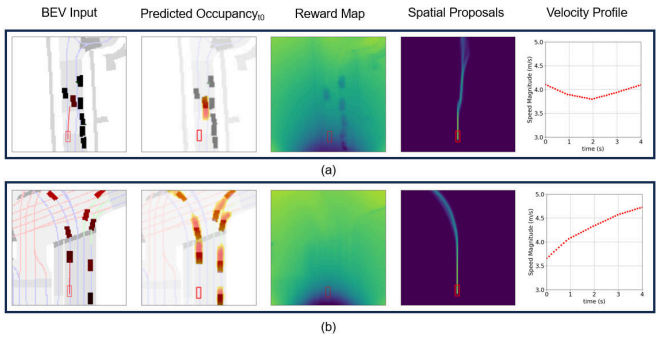

Fig. 4: Explainability Cases. (a) and (b) illustrate two distinct scenarios. For each sub-figure (from left to right): the input bird's eye view (BEV) raster with the planned trajectory, the predicted dynamic occupancy grid for a 1-second future horizon, the generated reward map, the spatial policy proposals, and the velocity profile corresponding to the selected optimal proposal are shown.

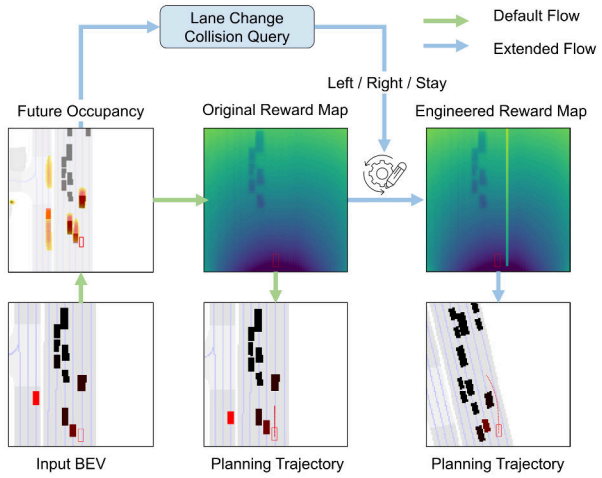

Fig. 5: Flexibility in adjusting driving styles. The default workflow (green arrow) halts at a traffic jam. In contrast, the extended workflow (blue arrow) modifies the reward map to enable a lane change for weaving in traffic, demonstrating adaptive behavior without retraining.

overly conservative strategy and stops, while PDM and PlanTF fail to account for spatial constraints, resulting in collisions. HiTail effectively balances safety and progress, demonstrating its robustness in real-world navigation.

HiTail planner learns interpretable spatial decisions as reward maps that attend to semantically meaningful regions on the BEV. Figure 4 presents intermediate outputs from two distinct scenarios. Each sub-figure includes the input BEV raster with the planning trajectory, the predicted future dynamic occupancy grid, the reward map, the spatial policy derived from value iteration, and the velocity profile attached to the selected clothoid curve. In Figure 4(a), the ego vehicle navigates a narrow, two-lane, one-way road. The reward map assigns low values to the stationary vehicle ahead, effectively identifying it as an obstacle. The spatial policy derived from the reward suggests bypassing the obstruction using the opposite lane. The throttle controller correspondingly reduces speed to avoid collision with the oncoming vehicle. In Figure 4(b), the ego vehicle follows a leading car. Here, the reward map emphasizes the drivable area and centerline while de-emphasizing the moving vehicle ahead, indicating that it does not impede the navigation task. The throttle controller produces a gradually increasing speed profile, ensuring safe following distances while maintaining progress.

