LOOKAHEAD-GUIDED CLOSED-LOOP FINE-TUNING FOR MULTI-AGENT TRAFFIC SIMULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Traffic simulation is crucial for the development of autonomous driving systems. Recent autoregressive approaches formulate this task as sequential motion token prediction, yet inherently suffer from covariate shift between open-loop training and closed-loop deployment. Closed-loop fine-tuning (CLFT) mitigates this issue by sampling pseudo expert trajectories. However, most existing token-level methods rely on greedy single-step sampling, and locally closest matches do not guarantee global trajectory alignment. Inspired by multi-token prediction in large language models, we propose Lookahead-Guided Closed-Loop Fine-Tuning (LG-CLFT), a simple yet effective framework that enables long-horizon matching for pseudo expert trajectory construction. Specifically, we introduce auxiliary multi-step prediction heads to approximate future motion token distributions during open-loop pretraining. During CLFT, we leverage this long-horizon prediction capability to perform beam search over future motion token sequences, evaluating their overall alignment with expert trajectories. This lookahead mechanism enables sampling superior pseudo expert trajectories within the model policy distribution. On the WOSAC 2025 leaderboard, LG-CLFT achieves a Realism Meta score of 0.7854 and a minADE of 1.2865, outperforming prior supervised CLFT methods in both simulation realism and trajectory alignment.

1 INTRODUCTION

Traffic simulation plays a critical role in both the development and evaluation of autonomous driving systems (Dosovitskiy et al., 2017). Traditional log replay methods lack the capacity for reactive feedback, while rule-based models such as the intelligent driver model (Treiber et al., 2000) struggle to capture the diversity of agent interaction behaviors. Recently, learning-based approaches (Seff et al., 2023; Philion et al., 2024; Wu et al., 2024) have demonstrated promising capabilities in this domain. These methods aim to learn a policy that generates subsequent agent motions conditioned on historical agent motions and scene context, including map information and traffic light states.

Recent works formulate the simulation policy as multiple agents autoregressively generating motion tokens over time (Wu et al., 2024). However, autoregressive (AR) models are typically trained under teacher forcing on expert trajectories but deployed through self-conditioned closed-loop rollouts over their own generated histories, resulting in a train–deployment discrepancy. This discrepancy induces *covariate shift* (Shimodaira, 2000; Sugiyama et al., 2007; Quiñonero-Candela et al., 2009), causing minor behavioral deviations to accumulate sequentially, eventually driving rollouts away from the nominal distribution and further changing the inputs used for subsequent motion prediction.

To mitigate covariate shift, recent works (Zhang et al., 2025a; Lin et al., 2026; Garcia-Cobo et al., 2026) introduce closed-loop fine-tuning (CLFT) after pretraining, which trains the model on pseudo expert trajectories sampled from its current policy while remaining aligned with offline expert demonstrations. Most existing token-level CLFT methods perform single-step matching: they select the high-probability motion token whose corresponding short-horizon motion is closest to the expert trajectory (Figure 1 (a)). However, using only a short-horizon expert trajectory as the reference cannot ensure the selection of pseudo labels that yield reasonable long-term behavior. As illustrated in Figure 1 (c), a token that achieves the best short-term match changes the subsequent conditioning history and may induce future motions that increasingly diverge from the expert trajectory.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

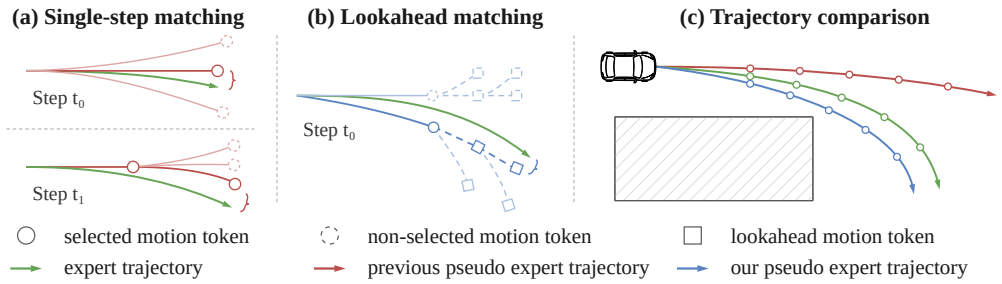

Figure 1: **Greedy vs. lookahead pseudo-label selection.** (a) Greedy matching selects the locally closest motion token. (b) LG-CLFT compares multi-step continuations and selects the first token of the best-aligned sequence. (c) Repeated local matching can lead to trajectory drift, while lookahead improves alignment over repeated matching steps in this schematic example.

We propose **Lookahead-Guided Closed-Loop Fine-Tuning (LG-CLFT)**, which consists of two stages. During open-loop pretraining, we augment the AR model with serial multi-step prediction (MSP) heads. Inspired by multi-token prediction in large language models (Gloeckle et al., 2024), these heads form a causal chain that combines current-scene agent features with preceding token inputs to predict motion-token distributions over multiple future steps (Figure 1 (b)). Through multi-step supervision, this design improves the base AR simulator while enabling efficient long-horizon prediction without explicit autoregressive rollouts. During CLFT, we use the pretrained MSP heads to evaluate the long-term consequences of candidate motion tokens when constructing pseudo expert trajectories. Beam search produces high-probability motion-token sequences over multiple future steps, whose trajectory-level discrepancies from the expert trajectory are then evaluated. The first token of the best-aligned sequence is selected as the pseudo label for the current step, allowing pseudo-label selection to account for long-term trajectory alignment rather than only the next token's local match to the expert trajectory from the current generated state.

Experimental results on the Waymo Open Sim Agents Challenge (WOSAC) benchmark (Montali et al., 2023) demonstrate that LG-CLFT achieves a Realism Meta score of 0.7854 and a minADE of 1.2865 on the 2025 leaderboard test split, indicating more realistic closed-loop behaviors and better alignment to expert trajectories across the generated multi-agent rollouts.

Our main contributions are summarized as follows:

- We introduce serial MSP heads during open-loop pretraining to improve the AR simulator and enable efficient multi-step motion prediction.
- We propose LG-CLFT, which leverages multi-step predictions to guide pseudo expert trajectory sampling through distance-oriented beam search.
- Extensive experiments on the WOSAC benchmark demonstrate improved closed-loop realism and trajectory alignment over prior supervised CLFT methods.

2 Related Work

Multi-agent traffic simulation. Traffic simulation supports autonomous-driving development by allowing agents to respond to a tested driving policy, beyond replaying recorded trajectories (Zhou et al., 2024). Recent learning-based models incorporate behavior modeling (Zheng et al., 2021; Liao et al., 2024a), causal reasoning (Wang et al., 2025a), and cognitive-driven mechanisms (Liao et al., 2024b). The Waymo Open Motion Dataset (WOMD) (Ettinger et al., 2021) provides large-scale multi-agent logs for training models to generate long-horizon joint trajectories that match real-world traffic distributions (Montali et al., 2023). Inspired by language modeling, tokenized simulators (Philion et al., 2024; Seff et al., 2023; Peng et al., 2026; Xu et al., 2026; Xiao et al., 2026) discretize continuous motion and train Transformers (Vaswani et al., 2017) with next-token prediction (NTP) or grouped token prediction for joint closed-loop generation. SMART (Wu et al., 2024) further tokenizes map polylines and agent motions, providing the discrete autoregressive backbone on which we build both open-loop pretraining and closed-loop fine-tuning.

