SEQUENTIAL GROUP-RELATIVE TEST-TIME ADAPTATION FOR CLOSED-LOOP LLM PLANNERS ON A SINGLE DRIVING TIMELINE

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM-based motion planners still lose substantial closed-loop performance in unseen driving domains, motivating online test-time adaptation from reward feedback. Two structural obstacles stand in the way: per-step gradient updates on a multi-billion-parameter policy are expensive, and critic-free group-relative methods such as GRPO normalize each rollout against other rollouts from the same state, which a deployed planner never observes on its single, irreversible driving timeline. We propose **TeTO** (Test-time Trajectory-level Optimization). Its core, **Sequential Grouping**, reformulates group-relative optimization along time, normalizing each rollout against the planner's immediately preceding rollouts in short, non-overlapping windows. The deviation of this baseline from the unobservable same-state baseline is bounded by the local reward drift, which is small under local smoothness, plus a sampling term; empirically, a three-step window matches the reward variance of same-observation groups, and its advantages agree in sign with a same-state reference in 92.3% of logged windows without abrupt time-to-collision changes. **Reward Truncation** further restricts updates to windows whose weighted per-component minimum reward falls below a threshold, and only LoRA adapters are updated, with the frozen backbone serving as the KL reference. On nuPlan, TeTO improves the supervised planner by up to 16.7% in an out-of-domain city while updating on as few as 7.8% of steps, with consistent gains across two LLM backbones, three unseen cities, and a planner trained on mixed nuPlan–Waymo data. To our knowledge, TeTO is the first critic-free group-relative policy optimization method for a single executed driving timeline whose baseline has a bounded deviation from the same-state baseline.

1 INTRODUCTION

Critic-free alignment methods for large language models, such as Group Relative Policy Optimization (GRPO) (Shao et al., 2024b), estimate advantages by comparing several responses sampled for the same input, without learning a value function. This is natural in language tasks, where many responses to a fixed prompt can be sampled and scored independently. It breaks down in closed-loop embodied control, where each executed action changes the world state and only one action can be executed per state; autonomous driving is a safety-critical instance of this mismatch.

Reasoning-capable LLMs are increasingly used as driving policies for their world knowledge and their ability to reason over long-tail traffic scenarios (Mao et al., 2023; Chen et al., 2024b; Zhang

Table 1: **The ID–OOD gap on nuPlan Test14-Hard.** ID: Las Vegas; OOD: Singapore. NR/R denote nonreactive/reactive traffic.

		ID		OOD	
Planner	Type	NR	R	NR	R
Diffusion Planner (Zheng et al., 2025b)	Non-LLM	80.74	78.98	46.10	51.70
InstructDriver-style SFT (Zhang et al., 2024)	LLM	78.72	80.87	62.09	72.33

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.

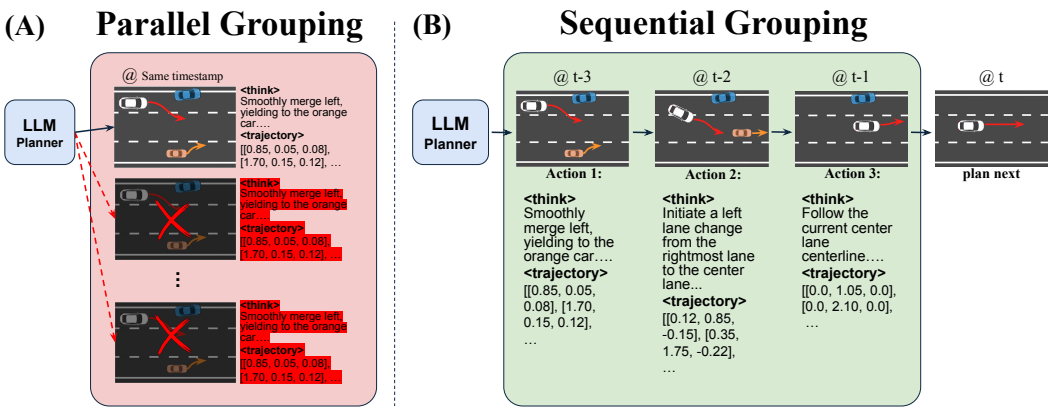

Figure 1: **Parallel and Sequential Grouping.** (A) Parallel grouping (GRPO) compares alternative rollouts from the same state. On a single driving timeline only one of them can be executed; evaluating the others (red, crossed out) requires restoring the joint traffic state. (B) Sequential Grouping uses the plans executed at $t-3$, $t-2$, and $t-1$ as the reference group. After each completed window, the planner receives at most one critic-free update before it plans at t.

et al., 2024; Sima et al., 2025b; Shao et al., 2024a; NVIDIA et al., 2025). One might expect their strong generalization in language to transfer to planning, but Table 1 shows that this only partially holds: the LLM planner degrades less than a diffusion-based planner when moved from the training city (Las Vegas) to an unseen one (Singapore), yet it still loses 16.6 NR points. A linguistic interface alone does not guarantee policy-level robustness; after supervised fine-tuning (SFT) on a limited corpus, the planner remains tied to the state–action distribution of its training domain (Cheng et al., 2024b; Li et al., 2024; Karnchanachari et al., 2024).

Cross-domain deployment is a recognized challenge for learned driving policies (Yasarla et al., 2026), and broader offline data narrows this gap without closing it (Appendix E.3). A more direct remedy is to adapt the planner at test time. Such methods for driving adapt trajectory predictors, world models, or end-to-end planners (Park et al., 2024; Wang et al., 2025; Sima et al., 2025a), or revise plans without updating the policy (Li et al., 2024); to our knowledge, reward-driven test-time adaptation of an LLM planner's parameters remains unexplored and raises two challenges. First, gradient updates on a multi-billion-parameter policy at every step are expensive and can destabilize it. Second, GRPO-style estimators need several rollouts from the same state, but a deployed planner executes one plan per state: each plan changes the ego state and the surrounding traffic, so the other members of a same-state group are never observed without restoring the world state (Figure 1).

We propose **TeTO** (Test-time Trajectory-level Optimization), a test-time adaptation framework for closed-loop LLM planners. The key observation is that group-relative optimization needs a low-variance, locally calibrated baseline, which the agent's own recent rollouts can provide. **Sequential Grouping** (SG) therefore builds the group along the *temporal* axis of the executed timeline, normalizing each rollout against the mean and spread of the preceding rollouts in a short, non-overlapping window collected under a fixed policy. The deviation of this baseline from the unobservable same-state baseline is bounded by the local reward drift, which local smoothness keeps small, plus a sampling term (Equation (5)). A three-step window matches the reward variance of same-observation parallel groups, and its advantages agree in sign with a same-state reference in 92.3% of logged windows without abrupt changes in time-to-collision (TTC). The result is a critic-free, GRPO-style update that needs neither state restoration nor extra plan sampling.

Updating on every window is often unnecessary: because advantages are standardized within each window, a window of uniformly high scores still yields unit-scale advantages driven by small, noise-dominated reward differences. **Reward Truncation** (RT) therefore summarizes each window by R_{rep}, a weighted aggregate of per-component minima, and issues an update only when R_{rep} falls below a threshold. As a bottleneck statistic, R_{rep} selects windows with degraded behavior without altering the gradient of an issued update. For efficiency, TeTO updates only LoRA adapters (Hu et al., 2022) and obtains the KL reference by temporarily disabling them (Santacroce et al., 2023).

On nuPlan (Caesar et al., 2021; Karnchanachari et al., 2024), TeTO improves closed-loop planning in and out of domain, by up to 16.7% in Singapore while updating on as few as 7.8% of steps, and the gains carry over to Qwen2.5-7B, Boston, Pittsburgh, and a planner trained on mixed nuPlan–Waymo data. At a matched objective and update count, the local baseline outperforms a running mean, isolating the contribution of Sequential Grouping. Our contributions are:

- We identify a structural mismatch between critic-free group-relative policy optimization and closed-loop test-time adaptation: same-state groups cannot be formed on the single executed timeline of a deployed planner.
- We propose **Sequential Grouping**, a temporal reformulation of group-relative optimization whose baseline approximates the unobservable same-state baseline up to local reward drift and sampling error, and we validate it against a same-state reference.
- We instantiate it in **TeTO** with Reward Truncation and adapter-only LoRA updates, improving out-of-domain closed-loop planning on nuPlan across two backbones, three unseen cities, and mixed-data training with sparse updates.

2 Related work

LLM-based motion planners. Driving policies have progressed from modular pipelines (Yurtsever et al., 2020) and end-to-end imitation (Pomerleau, 1988; Bojarski et al., 2016; Codevilla et al., 2019; Chen et al., 2024a) to language-guided reasoning and planning (Yang et al., 2023). GPT-Driver and DriveGPT4 formulate motion generation through language models (Mao et al., 2023; Xu et al., 2024); LanguageMPC, LLM-Assist, and PlanAgent use language reasoning to guide decisions and closed-loop planning (Sha et al., 2023; Sharan et al., 2023; Zheng et al., 2026); and object-level representations, graph question answering, vision–language alignment, instruction tuning, and sensor–language training connect driving context to these models (Chen et al., 2024b; Sima et al., 2025b; Pan et al., 2024; Zhang et al., 2024; Shao et al., 2024a), drawing on broader vision–language–action policies (Brohan et al., 2023; Reed et al., 2022). Recent large-model planners add BEV-informed intent reasoning (Wang et al., 2026b), 2D/3D vision–language alignment (Zhou et al., 2026), reasoning chains with self-reflection or unified perception–prediction–planning trained by reinforcement learning (Yuan et al., 2026; Ye et al., 2026), adaptive or explicit reasoning (Zhou et al., 2025; Luo et al., 2026; NVIDIA et al., 2025), and shared language–action representations (Wang et al., 2026a). These methods improve the planner offline, where several trajectories can be sampled and scored for the same logged scene; TeTO addresses deployment, where the planner meets each state once and adapts from its own closed-loop feedback.

Critic-free policy optimization. Reward-based fine-tuning of language models (Christiano et al., 2017; Ouyang et al., 2022) typically relies on PPO and its learned critic (Schulman et al., 2017). Critic-free alternatives estimate advantages from reward baselines, as in REINFORCE, reward-weighted regression, and REINFORCE++ (Williams, 1992; Peters & Schaal, 2007; Štrupl et al., 2022; Hu et al., 2025), or learn from preference pairs, as in DPO (Rafailov et al., 2023). GRPO and GSPO normalize rewards within groups of responses to a common prompt, and RLOO uses a leave-one-out baseline (Shao et al., 2024b; Zheng et al., 2025a; Ahmadian et al., 2024); Plan-R1 and Drive-R1 bring such objectives to driving (Tang et al., 2026; Li et al., 2026), and MindDrive applies online reinforcement learning to an LLM driving policy during training in simulation (Fu et al., 2025). S-GRPO forms serial groups at early-exit positions within a single reasoning generation, so all members share one input and environment state (Dai et al., 2025); TeTO groups executed plans across state transitions, where cross-state drift is the central difficulty. Reward baselines (Williams, 1992) and global advantage normalization (Hu et al., 2025) also pool samples from different states. Sequential Grouping differs in its locality: a short, non-overlapping window collected under a fixed policy, whose mean serves as a local surrogate for the same-state group mean. Section 4.3 compares this local baseline with a running mean at a matched update count.

Test-time adaptation under domain shift. Beyond training-time domain adaptation (Ganin et al., 2016; Hoffman et al., 2018; Tobin et al., 2017), test-time methods adapt to the target stream through entropy minimization and continual self-training (Wang et al., 2021; 2022). Test-time reinforcement learning extends this idea to vision–language models, where TTRV applies GRPO to repeated samples of each test input (Singh et al., 2026). In driving, cross-city deployment remains a practical challenge

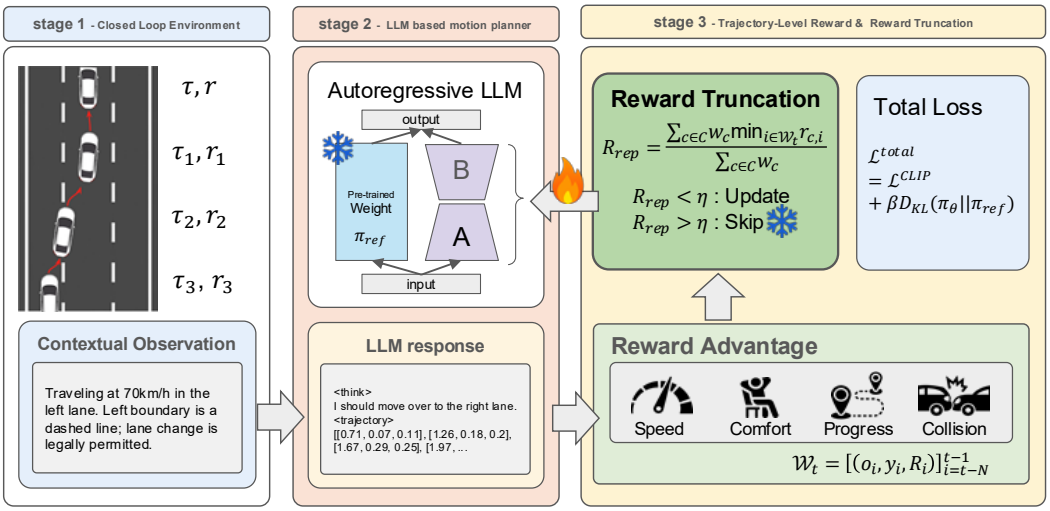

Figure 2: **TeTO framework.** Recent closed-loop interactions form a sequential group whose rewards yield relative advantages. Reward Truncation decides whether the group updates the LoRA adapter, and the frozen backbone serves as the KL reference. The collision icon denotes the TTC component.

for end-to-end planners (Yasarla et al., 2026). LLaDA adapts motion plans to local traffic rules (Li et al., 2024), T4P adapts trajectory prediction (Park et al., 2024), AdaWM selectively fine-tunes a world model or policy (Wang et al., 2025), and Centaur updates an end-to-end planner by minimizing the uncertainty of its decisions (Sima et al., 2025a). None of these driving methods updates an LLM planner from closed-loop reward feedback at test time. TeTO addresses this setting, using low-rank adaptation and adapter-based reference sharing (Hu et al., 2022; Santacroce et al., 2023) to keep these updates practical on a 7B backbone.

