LEARNING HOW TO DRIVE AT TEST TIME

Anonymous authors
Paper under double-blind review

ABSTRACT

Driving policies are often deployed with verifiers, i.e. safety checks scoring candidate trajectories against drivable area, occupancy, and comfort criteria. We argue that such verifiers are good rewards, and that policies can learn from them at test time. We use verifiers for rejection sampling and optimize driving policies with one gradient step toward accepted trajectories. Our method requires no human demonstrations and no differentiable simulator. It is agnostic to where the samples come from: with a teacher policy it performs online distillation, with the policy's own samples online self-distillation. We show that both regimes support test-time fine-tuning of pre-trained Alpamayo 1.5, AutoVLA, and OneVL policies, improving composite driving scores by up to 24% and lowering displacement error by up to 42%. VADv2, CILRS, and CoverNet policies can even be learned from scratch at test time, given a reasonable initial trajectory distribution. On the KITScenes LongTail and the Waymo Open E2E dataset, our method compensates for domain shifts from nominal to unusual driving scenarios, outperforming all submissions to the 2026 KITScenes Few-Shot Challenge.

1 INTRODUCTION

Offline-trained driving policies fail to adapt to unusual long-tail scenarios. End-to-end driving policies are trained offline on large collections of human demonstrations and then deployed with frozen weights (e.g., Hu et al., 2023; Jiang et al., 2026; Zhou et al., 2025). This works well as long as deployment resembles training. It breaks down in unusual long-tail scenarios, such as unusual road layouts, rare agent behavior, or adverse weather conditions. Recent benchmarks show that even large reasoning-capable driving models degrade substantially in such scenarios (Wagner et al., 2026; Xu et al., 2026). Moreover, collecting and annotating more data scales poorly, since the scenarios that matter most are the ones that are hardest to collect.

Safety verifiers are used at test time as filters, but not to learn online. Driving policies are typically deployed with verifiers, i.e., safety checks that score candidate trajectories against drivable area, occupancy, and comfort criteria (Hu et al., 2023; Dauner et al., 2023; Guo et al., 2026). However, these verifiers are used purely as filters: a rejected trajectory is discarded or replaced by a fallback maneuver, and the policy that proposed it remains unchanged. The policy will therefore make the same mistake the next time it encounters a similar scenario. We argue that this neglects a learning signal that is available at test time. Verifying a trajectory is easier than generating one, and a safety verifier enforces the properties we want a driving policy to have.

Building on this observation, we propose to learn how to drive at test time. For each test scenario, we sample candidate trajectories, keep those that the verifier accepts, and take one gradient step toward the accepted trajectories (see fig. 1). This is rejection sampling-based learning (Dong et al., 2023; Lambert, 2025; Liu et al., 2024) applied online. Unlike policy-gradient and preference optimization methods (Schulman et al., 2017; Rafailov et al., 2023; Shao et al., 2024), it needs neither advantage estimates nor preference pairs, and a binary accept/reject signal suffices. It requires no annotations and no differentiable simulator, and it uses only the observations of the test split, never its ground-truth trajectories. Our method is also agnostic to where the candidate trajectories come from. If they are sampled from a teacher policy, it performs online distillation; if they are sampled from the policy itself, it performs online self-distillation.

We fine-tune pre-trained policies, or learn policies from scratch at test time. As *test-time fine-tuning*, we fine-tune pre-trained policies on the test split to compensate for the domain shift from

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.

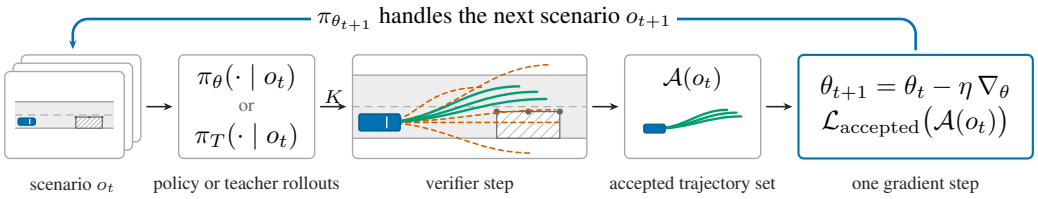


Figure 1: **One step of test-time learning.** For test scenario o_t, we sample K candidate trajectories from the policy π_θ or a teacher π_T, the verifier V marks each as accepted or rejected, and we take one gradient step toward the accepted set $\mathcal{A}(o_t)$ before the next scenario arrives. Policy rollouts yield online self-distillation; teacher rollouts yield online distillation.

nominal to long-tail driving, and improve Alpamayo 1.5 (Wang et al., 2025), AutoVLA (Zhou et al., 2025), and OneVL (Lu et al., 2026) in both the distillation and the self-distillation setting. As *test-time learning*, we go one step further and drop pre-training altogether: given a reasonable initial trajectory distribution generated with simple kinematic heuristics, VADv2 (Jiang et al., 2026), CILRS (Codevilla et al., 2019), and CoverNet (Phan-Minh et al., 2020) policies can be learned from scratch on the test set, supervised by just the safety verifiers.