Closed-loop fine-tuning. Closed-loop simulators condition predictions on generated trajectories, causing covariate shift (Shimodaira, 2000; Quiñonero-Candela et al., 2009), compounding errors, and divergence from expert demonstrations (Bengio et al., 2015). CLFT mitigates this by training on samples from the model's own policy distribution. For example, DAgger (Ross et al., 2011) queries an expert on aggregated on-policy states, while DAD (Venkatraman et al., 2015) pairs rollout inputs with recorded targets. For tokenized simulation, CAT-K (Zhang et al., 2025a) constructs expert-proximal rollout states through stepwise matching among high-probability actions and supervises them with recovery targets; RoaD (Garcia-Cobo et al., 2026) directly treats expert-guided rollout actions as demonstrations. UniMM (Lin et al., 2026) uses temporal disentanglement and alignment to address shortcut and off-policy learning across mixture-model variants. ForSim (Chen et al., 2026) explores reward-based fine-tuning by coupling stepwise, interaction-aware forward simulation with group-relative policy optimization. ForSim maintains branch consistency by aligning later predictions with each branch's initial candidate trajectory. LG-CLFT uses supervised candidate selection: serial prediction heads supply short token continuations, which are ranked against the recorded expert future before the first token is selected as the pseudo label for the current step.

Multi-token prediction. Multi-token prediction (MTP) extends NTP by predicting multiple future tokens from the current prefix (Gloeckle et al., 2024); lookahead-based training also supports planning (Thankaraj et al., 2025). Parallel MTP predicts future offsets independently from a shared hidden feature (Gloeckle et al., 2024), whereas serial MTP conditions later predictions on earlier token inputs through a causal dependency chain (DeepSeek-AI, 2024). JTP (Ahn et al., 2025) instead models future tokens jointly through teacher forcing and a representation bottleneck. MuToR (Gerontopoulos et al., 2025) inserts learnable register tokens to predict future targets during pretraining and fine-tuning, while future summary prediction (Mahajan et al., 2026) trains an auxiliary head to predict a compact representation of the longer-term future. Token order prediction (Zuhri et al., 2026) uses a learning-to-rank objective to order upcoming tokens by proximity, providing an alternative to predicting each exact future token. Aynetdinov & Akbik (2025) study training curricula that gradually vary the number of prediction targets, moving from NTP to MTP or in the reverse order in small language models. Multi-token heads can also draft blocks for speculative decoding, which verifies proposed tokens with the target model to reduce serial decoding calls (Leviathan et al., 2023; Chen et al., 2023; DeepSeek-AI, 2024; LLM-Core Xiaomi, 2025).

In contrast, our method integrates multi-step prediction (MSP) into CLFT for traffic simulation: when generating trajectories from the policy distribution, we predict multiple future motion tokens to evaluate long-horizon similarity to expert trajectories, thereby selecting higher-quality trajectories as reliable closed-loop supervision under generated histories rather than recorded expert prefixes.

3 METHODOLOGY

3.1 PRELIMINARIES

Problem formulation. We study *closed-loop generation* for multi-agent traffic simulation. Let C denote the scene context (*e.g.*, HD maps, agent types, and road topology), and $S_t = \{s_t^i\}_{i=1}^M$ the joint state of M agents at time step t, where s_t^i represents the motion state of agent i (*e.g.*, position, heading, and velocity). Given the initial observation S_0 and C, the goal is to generate realistic multi-agent behaviors over a horizon T. We model $P_\theta(S_{1:T} \mid S_0, C)$ autoregressively as the product of transitions $\pi_\theta(S_t \mid S_{<t}, C)$ over $t = 1, \ldots, T$, with scene context C held fixed.

Covariate shift. In temporal AR models, open-loop training and closed-loop deployment use different prefix distributions. Open-loop teacher forcing conditions on the expert history $S_{<t}$ and minimizes the expected sum of per-step prediction losses $\ell(\pi_\theta(\cdot \mid S_{<t}, C), S_t)$ over $(C, S) \sim \mathcal{D}_{\text{expert}}$. During closed-loop deployment, the model instead conditions on its generated trajectory history $\hat{S}_{<t}$ and produces $\hat{S}_t \sim \pi_\theta(\cdot \mid \hat{S}_{<t}, C)$. This mismatch between $\hat{S}_{<t}$ and $S_{<t}$ causes small prediction errors to accumulate, leading to unrealistic long-horizon traffic rollouts.

Closed-loop fine-tuning. CLFT constructs pseudo expert trajectories from model-proposed candidates under generated prefixes, using the logged expert future as a matching reference.

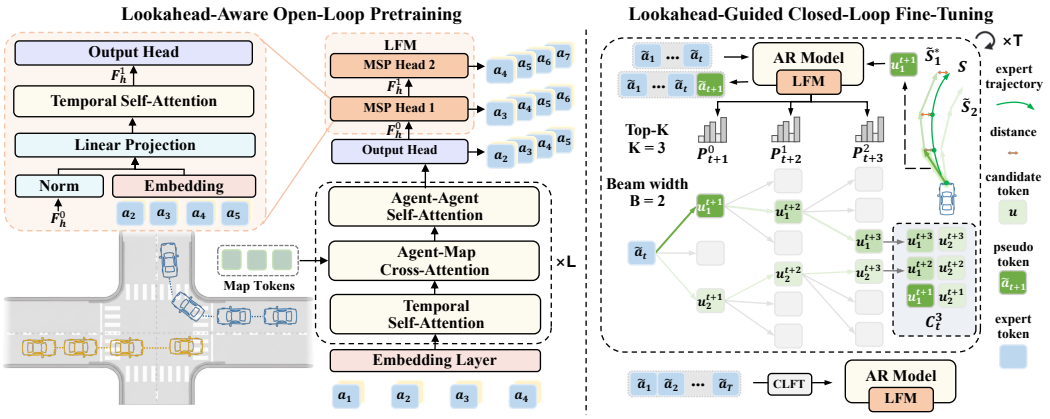

Figure 2: **Overview of our LG-CLFT framework.** *Left:* The SMART backbone augmented with the lookahead forecasting module (LFM), which consists of N serial multi-step prediction (MSP) heads. These heads form a causal dependency chain across prediction depths, enabling efficient multi-step lookahead without explicit rollouts. *Right:* The distance-oriented beam search (DBS) employed in LG-CLFT, which expands candidate action sequences over multiple future steps and selects the current action by minimizing the future-window trajectory distance to the expert trajectory.

Under generated histories, recorded expert tokens need not recover expert states, so matching selects targets relative to generated states. These trajectories form $\mathcal{D}_{\text{pseudo}} = \{(\tilde{S}_{1:T}, C)\}$ for training on closed-loop histories. Existing token-level methods choose the expert-closest next motion from top-K candidates, although a worse local match can yield a better-aligned continuation.

Method overview. LG-CLFT changes pseudo-label selection from immediate token matching to multi-step trajectory alignment. In Figure 2, Stage I trains the backbone and serial MSP heads to propose future motion tokens (Section 3.2). Stage II uses DBS to rank continuations by expert alignment, commits the first token of the best-ranked sequence, and repeats from the updated prefix (Section 3.3). The completed pseudo trajectory supervises all heads, and subsequent target construction uses the updated policy. The two stages share a model but differ in their source of supervision: recorded expert tokens in Stage I and online matched pseudo tokens in Stage II. Lookahead guides training-target construction; WOSAC evaluation uses NTP sampling from the learned policy.

3.2 Lookahead-Aware Open-Loop Pretraining

Autoregressive model. Following SMART (Wu et al., 2024), we use a discrete AR model to parameterize π_θ. We segment continuous agent trajectories at fixed time intervals and convert each segment into a motion token a_t^i using K-disks clustering. We partition map polylines by length into road vectors, which a map encoder converts into map tokens m. The model processes these tokens through a sequence of attention layers. Temporal self-attention captures dependencies in each agent's motion history. Agent–map cross-attention incorporates road topology and geometry into agent representations, and agent–agent self-attention exchanges information among traffic participants. These operations give each agent access to the executed scene history before an MLP projects its final features to next-token logits. At each simulation step, the shared policy predicts a categorical distribution for each agent conditioned on the joint action history and map, factorizing the joint action distribution across agents. DBS searches separately for each agent, conditioned on the executed joint history; it does not enumerate joint multi-agent futures. Below, we omit the agent index and leave the other agents' executed histories implicit in the search notation.