3 Method

Figure 2 shows TeTO's closed-loop pipeline: the supervised planner, extended with an online LoRA adapter, acts in the environment, Sequential Grouping turns its recent interactions into relative advantages, and Reward Truncation decides which windows update the adapter.

3.1 Closed-loop planning and trajectory feedback

At step i, the planner observes $o_i = f(s_i)$, a serialization of ego history, nearby agents, map information, and route context. The autoregressive policy generates a response $y_i \sim \pi_\theta(\cdot \mid o_i)$ containing reasoning and structured waypoints. A deterministic parser extracts the plan $\tau_i = \mathrm{parse}(y_i)$, and the controller advances the simulator according to $P(s_{i+1} \mid s_i, \tau_i)$. Each plan contains 16 poses at 0.5-second intervals; replanning occurs every 0.1 seconds of simulation time.

The reward is a state-conditioned trajectory utility over comfort, TTC, speed adherence, and route progress, following the reward design of Plan-R1 (Tang et al., 2026). With $\mathcal{C}$ denoting these components,

$$R_i = \mathrm{clip}\left(\frac{\sum_{c\in\mathcal{C}} w_c r_{c,i}}{\sum_{c\in\mathcal{C}} w_c}, 0, 1\right), \quad (w_{\text{comfort}}, w_{\text{ttc}}, w_{\text{speed}}, w_{\text{progress}}) = (2, 5, 4, 2). \tag{1}$$

A trajectory evaluator computes each component from the current simulator state and the selected 8-second plan, including its predicted TTC. Appendix A defines the components and the observation and prompt interface.

3.2 SEQUENTIAL GROUPING

For a state s_t, GRPO samples G responses and normalizes each reward by the group mean μ_G and standard deviation σ_G (Shao et al., 2024b):

$$A^{(g)} = \frac{R(s_t, \tau^{(g)}) - \mu_G}{\sigma_G}, \qquad g = 1, \ldots, G, \tag{2}$$

so that μ_G estimates the expected reward at s_t under the current policy.

Why not a critic, a world model, or same-state sampling? A PPO-style critic (Schulman et al., 2017) is fit on the training domain, so its values are least reliable under the shift that motivates adaptation, and a 150-step scenario is too short to refit it. A world model that simulates alternative plans (Ha & Schmidhuber, 2018; Hu et al., 2023) faces the same shift, and its errors bias the advantages. Same-state groups require restoring the world state, which only a simulator allows, or sampling G plans per step and scoring the unexecuted candidates through predicted agent motion, which multiplies autoregressive decoding, the dominant planner cost, by G (Appendix B.1).

Sequential Grouping instead builds the group along the temporal axis of the executed timeline, from the N most recent interactions:

$$\mathcal{W}_t = \{(o_i, y_i, R_i, \{r_{c,i}\}_{c \in \mathcal{C}})\}_{i=t-N}^{t-1}. \tag{3}$$

We use non-overlapping windows with $N = 3$ and hold the policy fixed while a window is collected. The baseline, scale, and advantages are

$$b_t = \frac{1}{N} \sum_{i \in \mathcal{W}_t} R_i, \qquad \sigma_t = \sqrt{\frac{1}{N} \sum_{i \in \mathcal{W}_t} (R_i - b_t)^2}, \qquad A_i = \frac{R_i - b_t}{\sigma_t + \epsilon}, \tag{4}$$

where $\epsilon > 0$ stabilizes division. Advantages are fixed targets during optimization, and the buffer is cleared after each update decision, including one skipped by Reward Truncation (Section 3.4), so every interaction belongs to exactly one group. The group reuses plans that the planner generates and executes anyway, adding neither decoding nor a critic.

Local calibration of the temporal baseline. Let $m_i = \mathbb{E}[R \mid s_i, \bar{\pi}_t]$ be the expected reward at s_i under the window's fixed policy $\bar{\pi}_t$, the quantity that μ_G estimates in Equation (2), and let $e_i = R_i - m_i$. For every realized window, the triangle inequality gives

$$|m_i - b_t| \le \underbrace{\frac{1}{N} \sum_{j \in \mathcal{W}_t} |m_i - m_j|}_{\text{local context drift}} + \underbrace{\left| \frac{1}{N} \sum_{j \in \mathcal{W}_t} e_j \right|}_{\text{sampling error}}. \tag{5}$$

Under a local L-Lipschitz condition, the first term is at most $LN^{-1} \sum_j d(s_i, s_j)$, which a short window keeps small. The second term corresponds to the finite-sample error of a same-state group of size N, here with serial dependence (derivation in Appendix B).

Figure 3 compares the two groupings, with parallel groups drawn from 20 plans sampled at each initial observation; beyond $n = 10$, the sequential variance grows as a longer window spans more diverse contexts. We use $N = 3$, where the two curves coincide; varying N from 2 to 4 changes the closed-loop score by at most 0.25 points (Appendix E.4).

3.3 REFERENCE-REGULARIZED ADAPTER UPDATES

The supervised checkpoint defines the frozen reference π_{ref}, and a new LoRA adapter defines the adapting policy π_θ. Over the generated positions $\mathcal{G}_i$ of a response, including reasoning and waypoints, we compute the response-level ratio of geometric-mean token probabilities

$$\bar{\ell}_\theta(y_i \mid o_i) = \frac{1}{|\mathcal{G}_i|} \sum_{j \in \mathcal{G}_i} \log \pi_\theta(y_{i,j} \mid o_i, y_{i,<j}), \qquad \rho_i = \exp\big(\bar{\ell}_\theta(y_i \mid o_i) - \bar{\ell}_{\text{ref}}(y_i \mid o_i)\big), \tag{6}$$

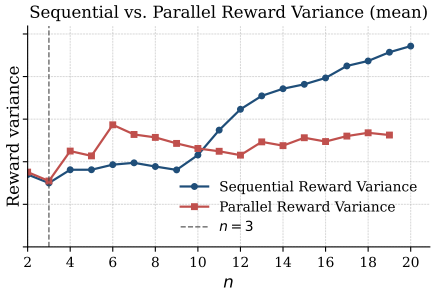

Figure 3: **Reward variance across group sizes n (scenario average).** Parallel groups consist of n plans sampled from the same initial observation and scored without advancing the simulator; sequential groups consist of n consecutive executed plans. The two curves coincide at $n = 3$, the sequential variance stays at or below the parallel variance up to $n = 10$, and it grows steadily beyond.

and minimize the clipped, reference-regularized objective

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i\in\mathcal{W}_t} \min(\rho_i A_i, \mathrm{clip}(\rho_i, 1-\epsilon_c, 1+\epsilon_c)A_i) + \frac{\beta}{N}\sum_{i\in\mathcal{W}_t} K_i(\theta, \mathrm{ref}), \tag{7}$$

where K_i is the full-response KL divergence to the reference, $\epsilon_c = 0.2$, and $\beta = 0.02$. The ratio is taken against π_{ref} rather than the collecting policy: that policy is fixed during collection and each window receives a single gradient step, so a ratio against it would equal one at every update and the clip would never act. Against π_{ref}, the clip forms a trust region around the supervised planner, complementing the smooth KL penalty. Reasoning tokens condition the subsequent waypoints, so trajectory-derived credit is assigned to the complete response (Appendix D.3). Only the online LoRA parameters are optimized; temporarily disabling the adapter yields reference probabilities from the same frozen backbone (Santacroce et al., 2023), into which the SFT adapter is merged beforehand, so no second checkpoint is stored.

3.4 Reward Truncation

Updating after every window is often unnecessary. Because Equation (4) standardizes rewards within each window, a window whose interactions all score highly still yields unit-scale advantages unless its rewards are nearly identical. These advantages are driven by small differences that carry little information about action quality, and updating on them can perturb a policy that already behaves well. RT therefore summarizes each window by a bottleneck statistic, the weighted minimum of each reward component:

$$R_{\mathrm{rep}} = \frac{\sum_{c\in\mathcal{C}} w_c \min_{i\in\mathcal{W}_t} r_{c,i}}{\sum_{c\in\mathcal{C}} w_c}, \qquad g_t = \mathbb{I}[R_{\mathrm{rep}} \le \eta], \qquad \eta = 0.70. \tag{8}$$

A single degraded component anywhere in the window lowers R_{rep} even when the others are saturated, whereas a window mean can hide it. The gate only selects windows: TeTO updates on every complete window, TeTO + RT on the selected ones, and each issued update uses the unchanged advantages of Equation (4). The planner still replans at every simulation step (Algorithm 1).

4 Experiments

4.1 Experimental setup

Backbones and evaluation. Our default planner is Llama-2-7B (Touvron et al., 2023), fine-tuned with SFT on 0.3M Las Vegas training examples with InstructChain prompting (Zhang et al., 2024). We evaluate in the nuPlan closed-loop simulator (Caesar et al., 2021; Karnchanachari et al., 2024) on Test14-Hard (Cheng et al., 2024b): Las Vegas is in-domain (ID), and Singapore is out-of-domain (OOD). Nonreactive (NR) agents follow logged motion; reactive (R) agents can respond during simulation. Closed-loop scores (CLS) use a 0–100 scale. We also evaluate a Qwen2.5-7B planner

Table 2: **Test14-Hard with Llama-2-7B.** Top: closed-loop scores; bottom: average updates per scenario (fewer is better). Best in bold, second best underlined.

	Las Vegas (ID)		Singapore (OOD)	
Method	NR	R	NR	R
SFT	78.72	80.87	62.09	72.33
REINFORCE	80.01	80.31	71.61	73.15
REINFORCE++	79.87	78.11	72.21	75.55
RLOO	78.28	78.58	68.64	75.66
RWR	79.54	78.46	<u>72.39</u>	76.06
DPO	79.50	79.65	70.79	75.94
TENT	78.14	80.91	71.64	75.07
TeTO	**81.33**	<u>81.59</u>	71.25	<u>77.92</u>
TeTO + RT	<u>80.36</u>	**82.88**	**72.47**	**77.96**
Average number of updates				
Dense baselines	149.2	149.1	149.3	149.3
TeTO	<u>49.1</u>	<u>49.1</u>	<u>49.1</u>	<u>49.1</u>
TeTO + RT	**20.9**	**29.0**	**11.7**	**28.3**

(Qwen et al., 2024), additional cities on Test14-Random, and a planner trained on mixed nuPlan–Waymo data.

Baselines and implementation. We adapt six objectives to the same single-timeline interface: REINFORCE (Williams, 1992), REINFORCE++ (Hu et al., 2025), RLOO (Ahmadian et al., 2024), reward-weighted regression (RWR; Peters & Schaal, 2007), DPO (Rafailov et al., 2023), and TENT (Wang et al., 2021). All use the same backbone, online LoRA, optimizer, and decoding settings; REINFORCE++, RLOO, and DPO form their groups from the three most recent interactions, and TENT minimizes response entropy without reward. PPO and vanilla GRPO are not included because their native forms require an online critic or same-state groups (Section 3.2). Appendix C specifies the implemented objectives.

Online LoRA has rank 16, scaling 32, and dropout 0.05 in the query and value projections. AdamW uses learning rate 10^{-4}, weight decay 0.01, and gradient clipping at 1.0. Adapters reset at each scenario; sampling uses temperature 0.75 and top-$p = 0.9$. Experiments use one NVIDIA A6000 48GB. Appendix A gives the training settings, prompts, and interaction algorithm.

4.2 Planning quality and update efficiency

Adaptation under domain shift. Table 2 shows gains over the SFT planner in all four settings. In Singapore, TeTO + RT raises NR-CLS from 62.09 to 72.47 (+10.38, a 16.7% relative gain) and R-CLS from 72.33 to 77.96 (+5.63). TeTO performs about 49 updates per scenario, and RT further reduces the count to 11.7 in OOD-NR and 28.3 in OOD-R. The OOD-NR count is 7.8% of dense adaptation, a 92.2% reduction. TeTO + RT is best in both Singapore settings, both TeTO variants exceed every baseline in Singapore reactive traffic, and in Las Vegas TeTO and TeTO + RT are best in NR and R, respectively.

Figure 4 shows the gate in a Singapore episode: updates occur while rewards are low early in the episode, after which TeTO + RT follows the curved road, whereas SFT drifts across the lane boundary by t=85. In further Singapore scenarios (Appendix G), SFT departs from its lane, REINFORCE fails to stabilize, and both TeTO variants produce lane-consistent trajectories.

In Figure 5, both TeTO variants score above every dense baseline with a fraction of the updates. In the reward trace, REINFORCE's reward falls sharply after timestep 110, consistent with policy drift under per-step updates, while both TeTO variants stay near 0.9.