Related work targets different applications or focuses on uncertainty rather than safety criteria. Test-time adaptation and test-time learning have been studied extensively in image classification (Wang et al., 2021; Niu et al., 2022; Chen et al., 2024) and language modeling (Hardt & Sun, 2024; Hübotter et al., 2025; Hu et al., 2025), where the learning signal is typically prediction entropy, perplexity, or retrieved neighbors. In end-to-end driving, test-time compute has mostly been spent on selecting or refining trajectories with frozen policy weights (Hu et al., 2023; Dauner et al., 2023; Guo et al., 2026). Centaur (Sima et al., 2025) is most related to our work and minimizes an uncertainty measure over the policies' outputs. Uncertainty, however, is a proxy: a policy can become more confident without becoming more correct, which is precisely the failure mode to expect under domain shift. Our learning signal is instead grounded in explicit driving criteria. To the best of our knowledge, we are the first to use verifiers as rewards for test-time learning of driving policies, and the first to show that driving policies can be learned from scratch at test time.

Our main contributions are:

- We propose a rejection-sampling-based method for learning driving policies at test time, which uses safety verifiers to accept and reject trajectories. Our method requires no annotations, no differentiable simulator, and no learned reward model (see Figure 1).
- We show that our method is agnostic to the source of candidate trajectories and supports both online distillation from a teacher and online self-distillation, improving Alpamayo 1.5, AutoVLA, and OneVL under domain shift (see Section 4.3.1 and Section 4.3.2).
- We show that VADv2, CILRS, and CoverNet policies can be learned from scratch at test-time, given an initial trajectory distribution from kinematic heuristics (see Section 4.4).
- On KITScenes LongTail and the Waymo Open E2E dataset, our method compensates for the shift from nominal to unusual driving scenarios and outperforms all submissions to the 2026 KITScenes Few-Shot Challenge (see Table 2).

2 Related Work

End-to-end driving policies map sensor observations and driving commands to driving actions or trajectories, bypassing traditional modular pipelines (e.g., Hu et al., 2023; Codevilla et al., 2019; Jiang et al., 2026). Such policies are typically trained offline on large-scale human driving datasets and deployed with frozen weights. While effective in nominal conditions, they struggle to generalize to unusual, long-tail scenarios such as rare agent behaviors, unusual road layouts, or adverse weather (Xu et al., 2026; Wagner et al., 2026). To improve robustness and interpretability in these edge cases, recent reasoning-capable policies (e.g., Wang et al., 2025; Zhou et al., 2025; Lu et al., 2026) perform text-based reasoning to explain and guide their decision-making. However, Wagner et al. (2026)

show that even these large reasoning models degrade substantially under domain shift. Furthermore, scaling these models by simply collecting and annotating more data is inefficient, as the most critical edge cases are the hardest to record. Our work addresses this limitation by enabling driving policies to adapt online at test time without requiring additional offline data collection or annotations.

Test-time learning and fine-tuning optimizes machine learning models at test time based on incoming test data before making a final prediction. It has been extensively studied in image classification (Wang et al., 2021; Niu et al., 2022; Chen et al., 2024) and language modeling (Hardt & Sun, 2024; Hübotter et al., 2025; Hu et al., 2025), where the learning signal is prediction entropy, perplexity, or retrieved nearest neighbors. However, in end-to-end driving, test-time compute has mostly been spent on selecting or refining trajectories without optimizing the underlying policies (Hu et al., 2023; Dauner et al., 2023; Guo et al., 2026). Most related to our work is the Centaur (Sima et al., 2025) method, which minimizes an uncertainty measure over the policy's outputs at test time. However, uncertainty is merely a proxy: a policy can become more confident without becoming more correct, which is a failure expected under domain shift or when learning from scratch. In contrast, our learning signal is grounded in explicit, interpretable driving criteria (drivable area, occupancy, and comfort), ensuring that the policy adapts toward safe and physically valid behaviors.

3 Method: Test-time learning via rejection sampling (TTLR)

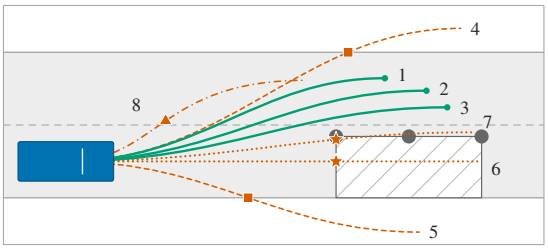
We propose to use rejection sampling with safety verifiers for learning driving policies at test time. For each test scenario, we sample K candidate trajectories τ with $\tau_k \sim \pi_\theta(\cdot \mid o_t)$, where $k \in 1, ..., K$ and π_θ is a driving policy parameterized by θ. o_t is the observation data at time step t, including camera images, past trajectories, and driving commands. We accept and reject candidate trajectories using a binary verifier $V(\cdot)$ that evaluates a set of 3 safety criteria c:

$$V(o_t, \tau) = \mathbf{1}\left[\begin{array}{l} c_{\text{drivable}}(o_t, \tau) \leq 0, \\ c_{\text{occupancy}}(o_t, \tau) \leq 0, \\ c_{\text{comfort}}(o_t, \tau) \leq 0 \end{array}\right]. \tag{1}$$