Lookahead forecasting module. The *lookahead forecasting module (LFM)* augments the backbone with N serial MSP heads; head 0 denotes the backbone's NTP head. Let $\mathbf{F}_h^0 = [h_0^0, \ldots, h_T^0]$ be the backbone features and $\mathbf{E} = [e_0, \ldots, e_T]$ the motion-token input features, including geometric context. For $n = 0, \ldots, N-1$, head $n+1$ fuses the preceding features with inputs shifted by $n+1$:

$$\mathbf{F}_h^{n+1} = \text{CausalAttn}_{n+1}(\text{Linear}_{n+1}(\mathbf{F}_h^n \| \text{Shift}_{n+1}(\mathbf{E}))), \tag{1}$$

where $\text{Shift}_k(\mathbf{E})[r] = e_{r+k}$ on valid positions and $\|$ denotes concatenation. Temporal attention uses only the valid causal history of the fused sequence. Here, the feature $\mathbf{F}_h^n[r]$ predicts a_{r+n+1}:

head 0 predicts a_{r+1}, while head 1 additionally consumes e_{r+1} to predict a_{r+2}. At the same anchor r, head 2 incorporates e_{r+2} and predicts a_{r+3}. Each head consumes tokens preceding its own target; the shifted input does not contain the token being predicted. The output distribution is $P_t^n = \mathrm{Softmax}(\mathrm{Out}_n(\mathbf{F}_h^n[t-n-1]))$ for $t > n$, where Out_n is the head's output projection. During Stage I, shifted inputs are teacher-forced expert tokens. During Stage II search, they come from candidate tokens proposed at preceding depths, so each branch supplies its own continuation features without a full backbone rollout. In Stage II, the model is supervised only by the completed pseudo trajectory; candidate continuations discarded during search do not become training targets.

Training objective with entropy regularization. We jointly supervise all heads over the stage-specific training distribution $\mathcal{D}$, with the n-th head predicting target token a_t at time step t. Here, a_t is the expert token in Stage I and a token from the completed pseudo trajectory in Stage II:

$$\mathcal{L}_{\mathrm{sup}} = \mathbb{E}_{(m,a)\sim\mathcal{D}} \sum_{n=0}^{N} \sum_{t=n+1}^{T} \mathrm{CE}(P_t^n, a_t)\,, \tag{2}$$

where $\mathrm{CE}(\cdot,\cdot)$ is categorical cross-entropy; only valid input–target positions contribute. During closed-loop fine-tuning, concentrated token distributions can limit candidate diversity in subsequent pseudo-label construction. We define predictive entropy as $\mathcal{H}(P) = -\sum_{v\in\mathcal{V}} P(v)\log P(v)$, where $\mathcal{V}$ is the motion-token set. The regularizer $\mathcal{L}_{\mathrm{ent}}$ is the negative expected sum of $\mathcal{H}(P_t^n)$ over the same head and time indices as $\mathcal{L}_{\mathrm{sup}}$. The total objective is $\mathcal{L} = \mathcal{L}_{\mathrm{sup}} + \lambda_{\mathrm{ent}}\mathcal{L}_{\mathrm{ent}}$. We set $\lambda_{\mathrm{ent}}{=}0$ during open-loop pretraining and $\lambda_{\mathrm{ent}}{=}0.1$ during closed-loop fine-tuning; the entropy term encourages broader token proposals, while cross-entropy fits the selected pseudo labels.

3.3 Lookahead-Guided Closed-Loop Fine-Tuning

Rolling pseudo-label construction. Starting from $\tilde{s}_0 = s_0$, we construct $\tilde{a}^\star_{1:T}$ under the current policy by repeatedly comparing candidate futures with the expert trajectory. Executing a selected token updates the generated state as $\tilde{s}_{t+1} = f(\tilde{s}_t, \tilde{a}^\star_{t+1})$, where f decodes the token from the current state. Only the first token of each selected continuation is committed. Finite lookahead and beam pruning provide a rolling approximation, with no full-horizon optimality or cumulative-probability guarantee. The expert trajectory supplies the reference for ranking candidates, while candidate probabilities are conditioned on the generated history and map. All candidates start from the generated state, so their expert-distance scores compare continuations of the same executed rollout history.

Distance-oriented beam search. DBS compares up to H future tokens using the backbone and MSP proposals, where $1 \le H \le N+1$. The branching factor K controls proposals per retained prefix, while the beam width B limits the sequences retained across prefixes; they need not be equal. Each motion token spans five frames at 10 Hz, so the default $H{=}2$ compares approximately one second of continuation and commits only the first half-second token. At step t, we initialize $\mathcal{C}_t^0 = \{\emptyset\}$ and build candidate sequences by expanding and pruning at depths $n = 1, \ldots, H$.

(i) Probability-based tree expansion: Each retained prefix $\mathbf{u} = [u^1, \ldots, u^{n-1}]$ is extended with the top-K tokens from head $n-1$. The distribution $P_{t+n}^{n-1}(\cdot \mid \tilde{a}^\star_{1:t}, \mathbf{u}, m)$ conditions on the executed history, candidate prefix, and map. Suppressing the executed history and map, we write

$$\widetilde{\mathcal{C}}_t^n = \bigcup\nolimits_{\mathbf{u}\in\mathcal{C}_t^{n-1}} \left\{ [\mathbf{u}, v] \mid v \in \mathrm{TopK}\big(P_{t+n}^{n-1}(\cdot \mid \mathbf{u})\big) \right\}. \tag{3}$$

The backbone supplies the first token; MSP heads propose later tokens conditioned on each branch. These are approximate continuations rather than full autoregressive backbone rollouts. Top-K expansion restricts which tokens can enter the search, while the distance score determines which expanded sequences survive. A retained sequence need not maximize proposal probability.

(ii) Distance-oriented tree pruning: We decode each candidate recursively from $\tilde{s}_t$ using $\tilde{s}_{t+j} = f(\tilde{s}_{t+j-1}, u^j)$. Its score $d_{t,n}(\mathbf{u})$ aggregates distances to the expert at corresponding future token endpoints, using only valid reference positions within the recorded horizon. When the expert future is unavailable, we instead use negative cumulative proposal log probability, $d_{t,n}(\mathbf{u}) = -\sum_{j=1}^{n} \log P_{t+j}^{j-1}(u^j \mid u^{1:j-1})$. This score remains defined near scenario boundaries without accessing unrecorded expert states. The up to B lowest-scoring sequences form $\mathcal{C}_t^n$. After depth H, we select the best retained continuation and commit its first token:

$$\mathbf{u}^\star = \arg\min_{\mathbf{u}\in\mathcal{C}_t^H} d_{t,H}(\mathbf{u}), \qquad \tilde{a}^\star_{t+1} = u^{\star,1}. \tag{4}$$

Algorithm 1 LG-CLFT training

Input: pretrained policy π_θ, dataset $\mathcal{D}$, lookahead H, rollout horizon T

repeat
  Sample a scenario $(s_{0:T}, m) \sim \mathcal{D}$.
  Initialize $\tilde{a}^\star_{1:0} \leftarrow \emptyset$ and $\tilde{s}_0 \leftarrow s_0$.
  for t in $[0, \dots, T-1]$ **do**
    Initialize candidate set $\mathcal{C}^0_t \leftarrow \{\emptyset\}$.
    for n in $[1, \dots, H]$ **do**
      Expand candidates by Eq. (3).
      Keep up to B lowest-$d_{t,n}$ sequences.
    end for
    Select each agent's first token by Eq. (4).
    Execute tokens; update states and history.
  end for
  Update θ on $\tilde{a}^\star_{1:T}$ by minimizing $\mathcal{L}$.
until convergence

Algorithm 1: Training overview. At each token step, all agents share the executed joint history but search separately, without enumerating joint futures. The backbone proposes the first token; serial MSP heads propose later tokens conditioned on each candidate prefix. At every depth, DBS expands top-K proposals and retains up to B continuations with the lowest valid-aware score $d_{t,n}$. Only the first token of each selected sequence is executed. All agents' choices then update the joint state and history before search restarts. The completed pseudo rollout provides supervision for the backbone and MSP heads. Expert futures guide training labels only; the policy conditions on the generated history and the HD map when deployed.