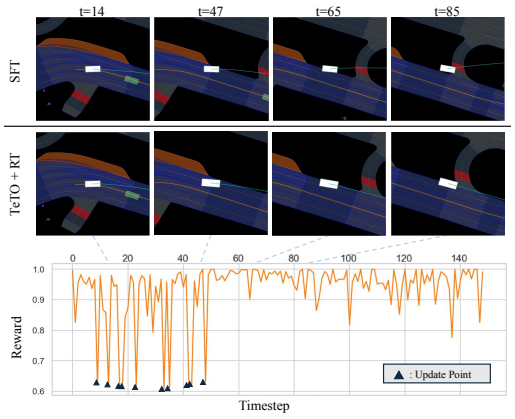

Figure 4: **Reward Truncation during an OOD episode.** Top: SFT and TeTO + RT trajectories at four timesteps. Bottom: reward evolution, with triangles marking parameter updates. The gate concentrates updates on low-reward windows while planning continues through the episode.

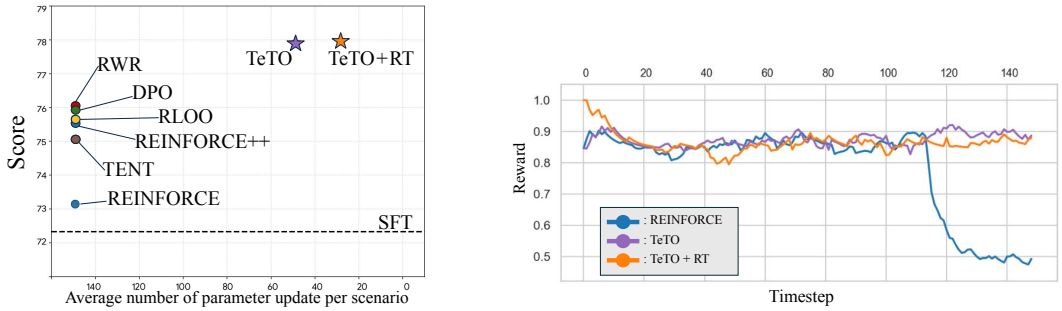

Figure 5: **Efficiency and reward evolution.** Left: Singapore reactive score versus average updates per scenario. Right: reward history in an OOD episode. TeTO variants retain high rewards after timestep 110, when REINFORCE's reward falls.

Transfer across backbones and cities. With a Qwen2.5-7B backbone (Table 3), TeTO + RT improves Singapore NR/R by 10.43/3.22 points over its SFT initialization and achieves the best OOD scores of all compared planners, whereas PlanTF and Plan-R1, the strongest planners in Las Vegas, fall below 40 in Singapore. On Test14-Random, a TeTO variant achieves the highest reactive score in each of Las Vegas, Singapore, Pittsburgh, and Boston. With broader SFT data comprising 0.3M nuPlan and 0.3M Waymo Open Motion Dataset examples (Ettinger et al., 2021), TeTO improves nuPlan Singapore R-CLS from 74.05 to 77.31 (Appendix E).

4.3 What contributes to adaptation?

Local baseline at a matched update count. At a fixed window and update count (Table 4), replacing SG's local baseline with a running mean under the same objective lowers R-CLS from 77.92 to 76.77, and windowed RWR over the same responses reaches 76.87, so the local baseline itself contributes to the gain.

Signal consistency within short windows. An offline cross-evaluation of 491 complete windows compares SG with a same-state plan-scoring reference. In the 95.9% of windows whose TTC component stays constant, the SG and same-state advantages agree in sign in 92.3% of cases and the baselines differ by 0.0036 on average; on the windows where RT issues updates, agreement is 89.5%. Agreement drops only in the 4.1% of windows where the TTC component switches, which make up 5.3% of the windows selected by RT (Appendix B).

Table 3: **Comparison with established planners on Test14-Hard.** TeTO rows adapt a Qwen2.5-7B planner fine-tuned on Las Vegas data. Best in bold, second best underlined.

	Las Vegas (ID)		Singapore (OOD)	
Method	NR	R	NR	R
IDM (Treiber et al., 2000)	66.43	72.99	38.08	47.04
PlanTF (Cheng et al., 2024b)	**86.55**	87.07	38.47	36.81
PLUTO (Cheng et al., 2024a)	74.47	73.06	70.10	68.88
Diffusion Planner (Zheng et al., 2025b)	80.74	78.98	46.10	51.70
Plan-R1 (Tang et al., 2026)	85.10	**93.20**	26.15	22.40
Qwen-SFT	80.86	81.12	75.32	76.93
TeTO	<u>85.97</u>	<u>89.38</u>	<u>85.42</u>	<u>79.02</u>
TeTO + RT	85.08	87.69	**85.75**	**80.15**

Table 4: **Matched-update controls in Singapore reactive traffic.** The running-mean control shares TeTO's objective and optimizer; windowed RWR shares its three-response window and update count.

Method	Updates/scenario	R-CLS
Windowed RWR	49.1	76.87
TeTO with running mean	49.1	76.77
TeTO with Sequential Grouping	49.1	<u>77.92</u>
TeTO + RT	**28.3**	**77.96**

Gating and response credit. The component-minimum gate reaches 77.96 R-CLS with 28.3 updates, compared with 76.98 at 29.3 updates for a window-mean gate, and 75.63 versus 75.19 under TTC reward noise with standard deviation 0.3. The window-level gate also outperforms thresholding the instantaneous reward in all four ID/OOD settings, and crediting the full response, including its reasoning tokens, stabilizes adaptation (Appendix D).

4.4 Synchronized adaptation cost

We measure adaptation cost with CUDA synchronization around each GPU phase in a separate rerun of the Test14-Random Singapore evaluation. Amortized over all control steps, including forward and backward passes through the full 6.75B-parameter backbone, gradient clipping, and optimizer steps, dense REINFORCE spends 945.9 ms per step on adaptation and TeTO + RT 64.6 ms, 93.2% less. The saving comes from updating rarely: TeTO + RT issues 7.8 updates per scenario in this run, 5.2% of the dense count. Autoregressive generation is shared by all methods and unaffected by TeTO; Appendix F gives the phase-level breakdown.

5 Conclusion

We presented TeTO, which makes critic-free, group-relative policy optimization available on a single executed driving timeline. Sequential Grouping replaces unobtainable same-state rollouts with a temporally local baseline whose deviation is bounded by local reward drift and sampling error, and Reward Truncation concentrates updates on windows with degraded components without changing the gradient of an issued update. Together, they improve out-of-domain closed-loop planning across two LLM backbones, three unseen cities, and mixed-data training while updating on as few as 7.8% of steps. An LLM planner can thus be adapted online at a small fraction of the dense update cost.

Limitations and future work. The reward shares kinematic and safety criteria with the nuPlan score, and our evaluation is conducted in simulation. Adaptive windows that shorten around abrupt reward changes, learned reward models, and other embodied settings with a single irreversible timeline are natural extensions.

AI USE STATEMENT

We used large language models to aid and polish the writing of this paper. All research content is the authors' own; the authors reviewed every AI-assisted edit and take full responsibility for the paper.

REPRODUCIBILITY STATEMENT

Appendix A provides the observation schema, prompts, training settings, rewards, and online algorithm. Appendices B–F provide estimator diagnostics, baseline objectives, ablations, extended evaluation, and synchronized timing protocols.

ETHICS STATEMENT

This study evaluates autonomous driving in simulation. Online optimization can amplify perception, prediction, map, or reward errors, so physical deployment would require independently validated feedback and safety evaluation beyond the reported experiments.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to Basics: Revisiting REINFORCE-Style Optimization for Learning from Human Feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024. doi: 10.18653/v1/2024.acl-long.662. URL https://aclanthology.org/2024.acl-long.662/.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to End Learning for Self-Driving Cars. *arXiv preprint arXiv:1604.07316*, 2016. URL https://arxiv.org/abs/1604.07316.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. NuPlan: A closed-loop ML-based planning benchmark for autonomous vehicles. In *CVPR ADP3 Workshop*, 2021. URL https://arxiv.org/abs/2106.11810.

Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-End Autonomous Driving: Challenges and Frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10164–10183, 2024a. doi: 10.1109/TPAMI.2024.3435937. URL https://doi.org/10.1109/tpami.2024.3435937.

Long Chen, Oleg Sinavski, Jan Hünermann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. Driving with LLMs: Fusing Object-Level Vector Modality for Explainable Autonomous Driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14093–14100. IEEE, 2024b. doi: 10.1109/ICRA57147.2024.10611018. URL https://doi.org/10.1109/icra57147.2024.10611018.

Yuan Chen, Zi-han Ding, Ziqin Wang, Yan Wang, Lijun Zhang, and Si Liu. Asynchronous Large Language Model Enhanced Planner for Autonomous Driving. In *Computer Vision – ECCV 2024*, volume 15094 of *Lecture Notes in Computer Science*, pp. 22–38. Springer Nature Switzerland, 2025. doi: 10.1007/978-3-031-72764-1_2. URL https://link.springer.com/chapter/10.1007/978-3-031-72764-1_2.

Jie Cheng, Yingbing Chen, and Qifeng Chen. PLUTO: Pushing the Limit of Imitation Learning-based Planning for Autonomous Driving. *arXiv preprint arXiv:2404.14327*, 2024a. URL https://arxiv.org/abs/2404.14327.

Jie Cheng, Yingbing Chen, Xiaodong Mei, Bowen Yang, Bo Li, and Ming Liu. Rethinking Imitation-based Planners for Autonomous Driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14123–14130, 2024b. doi: 10.1109/ICRA57147.2024.10611364. URL https://ieeexplore.ieee.org/document/10611364/.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep Reinforcement Learning from Human Preferences. In *Advances in Neural Information Processing Systems*, volume 30, pp. 4299–4307. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf.

Felipe Codevilla, Eder Santana, Antonio M. Lopez, and Adrien Gaidon. Exploring the Limitations of Behavior Cloning for Autonomous Driving. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9328–9337. IEEE, 2019. doi: 10.1109/ICCV.2019.00942. URL https://doi.org/10.1109/iccv.2019.00942.

Muzhi Dai, Chenxu Yang, and Qingyi Si. S-GRPO: Early Exit via Reinforcement Learning in Reasoning Models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 48178–48204, 2025. doi: 10.52202/085713-1608. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/44f09d1973a88529b40029e6c3693ded-Abstract-Conference.html.

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*, pp. 9710–9719, 2021. doi: 10.1109/ICCV48922.2021.00957. URL https://openaccess.thecvf.com/content/ICCV2021/html/Ettinger_Large_Scale_Interactive_Motion_Forecasting_for_Autonomous_Driving_The_Waymo_ICCV_2021_paper.html.

Elias Frantar and Dan Alistarh. SparseGPT: Massive Language Models Can Be Accurately Pruned in One-Shot. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10323–10337, 2023. URL https://proceedings.mlr.press/v202/frantar23a.html.

Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Hongwei Xie, Bing Wang, Guang Chen, Hangjun Ye, Dingkang Liang, and Xiang Bai. MindDrive: A Vision-Language-Action Model for Autonomous Driving via Online Reinforcement Learning. *arXiv preprint arXiv:2512.13636*, 2025. URL https://arxiv.org/abs/2512.13636.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-Adversarial Training of Neural Networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016. URL https://www.jmlr.org/papers/v17/15-239.html.

David Ha and Jürgen Schmidhuber. Recurrent World Models Facilitate Policy Evolution. In *Advances in Neural Information Processing Systems*, volume 31, pp. 2450–2462. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/2de5d16682c3c35007e4e92982f1a2ba-Paper.pdf.

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. CyCADA: Cycle-Consistent Adversarial Domain Adaptation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1989–1998. PMLR, 2018. URL https://proceedings.mlr.press/v80/hoffman18a.html.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A Generative World Model for Autonomous Driving. *arXiv preprint arXiv:2309.17080*, 2023. URL https://arxiv.org/abs/2309.17080.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. REINFORCE++: Stabilizing Critic-Free Policy Optimization with Global Advantage Normalization. *arXiv preprint arXiv:2501.03262*, 2025. URL https://arxiv.org/abs/2501.03262.

Napat Karnchanachari, Dimitris Geromichalos, Kok Seang Tan, Nanxiang Li, Christopher Eriksen, Shakiba Yaghoubi, Noushin Mehdipour, Gianmarco Bernasconi, Whye Kit Fong, Yiluan Guo, and Holger Caesar. Towards learning-based planning: The nuPlan benchmark for real-world autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 629–636, 2024. doi: 10.1109/ICRA57147.2024.10610077. URL https://ieeexplore.ieee.org/document/10610077/.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023. doi: 10.1145/3600006.3613165. URL https://dl.acm.org/doi/10.1145/3600006.3613165.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast Inference from Transformers via Speculative Decoding. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19274–19286, 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

Boyi Li, Yue Wang, Jiageng Mao, Boris Ivanovic, Sushant Veer, Karen Leung, and Marco Pavone. Driving Everywhere with Large Language Model Policy Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14948–14957, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Li_Driving_Everywhere_with_Large_Language_Model_Policy_Adaptation_CVPR_2024_paper.html.

Yue Li, Meng Tian, Dechang Zhu, Jiangtong Zhu, Zhenyu Lin, Zhiwei Xiong, and Xinhai Zhao. Drive-R1: Bridging Reasoning and Planning in VLMs for Autonomous Driving with Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(8):6708–6716, 2026. doi: 10.1609/aaai.v40i8.37602. URL https://ojs.aaai.org/index.php/AAAI/article/view/37602.