Figure 2 shows how our verifier scores trajectories against drivable area, occupancy, and comfort criteria. Trajectories 4 and 5 are rejected because they leave the drivable area (i.e., drive off the road). We detect that using two stages: (1) we use SAM 3 (Carion et al., 2026) to segment roads, off-road areas, and road markings, and UniDepthV2 (Piccinelli et al., 2025) to estimate metric depth in the front-view image; (2) we project roads, off-road areas, and road markings into bird's-eye-view (BEV), and determine off-road driving using an ego lane estimate and the road boundaries. Trajectories 6 and 7 are rejected since they intersect with the construction zone (i.e., intersecting an occupied area). We detect that by using SAM to segment vehicles and obstacles and project them to BEV as before to generate a basic occupancy grid. Afterwards, we reject trajectories that intersect with occupied cells. Trajectory 8 is rejected for comfort criteria, i.e. excessive tortuosity or jerk (inspired by the MMS metric (Wagner et al., 2026)).

For trajectories 1 to 3 in Figure 2, all criteria are fulfilled, thus they form a set of accepted trajectories,

$$\mathcal{A}(o_t) = \{\tau : V(o_t, \tau) = 1\}.$$



k	drivable	occupancy	comfort	V
1	✓	✓	✓	✓
2	✓	✓	✓	✓
3	✓	✓	✓	✓
4	■	✓	✓	✗
5	■	✓	✓	✗
6	✓	★	✓	✗
7	✓	★	✓	✗
8	✓	✓	▲	✗

Figure 2: **Our verifiers check explicit safety criteria.** Left: the candidate trajectories of Figure 1 for a lane blocked by a construction zone (shown as white area with gray stripes). Right: $V(o_t, \tau) = 1$ only if a candidate stays in the drivable area, does not overlap occupied space, and satisfies comfort limits; markers show which criterion a candidate violates, and where.

Afterwards, we optimize the policy weights θ with one gradient step toward the accepted trajectories

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}_{\text{accepted}}\big(\mathcal{A}(o_t)\big),$$

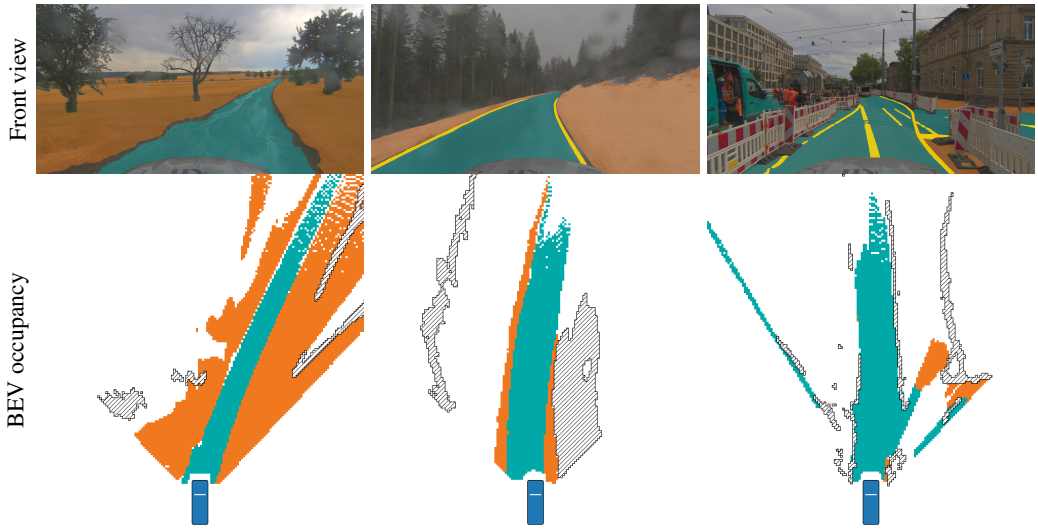
where $\mathcal{L}_{\text{accepted}}$ is the motion planning loss specific to the policy and η is the learning rate.

4 RESULTS

In this section, we evaluate how our test-time learning method generalizes across different policy types, i.e. reasoning-capable driving policies (see Table 1 and Table 2) and standard driving policies (see Table 3). Additionally, we evaluate how our method improves pre-trained policies (i.e., test-time fine-tuning in Section 4.3.1 and Section 4.3.2) and whether driving policies can be learned from scratch at test time (i.e., test-time learning from scratch in Section 4.4).

4.1 COMMON VERIFIER SETUP

For the drivable area and occupancy criteria (see Equation (1)), we use a SAM 3 (Carion et al., 2026) segmentation model with 900M parameters, and a UniDepthV2 (Piccinelli et al., 2025) depth estimation model with a ViT-L backbone, a patch size of 14×14 pixels, and 400M parameters. Afterwards, we sweep the vehicle's footprint along sampled candidate trajectories over a per-scenario occupancy grid with a cell size of $0.25\text{m} \times 0.25\text{m}$. Figure 3 shows a qualitative example. A candidate trajectory is rejected if it drives off-road or if the footprint overlaps with an obstacle.