All agents' selected tokens update the joint state and history before the backbone is evaluated again. Unexecuted continuations are discarded, and search restarts from the updated prefix. This rolling update lets the next search incorporate the motions just executed by other agents. The later tokens influence which first token is selected through the distance score, but they do not fix the actions executed in later steps. For $H = 1$, this reduces to one-step expert-distance matching within the retained top-K candidates, without evaluating alignment beyond the next token interval.

4 Experiments

4.1 Experimental Setup

Benchmark. We evaluate on the Waymo Open Sim Agents Challenge (WOSAC) benchmark. Following the official setup, the simulator takes 1 s of agent history together with the HD map context and generates R=32 possible multi-agent future trajectory rollouts. Each rollout spans an 8 s horizon at 10 Hz (80 steps), and we output the per-agent future motion states required by the challenge protocol (*e.g.*, position and heading) for up to 128 agents in each scenario.

Metrics. We adopt the official WOSAC distribution-matching metrics. The evaluator scores the expert trajectory by its likelihood under a history-conditioned density estimated from the simulated rollouts. We report metrics in three groups: **Kinematic** (speed and acceleration), **Interactive** (proximity, collision, and time-to-collision), and **Map-based**, which measures compliance with map constraints such as off-road and road-edge behavior. The overall **Realism Meta** score is a weighted combination of these groups and is the primary evaluation criterion. We also report **minADE** as an auxiliary measure of displacement error; it is not included in Realism Meta.

Implementation details. All in-house variants share the SMART-tiny backbone (7M parameters) and discrete tokenization scheme. Unless otherwise specified, we train N=2 auxiliary MSP heads and apply entropy regularization in CLFT with λ_{ent}=0.1. For pseudo-label construction, we use rolling matching with lookahead horizon H, top-K branching (K=32), and beam width B=32; the main results in Table 1 use N=2 and H=2. We use 8 NVIDIA A100 GPUs for 32-epoch pretraining and 4-epoch CLFT. As detailed in Appendix A, both stages use AdamW with a total batch size of 80 and cosine learning-rate decay, starting at 5×10^{-4} for pretraining and 5×10^{-5} for CLFT.

Inference and validation. For local WOSAC validation (Montali et al., 2023), we sample from the main backbone's next-token head using top-48 sampling at temperature 1.0. The R=32 rollouts are drawn sequentially from a seeded random-number stream. Generating the full validation split takes about 3 hours; local metric computation takes about 8 hours. The MSP heads and DBS construct Stage II training targets. For leaderboard and validation evaluation, we sample the main NTP head using the learned policy, without running the expert-guided lookahead search.

Table 1: WOSAC 2025 leaderboard test results. RMM denotes the official Realism Meta ranking score; the remaining columns report its three metric groups and auxiliary minADE.

Method	RMM ↑	Kinematic ↑	Interactive ↑	Map-based ↑	minADE ↓
SMART-tiny (Wu et al., 2024)	0.7814	0.4854	0.8089	0.9153	1.3931
SimFormer (Wang et al., 2025b)	0.7820	0.4920	0.8060	0.9167	1.3221
UniMM (Lin et al., 2026)	0.7829	0.4914	0.8089	0.9161	1.2949
comBOT (Rössert et al., 2025)	0.7837	0.4899	0.8102	0.9175	1.3687
CAT-K (Zhang et al., 2025a)	0.7846	0.4931	0.8106	0.9177	1.3065
RoaD (Garcia-Cobo et al., 2026)	0.7847	0.4932	0.8106	0.9178	1.3042
UniMotion (Song et al., 2025)	0.7851	**0.4943**	0.8105	0.9187	1.3036
TrajTok (Zhang et al., 2025b)	0.7852	0.4887	**0.8116**	**0.9207**	1.3179
LG-CLFT (ours)	**0.7854**	0.4920	0.8107	0.9205	**1.2865**

Table 2: Full WOSAC validation. LS/LA: linear speed and acceleration; AS/AA: angular speed and acceleration; OD: object distance; Col.: collision; TTC: time-to-collision; RE: road edge; OR: offroad. SMART w/ LFM denotes pretraining with auxiliary MSP heads.

Method	Kinematic				Interactive			Map-based		Overall
	LS ↑	LA ↑	AS ↑	AA ↑	OD ↑	Col. ↑	TTC ↑	RE ↑	OR ↑	RMM ↑
SMART	0.3781	0.4033	0.5141	0.6557	0.3900	0.9649	0.8366	0.6796	0.9493	0.7667
SMART w/ LFM	0.3890	0.4062	0.5162	0.6556	0.3916	0.9634	0.8370	0.6823	0.9520	0.7683
CAT-K	0.3876	0.4075	0.5163	**0.6578**	0.3923	**0.9695**	0.8375	0.6821	0.9511	0.7698
LG-CLFT	**0.3914**	**0.4087**	**0.5188**	0.6568	**0.3937**	0.9677	**0.8376**	**0.6846**	**0.9549**	**0.7710**

4.2 Quantitative Results

Leaderboard comparison. On the WOSAC 2025 test split (Table 1), LG-CLFT achieves the highest Realism Meta (0.7854) and lowest minADE (1.2865) among the listed entries. Compared with SMART-tiny, it improves all three metric groups while reducing minADE from 1.3931 to 1.2865. Relative to CAT-K, LG-CLFT improves Realism Meta from 0.7846 to 0.7854 and reduces minADE from 1.3065 to 1.2865, with higher Interactive and Map-based scores. These test results show gains in both realism and geometric alignment; the validation results below identify improvements from pretraining and closed-loop fine-tuning. Its Interactive and Map-based scores both rank second among the listed methods, while its Kinematic score remains below the best entry.

Full-validation breakdown. Table 2 compares SMART-based methods on the full WOSAC validation split. Realism Meta rises from 0.7667 for SMART to 0.7683 with the LFM and 0.7710 with LG-CLFT, showing gains at both training stages. The improvements span motion and map compliance: the linear-speed score rises from 0.3781 to 0.3914, and the offroad score from 0.9493 to 0.9549. LG-CLFT leads on seven of the nine listed sub-metrics; CAT-K retains the highest angular-acceleration and collision scores. LG-CLFT also improves every listed sub-metric over LFM pretraining alone, indicating further gains from fine-tuning on generated histories. These gains are obtained with standard next-token decoding of the learned policy; expert-guided search is used only to construct training targets. Against CAT-K, offroad rises from 0.9511 to 0.9549 and road-edge from 0.6821 to 0.6846; time-to-collision changes only slightly.

4.3 Ablation Study

Following CAT-K (Zhang et al., 2025a), we ablate LG-CLFT on a 2% validation subset to examine how pretraining and target construction affect closed-loop performance. Realism Meta is the primary selection criterion, with minADE as a complementary diagnostic. Tables 3–8 report the tested settings. We fix $K{=}32$ and retain the SMART/CAT-K architecture and optimization settings. Unless stated otherwise, each in-house configuration uses a single training run with global seed 817; deterministic GPU kernels are not enforced during any of these training runs.

Table 3: MSP pretraining depth.

Method	RMM ↑	minADE ↓
SMART (N=0)	0.7669	1.3746
MSP (N=1)	**0.7693**	**1.2943**
MSP (N=2)	0.7690	1.3065
MSP (N=3)	0.7689	1.2987

Table 4: Pseudo-trajectory quality.

Setting	RMM ↑	ADE ↓
H=1	0.7923	0.1755
H=2	0.7952	0.1608
H=3	**0.7953**	**0.1597**
H=4	0.7943	0.1714

Table 5: Lookahead horizon H in CLFT.