Yuechen Luo, Fang Li, Shaoqing Xu, Zhiyi Lai, Lei Yang, Qimao Chen, Ziang Luo, Zixun Xie, Shengyin Jiang, Jiaxin Liu, Long Chen, Bing Wang, and Zhi-Xin Yang. AdaThinkDrive: Adaptive Thinking via Reinforcement Learning for Autonomous Driving. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026. URL https://ras.papercept.net/conferences/conferences/ICRA26/program/ICRA26_ContentListWeb_4.html#wei2i_303.

Jiageng Mao, Yuxi Qian, Junjie Ye, Hang Zhao, and Yue Wang. GPT-Driver: Learning to Drive with GPT. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023. URL https://openreview.net/forum?id=Pvjk9lxLJK.

NVIDIA, Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, Liang Feng, Greg Heinrich, Jack Huang, Peter Karkus, Boyi Li, Pinyi Li, Tsung-Yi Lin, Dongran Liu, Ming-Yu Liu, Langechuan Liu, Zhijian Liu, Jason Lu, Yunxiang Mao, Pavlo Molchanov, Lindsey Pavao, Zhenghao Peng, Mike Ranzinger, Ed Schmerling, Shida Shen, Yunfei Shi, Sarah Tariq, Ran Tian, Tilman Wekel, Xinshuo Weng, Tianjun Xiao, Eric Yang, Xiaodong Yang, Yurong You, Xiaohui Zeng, Wenyuan Zhang, Boris Ivanovic, and Marco Pavone. Alpamayo-R1: Bridging Reasoning and Action Prediction for Generalizable Autonomous Driving in the Long Tail. *arXiv preprint arXiv:2511.00088*, 2025. URL https://arxiv.org/abs/2511.00088.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. doi: 10.52202/068431-2011. URL https://proceedings.neurips.cc/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Chenbin Pan, Burhaneddin Yaman, Tommaso Nesti, Abhirup Mallik, Alessandro G Allievi, Senem Velipasalar, and Liu Ren. VLP: Vision Language Planning for Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14760–14769, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Pan_VLP_Vision_Language_Planning_for_Autonomous_Driving_CVPR_2024_paper.html.

Daehee Park, Jaeseok Jeong, Sung-Hoon Yoon, Jaewoo Jeong, and Kuk-Jin Yoon. T4P: Test-Time Training of Trajectory Prediction via Masked Autoencoder and Actor-specific Token Memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15065–15076, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Park_T4P_Test-Time_Training_of_Trajectory_Prediction_via_Masked_Autoencoder_and_CVPR_2024_paper.html.

Jan Peters and Stefan Schaal. Reinforcement Learning by Reward-Weighted Regression for Operational Space Control. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 745–750, 2007. doi: 10.1145/1273496.1273590. URL https://doi.org/10.1145/1273496.1273590.

Dean A. Pomerleau. ALVINN: An Autonomous Land Vehicle in a Neural Network. In *Advances in Neural Information Processing Systems*, volume 1, pp. 305–313. Morgan-Kaufmann, 1988. URL https://proceedings.neurips.cc/paper/1988/hash/812b4ba287f5ee0bc9d43bbf5bbe87fb-Abstract.html.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741. Curran Associates, Inc., 2023. doi: 10.52202/075280-2338. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Giménez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals,

Mahyar Bordbar, and Nando de Freitas. A Generalist Agent. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=1ikK0kHjvj.

Michael Santacroce, Yadong Lu, Han Yu, Yuanzhi Li, and Yelong Shen. Efficient RLHF: Reducing the Memory Usage of PPO. *arXiv preprint arXiv:2309.00754*, 2023. URL https://arxiv.org/abs/2309.00754.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Hao Sha, Yao Mu, Yuxuan Jiang, Li Chen, Chenfeng Xu, Ping Luo, Shengbo Eben Li, Masayoshi Tomizuka, Wei Zhan, and Mingyu Ding. LanguageMPC: Large Language Models as Decision Makers for Autonomous Driving. *arXiv preprint arXiv:2310.03026*, 2023. URL https://arxiv.org/abs/2310.03026.

Hao Shao, Yuxuan Hu, Letian Wang, Guanglu Song, Steven L. Waslander, Yu Liu, and Hongsheng Li. LMDrive: Closed-Loop End-to-End Driving with Large Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15120–15130, 2024a. URL https://openaccess.thecvf.com/content/CVPR2024/html/Shao_LMDrive_Closed-Loop_End-to-End_Driving_with_Large_Language_Models_CVPR_2024_paper.html.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024b. URL https://arxiv.org/abs/2402.03300.

SP Sharan, Francesco Pittaluga, Vijay Kumar B G, and Manmohan Chandraker. LLM-Assist: Enhancing Closed-Loop Planning with Language-Based Reasoning. *arXiv preprint arXiv:2401.00125*, 2023. URL https://arxiv.org/abs/2401.00125.

Chonghao Sima, Kashyap Chitta, Zhiding Yu, Shiyi Lan, Ping Luo, Andreas Geiger, Hongyang Li, and Jose M. Alvarez. Centaur: Robust End-to-End Autonomous Driving with Test-Time Training. *arXiv preprint arXiv:2503.11650*, 2025a. URL https://arxiv.org/abs/2503.11650.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. DriveLM: Driving with Graph Visual Question Answering. In *Computer Vision – ECCV 2024*, volume 15110 of *Lecture Notes in Computer Science*, pp. 256–274. Springer Nature Switzerland, 2025b. doi: 10.1007/978-3-031-72943-0_15. URL https://link.springer.com/chapter/10.1007/978-3-031-72943-0_15.

Akshit Singh, Shyam Marjit, Wei Lin, Paul Gavrikov, Serena Yeung-Levy, Hilde Kuehne, Rogerio Feris, Sivan Doveh, James Glass, and M. Jehanzeb Mirza. TTRV: Test-Time Reinforcement Learning for Vision Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33153–33163, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Singh_TTRV_Test-Time_Reinforcement_Learning_for_Vision_Language_Models_CVPR_2026_paper.html.

Xiaolong Tang, Meina Kan, Shiguang Shan, and Xilin Chen. Plan-R1: Safe and Feasible Trajectory Planning as Language Modeling. In *International Conference on Learning Representations*, volume 2026, pp. 23715–23735, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/28a80012e6f564c0cc8e7661e1db83fe-Abstract-Conference.html.

Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, XianPeng Lang, and Hang Zhao. DriveVLM: The Convergence of Autonomous Driving and Large Vision-Language Models. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4698–4726. PMLR, 2025. URL https://proceedings.mlr.press/v270/tian25c.html.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30, 2017. doi: 10.1109/IROS.2017.8202133. URL https://doi.org/10.1109/IROS.2017.8202133.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*, 2023. URL https://arxiv.org/abs/2307.09288.

Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested Traffic States in Empirical Observations and Microscopic Simulations. *Physical Review E*, 62(2):1805–1824, 2000. doi: 10.1103/PhysRevE.62.1805. URL https://journals.aps.org/pre/abstract/10.1103/PhysRevE.62.1805.

Miroslav Štrupl, Francesco Faccio, Dylan R. Ashley, Rupesh Kumar Srivastava, and Jürgen Schmidhuber. Reward-Weighted Regression Converges to a Global Optimum. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(8):8361–8369, 2022. doi: 10.1609/aaai.v36i8.20811. URL https://ojs.aaai.org/index.php/AAAI/article/view/20811.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully Test-Time Adaptation by Entropy Minimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uXl3bZLkr3c.

Hang Wang, Xin Ye, Feng Tao, Chenbin Pan, Abhirup Mallik, Burhan Yaman, Liu Ren, and Junshan Zhang. AdaWM: Adaptive World Model based Planning for Autonomous Driving. In *International Conference on Learning Representations*, pp. 85591–85615, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/d4c745dcbaf8ef0d7e145754e31b1516-Abstract-Conference.html.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual Test-Time Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7201–7211, 2022. doi: 10.1109/CVPR52688.2022.00706. URL https://openaccess.thecvf.com/content/CVPR2022/html/Wang_Continual_Test-Time_Domain_Adaptation_CVPR_2022_paper.html.

Xinyang Wang, Qian Liu, Wenjie Ding, Zhao Yang, Wei Li, Chang Liu, Bailin Li, Kun Zhan, Xianpeng Lang, and Wei Chen. Unifying Language-Action Understanding and Generation for Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25193–25203, 2026a. URL https://openaccess.thecvf.com/content/CVPR2026/html/Wang_Unifying_Language-Action_Understanding_and_Generation_for_Autonomous_Driving_CVPR_2026_paper.html.

Yinuo Wang, Mining Tan, Yuanxin Zhong, Zhitao Wang, and Siyuan Cheng. Bird's-eye-view Informed Reasoning Driver. In *International Conference on Learning Representations*, volume 2026, pp. 105888–105905, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/acb18f946cde3cc29864e6df7df54d11-Paper-Conference.pdf.

Ronald J. Williams. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 8(3):229–256, 1992. doi: 10.1007/BF00992696. URL https://link.springer.com/article/10.1007/BF00992696.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 38087–38099, 2023. URL https://proceedings.mlr.press/v202/xiao23c.html.

Minhao Xiong, Zichen Wen, Zhuangcheng Gu, Xuyang Liu, Rui Zhang, Hengrui Kang, Jiabing Yang, Junyuan Zhang, Weijia Li, Conghui He, and Linfeng Zhang. Prune2Drive: A Plug-and-Play Framework for Accelerating Vision-Language Models in Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25215–25224, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Xiong_Prune2Drive_A_Plug-and-Play_Framework_for_Accelerating_Vision-Language_Models_in_Autonomous_CVPR_2026_paper.html.

Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K. Wong, Zhenguo Li, and Hengshuang Zhao. DriveGPT4: Interpretable End-to-End Autonomous Driving Via Large Language Model. *IEEE Robotics and Automation Letters*, 9(10):8186–8193, 2024. doi: 10.1109/LRA.2024.3440097. URL https://ieeexplore.ieee.org/document/10629039/.

Zhenjie Yang, Xiaosong Jia, Hongyang Li, and Junchi Yan. LLM4Drive: A Survey of Large Language Models for Autonomous Driving. *arXiv preprint arXiv:2311.01043*, 2023. URL https://arxiv.org/abs/2311.01043.

Rajeev Yasarla, Shizhong Han, Hsin-Pai Cheng, Apratim Bhattacharyya, Shweta Mahajan, Litian Liu, Yunxiao Shi, Risheek Garrepalli, Hong Cai, and Fatih Porikli. RoCA: Robust Cross-Domain End-to-End Autonomous Driving. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/64997.

Yuqi Ye, Zijian Zhang, Junhong Lin, Shangkun Sun, Changhao Peng, and Wei Gao. AutoDrive-P^3: Unified Chain of Perception-Prediction-Planning Thought via Reinforcement Fine-Tuning. In *International Conference on Learning Representations*, pp. 31007–31036, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/34515e20a0ec8c894c29fbd65d403678-Abstract-Conference.html.

Zhenlong Yuan, Chengxuan Qian, Jing Tang, Rui Chen, Zijian Song, Lei Sun, Xiangxiang Chu, Yujun Cai, Dapeng Zhang, and Shuo Li. AutoDrive-R^2: Incentivizing Reasoning and Self-Reflection Capacity for VLA Model in Autonomous Driving. In *International Conference on Learning Representations*, volume 2026, pp. 97934–97955, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/9f35ec2f7f403ef2c83d65b581df10bc-Paper-Conference.pdf.

Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A Survey of Autonomous Driving: Common Practices and Emerging Technologies. *IEEE Access*, 8:58443–58469, 2020. doi: 10.1109/ACCESS.2020.2983149. URL https://ieeexplore.ieee.org/document/9046805/.

Ruijun Zhang, Xianda Guo, Wenzhao Zheng, Chenming Zhang, Kurt Keutzer, and Long Chen. Instruct Large Language Models to Drive like Humans. *arXiv preprint arXiv:2406.07296*, 2024. URL https://arxiv.org/abs/2406.07296.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group Sequence Policy Optimization. *arXiv preprint arXiv:2507.18071*, 2025a. URL https://arxiv.org/abs/2507.18071.

Yinan Zheng, Ruiming Liang, Kexin Zheng, Jinliang Zheng, Liyuan Mao, Jianxiong Li, Weihao Gu, Rui Ai, Shengbo Eben Li, Xianyuan Zhan, and Jingjing Liu. Diffusion-Based Planning for Autonomous Driving with Flexible Guidance. In *International Conference on Learning Representations*, volume 2025, pp. 37207–37227, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/5c0cd43063f087eb32e572d040a19cd2-Paper-Conference.pdf.

Yupeng Zheng, Zebin Xing, Qichao Zhang, Bu Jin, Pengfei Li, Yuhang Zheng, Zhongpu Xia, Yaran Chen, and Dongbin Zhao. PlanAgent: A Multimodal Large Language Agent for Closed-Loop Vehicle Motion Planning. *IEEE Transactions on Cognitive and Developmental Systems*, 18(4): 1113–1126, 2026. doi: 10.1109/TCDS.2026.3664120. URL https://ieeexplore.ieee.org/document/11394788/.

Xingcheng Zhou, Xuyuan Han, Feng Yang, Yunpu Ma, Volker Tresp, and Alois Knoll. OpenDriveVLA: Towards End-to-end Autonomous Driving with Large Vision Language Action Model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(16):13782–13790, 2026. doi: 10.1609/aaai.v40i16.38386. URL https://ojs.aaai.org/index.php/AAAI/article/view/38386.