(a) Unpaved road (b) Snow & water droplets (c) Construction zone

Figure 3: **Qualitative examples on KITScenes LongTail**. Each column shows a different scenario. **Top:** front-view camera images overlaid with segmentation masks predicted by SAM 3 (Carion et al., 2026) for road, off-road, road markings, and obstacles. **Bottom:** the corresponding bird's-eye-view (BEV) occupancy maps, obtained by lifting the front-view masks into 3D using metric depth maps from UniDepthV2 (Piccinelli et al., 2025) and projecting them onto the ground plane. **(a)** The drivable area is correctly identified on unpaved roads without lane markings. **(b)** Our method remains robust in snowfall, producing accurate occupancy maps even with minor water droplets on the lens. **(c)** We accurately capture both a complex barrier layout and overlapping regular and temporary road markings. We include further qualitative results in Figure 6.

As comfort criteria, our verifier rejects trajectories that have a tortuosity value greater than $1.1\times$ the median or a jerk value $2.5\times$ greater than the median of the candidate trajectories. We use the median as robust baseline against standard operational variance and increase the threshold slightly compared to the MMS metric (Wagner et al., 2026). When applied to a smooth, nominal trajectory of $1.0\text{m}/\text{s}^3$, the $2.5\times$ jerk multiplier scales to a ceiling of $2.5\text{m}/\text{s}^3$, matching the limits established in ISO 15622:2018, where values up to $2.5\text{m}/\text{s}^3$ remain acceptable for decisive maneuvers like overtaking.

4.2 DATASETS AND METRICS

Our evaluation focuses on unusual, long-tail driving data since this setting resembles an open challenge in autonomous driving (Wagner et al., 2026; Xu et al., 2026). Additionally, this is an effective way to fine-tune policies after training on nominal driving conditions, compensating for the domain shift. We evaluate our method on two primary benchmarks: KITScenes LongTail (Wagner et al., 2026), a benchmark designed to assess driving models in rare, unusual scenarios and the basis for the 2026 KITScenes Few-Shot Challenge; and Waymo Open E2E (WOD-E2E, Xu et al. (2026)), a large-scale dataset specifically curated for end-to-end autonomous driving that emphasizes challenging, long-tail scenarios to test policy robustness in complex real-world environments.

Both datasets share a unified input structure and are accompanied by similar evaluation protocols. They provide multi-view camera images and high-level driving commands as inputs, and standardize the temporal horizon to a 4-second past trajectory context and a 5-second future prediction window. To assess driving policies, both benchmarks utilize pseudo-simulation metrics (i.e., multi-maneuver scores (MMS) and rater feedback scores (RFS)), alongside standard average displacement errors (ADE). To allow a direct comparison, all metrics are computed using greedy decoding.

4.3 TEST-TIME FINE-TUNING

Offline-trained driving policies often suffer from performance degradation when deployed in unusual, long-tail scenarios due to domain shifts from nominal training conditions. In this section, we evaluate the ability of our test-time learning method (TTLR) to fine-tune pre-trained policies on the fly. By leveraging safety verifiers and rejection sampling, our method adapts the policy using only the test observations, compensating for domain shifts without requiring ground-truth trajectories, differentiable simulators, or offline retraining.

4.3.1 DISTILLATION AT TEST-TIME

We first evaluate online distillation, where candidate trajectories are sampled from a strong teacher policy π_T (Alpamayo 1.5 (Wang et al., 2025)) and filtered by our safety verifier. The policy is then updated with one gradient step toward the accepted trajectories.

Hyperparameters. We sample $K = 120$ candidate trajectories for each test scenario using the teacher policy π_T. For the learning rate, we use the recommended values in the corresponding papers. Specifically, we set $\eta = 2 \times 10^{-5}$ for Alpamayo 1.5 (Wang et al., 2025), $\eta = 1 \times 10^{-4}$ for AutoVLA (Zhou et al., 2025) and OneVL (Lu et al., 2026). For all models, we use the publicly available weights as initialization, which are learned on predominantly nominal driving conditions. We pre-compute the candidate trajectories at the start of training. Thus, we perform regular distillation from Alpamayo to AutoVLA and OneVL, and distillation from an "older" initial version of Alpamayo to its current weights during training.

Fine-tuning with TTLR at test time consistently improves composite driving scores and displacement errors. Table 1 presents the results on the KITScenes LongTail and WOD-E2E datasets. On the KITScenes LongTail dataset, AutoVLA's ADE drops by 28.80%, from 22.12 to 15.75, while its MMS improves by 10.86%. Similarly, OneVL and Alpamayo 1.5 experience substantial reductions in ADE of 22.43% and 29.09%, respectively, alongside notable gains in MMS. This trend holds on the WOD-E2E dataset, where the fine-tuned Alpamayo 1.5 achieves a 12.85% increase in RFS to 6.41 and an 18.54% decrease in ADE to 3.47. These results demonstrate that verifier-guided distillation effectively compensates for domain shifts at test time, allowing pre-trained policies to rapidly adapt to long-tail scenarios using only a teacher's rollout distributions.