CLFT setting	RMM ↑	minADE ↓
CLFT w/ CAT-K	0.7712	1.2975
CLFT w/ H=1	0.7714	**1.2929**
CLFT w/ H=2	**0.7718**	1.2948
CLFT w/ H=3	0.7717	1.3150

Table 6: Beam width B in DBS.

Beam width B	RMM ↑	minADE ↓
8	0.7715	1.3168
16	0.7714	**1.2909**
32	**0.7718**	1.2948
64	0.7712	1.3008

Table 7: Distance-oriented beam search.

Selection rule	RMM ↑	minADE ↓
Greedy search (prob.)	0.7675	1.3335
Beam search (prob.)	0.7713	1.2968
Beam search (distance)	**0.7718**	**1.2948**

Table 8: Distance metric.

Distance metric	RMM ↑	minADE ↓
All trajectory points	0.7672	1.3221
Last token endpoint	0.7650	1.3526
Token endpoints	**0.7718**	**1.2948**

Effect of MSP pretraining. Table 3 shows that serial MSP heads consistently improve both Realism Meta and minADE over SMART, even when only the next-token head is used at inference. The auxiliary heads share backbone features with NTP, so their future-token losses train the main predictor's representation while inference retains next-token decoding. Among the tested pretraining depths, N=1 gives the best result on both metrics; adding more auxiliary heads does not further improve next-token rollout quality for the shared SMART-tiny backbone.

Effect of lookahead horizon for pseudo labels. Here, ADE measures the displacement error between the selected pseudo trajectory and the expert trajectory. Table 4 has the lowest matching ADE at H=3. Because N=$H-1$, these rows vary both MSP-head depth and the matching horizon. Matching ADE evaluates the selected pseudo trajectory before fine-tuning and is distinct from the final rollout minADE measured after closed-loop fine-tuning and rollout generation.

Effect of lookahead horizon in CLFT. Table 5 varies the CLFT lookahead horizon for pseudo-label construction, where H=1 corresponds to myopic matching. From H=1 to H=2, Realism Meta rises from 0.7714 to 0.7718, while minADE rises from 1.2929 to 1.2948; thus the two metrics favor different horizons in this ablation. The H=2 setting has the highest Realism Meta among the tested horizons, but lookahead does not monotonically reduce displacement error.

Effect of beam width. Among the evaluated settings in Table 6, B=32 achieves the highest Realism Meta (0.7718), while B=16 yields the lowest minADE (1.2909). We choose the default beam width by the primary Realism Meta criterion, selecting B=32 for subsequent experiments.

Effect of distance-oriented beam search. Table 7 tests whether expert-distance ranking improves pseudo-label construction when the beam-search budget is held fixed. Probability-based beam search improves both metrics over probability-based greedy search. At the same beam-search budget, distance ranking further improves both metrics over probability ranking. The two beam-search variants differ only in their ranking criterion, separating the contribution of expert-distance scoring from that of retaining multiple candidate sequences.

Effect of distance metric. Table 8 shows that token-endpoint distance yields the best Realism Meta and minADE among the three tested metrics on this subset. We use it as the default DBS metric: it aggregates decoded endpoint distances at corresponding future steps, including intermediate token boundaries rather than only the final position of the candidate continuation.

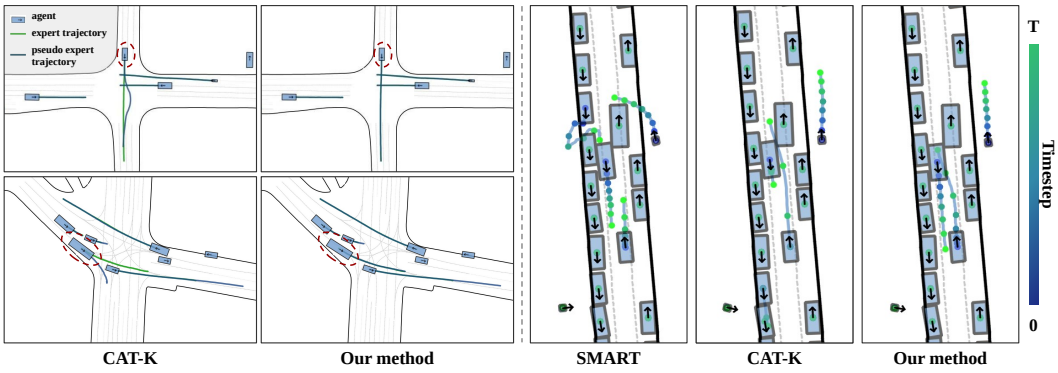

Figure 3: **Qualitative comparisons.** *Left:* pseudo expert trajectories under greedy and lookahead matching. *Right:* multi-agent closed-loop behavior in a narrow-lane passing scenario. Red dashed circles mark the compared agents on the left; the color bar indicates rollout time on the right.

4.4 Qualitative Results

Pseudo expert trajectory quality. The left panel of Figure 3 compares pseudo expert targets for agents marked by red dashed circles. In the top example, CAT-K's greedy matching makes a locally plausible next-token choice, but the resulting prefix leads to drift and abrupt corrections. LG-CLFT compares multi-step candidates over a future window and follows the expert trajectory more closely. The bottom example shows the same pattern: greedy errors accumulate, whereas LG-CLFT remains aligned. Both examples show how an initially close token can lead to a poor continuation.

Closed-loop interaction behaviors. The right panel of Figure 3 shows a challenging narrow-lane passing scenario with a blocking vehicle. SMART fails to coordinate under closed-loop rollout, while CAT-K improves stability but still attempts an aggressive pass and collides. LG-CLFT waits for the other vehicle to clear the narrow section before passing in this example. The trained policy produces this waiting-and-passing behavior without expert-guided search at inference.

4.5 Speculative Rollout Efficiency

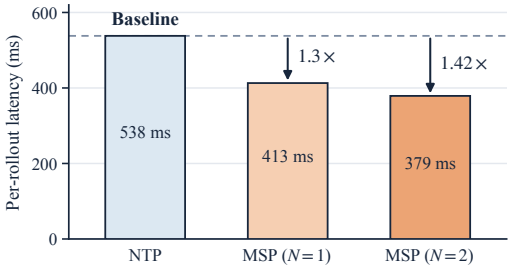

Figure 4: **Speculative rollout latency.** Deterministic ego-only timing on one V100: 538 ms for NTP, 413 ms with one draft head (1.30×), and 379 ms with two (1.42×).

The serial MSP heads draft a short sequence of motion tokens for ego-only speculative inference (Leviathan et al., 2023; Chen et al., 2023). The main decoder verifies each draft block in one forward pass against its top-1 choice. Accepted drafts advance the rollout without a separate backbone call for each token.

With two heads, target top-1 agreement is 81.73%/78.62%, with a mean of 1.30 accepted draft tokens per verification round. This preserves deterministic ego-only top-1 rollouts; WOSAC uses 32 stochastic multi-agent rollouts. Appendix D.2 reports training overhead.

5 Conclusion

Single-step matching in token-level closed-loop fine-tuning can select a token that matches the expert well over the immediate interval but leads to a poorly aligned continuation. We proposed Lookahead-Guided Closed-Loop Fine-Tuning (LG-CLFT), which combines serial MSP heads with distance-oriented beam search to select pseudo expert targets by finite-window alignment. At each step, the method commits only the first token of the selected continuation, then searches again from the updated generated history. WOSAC evaluation samples the main NTP head. On the WOSAC 2025 leaderboard test split, LG-CLFT achieves the highest Realism Meta and the lowest minADE among the listed methods, improving both rollout realism and geometric alignment.

AI USE STATEMENT

We used AI tools to assist with polishing the manuscript's language and presentation. The authors take responsibility for the accuracy, originality, and integrity of the final manuscript.