Zewei Zhou, Tianhui Cai, Seth Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. AutoVLA: A Vision-Language-Action Model for End-to-End Autonomous Driving with Adaptive Reasoning and Reinforcement Fine-Tuning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 27920–27956, 2025. doi: 10.52202/085713-0942. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/2843fccca5bedd369a4764848b9bd546-Abstract-Conference.html.

APPENDIX GUIDE

The appendix groups implementation details, estimator diagnostics, and supporting experiments by the question they address. Figures, tables, and equations continue the numbering of the main text.

Section	Contents
A	Training, observations, prompts, rewards, and online algorithm
B	Alternatives to the temporal baseline, baseline drift, and signal diagnostics
C	Adapted baseline objectives and comparison protocols
D	Gating, reward noise, token credit, and adaptation lag
E	Sparse controls, additional cities, mixed-data training, and window size
F	Synchronized timing protocol and phase breakdown
G	Selected qualitative examples

A IMPLEMENTATION AND EVALUATION PROTOCOL

A.1 SCENE REPRESENTATION AND SUPERVISED INITIALIZATION

The feature builder follows PlanTF (Cheng et al., 2024b). It uses 2.0 seconds of ego motion history and the same history for up to 20 nearby vehicles or vulnerable road users. Positions, headings, and velocities are represented in an ego-centric frame at the current timestep. The planner input serializes these features with map information, route context, traffic-light status, and driving instructions. The output follows InstructChain (Zhang et al., 2024): reasoning precedes a trajectory field containing 16 poses over 8 seconds, sampled at 0.5-second intervals. Each pose contains relative position and heading. Table 5 summarizes this interface; Appendix A.4 reproduces the instruction and serialized examples, followed by a complete generated response in Appendix A.5.

Table 5: Model-facing input and output schema. The model predicts waypoints; the simulator/controller advances the vehicle between replanning steps.

Field	Contents
Instruction	Driving task, coordinate convention, safety and traffic-rule objectives, and output format.
Perception	Nearby-agent type, shape, pose, velocity, and predictive context.
Ego-States	Ego shape, pose, velocity, acceleration, and historical trajectory.
Map Insights	Traffic lights, speed limit, lane centerline, boundaries, and route context.
Thoughts	Generated assessment of intended motion, nearby objects, and traffic conditions.
Trajectory	Structured future poses parsed into the controller's trajectory representation.

The default Llama-2-7B SFT checkpoint uses 0.3M Las Vegas training examples and next-token cross-entropy, with each example pairing a serialized scene and instruction with the future ego trajectory. The context window is 4,096 tokens; overlength examples are left-truncated. During SFT, LoRA with rank 16, scaling 32, and dropout 0.05 is attached to attention query, key, value, and output projections, as well as the gate, up, and down feed-forward projections. Training runs for one epoch on two NVIDIA H100 GPUs with batch size 16 per device and one accumulation step, giving an effective batch size of 32. The learning rate is 2×10^{-5}, weight decay is 0.01, warmup is disabled, and gradients are clipped at 1.0. This configuration gives approximately 9.4k optimizer steps. The SFT adapter is merged into the backbone before evaluation. Online adaptation creates a separate adapter on the frozen merged checkpoint. The Qwen2.5-7B planner follows the same SFT and online protocols, and Table 6 lists the online adaptation settings.

Table 6: Online adaptation settings shared by the compared methods unless their objective specifies otherwise.

Setting	Value
Online trainable modules	LoRA in `q_proj`, `v_proj`
LoRA rank / scaling / dropout	16 / 32 / 0.05
Optimizer	AdamW
Learning rate / weight decay	10^{-4} / 0.01
Gradient-norm clipping	1.0
TeTO window size	$N = 3$, non-overlapping
Response-ratio clipping	$\epsilon_c = 0.2$
Reference KL coefficient	$\beta = 0.02$
RT threshold	$\eta = 0.70$
Decoding temperature / top-p	0.75 / 0.9
Scenario reset	Restore initial online adapter
Evaluation hardware	One NVIDIA A6000, 48GB

A.2 Reward components

Equation (1) combines components based on the reward design of Plan-R1 (Tang et al., 2026). TeTO uses weights $(2, 5, 4, 2)$ for comfort, TTC, speed, and progress; they are fixed across target cities and were not tuned on them. Let $H = 8\,\mathrm{s}$ denote the plan horizon and Q the number of evaluated trajectory states.

Comfort, weight 2. The score is the proportion of evaluated states within the comfort region:

$$r_{\mathrm{comfort},i} = \frac{1}{Q}\sum_{k=1}^{Q} \mathbb{I}[z_{i,k} \in \mathcal{Z}_{\mathrm{comfort}}], \tag{9}$$

where $z_{i,k}$ is the k-th evaluated state of plan τ_i. The region $\mathcal{Z}_{\mathrm{comfort}}$ uses longitudinal acceleration in $[-4.05, 2.40]\,\mathrm{m/s^2}$, lateral acceleration in $[-4.89, 4.89]\,\mathrm{m/s^2}$, yaw acceleration in $[-1.93, 1.93]\,\mathrm{rad/s^2}$, and yaw rate in $[-0.95, 0.95]\,\mathrm{rad/s}$. Averaging across evaluated states provides graded feedback on trajectory smoothness.

TTC safety, weight 5. The score is zero if a nearby object yields predicted TTC at or below 0.95 seconds and one otherwise. The component provides a direct signal when the current plan approaches the evaluator's critical TTC boundary.

Speed adherence, weight 4. The score penalizes integrated overspeed using a 2.23 m/s tolerance:

$$r_{\mathrm{speed},i} = 1 - \frac{\int_0^H \max(0, v_i(u) - v_{\mathrm{limit}}(u))\,du}{H\,(2.23\,\mathrm{m/s})}. \tag{10}$$

Because this score can be negative, Equation (1) clips the weighted aggregate to $[0, 1]$.

Route progress, weight 2. With d_i the evaluated forward distance along the route,

$$r_{\mathrm{progress},i} = \min\left(\frac{d_i}{H\,(2.0\,\mathrm{m/s})}, 1\right). \tag{11}$$

The four components use the current simulator state and trajectory evaluator. At each replanning step, the evaluator scores the selected 8-second plan, and the controller advances the ego vehicle along the single closed-loop timeline.

The adaptation reward shares kinematic and safety criteria with the nuPlan score, as expected for signals that encode core driving objectives, but it is not identical to that score: collision, drivable-area, driving-direction, and other rule-compliance indicators enter the closed-loop score but are not optimized as reward components. TeTO never optimizes the aggregate score or uses future

Table 7: Information used by the four reward components and the corresponding vehicle-side sources. Our experiments obtain these quantities through the nuPlan simulator and trajectory evaluator.

Signal	Potential runtime source	Main error source
Comfort	Ego motion / IMU	Sensing and filtering
TTC	Detection, tracking, prediction	Perception and prediction
Speed adherence	Ego speed and map limit	Map and localization
Route progress	Localization and route map	Route and map errors

Algorithm 1 TeTO with optional Reward Truncation

Require: Frozen SFT reference π_{ref}; new online LoRA adapter; $N = 3$; optional RT gate

Restore initial online adapter; $\mathcal{W} \leftarrow \varnothing$
while scenario continues **do**
  Observe o_i; sample full response $y_i \sim \pi_\theta(\cdot \mid o_i)$
  Parse plan τ_i and advance the closed-loop interaction
  Obtain component feedback $\{r_{c,i}\}$ and reward R_i
  Append $(o_i, y_i, R_i, \{r_{c,i}\})$ to $\mathcal{W}$
  if $|\mathcal{W}| = N$ **then**
    Compute $b_t, \sigma_t, \{A_i\}$ using Equation (4)
    Compute g_t using Equation (8) if RT is enabled; otherwise $g_t \leftarrow 1$
    if $g_t = 1$ **then**
      Disable online adapter; evaluate reference response probabilities
      Re-enable adapter; compute mean-log-probability ratios (Equation (6))
      Compute clipped surrogate and full-response KL (Equation (7))
      Backpropagate, clip gradients, and take one AdamW adapter step
    end if
    $\mathcal{W} \leftarrow \varnothing$
  end if
end while

labels; each update uses only the feedback computed for the executed plans in its window. The results therefore measure test-time optimization from available task feedback under domain shift. Interpretable rule-based rewards of this kind are standard in driving alignment (Tang et al., 2026; Yuan et al., 2026; Zhou et al., 2025) and require neither human annotation nor a learned reward model; Table 7 lists their vehicle-side sources. The four-term reward is the same for all target cities and all compared methods.

A.3 Closed-loop execution and update algorithm

Scenarios contain approximately 150 frames at 10-Hz simulation time, spanning 15 seconds. NR and R evaluation use the corresponding nuPlan traffic protocols and its scoring aggregator, with the default ego dynamics and controller configuration shared by all methods. Nucleus sampling uses temperature 0.75 and top-$p = 0.9$ for both the frozen SFT planner and all adaptation methods. The planner generates a fresh plan from the latest observation at each control step. Appendix F reports the measured wall-clock computation separately from this simulation cadence.

Algorithm 1 holds the policy fixed during window collection and performs at most one adapter update per completed window. Clearing the buffer after a skipped window keeps every interaction in exactly one group, and an incomplete final window triggers no update.

A.4 Planner prompt and response format

The planner uses the Alpaca-style instruction wrapper below with InstructChain's Perception–Ego-States–Map Insights input and Thoughts–Trajectory response format (Zhang et al., 2024). The system instruction specifies the coordinate convention and the 16-pose, 8-second prediction horizon. Ellipses mark entries omitted for brevity. A complete response and its scene appear in Appendix A.5.

Alpaca-style wrapper

```
Below is an instruction that describes a task, paired with an input that
provides further context.  Write a response that appropriately completes
the request.
### Instruction:
{System Prompt:  Role, Coordinates, Objectives, etc.}
### Input:
{Input Data:  Perception, Ego-States, Map Insights}
### Response:
{Model Output:  Thoughts, Trajectory}
```

System instruction

```
**Autonomous Driving Planner**
Role:  You are the brain of an autonomous vehicle.  Plan a safe driving
trajectory.
Coordinate System:  Forward(+Y), Right(+X), Yaw(Counter-clockwise from
+X).
Objective:
- Identify critical objects and their effects.
- Produce a safe 8.0s trajectory (16 poses at 0.5s resolution).
- Avoid collisions (overlapping quadrilaterals).
```

Input example: perception and ego state

```
Perception:
- vehicle:
- Shape:  [[0.06, -5.62], [5.19, -5.84], [5.11, -7.86], [-0.03, -7.64]]
- Pose:  [2.58, -6.74, 6.24]
- Velocity:  [0.0, 0.0]
- Prediction:  [2.58, -6.74]
- pedestrian:
- Shape:  [[9.2, 0.67], [9.07, -0.23], [8.31, -0.12], [8.44, 0.78]
...  (omitted for brevity) ...
Ego-States:
- Shape:  [[2.59, -1.15], [2.59, 1.15], [-2.59, 1.15], [-2.59, -1.15]]
- Pose:  [0.0, 0.0, 0.0]
- Velocity:  [0.0, -0.0]
- Acceleration:  [0.0, -0.01]
Historical Trajectory:
[[0.0, 0.0, 0.0], [0.0, 0.0, 0.0], [0.0, 0.0, -0.0], [-0.0, 0.0, 0.0]]
```

Input example: map context

```
Map Insights:
- Traffic Light Status:  RED
- Speed Limit:  4.47
- Current Lane Centerline:  [[-5.88, 0.1], [-5.25, 0.1], [-2.72, 0.11],
[-0.19, 0.13], [2.33, 0.14], [4.86, 0.16], [6.13, 0.16]]
- Nearby Lane Boundaries:
[[-5.92, 1.92], [-5.29, 1.93], [-2.75, 1.97], [-0.22, 1.99], [2.32, 1.99],
[4.85, 1.99], [6.12, 1.99]]
[[-5.86, -1.73], [-5.23, -1.72], [-2.7, -1.71], [-0.17, -1.69], [2.36,
-1.68], [4.89, -1.66], [6.16, -1.66]]
```

Output example

```
<thoughts>
```

```
1. Preliminary Action: Follow lane centerline; proceed straight.
2. Collision Considerations: No hazards detected within 1.0s.
3. Transportation Considerations: Traffic Light RED to Stop.
4. Final Action: Remain stationary.
</thoughts>
<trajectory>
[0.0], [0.02, 0.0, -0.0], [0.11, 0.0, -0.0], [0.24, 0.01, -0.0],
... (omitted for brevity) ...
</trajectory>
```

A.5 Complete response and corresponding scene

Figure 6 pairs a generated response with its planned trajectory in a Singapore scenario. The response includes the current lane geometry, object-level collision assessments, traffic-light and speed-limit considerations, and all 16 trajectory poses; its full text follows the figure.