4.3.2 SELF-DISTILLATION AT TEST-TIME

Next, we investigate online self-distillation, where the policy generates its own candidate trajectories for the verifier, removing the need for an external teacher. Table 2 compares our self-distillation approach against the pre-trained Alpamayo 1.5 baseline and the winners of the 2026 KITScenes Few-Shot Challenge, namely TTVLM-Base (Azevedo et al., 2026), Kinematics-V2, and A.15 KE:SAI (based on Wang et al. (2025)). We report results for both the online evaluation during the continuous adaptation process and the final adapted policy.

Table 1: **Distillation at test time consistently improves driving performance.** We compare reasoning policies using their released base weights against the same policies trained with our method (TTLR), where only teacher trajectories accepted by the verifier are used for distillation. TTLR A1.5 refers to using Alpamayo 1.5 (Wang et al., 2025) as teacher policy π_T. We process the test split sequentially: for each scenario, the student takes one gradient step toward the accepted trajectories and then predicts a trajectory with the updated weights. Reported metrics are computed using online predictions, not predictions from the final weights, on the official KITScenes LongTail (Wagner et al., 2026) and WOD-E2E (Xu et al., 2026) test splits.

Dataset	Policy	Test-time learning	MMS ↑	RFS ↑	ADE ↓
KITScenes LongTail	AutoVLA (Zhou et al., 2025)	–	3.13	–	22.12
		TTLR A1.5 (ours)	3.47 +10.86%	–	15.75 -28.80%
	OneVL (Lu et al., 2026)	–	3.91	–	3.70
		TTLR A1.5 (ours)	4.00 +2.30%	–	2.87 -22.43%
	Alpamayo 1.5 (Wang et al., 2025)	–	4.69	–	3.30
		TTLR A1.5 (ours)	5.11 +8.96%	–	2.34 -29.09%
WOD-E2E	Alpamayo 1.5 (Wang et al., 2025)	–	–	5.68	4.26
		TTLR A1.5 (ours)	–	6.41 +12.85%	3.47 -18.54%

Hyperparameters. We sample $K = 120$ candidate trajectories for each test scenario using the current weights of the policy π_θ and set the learning rate $\eta = 2 \times 10^{-5}$. This allows for a direct comparison with the results reported in Table 1 (i.e., distillation vs. online self-distillation).

Test-time self-distillation via TTLR achieves strong performance on recent benchmarks. On the KITScenes LongTail dataset, the baseline Alpamayo 1.5 achieves an MMS of 4.69 and an ADE of 3.30, the final adapted policy boosts the MMS by 17.27% to 5.50 and slashes the ADE by 41.82% to 1.92. This allows our method to significantly outperform all challenge winners, which rely on complex few-shot setups. On the WOD-E2E dataset, the final adapted policy increases RFS by 23.94% to 7.04 and reduces ADE by 23.47% to 3.26 compared to the pre-trained baseline. These results confirm that driving policies can effectively learn from their own verifier-filtered rollouts. By treating safety verifiers as direct supervisory signals, self-distillation enables policies to achieve state-of-the-art robustness in long-tail scenarios without offline retraining or external supervision.

Table 2: **Test-time self-distillation achieves strong performance on both KITScenes LongTail and WOD-E2E benchmarks**. We perform test-time learning via rejection sampling (TTLR) with the same policy acting as both teacher and student. Unlike the static teacher in Table 1, the teacher here evolves with the student: for each scenario, candidate trajectories are sampled from the current weights and filtered by the verifier. Then, the policy is optimized with one gradient step toward the accepted trajectories. We evaluate Alpamayo 1.5, the strongest policy in Table 1, and include the top three submissions to the 2026 KITScenes Few-Shot Challenge as baselines. Eval. online refers to evaluating predictions made during a single pass over the test split, and eval. final refers to predictions from the weights obtained after this pass. We report the best values here and include means and stds in the appendix.

Dataset	Policy	Test-time learning	Eval.	MMS ↑	RFS ↑	ADE ↓
KITScenes LongTail	TTVLM-Base* (Azevedo et al., 2026)	–	–	5.15	–	2.48
	Kinematics-V2*	–	–	4.31	–	3.57
	A.15 KE:SAI* (Wang et al., 2025)	–	–	4.31	–	3.17
	Alpamayo 1.5 (Wang et al., 2025)	–	–	4.69	–	3.30
		TTLR (ours)	online	4.99 +6.40%	–	2.25 -31.8%
			final	5.50 +17.27%	–	1.92 -41.82%
WOD-E2E	Alpamayo 1.5 (Wang et al., 2025)	–	–	–	5.68	4.26
		TTLR (ours)	online	–	6.70 +17.96%	3.37 -20.89%
			final	–	7.04 +23.94%	3.26 -23.47%

*Winners of the 2026 KITScenes Few-Shot Challenge

4.4 TEST-TIME LEARNING FROM SCRATCH

Finally, we test the limit of our hypothesis: *Can driving policies be learned from scratch at test time using only a safety verifier?* We drop pre-training entirely and evaluate this on three well-established policy architectures: VADv2 (Jiang et al., 2026), CILRS (Codevilla et al., 2019), and CoverNet (Phan-Minh et al., 2020).