ETHICS STATEMENT

This work studies multi-agent traffic simulation using the existing Waymo Open Motion Dataset. More realistic simulation can support the offline evaluation of autonomous driving systems, but benchmark performance does not establish safety in real-world driving. The learned simulator may inherit biases and coverage limitations from the recorded traffic data, particularly for rare events and underrepresented road users or environments. Any use in safety-critical applications therefore requires validation beyond the reported benchmark.

REPRODUCIBILITY STATEMENT

The supplementary material includes anonymized code for the core two-stage training procedure, configuration files, a training launcher, and a qualitative comparison video. Section 3 describes the method, and Section 4.1 specifies the evaluation protocol. Appendix A provides the experimental environment and training details; Table 9 lists the optimizer, learning rates and schedule, batch size, training epochs, numerical precision, gradient clipping, and search settings for both stages. Appendix B describes the backbone architecture.

REFERENCES

Kwangjun Ahn, Alex Lamb, and John Langford. Efficient joint prediction of multiple future tokens. *arXiv preprint arXiv:2503.21801*, 2025.

Ansar Aynetdinov and Alan Akbik. Pre-training curriculum for multi-token prediction in language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

Keyu Chen, Wenchao Sun, Hao Cheng, Zheng Fu, and Sifa Zheng. ForSim: Stepwise forward simulation for traffic policy fine-tuning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

DeepSeek-AI. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning (CoRL)*, 2017.

Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R. Qi, Yin Zhou, Zoey Yang, Aurélien Chouard, Pei Sun, Jiquan Ngiam, Vijay Vasudevan, Alexander McCauley, Jonathon Shlens, and Dragomir Anguelov. Large scale interactive motion forecasting for autonomous driving: The Waymo Open Motion Dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Guillermo Garcia-Cobo, Maximilian Igl, Peter Karkus, Zhejun Zhang, Michael Watson, Yuxiao Chen, Boris Ivanovic, and Marco Pavone. RoaD: Rollouts as demonstrations for closed-loop supervised fine-tuning of autonomous driving policies. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Findings (CVPR Findings)*, 2026.

Anastasios Gerontopoulos, Spyridon Gidaris, and Nikos Komodakis. Multi-token prediction needs registers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Rozière, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

Haicheng Liao, Zhenning Li, Chengyue Wang, Huanming Shen, Dongping Liao, Bonan Wang, Guofa Li, and Chengzhong Xu. MFTraj: Map-free, behavior-driven trajectory prediction for autonomous driving. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, 2024a.

Haicheng Liao, Zhenning Li, Chengyue Wang, Bonan Wang, Hanlin Kong, Yanchen Guan, Guofa Li, and Zhiyong Cui. A cognitive-driven trajectory prediction model for autonomous driving in mixed autonomy environments. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, 2024b.

Longzhong Lin, Xuewu Lin, Kechun Xu, Haojian Lu, Lichao Huang, Rong Xiong, and Yue Wang. UniMM: A unified mixture model framework for multi-agent simulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

LLM-Core Xiaomi. MiMo: Unlocking the reasoning potential of language model – from pretraining to posttraining. *arXiv preprint arXiv:2505.07608*, 2025.

Divyat Mahajan, Sachin Goyal, Badr Youbi Idrissi, Mohammad Pezeshki, Ioannis Mitliagkas, David Lopez-Paz, and Kartik Ahuja. Beyond multi-token prediction: Pretraining LLMs with future summaries. In *Proceedings of the Fourteenth International Conference on Learning Representations (ICLR)*, 2026.

Nico Montali, John Lambert, Paul Mougin, Alex Kuefler, Nicholas Rhinehart, Michelle Li, Cole Gulino, Tristan Emrich, Zoey Yang, Shimon Whiteson, Brandyn White, and Dragomir Anguelov. The Waymo Open Sim Agents Challenge. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Zhenghao Peng, Yuxin Liu, and Bolei Zhou. SceneStreamer: Continuous scenario generation as next token group prediction. In *Proceedings of the Fourteenth International Conference on Learning Representations (ICLR)*, 2026.

Jonah Philion, Xue Bin Peng, and Sanja Fidler. Trajeglish: Traffic modeling as next-token prediction. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence (eds.). *Dataset Shift in Machine Learning*. The MIT Press, 2009.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

Christian Rössert, Johannes Drever, and Lukas Brostek. comBOT: An ensemble combination model combining results from SMART-tiny-CLSFT with a cognitive behavior model. Technical report, cogniBIT GmbH, 2025.

Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S. Refaat, Rami Al-Rfou, and Benjamin Sapp. MotionLM: Multi-agent motion forecasting as language modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 2000.

Nan Song, Junzhe Jiang, Jingyu Li, Xiatian Zhu, and Li Zhang. UniMotion: A unified motion framework for simulation, prediction and planning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 2007.

Abitha Thankaraj, Yiding Jiang, J. Zico Kolter, and Yonatan Bisk. Looking beyond the next token. *arXiv preprint arXiv:2504.11336*, 2025.

Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 2000.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Arun Venkatraman, Martial Hebert, and J. Andrew Bagnell. Improving multi-step prediction of learned time series models. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

Bonan Wang, Haicheng Liao, Chengyue Wang, Bin Rao, Yanchen Guan, Guyang Yu, Jiaxun Zhang, Songning Lai, Chengzhong Xu, and Zhenning Li. Beyond patterns: Harnessing causal logic for autonomous driving trajectory prediction. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, 2025a.

Sen Wang, Jianrong Xu, Xiaoyong Zhang, Fangqiao Hu, Zhijun Huang, JiaChen Luo, Kechen Zhu, Jiaxiang Zhu, Yong Zhou, and Zhenwu Chen. SimFormer: 1st place in the Waymo Open Scenario Generation Challenge 2025. Technical report, Shenzhen Urban Transport Planning Center Co., Ltd., 2025b.

Wei Wu, Xiaoxin Feng, Ziyan Gao, and Yuheng Kan. SMART: Scalable multi-agent real-time motion generation via next-token prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Yueh-Hua Wu, Nontawat Charoenphakdee, Han Bao, Voot Tangkaratt, and Masashi Sugiyama. Imitation learning from imperfect demonstration. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

Lingyu Xiao, Zexin Feng, and Xintao Yan. Long-term traffic simulation via structured autoregressive modeling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026.

Scott Xu, Dian Chen, Kelvin Wong, Chris Zhang, Kion Fallah, and Raquel Urtasun. Efficient equivariant transformer for self-driving agent modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Zhejun Zhang, Peter Karkus, Maximilian Igl, Wenhao Ding, Yuxiao Chen, Boris Ivanovic, and Marco Pavone. Closed-loop supervised fine-tuning of tokenized traffic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025a.

Zhiyuan Zhang, Xiaosong Jia, Guanyu Chen, Qifeng Li, and Junchi Yan. TrajTok: Technical report for 2025 Waymo Open Sim Agents Challenge. *arXiv preprint arXiv:2506.21618*, 2025b.

Guanjie Zheng, Hanyang Liu, Kai Xu, and Zhenhui Li. Objective-aware traffic simulation via inverse reinforcement learning. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

Zikang Zhou, Haibo Hu, Xinhong Chen, Jianping Wang, Nan Guan, Kui Wu, Yung-Hui Li, Yu-Kai Huang, and Chun Jason Xue. BehaviorGPT: Smart agent simulation for autonomous driving with next-patch prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Zayd Muhammad Kawakibi Zuhri, Erland Hilman Fuadi, and Alham Fikri Aji. Predicting the order of upcoming tokens improves language modeling. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

A Additional Implementation Details

Experimental environment. The experimental environment is based on Ubuntu 22.04 and CUDA 12.4.1, with Python 3.11.9, PyTorch 2.4.1, Torchvision 0.19.1, Lightning 2.4.0, TorchMetrics 1.4.1, TensorFlow 2.12.0, and Waymo Open Dataset 1.6.4. Training and GPU validation use 8 NVIDIA A100 GPUs, each with 80 GB of device memory.