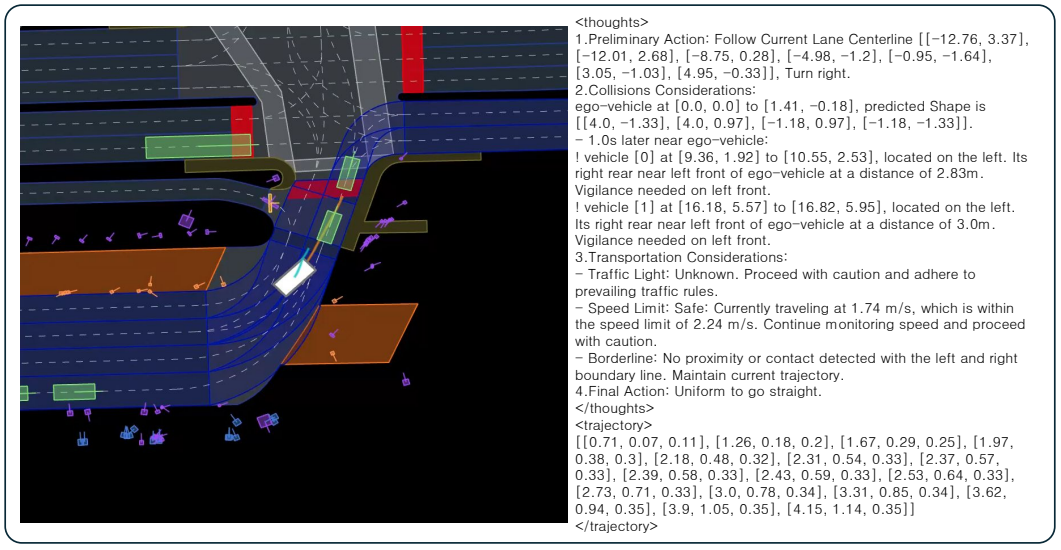

Figure 6: Output of TeTO + RT at one closed-loop timestep in Singapore: the scene with the predicted 8-second trajectory (left) and the generated response (right).

Complete generated response

```
<thoughts>
1.Preliminary Action: Follow Current Lane Centerline [[-12.76, 3.37],
[-12.01, 2.68], [-8.75, 0.28], [-4.98, -1.2], [-0.95, -1.64], [3.05,
-1.03], [4.95, -0.33]], Turn right.
2.Collisions Considerations:
ego-vehicle at [0.0, 0.0] to [1.41, -0.18], predicted Shape is
[[4.0, -1.33], [4.0, 0.97], [-1.18, 0.97], [-1.18, -1.33]].
- 1.0s later near ego-vehicle:
! vehicle [0] at [9.36, 1.92] to [10.55, 2.53], located on the left.
Its right rear near left front of ego-vehicle at a distance of 2.83m.
Vigilance needed on left front.
! vehicle [1] at [16.18, 5.57] to [16.82, 5.95], located on the left.
Its right rear near left front of ego-vehicle at a distance of 3.0m.
Vigilance needed on left front.
3.Transportation Considerations:
- Traffic Light: Unknown. Proceed with caution and adhere to prevailing
traffic rules.
```

```
- Speed Limit: Safe: Currently traveling at 1.74 m/s, which is within
the speed limit of 2.24 m/s. Continue monitoring speed and proceed with
caution.
- Borderline: No proximity or contact detected with the left and right
boundary line. Maintain current trajectory.
4.Final Action: Uniform to go straight.
</thoughts>
<trajectory>
[[0.71, 0.07, 0.11], [1.26, 0.18, 0.2], [1.67, 0.29, 0.25], [1.97, 0.38,
0.3], [2.18, 0.48, 0.32], [2.31, 0.54, 0.33], [2.37, 0.57, 0.33], [2.39,
0.58, 0.33], [2.43, 0.59, 0.33], [2.53, 0.64, 0.33], [2.73, 0.71, 0.33],
[3.0, 0.78, 0.34], [3.31, 0.85, 0.34], [3.62, 0.94, 0.35], [3.9, 1.05,
0.35], [4.15, 1.14, 0.35]]
</trajectory>
```

B Sequential Grouping analysis and diagnostics

B.1 Alternatives to the temporal baseline

Learned critic. Actor-critic methods such as PPO (Schulman et al., 2017) reduce gradient variance with a value function V_ϕ trained with the policy on the training distribution. In an unseen domain, target observations can fall outside the support on which V_ϕ was fit, so its advantages are least reliable exactly where adaptation is needed. Re-fitting the critic online is data-starved: a scenario provides about 150 interactions, and the critic and the policy would each learn from targets produced by the other on a non-stationary stream.

World model. A learned world model (Ha & Schmidhuber, 2018; Hu et al., 2023) could simulate the outcomes of alternative plans from the current state and supply a same-state baseline without execution. Its predictions, however, are subject to the same domain shift as the planner, and prediction errors bias the resulting advantages.

Same-state groups. GRPO normalizes rewards over G responses to one input (Equation (2)). A group of executed plans from the same state requires restoring the joint state of the ego vehicle and all traffic participants after each alternative. A physical deployment cannot do this, and a simulator that can multiplies the interaction cost of every decision by G. Because our evaluator scores a plan against the current state, one can instead sample G candidate plans at each step and score them without executing them, as the diagnostics in Appendix B.3–B.4 do. This multiplies autoregressive decoding, the dominant planner cost (Appendix F), by G, and it scores the unexecuted candidates only through predicted agent motion. Off-policy or replay-based variants relax the same-state requirement but need importance weights, which are hard to stabilize on a short, non-stationary stream.

These requirements are why PPO and vanilla GRPO are not included as online baselines: their native forms add an online critic or same-state groups on top of the objective under comparison. The adapted baselines in Appendix C share TeTO's single-timeline interface instead.

B.2 Local baseline calibration

For a realized window and its fixed policy snapshot, define m_i and e_i as in Section 3.2, with $\bar{m}_t = N^{-1}\sum_j m_j$. The identity

$$m_i - b_t = (m_i - \bar{m}_t) - \bar{e}_t, \qquad \bar{e}_t = \frac{1}{N}\sum_j e_j \tag{12}$$

separates changes in expected reward from sample variation. Applying the triangle inequality yields Equation (5); the mean-drift term satisfies

$$|m_i - \bar{m}_t| = \left|\frac{1}{N}\sum_j (m_i - m_j)\right| \le \frac{1}{N}\sum_j |m_i - m_j|. \tag{13}$$

With a local Lipschitz constant L and state metric d, this becomes $|m_i - \bar{m}_t| \le LN^{-1}\sum_j d(s_i, s_j)$. It describes the condition under which nearby contexts provide a useful local baseline. Discrete traffic events and thresholded TTC feedback can produce larger changes even over a short window, motivating the transition diagnostics below.

The sampling term has variance

$$\mathrm{Var}(\bar{e}_t) = \frac{1}{N^2}\sum_j \mathrm{Var}(e_j) + \frac{2}{N^2}\sum_{j<k} \mathrm{Cov}(e_j, e_k). \tag{14}$$

The covariance terms account for serial dependence along a closed-loop episode.

The scale estimate mixes action-level variation with changes in context. To illustrate the latter, consider a mixture that chooses a window index J uniformly and samples a reward from the corresponding state-conditioned distribution. The law of total variance gives

$$\mathrm{Var}(R) = \mathbb{E}_J[\mathrm{Var}(R \mid J)] + \mathrm{Var}_J(m_J). \tag{15}$$

The second term measures the variation introduced by different state-conditioned means; it grows as a window spans more diverse contexts, which is the behavior of the sequential curve in Figure 3 beyond $n = 10$. The bound above concerns the centering term. The effects of normalization, reference-based clipping, and gating on the full update are evaluated empirically in Appendix B.4, Appendix B.5, and Appendix D.

B.3 Fixed-observation reward variance

Figure 3 in the main text compares scenario-averaged reward variance across group sizes. For each evaluated scenario, the frozen SFT planner generates 20 candidate 8-second plans from the identical initial observation using temperature 0.75 and top-$p = 0.9$. All candidates share ego, map, and surrounding-agent context, with agent motion propagated from that context for scoring, and the simulator is not advanced during candidate generation. The first n candidates form the parallel group of size n, so this diagnostic measures open-loop, plan-level variance at a fixed observation. The sequential group of size n consists of n consecutive executed plans.

At $n = 3$, the mean parallel variance is 0.00853, with a 95% hierarchical-bootstrap interval of $[0.00446, 0.01517]$, and the sequential variance coincides with it. Up to $n = 10$ the sequential variance stays at or below the parallel variance; beyond that, longer windows span more diverse contexts and the sequential variance grows.

B.4 Logged-plan cross-evaluation

The second diagnostic uses 491 completed $N = 3$ windows. Let M_{ii} be the selected plan's score, b_t the sequential baseline of Equation (4), and b_i^{PG} the same-state plan-scoring reference. Their centered signals satisfy

$$M_{ii} - b_t = (M_{ii} - b_i^{\mathrm{PG}}) + (b_i^{\mathrm{PG}} - b_t). \tag{16}$$

The first term represents the selected plan's relative quality under the evaluator; the second captures the difference between temporal and same-state contexts. We measure the absolute baseline gap and agreement between the signs of the centered signals.

Table 8: Logged-window signal diagnostics. Top: all 491 complete windows. Bottom: the separately analyzed RT-triggered subset. Each subset percentage uses its corresponding analysis population.

Subset	Windows (%)	Mean absolute baseline gap	Sign agreement (%)
No TTC transition	95.9	0.0036	92.3
TTC transition	4.1	0.1607	62.3
RT-triggered, all	100.0	–	89.5
RT, no TTC transition	94.7	–	92.7
RT, TTC transition	5.3	–	61.7

Windows with an unchanged TTC component account for 95.9% of the sample (Table 8) and have high sign agreement (92.3%) and a small baseline gap (0.0036). A TTC transition is a change in the binary component within the window. Such windows have a gap of 0.1607 and 62.3% sign agreement, showing the effect of abrupt reward changes on the temporal comparison.

B.5 Diagnostics on windows selected by Reward Truncation

Table 9: RT-triggered window diagnostics. Each component's pair of rows partitions the RT population; partitions of different components overlap. Context share is the fraction of centered-signal variance explained by the context term of Equation (16).

Subset	Windows (%)	Sign agreement (%)	Context share (%)
All RT-triggered windows	100.0	89.5	15.8
No TTC transition	94.7	92.7	9.3
TTC transition	5.3	61.7	91.4
No comfort transition	49.0	82.1	17.5
Comfort transition	51.0	91.3	14.3
No speed transition	88.2	89.5	14.3
Speed transition	11.8	92.4	22.1
No progress transition	52.0	82.2	27.7
Progress transition	48.0	95.6	7.2

On the windows where RT issues updates, overall sign agreement is 89.5% (Table 9): 94.7% of these windows have an unchanged TTC component and reach 92.7% agreement, and TTC transitions account for only 5.3%.

For continuous components $c \in \{\text{comfort}, \text{speed}, \text{progress}\}$, a material transition is defined by

$$\mathbb{I}\left[\max_{j=1,2} |r_{c,j+1} - r_{c,j}| > \delta\right], \qquad \delta = 0.10. \tag{17}$$

Such transitions are common among RT-triggered windows, for example during intersection maneuvers or lane changes, yet agreement remains high: the comfort- and progress-transition subsets reach sign agreement of 91.3% and 95.6%. Table 10 shows how the prevalence of transitions depends on the threshold.

Table 10: Material-transition frequency among RT-triggered windows for different continuous-reward jump thresholds. Values are percentages.

Jump threshold δ	Comfort	Speed	Progress
0.05	68.1	18.1	58.2
0.10	51.0	11.8	48.0
0.20	34.9	7.9	36.5

In the all-window study, context shares are 10.3% for the constant-TTC subset and 85.4% for TTC-transition windows. Because the two terms of Equation (16) can covary, we use baseline gap and sign agreement as the primary measures of signal consistency.

C Adapted baseline objectives

C.1 Shared single-timeline interface

The baselines operate on the same sampled response y_i, parsed plan, scalar reward R_i, online LoRA configuration, and frozen reference as TeTO; Appendix B.1 explains why PPO and vanilla GRPO are not included. Let $\bar{\ell}_i = \bar{\ell}_\theta(y_i \mid o_i)$ denote the mean log-probability over all generated positions, and let $\bar{\ell}_i^{\text{ref}} = \bar{\ell}_{\text{ref}}(y_i \mid o_i)$ be its reference counterpart. Dense single-sample losses use the newest

interaction; dense grouped losses use a sliding buffer of the latest three interactions. The grouped buffer moves by one step after each interaction and may contain samples collected under different policy parameters. These are cross-state online adaptations of the original objectives.

TeTO clears its buffer after each complete window. Sparse controls update once per completed non-overlapping three-step window, and sparse RWR averages its loss over all three responses. All online adapters are reset at scenario boundaries. We report average update counts alongside scores and measure adaptation time separately in Appendix F.

C.2 Policy-gradient and reward-weighted controls

REINFORCE. The implemented single-sample loss is

$$\mathcal{L}_{\text{REINFORCE}} = -(R_t - b_t^{\text{base}})\bar{\ell}_t, \tag{18}$$

where b_t^{base} is the empirical reward baseline. This control uses the score-function form of Williams (1992).

REINFORCE++ adaptation. For the current sliding window $\mathcal{B}_t$, center rewards by its mean and normalize them using the standard deviation of a streaming buffer of centered rewards:

$$\tilde{R}_i = R_i - \frac{1}{N}\sum_{k\in\mathcal{B}_t} R_k, \qquad A_i^{++} = \frac{\tilde{R}_i}{\max(\sigma_{\text{global}}, \epsilon)}. \tag{19}$$

With $\Delta_{i,j} = \log \pi_\theta(y_{i,j} \mid o_i, y_{i,<j}) - \log \pi_{\text{ref}}(y_{i,j} \mid o_i, y_{i,<j})$, the loss is

$$\mathcal{L}_{++} = \frac{1}{N}\sum_{i\in\mathcal{B}_t}\left[-A_i^{++}\bar{\ell}_i + \frac{0.02}{2|\mathcal{G}_i|}\sum_{j\in\mathcal{G}_i}\Delta_{i,j}^2\right]. \tag{20}$$

The reference penalty is a squared log-probability difference. We adapt the global-normalization principle of Hu et al. (2025) to sequential groups and streaming reward statistics.