We specifically choose these driving policies because they perform motion planning as a classification problem over a predefined trajectory vocabulary. This formulation enables us to initialize trajectories in a reasonable, physically plausible way, and focuses this experiment entirely on learning to choose the correct trajectory. Crucially, this removes the need for the policy to learn how to generate continuous trajectories, which is challenging from limited examples and without offline pre-training.

To adapt these models to our end-to-end test-time setting, we make two specific architectural modifications. First, we adapt CoverNet from motion forecasting to end-to-end motion planning by using front-view camera images as inputs. Second, we replace the continuous output representation of CILRS with a discrete trajectory vocabulary to align it with the classification-based planning paradigm. Additionally, we compare our TTLR method against Centaur (Sima et al., 2025), the current state-of-the-art for test-time training in autonomous driving, which minimizes an uncertainty proxy over the policy's outputs.

Hyperparameters. We provide each policy with a trajectory vocabulary of 4096 trajectories. For VADv2 and CILRS, we use the static trajectory initialization from Jiang et al. (2026), which is based on a kinematic model. For CoverNet, we use it's dynamic trajectory initialization, considering the current speed. For TTLR, we set the learning rate $\eta = 2 \times 10^{-4}$ for all policies. For Centaur (Sima et al., 2025), we set VADv2: $\eta = 1 \times 10^{-3}$ and use a buffer of 4; CILRS: $\eta = 5 \times 10^{-4}$ and use a buffer of 4; CoverNet: $\eta = 5 \times 10^{-4}$ and use a buffer of 16.

A driving policy can become more confident without becoming more correct. Table 3 reveals a stark contrast between uncertainty minimization and our rejection sampling-based test-time learning. Centaur yields marginal improvements for VADv2 but fails for CILRS and CoverNet (ADE actually worsens). This resembles a case where a driving policy can become more confident without becoming more correct (see Figure 4). Minimizing uncertainty of an uncalibrated model just sharpens the dominant mode without improving the prediction.

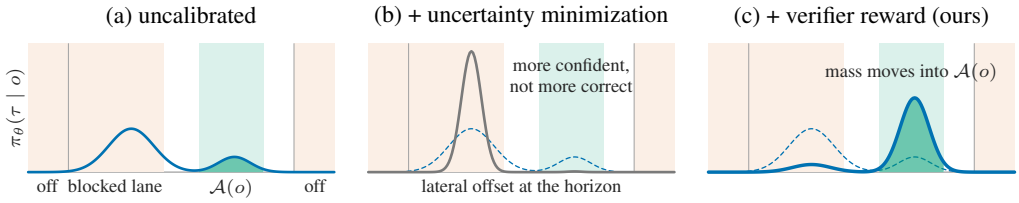


Figure 4: **A driving policy can become more confident without becoming more correct.** Densities over the lateral offset of the trajectory at the horizon for the scene in Figure 2; orange bands are rejected, the green band is the accepted region $\mathcal{A}(o)$, and the filled area is the probability of sampling an accepted trajectory. Minimizing uncertainty (Wang et al., 2021; Sima et al., 2025) sharpens the dominant mode in place (b), whereas fitting accepted samples moves probability mass into $\mathcal{A}(o)$ (c), provided the proposals put some mass there. Dashed: the pre-trained policy from (a).

Conversely, TTLR drastically improves all three policies from scratch. For VADv2, TTLR increases MMS by 18.53% and reduces ADE by 67.31%. For CILRS, our method increases MMS by 7.12% and reduces ADE by 27.23%. For CoverNet, TTLR increases MMS by 21.92% and reduces ADE by 57.45%. Because the verifier enforces explicit physical and safety constraints rather than a statistical proxy, the gradient steps consistently push the policy's probability mass into the accepted safe region (see Figure 4 (c)), enabling test-time learning of complex driving behaviors without any human annotations. Additionally, our method outperforms well-established baselines, which were pre-trained on nominal conditions in nuScenes (Caesar et al., 2020).

Table 3: **Driving policies can be learned from scratch at test time, given a reasonable trajectory vocabulary.** VADv2, CILRS, and CoverNet are probabilistic planning policies, here trained without any pre-training. As in Table 2, candidate trajectories are sampled from the current policy and filtered by the verifier, and the policy is optimized with one gradient step toward the accepted trajectories. We compare TTLR against Centaur (Sima et al., 2025), which minimizes an uncertainty measure over the policy's outputs at test time. For reference, gray rows show end-to-end policies pre-trained on nuScenes (Caesar et al., 2020) under nominal conditions and zero-shot evaluated without test-time learning. We report the best values here and include means and stds in the appendix.