Model and training. We retain the SMART-tiny architecture and the training settings used by CAT-K unless otherwise specified. In Stage I (lookahead-aware open-loop pretraining), we jointly train the backbone, the NTP head, and two auxiliary MSP heads for 32 epochs (about 64 hours) with a cosine learning-rate schedule and an initial learning rate of 5×10^{-4}. We use the SMART-tiny backbone with hidden_dim 128, 8 attention heads (head_dim 16), 3 map layers, and 6 agent layers. Neighborhood radii are set to 10 (polyline-to-polyline), 30 (polyline-to-agent), and 60 (agent-to-agent). In Stage II (lookahead-guided closed-loop fine-tuning), we initialize from the Stage I checkpoint and fine-tune for 4 epochs (about 12 hours) with a cosine schedule and an initial learning rate of 5×10^{-5}. During Stage II, we freeze the map-related modules (map encoder and map layers) and update the agent-motion backbone together with our added components. During Stage II, we construct pseudo labels online using $N{=}2$ auxiliary MSP heads. At each time step, we expand candidates with top-K branching (default $K{=}32$) and maintain a beam of width B (default $B{=}32$) over a lookahead horizon of $H{=}2$ tokens. We score trajectories by aggregating distances at the decoded token endpoints over the lookahead window, using corresponding future time indices. The first token of the best-scoring sequence becomes the pseudo label.

Table 9: Final optimization and search settings. Settings inherited from CAT-K are retained unless explicitly overridden for LG-CLFT. AdamW uses its default weight decay of 0.01; the learning rate follows an epoch-wise cosine schedule without warm-up.

Setting	**Stage I**	**Stage II**
Optimizer	AdamW	AdamW
Epochs	32	4
Total batch size	80	80
Initial learning rate	$5{\times}10^{-4}$	$5{\times}10^{-5}$
Minimum LR ratio	0.01	0.01
Warm-up epochs	0	0
Numerical precision	FP32	FP32
Gradient clipping	0.5	0.5
MSP heads N	2	2
Entropy weight	0	0.1
DBS (H, K, B)	–	$(2, 32, 32)$

B Model Backbone

Inputs. We represent each driving scenario using (i) discrete motion tokens for all agents and (ii) road-vector map tokens extracted from the scene context. For agent i at time step t, the discretized motion token is $a_t^i \in \mathcal{V}$, and $e_t^i \in \mathbb{R}^d$ denotes its input feature together with geometric and agent context. HD map polylines are partitioned into road vectors and embedded as $m_j = \mathrm{Emb}(r_j) \in \mathbb{R}^d$.

Relative positional encoding (RPE). Relative positional encoding represents temporal lags and agent-centric geometric relations in attention. Temporal RPE $\mathrm{RPE}_{t,t'}$ encodes relative distance, bearing, relative heading, and the signed time offset $(t - t')$. Spatial RPE $\mathrm{RPE}_{i,j}$ encodes the relative distance, bearing, and orientation of source entity j with respect to the receiving entity i in the receiver-centric coordinate frame. A Fourier embedding maps each descriptor to an edge feature, whose learned projections are added to the attention keys and values.

Map encoding. We first contextualize map tokens via multi-head self-attention (MHSA):

$$\hat{m}_j = \mathrm{MHSA}\Big(q(m_j), \{k(m_{j'}, \mathrm{RPE}_{j,j'}),\ v(m_{j'}, \mathrm{RPE}_{j,j'})\}_{j' \in \mathcal{R}(j)}\Big), \tag{5}$$

where $\mathcal{R}(j)$ indexes the road tokens in the local neighborhood of map token m_j.

Motion backbone (factorized attention). Each agent layer sequentially applies temporal self-attention, agent–map cross-attention, and agent–agent self-attention.

(a) Temporal self-attention. At time t, each agent attends to its own historical motion tokens in a window of size $\tau > 0$, with past time indices restricted to $t - \tau \le t' < t$:

$$\tilde{e}_t^i = \mathrm{MHSA}\Big(q(e_t^i), \{k(e_{t'}^i, \mathrm{RPE}_{t,t'}),\ v(e_{t'}^i, \mathrm{RPE}_{t,t'})\}_{t-\tau \le t' < t}\Big). \tag{6}$$

(b) Agent–map cross-attention. We fuse road context from local map tokens in $\mathcal{M}(i)$:

$$\hat{e}_t^i = \mathrm{MHCA}\Big(q(\tilde{e}_t^i), \{k(\hat{m}_j, \mathrm{RPE}_{i,j}),\ v(\hat{m}_j, \mathrm{RPE}_{i,j})\}_{j \in \mathcal{M}(i)}\Big). \tag{7}$$

(c) Agent–agent self-attention. At time t, we attend to neighboring agents in $\mathcal{N}(i)$:

$$h_t^i = \mathrm{MHSA}\Big(q(\hat{e}_t^i), \{k(\hat{e}_t^j, \mathrm{RPE}_{i,j}),\ v(\hat{e}_t^j, \mathrm{RPE}_{i,j})\}_{j \in \mathcal{N}(i)}\Big). \tag{8}$$

Here, MHSA and MHCA denote multi-head self- and cross-attention. The neighborhoods $\mathcal{M}(i)$ and $\mathcal{N}(i)$ contain local map tokens and agents, respectively (e.g., within a spatial radius). Stacking L=6 layers of these three attention operations yields the backbone features $\{h_t^i\}$.

Backbone output. We collect the final-layer hidden features of all agents at each time step as $h_t = \{h_t^i\}_{i=1}^M$. Stacking them over time yields $\mathbf{F}_h^0 = [h_0, \ldots, h_T]$, the backbone output supplied to the lookahead forecasting module (LFM) for multi-step motion-token prediction.

C Entropy Regularization Ablation

Table 10: Entropy-weight ablation.

λ_{ent}	**RMM** ↑	**minADE** ↓
0 (w/o entropy)	0.7718	1.3156
0.05	0.7721	1.3131
0.1	**0.7725**	**1.3098**
0.2	0.7674	1.3417

Effect of entropy regularization. We use entropy regularization to discourage overly concentrated token distributions and retain candidate diversity during matching. Table 10 uses a separate experimental setting from the main ablations in Tables 5–8. Within this sweep, both metrics are best at λ_{ent}=0.1. Increasing the weight to 0.2 lowers Realism Meta and raises minADE relative to training without entropy regularization.

D Robustness and Cost Analyses

D.1 Robustness to Noisy Expert Logs

Imperfect or noisy demonstrations are a long-standing challenge in imitation learning (Wu et al., 2019). LG-CLFT remains dependent on expert-data quality: Stage I trains directly on expert trajectories, and Stage II uses the expert future as the matching reference.

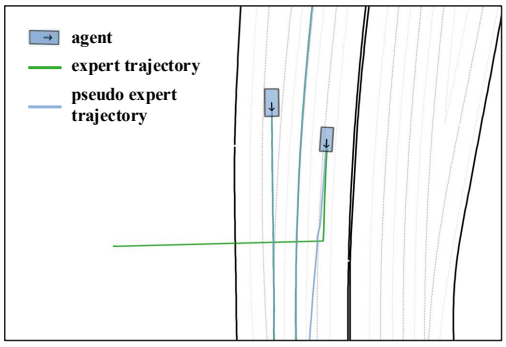

Figure 5: **Illustrative noisy-label case.** *Green:* recorded expert trajectory. *Blue:* selected pseudo expert trajectory.

At each Stage II step, the target is selected from high-probability, policy-reachable candidates by aligning their trajectories with the expert future, rather than copying the recorded trajectory. This may avoid a local artifact when a plausible nearby candidate exists, but does not establish robustness to noisy logs. Figure 5 shows one illustrative case: the recorded expert trajectory contains an abrupt off-lane segment, whereas the selected pseudo expert remains smooth and lane-aligned. Search remains limited to motions in the candidate set. The recorded expert future also provides the distance reference, so annotation errors can still influence the selected motion at each step of training.