HiTail planner enables flexible driving strategy adjustments by modifying the reward map without retraining. Figure 5 illustrates our extended planner. The green arrow represents the default workflow, while the blue arrow shows our extended flow, which involves modifying the intermediate reward map for specific purposes. For the original planner, the ego vehicle slows down to stop when encountering a traffic jam. In contrast, the extended planner flexibly integrates a manual lane-change mechanism within our framework. This lane change query mechanism operates as follows: when the throttle controller issues a deceleration command, it checks adjacent lanes for valid lane-change opportunities using the predicted occupancy and drivable area map. If suitable conditions are identified, the reward map is adjusted to assign a higher reward to the centerline of the target lane. Experimental results confirm that this modification enables the ego vehicle to adopt a more aggressive driving style, navigating around stationary vehicles to maximize forward progress while ensuring safety. Crucially, this adaptive behavior is achieved without retraining, showcasing the planner's flexibility compared to other learning-based methods. A complete demonstration is available in our accompanying video.

TABLE II: Self-ablation study

ID	spatial	occ	temporal	NR-CLS ↑	NC↑	PR ↑
1	✓			66.2	77.2	82.1
2	✓		✓	77.8	90.1	87.8
3		✓	✓	81.1	95.8	78.6
4	✓	✓	✓	86.5	97.4	88.9

The spatial planning aims to maximize progress while ensuring safe distances from stationary vehicles, while the temporal planning enhances safety in dynamic environments by leveraging future occupancy predictions, resulting in a 25% safety enhancement. To evaluate these components, we conducted a self-ablation study with three configurations, as summarized in Table II. The ID1 model includes only the spatial planning module, where the output proposals are directly flattened and processed through a simple multilayer perceptron (MLP) to generate waypoints. The results show that excluding temporal planning significantly reduces the no-collision rate, leading to a substantial decline in overall performance. The ID2 model omits the future occupancy grid predictor, relying solely on the network to infer dynamics implicitly from current BEV data. This configuration highlights the importance of incorporating an independent occupancy prediction module, as its inclusion

notably enhances safety by enabling the planner to anticipate and react to future environmental changes, thereby mitigating collision risks. In the ID3 model, centerline-based proposals replace the spatial planning module, illustrating the advantages of MaxentIRL-based techniques. The results demonstrate that the inductive bias introduced by spatial proposals leads to trajectories more effectively conforming to scene constraints. Overall, the findings underscore that the spatial planning module handles high-level decision-making tasks like lane changes and obstacle avoidance. In contrast, the temporal planning module refines control to ensure smooth and safe navigation. Integrating these two modules effectively addresses distinct challenges in autonomous driving, resulting in a more robust and adaptive planning framework.

VI. Conclusion

In this work, we propose a hierarchical neural planner that is adaptive and flexible for long-tail scenarios. We introduce spatial planning, which learns an interpretable reward map, and temporal planning, which learns corresponding velocity profiles. Our proposed planner outperforms other learning-based methods, showing superior adaptability in long-tail scenarios. Additionally, we explored our planner's flexibility, highlighting its ability to integrate manually defined rule sets and adapt to different driving styles.

References

[1] M. Bansal, A. Krizhevsky, and A. Ogale, "Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst," *arXiv preprint arXiv:1812.03079*, 2018.

[2] K. T. e. a. H. Caesar, J. Kabzan, "Nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles," in *CVPR ADP3 workshop*, 2021.

[3] 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, pp. 1268–1281.

[4] M. Hallgarten, J. Zapata, M. Stoll, K. Renz, and A. Zell, "Can vehicle motion planning generalize to realistic long-tail scenarios?" *arXiv preprint arXiv:2404.07569*, 2024.

[5] A. Jain, L. Del Pero, H. Grimmett, and P. Ondruska, "Autonomy 2.0: Why is self-driving always 5 years away?" *arXiv preprint arXiv:2107.08142*, 2021.

[6] 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.

[7] 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.

[8] R. Xin, J. Cheng, and J. Ma, "Planscope: Learning to plan within decision scope does matter," 2024. [Online]. Available: https://arxiv.org/abs/2411.00476

[9] A. Mandlekar, F. Ramos, B. Boots, S. Savarese, L. Fei-Fei, A. Garg, and D. Fox, "Iris: Implicit reinforcement without interaction at scale for learning control from offline robot manipulation data," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4414–4420.

[10] Y. Ding, C. Florensa, P. Abbeel, and M. Phielipp, "Goal-conditioned imitation learning," *Advances in neural information processing systems*, vol. 32, 2019.