Dataset	Policy	Pre-trained	Test-time learning	MMS ↑	ADE ↓
KITScenes LongTail	UniAD (Hu et al., 2023)	✓	–	3.29	10.90
	DMAD (Shen et al., 2025)	✓	–	3.52	10.04
	Epona (Zhang et al., 2025)	✓	–	2.74	17.67
	VADv2 (Jiang et al., 2026)	✗	–	3.13	28.42
		✗	Centaur (Sima et al., 2025)	3.43 +9.58%	21.99 -22.62%
		✗	TTLR (ours)	3.71 +18.53%	9.29 -67.31%
	CILRS (Codevilla et al., 2019)	✗	–	3.37	29.38
		✗	Centaur (Sima et al., 2025)	3.38 +0.30%	29.48 +0.34%
		✗	TTLR (ours)	3.61 +7.12%	21.38 -27.23%
	CoverNet (Phan-Minh et al., 2020)	✗	–	3.33	8.79
		✗	Centaur (Sima et al., 2025)	3.34 +0.30%	8.83 +0.46%
		✗	TTLR (ours)	4.06 +21.92%	3.74 -57.45%

5 Conclusion

We introduced test-time learning via rejection sampling (TTLR), a simple yet highly effective framework for adapting driving policies at test time. By re-purposing existing safety verifiers, which check trajectories against drivable area, occupancy, and comfort criteria, for rejection sampling, we enable driving policies to learn continuously from accepted trajectories. Our method requires no human annotations or differentiable simulators.

Our experiments demonstrate that TTLR effectively compensates for domain shifts in unusual, long-tail driving scenarios. In the test-time fine-tuning regime, it significantly improves state-of-the-art pre-trained policies like Alpamayo 1.5, AutoVLA, and OneVL through both online distillation and self-distillation. Furthermore, we showed that probabilistic policies such as VADv2, CILRS, and CoverNet can be learned entirely from scratch at test time when initialized with a reasonable trajectory vocabulary. By outperforming uncertainty-minimization with Centaur and all submissions to the 2026 KITScenes Few-Shot Challenge, our work highlights that safety verifiers already deployed in autonomous vehicles contain learning signals that can be leveraged for robust, online learning.

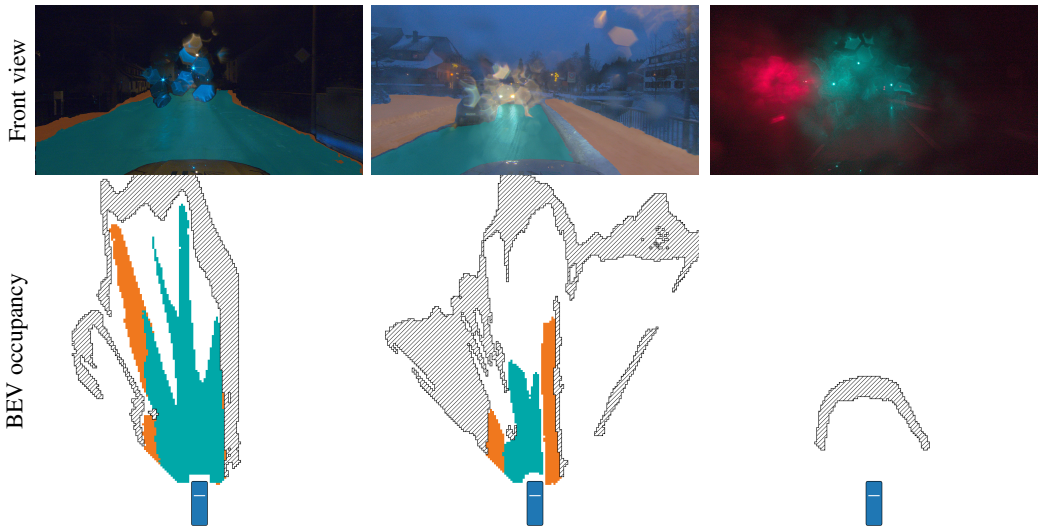
5.1 Limitations and Future Work

While our approach demonstrates strong empirical results, we acknowledge the following limitations, which can be addressed in future work.

Perception failures in extreme weather conditions. Our verifier relies on the robustness of the underlying zero-shot perception models. In extreme visual conditions, such as severe glare from oncoming traffic lights, or heavy rain combined with nighttime driving, the semantic segmentation (SAM 3) and metric depth estimation (UniDepthV2) models can degrade (see Figure 5). This can lead to noisy or incorrect accept/reject signals, which in turn provides poor gradient updates to the policy. Future work could integrate temporal smoothing or fallback heuristics to handle extreme perception failures.

Goodhart's law and verifier overfitting. Finally, we should consider Goodhart's law (Goodhart, 1984): *when a measure becomes a target, it ceases to be a good measure.* Because our policy is optimized directly against the verifier's explicit geometric and kinematic criteria, there is a risk that the policy may learn to "game" the verifier (e.g., finding edge cases in the occupancy grid or exploiting the median-based comfort thresholds) rather than learning universally safe driving

behaviors. Over extended deployment, this over-optimization could cause the verifier's acceptance score to become less informative about the true, holistic safety of the driving policy.