RLOO adaptation. The leave-one-out baseline and normalized reward signal are

$$b_i^{\text{LOO}} = \frac{1}{N-1}\sum_{k\in\mathcal{B}_t,\, k\neq i} R_k, \qquad A_i^{\text{LOO}} = \frac{R_i - b_i^{\text{LOO}}}{\max(\text{std}_{k\in\mathcal{B}_t}(R_k - b_k^{\text{LOO}}), \epsilon)}. \tag{21}$$

We use $\mathcal{L}_{\text{LOO}} = -N^{-1}\sum_{i\in\mathcal{B}_t} A_i^{\text{LOO}}\bar{\ell}_i$. The leave-one-out comparison of Ahmadian et al. (2024) is applied here to responses from successive observations.

Reward-weighted regression. Dense RWR uses $\mathcal{L}_{\text{RWR}} = -\exp(R_t/\lambda)\bar{\ell}_t$ with temperature $\lambda = 1$. Its sparse variant uses

$$\mathcal{L}_{\text{sparse RWR}} = -\frac{1}{N}\sum_{i\in\mathcal{W}_t}\exp(R_i/\lambda)\bar{\ell}_i. \tag{22}$$

RWR assigns every response a positive reward-dependent imitation weight (Peters & Schaal, 2007; Štrupl et al., 2022), so even a response that scores below its local context is reinforced, and the gradient depends on the local reward offset and scale. SG centers and scales the rewards within a window, reinforcing responses above the window mean and suppressing those below it.

C.3 Preference and entropy controls

DPO adaptation. The highest- and lowest-reward interactions in the sliding window form the positive and negative pair, respectively, each retaining its own observation. With $h_i = \bar{\ell}_i - \bar{\ell}_i^{\text{ref}}$, the loss is

$$\begin{aligned}\mathcal{L}_{\text{DPO}} &= -\log \text{sigmoid}(\beta_{\text{DPO}}(h_{i^+} - h_{i^-}))\,,\\ i^+ = \arg\max_i R_i, &\quad i^- = \arg\min_i R_i, \quad \beta_{\text{DPO}} = 1.\end{aligned} \tag{23}$$

Equal-reward pairs are skipped. This adaptation of Rafailov et al. (2023) forms preference pairs from the executed timeline. Each response retains its own observation, so the reward ordering reflects both action quality and local context.

TENT adaptation. We retain the entropy-minimization principle of Wang et al. (2021), applying it to the generated token distributions of the newest response:

$$\mathcal{L}_{\text{TENT}} = \frac{1}{|\mathcal{G}_t|} \sum_{j \in \mathcal{G}_t} \mathcal{H}(\pi_\theta(\cdot \mid o_t, y_{t,<j})) \,, \tag{24}$$

where $\mathcal{H}$ denotes the entropy of the next-token distribution. The entropy objective updates the online LoRA parameters and provides a reward-free control for the language-model planner.

C.4 Objective-matched running mean

The running-mean control in Table 4 retains TeTO's non-overlapping three-sample windows, response-ratio clipping, KL term, optimizer, and 49.1 average updates per scenario. It uses a running reward mean for the baseline. In the OOD-R comparison, local SG centering improves the score from 76.77 to 77.92 under the shared objective and update schedule.

D Additional ablations

Unless otherwise stated, the following experiments use the Llama-2-7B planner on Test14-Hard Singapore reactive scenarios.

D.1 Gate threshold and reward noise

Table 11: Sensitivity to the RT threshold η. Best per row in bold, second best underlined; fewer updates is better.

Threshold η	0.50	0.60	0.70	0.80	0.90
R-CLS	<u>77.80</u>	75.59	**77.96**	77.30	75.20
Updates/scenario	**10.6**	<u>11.7</u>	28.3	32.2	33.5

With $\eta = 0.70$, TeTO + RT reaches 77.96 R-CLS with 28.3 updates (Table 11). The stricter $\eta = 0.50$ retains nearly the same score (77.80) with only 10.6 updates, whereas $\eta = 0.80$ and 0.90 add updates without improving the score.

Table 12: Gaussian corruption of the TTC reward channel with TeTO + RT. The perturbation is applied to the feedback signal. Best in bold, second best underlined.

Noise standard deviation	0.0	0.1	0.2	0.3
R-CLS	<u>77.96</u>	**80.12**	77.02	75.63

Even severe noise on the TTC channel (standard deviation 0.3; Table 12) lowers R-CLS by only 2.33 points, to 75.63, consistent with the multi-component reward limiting the influence of a single corrupted channel; mild noise (0.1) does not reduce the score.

D.2 Minimum, mean, and instantaneous gates

At nearly equal clean update counts (28.3 and 29.3), the component-minimum gate scores 0.98 points above the window-mean gate, and it stays 0.44 points above it under noise (Table 13). A mean can hide one degraded component behind saturated ones, whereas the minimum exposes it; under noise, the minimum gate issues somewhat more updates (34.2 versus 31.2) and keeps the higher score. Both window gates outperform the ungated variant under noise.

The window gate outperforms the instantaneous gate in all four settings (Table 14). It matches the granularity of the update itself and retains low component values anywhere in the window.

Table 13: Gate comparison with approximately matched clean update counts and fixed thresholds under TTC noise with standard deviation 0.3. Counts are average updates per scenario. Best in bold, second best underlined; fewer updates is better.

Gate	Clean R-CLS	Noisy R-CLS	Clean updates	Noisy updates
No gate	<u>77.92</u>	74.42	49.1	49.1
Window mean	76.98	<u>75.19</u>	<u>29.3</u>	**31.2**
Component minima	**77.96**	**75.63**	**28.3**	<u>34.2</u>

Table 14: Thresholding the current scalar reward (instantaneous gate) versus the weighted component-wise window minima (RT). Best in bold.

	Las Vegas		Singapore	
Gate	NR	R	NR	R
Instantaneous reward	79.18	77.83	68.10	74.72
Component minima	**80.36**	**82.88**	**72.47**	**77.96**

D.3 Which response tokens receive policy credit?

There is one autoregressive policy for both reasoning and waypoints. If $y = (y^{\text{reason}}, y^{\text{traj}})$, then

$$\log \pi(y \mid o) = \log \pi(y^{\text{reason}} \mid o) + \log \pi(y^{\text{traj}} \mid o, y^{\text{reason}}). \tag{25}$$

The reasoning prefix thus affects the conditional waypoint distribution. Masking it out of the policy loss holds the sampled prefix fixed for that update. Our full-response surrogate length-normalizes the selected token log-probabilities as in Equation (6).

Table 15: Policy-loss token ablation. Both variants retain full-response KL and average over their selected policy-loss positions. Best in bold.

Policy-loss positions	R-CLS	Mean KL to SFT
Trajectory only	10.04	0.0330
Full response	**77.92**	**0.0234**

Full-response credit achieves 77.92 with mean KL 0.0234 (Table 15). The trajectory-only run collapses to 10.04 with a larger mean KL of 0.0330, mainly through zero drivable-area compliance. Crediting the full response, which matches the joint reasoning–trajectory policy, therefore stabilizes adaptation.

D.4 Window completion and adaptation lag

The window delays only parameter updates: the planner receives the current observation and generates a new plan at every simulation step. For $N = 3$ and a 0.1-second step, the earliest member of a window waits at most 0.3 seconds of simulation time for its update, and no observation-to-action latency is added. Wall-clock inference and adaptation costs are reported in Appendix F.

Table 16: A per-step EMA control and the non-overlapping SG schedule on Singapore reactive scenarios. Best in bold.

Schedule	N	Update rate (%)	R-CLS	TTC violations (%)
Per-step EMA	1	100.0	74.13	26.30
Non-overlapping SG	3	**32.9**	**77.92**	**25.00**

With about one third of the updates, the three-step SG schedule (Table 16) achieves 77.92 with a 25.00% planned-TTC violation rate, compared with 74.13 and 26.30% for a per-step control with

an exponential-moving-average (EMA) baseline. The window therefore delays parameter updates without increasing planned-TTC violations.

E EXTENDED EVALUATION

E.1 SPARSE-UPDATE CONTROLS

Table 17: Reactive Test14-Hard scores for dense and sparse variants of the adapted baselines. Dense variants update each step; sparse variants update every three steps. The TeTO rows use their own window/gating schedules and are shown once per city. Best in bold, second best underlined.

	Las Vegas (ID)		Singapore (OOD)	
Method	Dense	Sparse	Dense	Sparse
REINFORCE	80.31	78.82	73.15	76.25
REINFORCE++	78.11	77.74	75.55	76.28
RLOO	78.58	78.59	75.66	75.03
RWR	78.46	79.57	76.06	76.87
DPO	79.65	78.75	75.94	75.63
TENT	80.91	76.28	75.07	75.59
TeTO	<u>81.59</u>		<u>77.92</u>	
TeTO + RT	**82.88**		**77.96**	

The sparse baselines reach OOD-R scores between 75.03 and 76.87, while TeTO and TeTO + RT achieve 77.92 and 77.96 (Table 17). Sparse updating helps some objectives and hurts others, but none of them reaches TeTO, so its advantage persists at a matched update frequency. The matched-objective control in Table 4 isolates the local baseline under a shared loss and update schedule.

E.2 FOUR-CITY TEST14-RANDOM EVALUATION

Table 18: Reactive CLS on Test14-Random. Las Vegas is the SFT training city; the other three cities are unseen during training. Best in bold, second best underlined.

Method	Las Vegas	Singapore	Pittsburgh	Boston
SFT	81.30	77.19	77.66	80.57
REINFORCE	80.92	77.48	77.65	79.52
REINFORCE++	<u>84.22</u>	73.57	81.24	80.68
RLOO	82.43	77.17	80.11	81.53
RWR	83.48	76.85	81.27	82.25
DPO	82.53	<u>78.35</u>	73.57	79.27
TENT	82.27	76.17	<u>81.93</u>	<u>83.63</u>
TeTO	**85.06**	78.08	80.74	**86.07**
TeTO + RT	83.77	**78.38**	**82.01**	82.43

Test14-Random is less curated than Test14-Hard and probes whether the trends hold under broader scenario coverage. A TeTO variant achieves the highest score in each city: TeTO in Las Vegas (85.06) and Boston (86.07), and TeTO + RT in Singapore (78.38) and Pittsburgh (82.01).

E.3 MIXED NUPLAN–WAYMO SUPERVISED TRAINING

To examine whether adaptation remains useful with broader offline coverage, an additional backbone is trained on 0.6M examples: 0.3M from nuPlan Las Vegas and 0.3M from the Waymo Open Motion Dataset (WOMD; Ettinger et al., 2021). Agent and local-map features are converted into the same model-facing representation, using an ego-rear-axle-centered, ego-heading-aligned frame and

consistent physical units. Both datasets use the Perception, Ego-States, and Map Insights input fields and the Thoughts–Trajectory output format. Evaluation remains in the reactive nuPlan simulator on Las Vegas and Singapore.

Table 19: Reactive nuPlan evaluation after broader offline training and online adaptation. Values in parentheses are score gains over the corresponding SFT checkpoint. Best in bold, second best underlined.

Training data	Method	Las Vegas R-CLS	Singapore R-CLS
nuPlan, 0.3M	SFT	<u>80.87</u>	72.33
nuPlan, 0.3M	TeTO	**81.59** (+0.72)	**77.92** (+5.59)
nuPlan + WOMD, 0.6M	SFT	78.99	74.05
nuPlan + WOMD, 0.6M	TeTO	80.81 (+1.82)	<u>77.31</u> (+3.26)

The Las Vegas-only SFT protocol of the main experiments isolates city-level domain shift; this experiment (Table 19) tests whether TeTO remains useful when the base planner sees broader offline data. The mixed-data checkpoint raises the OOD SFT score by 1.72 points but lowers the ID score by 1.88 points. TeTO then improves both, adding 1.82 points in Las Vegas and 3.26 points in Singapore. TeTO on the 0.3M nuPlan planner (77.92) also exceeds SFT on the twice larger mixed corpus (74.05), and both adapted planners reach 77–78 in Singapore.

Table 20: Descriptive statistics for the analyzed nuPlan and WOMD subsets.

Metric	nuPlan Vegas	WOMD	nuPlan Singapore
Median ego speed (m/s)	0.94	3.21	3.46
Median final travel distance (m)	13.85	28.95	31.10
Stopping trajectory ratio (%)	38.60	22.90	24.30
Red traffic-light state (%)	35.40	29.90	28.50
Mean detected agents	50.25	29.93	9.43
Mean pedestrians	7.74	1.34	3.33
Traffic convention	Right-hand	Right-hand	Left-hand

Table 20 compares the analyzed subsets. WOMD is much closer to Singapore than Las Vegas is in median ego speed, final travel distance, and stopping-trajectory ratio, consistent with the higher OOD score of the mixed-data SFT planner. The remaining differences are structural: Singapore uses left-hand traffic, which neither training corpus contains, and its scenes are far sparser. Broader offline data thus widens the prior without aligning it to the target domain, which is the gap that test-time adaptation addresses.