[11] Z. Zhang, S. Tang, Y. Zhang, T. Fu, Y. Wang, Y. Liu, D. Wang, J. Shao, L. Wang, and H. Lu, "Ad-h: Autonomous driving with hierarchical agents," 2024. [Online]. Available: https://arxiv.org/abs/2406.03474

[12] X. Tian, J. Gu, B. Li, Y. Liu, Y. Wang, Z. Zhao, K. Zhan, P. Jia, X. Lang, and H. Zhao, "Drivevlm: The convergence of autonomous driving and large vision-language models," *arXiv preprint arXiv:2402.12289*, 2024.

[13] 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," 2018. [Online]. Available: https://arxiv.org/abs/1807.08048

[14] X. Li, Z. Sun, Z. He, Q. Zhu, and D. Liu, "A practical trajectory planning framework for autonomous ground vehicles driving in urban environments," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, 2015, pp. 1160–1166.

[15] E. Bronstein, M. Palatucci, D. Notz, B. White, A. Kuefler, Y. Lu, S. Paul, P. Nikdel, P. Mougin, H. Chen, J. Fu, A. Abrams, P. Shah, E. Racah, B. Frenkel, S. Whiteson, and D. Anguelov, "Hierarchical model-based imitation learning for planning in autonomous driving," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 8652–8659.

[16] Y. Qi, B. He, R. Wang, L. Wang, and Y. Xu, "Hierarchical motion planning for autonomous vehicles in unstructured dynamic environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 496–503, 2022.

[17] M. Herman, V. Fischer, T. Gindele, and W. Burgard, "Inverse reinforcement learning of behavioral models for online-adapting navigation strategies," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3215–3222.

[18] H. Wang, P. Cai, Y. Sun, L. Wang, and M. Liu, "Learning interpretable end-to-end vision-based motion planning for autonomous driving with optical flow distillation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 731–13 737.

[19] W. Zeng, W. Luo, S. Suo, A. Sadat, B. Yang, S. Casas, and R. Urtasun, "End-to-end interpretable neural motion planner," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8660–8669.

[20] S. Teng, L. Chen, Y. Ai, Y. Zhou, Z. Xuanyuan, and X. Hu, "Hierarchical interpretable imitation learning for end-to-end autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 673–683, 2022.

[21] W. Zeng, S. Wang, R. Liao, Y. Chen, B. Yang, and R. Urtasun, "Dsdnet: Deep structured self-driving network," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*. Springer, 2020, pp. 156–172.

[22] S. Jiwani, X. Li, S. Karaman, and D. Rus, "Risk-aware neural navigation from bev input for interactive driving," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5659–5665.

[23] Z. Xu, Y. Zhang, E. Xie, Z. Zhao, Y. Guo, K.-Y. K. Wong, Z. Li, and H. Zhao, "Drivegpt4: Interpretable end-to-end autonomous driving via large language model," *IEEE Robotics and Automation Letters*, 2024.

[24] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey *et al.*, "Maximum entropy inverse reinforcement learning," in *Aaai*, vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.

[25] D. H. Shin, S. Singh, and W. Whittaker, "Path generation for a robot vehicle using composite clothoid segments," *IFAC Proceedings Volumes*, vol. 25, no. 6, pp. 443–448, 1992.

[26] E. Bertolazzi and M. Frego, "G1 fitting with clothoids," *Mathematical Methods in the Applied Sciences*, vol. 38, no. 5, pp. 881–897, 2015.

[27] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[28] nicolas ballas, li yao, chris pal, and A. C. Courville, "Delving deeper into convolutional networks for learning video representations," in *International Conference on Learning Representations*, 2015.

[29] D. Nachiket and T. M. M., "Trajectory forecasts in unknown environments conditioned on grid-based plans," *CoRR*, 2020.

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.

[31] X. Ren, T. Yang, L. E. Li, A. Alahi, and Q. Chen, "Safety-aware motion prediction with unseen vehicles for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 731–15 740.

[32] 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.