D.2 Training Overhead and Scalability

LG-CLFT's added cost is concentrated in Stage II online pseudo-label construction; Stage I also trains the lightweight MSP heads. Table 11 compares per-step strategies under a controlled budget of 8 candidate trajectories, separate from default DBS: greedy matching, CAT-K matching (Zhang et al., 2025a), and MSP-based lookahead cost 24.69, 27.65, and 29.26 ms, respectively. Each MSP head adds one attention module, whereas the six-layer agent decoder has three factorized attention modules per layer (18 total). Rerunning the full backbone for the same candidates costs 197.52 ms, about 6.7× the MSP cost. These timings cover Stage II label construction; WOSAC inference remains NTP-only. Pretraining requires 32 epochs on 8 A100 GPUs (about 64 hours), and fine-tuning reaches the reported performance in 4 epochs (about 12 hours).

Table 11: Per-step cost of online label construction on a single A100 GPU under a matched budget of 8 candidate trajectories for all lookahead methods. This controlled timing configuration is distinct from the default DBS setting used in the main experiments.

Label-construction strategy	**Time / step**
Greedy matching	24.69 ms
CAT-K matching	27.65 ms
MSP-based lookahead (ours)	29.26 ms
Full-backbone lookahead	197.52 ms

D.3 Speculative Rollout Efficiency

We evaluate an optional deterministic ego-only rollout setting separately from the stochastic multi-agent WOSAC protocol. Following speculative decoding (Leviathan et al., 2023; Chen et al., 2023), the serial MSP heads draft future ego tokens from the current prefix, and the main backbone verifies the block in one forward pass, reducing serial target-backbone calls. Table 12 reports speedup from synchronized timing on one NVIDIA V100 GPU and target top-1 agreement at each draft depth (Draft-1/Draft-2).

Table 12: Synchronized ego-only speculative-rollout efficiency on one NVIDIA V100 GPU. "Draft-1/Draft-2" gives target top-1 agreement at the two draft depths; "Avg. accepted" is the average number of draft tokens committed per sequential verification round.

Method	**Speedup**	**Draft-1/Draft-2**	**Avg. accepted**
NTP top-1	1.000×	–	–
MSP (N=1)	1.301×	82.81% / –	0.83
MSP (N=2)	1.420×	81.73% / 78.62%	1.30

For $N{=}2$, Draft-1 and Draft-2 agree with the target top-1 prediction in 81.73% and 78.62% of the corresponding measurements, respectively. These token-level argmax agreement rates do not establish that the full predictive distributions or unverified ego trajectories match those of the target model. An average of 1.30 draft tokens is committed per round, and latency decreases from 537.74 ms to 378.69 ms ($1.42\times$). The difference between the two agreement rates is descriptive and does not establish a general depth-dependent trend. This optional acceleration applies to deterministic ego-only rollout; WOSAC uses stochastic NTP decoding.

E Additional Qualitative Results

Figure 6 shows additional closed-loop rollouts. Each row corresponds to one scenario, and each column shows one stochastic closed-loop rollout from the same initial condition. The samples differ in maneuver timing, speed profiles, and path selection while remaining visually consistent with the local map geometry. These examples show sample-level variation but do not measure behavior frequencies, multimodality, or interaction quality systematically.

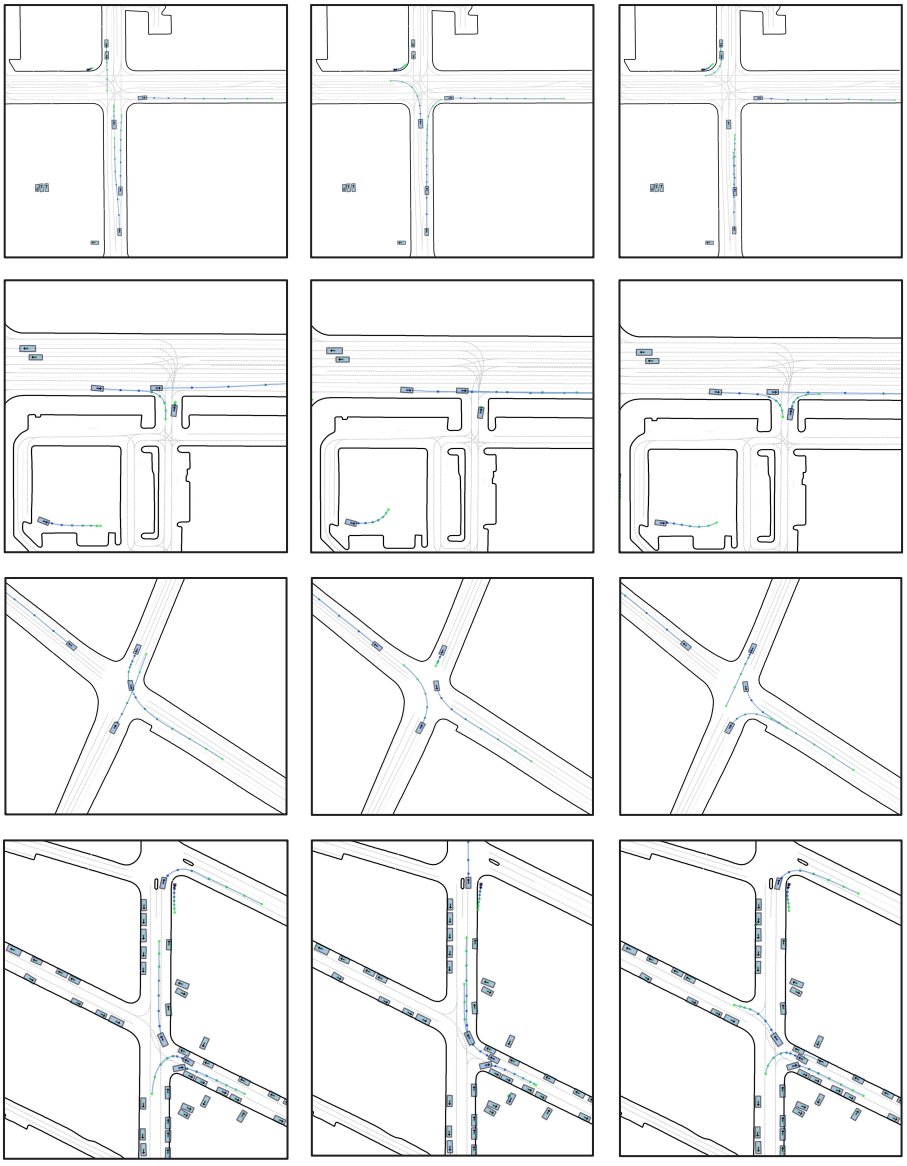

Figure 6: **Additional qualitative examples from closed-loop rollouts.** Rows show different scenarios; columns show stochastic rollouts sampled from the same initial condition.

F LIMITATIONS AND FUTURE WORK

Limitations. LG-CLFT depends on both the policy support established during pretraining and the quality and coverage of the expert demonstrations used for supervision.

First, pseudo expert targets are typically constructed within the policy-reachable, high-probability support to keep supervision distribution-consistent and training stable. Consequently, closed-loop refinement depends on the interaction modes covered during pretraining, which can limit recovery of rare or strongly coupled behaviors outside the policy's high-probability support.

Second, rollout quality depends on the reliability and representativeness of expert data. Noisy logs, tracking/annotation errors, inconsistent driving styles, or limited coverage of rare interactions can bias Stage I learning and make Stage II matching less reliable, potentially reducing robustness and realism.

Future work. Future work includes *metric-aware* pseudo-label construction and *joint* multi-agent target selection. Evaluation-aligned criteria such as map compliance, safety, and interaction comfort could complement geometric distance when ranking candidate trajectories. Joint multi-agent pseudo labels could account for coupled decisions across agents and improve coordination in dense traffic.