E.4 Five-scenario evaluation across cities and window sizes

This evaluation samples one scenario from each of five scenario types per city and reports reactive closed-loop scores. Both TeTO variants outperform SFT and REINFORCE in every city, while TeTO + RT updates on 1.3–17.0% of steps (Table 21). On the Singapore scenarios, Table 22 varies the window size of TeTO; its $N = 3$ column corresponds to the Singapore TeTO entry of Table 21. The aggregate score changes by at most 0.25 points across $N \in \{2, 3, 4\}$ and no scenario type changes by more than 0.66 points, so performance is insensitive to N in this range. We use $N = 3$, where the sequential and parallel reward variances coincide (Figure 3).

Table 21: Five-scenario evaluation per city (reactive CLS). The last column reports the RT update count per scenario and its fraction of the dense count. Best per city in bold, second best underlined.

City	SFT	REINFORCE	TeTO	TeTO + RT	RT updates
Las Vegas	76.33	77.66	**85.16**	82.87	17.4 (11.7%)
Singapore	75.91	80.58	80.89	**83.34**	2.0 (1.3%)
Pittsburgh	68.43	69.76	70.86	**71.84**	25.4 (17.0%)
Boston	68.70	69.27	72.45	**72.67**	12.8 (8.6%)

Table 22: Window size of TeTO on the five Singapore scenarios of Table 21. The first row is the aggregate reactive CLS; the remaining rows are per-type scores. Best per row in bold, second best underlined.

Window size N	2	3	4
Aggregate	**80.96**	80.89	80.71
High magnitude speed	**78.68**	78.19	78.10
On intersection	63.89	**63.90**	**63.90**
Stationary	87.50	87.50	87.50
Stationary at traffic light	**81.54**	81.30	80.88
Traversing intersection	93.18	**93.55**	93.18

F SYNCHRONIZED RUNTIME MEASUREMENTS

We measure GPU completion with `torch.cuda.synchronize()` immediately before and after each timed GPU phase. Measurements use one NVIDIA A6000, the full 6.75B-parameter BF16 backbone computation graph, and 8.39M trainable online LoRA parameters in query and value projections. Gradient checkpointing is disabled. The measurements come from a separate rerun of the Singapore evaluation on Test14-Random (Table 18), in which TeTO + RT issues 7.8 updates per scenario.

Table 23: Adaptation phase costs, amortized over all control steps in the synchronized rerun (ms/step). A dash indicates no RT gate. Lowest per row in bold.

Phase	REINFORCE	TeTO + RT
Loss forward	373.3	**25.7**
Backward	541.6	**37.5**
Gradient clipping	3.04	**0.18**
AdamW	3.76	**0.21**
Gate evaluation	–	0.06
Other update overhead	24.20	**0.95**
Total adaptation	945.9	**64.6**
Updates/scenario	149.2	**7.8**

Table 23 breaks down the adaptation cost by phase; other update overhead includes gradient zeroing, loss scaling, autograd bookkeeping, and related update operations. TeTO + RT reduces the average incremental adaptation cost by 93.2% relative to dense REINFORCE because it updates on only a small fraction of control steps.

Table 24 reports the remaining pipeline components; token-ID detokenization is the CPU conversion of generated IDs into text. Differences in these shared components reflect the independent closed-loop trajectory of each method, and the adaptation savings in Table 23 come from fewer updates. Autoregressive generation with the unchanged 7B backbone exceeds one second per control step in our implementation and dominates the shared inference cost, consistent with prior LLM-based

Table 24: Other measured pipeline components (ms/control step), excluding autoregressive generation. These shared stages are measured along each method's independently generated closed-loop trajectory. Lowest per row in bold, second lowest underlined.

Component	REINFORCE	TeTO	TeTO + RT
Prompt construction	62.36	<u>36.95</u>	**26.74**
Tokenization	4.72	<u>3.82</u>	**3.64**
Token-ID detokenization	0.65	**0.49**	<u>0.50</u>
Parsing	3.98	**3.32**	<u>3.35</u>
Reward computation	20.94	<u>15.71</u>	**14.24**
Simulator execution	19.95	<u>15.06</u>	**13.68**

driving systems that decouple slow language reasoning from real-time control (Sha et al., 2023; Chen et al., 2025). The reported savings concern the incremental cost of online adaptation.

Combining adaptation with faster inference. Efficient serving and speculative decoding reduce the cost of language-model generation (Kwon et al., 2023; Leviathan et al., 2023); quantization and pruning reduce inference resource demands (Xiao et al., 2023; Frantar & Alistarh, 2023). Driving-specific systems also distribute computation across planning components or adjust model execution, including asynchronous LLM planning, DriveVLM, and Prune2Drive (Chen et al., 2025; Tian et al., 2025; Xiong et al., 2026). TeTO is orthogonal to these techniques and can be combined with an accelerated, quantized, or smaller base planner.

G QUALITATIVE EXAMPLES

The following figures extend the main-paper visualizations with additional Singapore road geometries and longer rollouts. In the scenario of Figure 7, SFT departs from its lane, REINFORCE attempts a correction but fails to stabilize, and both TeTO variants recover smooth, lane-consistent trajectories. Figure 8 shows a high-curvature turn, a complex intersection, and a constrained lane with nearby obstacles; again the TeTO variants track the lane geometry, while SFT and REINFORCE depart from the lane or correct insufficiently.

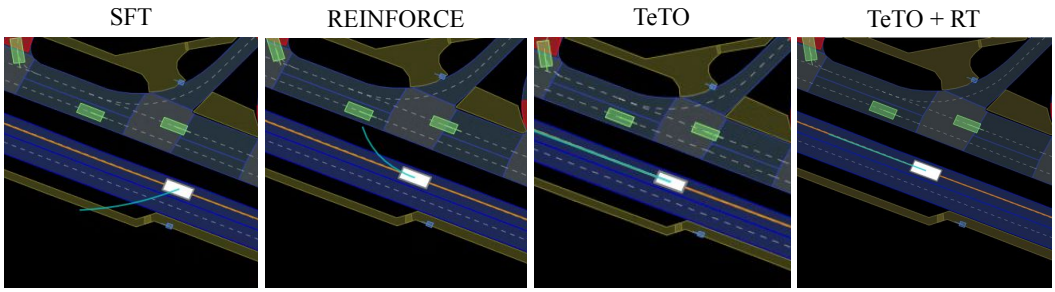

Figure 7: Planning in an unseen domain. From left to right: SFT, REINFORCE, TeTO, and TeTO + RT in a Singapore scenario.

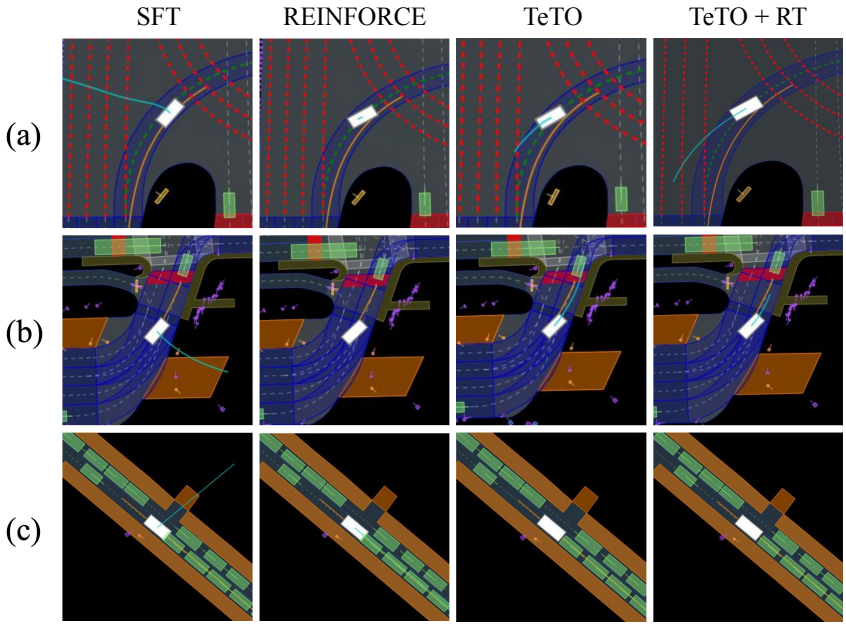

Figure 8: Additional qualitative comparisons across selected road geometries: (a) a high-curvature turn, (b) an intersection, and (c) a constrained lane with nearby obstacles. Columns show SFT, REINFORCE, TeTO, and TeTO + RT.

G.1 TRAJECTORY EVOLUTION OVER A COMPLETE SCENARIO

Figures 9 and 10 visualize the same four planners over two out-of-domain scenarios, from the initial observation to the end of the 150-frame episode. Snapshots at $t = 0, 29, 59, 89, 119, 149$ show how the trajectories evolve as the closed-loop scene progresses. The high-magnitude-speed example highlights lane keeping over sustained motion, while the intersection example highlights the changing drivable corridor through lane connectors.

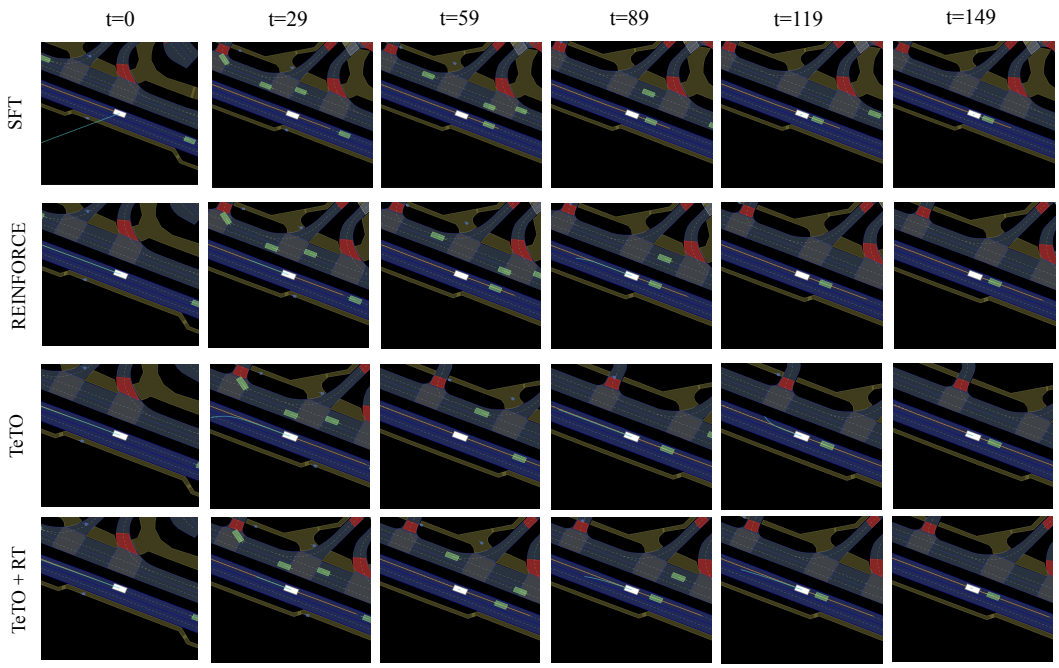

Figure 9: Trajectory evolution in the high-magnitude-speed Singapore scenario. Rows show SFT, REINFORCE, TeTO, and TeTO + RT; columns show successive points along the closed-loop rollout. The TeTO variants maintain continuous lane-following trajectories through the episode.

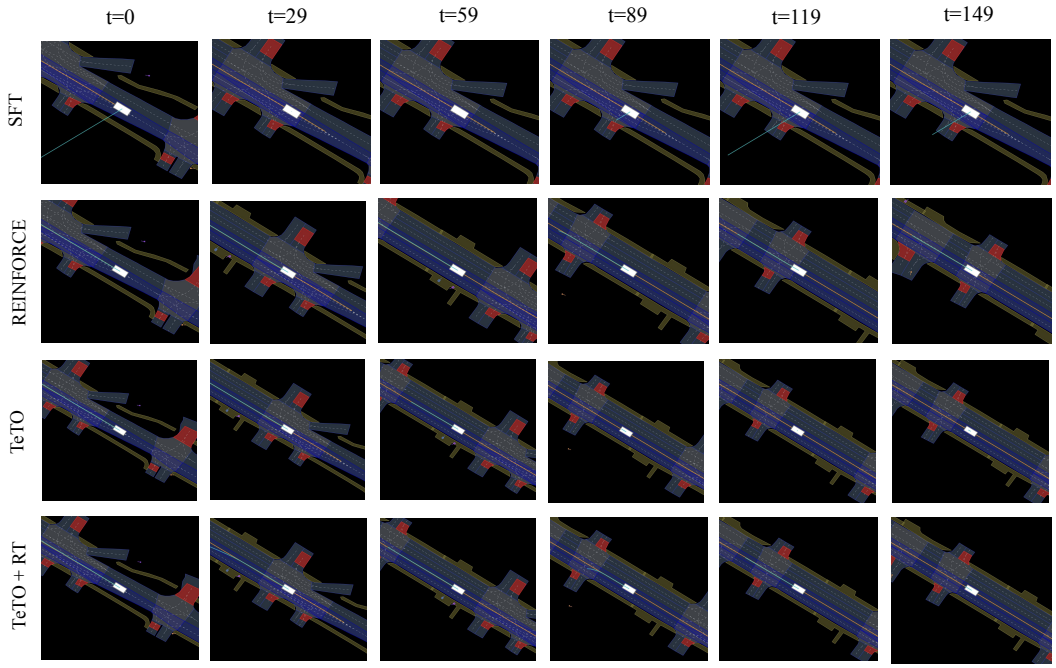

Figure 10: Trajectory evolution in the traversing-intersection Singapore scenario. The TeTO variants follow the drivable corridor across the intersection as the geometry changes over the episode. Planner order and snapshot times match Figure 9.