(a) Night-time & droplet artifact (b) Snow & droplet artifact (c) Night-time & light haze artifact

Figure 5: **Qualitative examples of failure cases on KITScenes LongTail**. Each column shows a different scenario. **Top:** front-view camera images overlaid with segmentation masks predicted by SAM 3 (Carion et al., 2026) (we follow the same color scheme as in Figure 3). **Bottom:** the corresponding bird's-eye-view (BEV) occupancy maps, obtained by lifting the front-view masks into 3D using metric depth maps from UniDepthV2 (Piccinelli et al., 2025) and projecting them onto the ground plane. **(a)** Water droplets on the lens corrupt both the segmentation masks and the depth estimates, and are lifted into a spurious wall-like obstacle in front of the ego-vehicle. **(b)** Similarly, droplets combined with light haze produce erroneous segmentation masks and depth estimates, resulting in phantom obstacles in the BEV map. **(c)** The haze is so dense that no road or off-road masks are produced at all, while the depth estimate places a spurious wall only a few meters ahead of the ego-vehicle.

REFERENCES

Caio Azevedo, Stefano Sabatini, Sascha Hornauer, and Fabien Moutarde. Towards zero-shot transfer across embodiments for driving vlas, 2026. URL https://arxiv.org/abs/2609.02341.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11618–11628. IEEE, 2020.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

Yaofo Chen, Shuaicheng Niu, Yaowei Wang, Shoukai Xu, Hengjie Song, and Mingkui Tan. Towards robust and efficient cloud-edge elastic model adaptation via selective entropy distillation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=vePdNU3u6n.

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

Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *Conference on Robot Learning*, pp. 1268–1281. PMLR, 2023.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, KaShun SHUM, and Tong Zhang. RAFT: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=m7p5O7zblY.

C. A. E. Goodhart. *Problems of Monetary Management: The UK Experience*, pp. 91–121. Macmillan Education UK, London, 1984. ISBN 978-1-349-17295-5. doi: 10.1007/978-1-349-17295-5_4. URL https://doi.org/10.1007/978-1-349-17295-5_4.

Ke Guo, Haochen Liu, Xiaojun Wu, Jia Pan, and Chen Lv. ipad: Iterative proposal-centric end-to-end autonomous driving. *IEEE Robotics and Automation Letters*, 2026.

Moritz Hardt and Yu Sun. Test-time training on nearest neighbors for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=CNL2bku4ra.

Jinwu Hu, Zitian Zhang, Guohao Chen, Xutao Wen, Chao Shuai, Wei Luo, Bin Xiao, Yuanqing Li, and Mingkui Tan. Test-time learning for large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24823–24849. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/hu25z.html.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17853–17862. IEEE, 2023.

Jonas Hübotter, Sascha Bongni, Ido Hakimi, and Andreas Krause. Efficiently learning at test-time: Active fine-tuning of LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=NS1G1Uhny3.

ISO 15622:2018. Intelligent transport systems – Adaptive cruise control systems – Performance requirements and test procedures. Standard, International Organization for Standardization, Geneva, CH, September 2018.

Bo Jiang, Shaoyu Chen, Hao Gao, Bencheng Liao, Qian Zhang, Wenyu Liu, and Xinggang Wang. VADv2: End-to-end vectorized autonomous driving via probabilistic planning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Nathan Lambert. Reinforcement learning from human feedback, 2025. URL https://arxiv.org/abs/2504.12501.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. Statistical rejection sampling improves preference optimization. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=xbjSwwrQOe.

Jinghui Lu, Jiayi Guan, Zhijian Huang, Jinlong Li, Guang Li, Lingdong Kong, Yingyan Li, Han Wang, Shaoqing Xu, Yuechen Luo, et al. Xiaomi onevl: One-step latent reasoning and planning with vision-language explanation. *arXiv preprint arXiv:2604.18486*, 2026.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16888–16905. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/niu22a.html.

Tung Phan-Minh, Elena Corina Grigore, Freddy A Boulton, Oscar Beijbom, and Eric M Wolff. Covernet: Multimodal behavior prediction using trajectory sets. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14062–14071. IEEE, 2020.

Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. Unidepthv2: Universal monocular metric depth estimation made simpler. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *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.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

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, 2024. URL https://arxiv.org/abs/2402.03300.

Yinzhe Shen, Omer Sahin Tas, Kaiwen Wang, Royden Wagner, and Christoph Stiller. Divide and Merge: Motion and Semantic Learning in End-to-End Autonomous Driving. *TMLR*, 2025.

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*, 2025.

Royden Wagner, Omer Sahin Tas, Jaime Villa, Felix Hauser, Yinzhe Shen, Marlon Steiner, Dominik Strutz, Carlos Fernandez, Quentin Delfosse, Christoph Weinhuber, Christian Kinzig, Guillermo S. Gutierrez-Cabello, Hendrik Königshof, Fabian Immel, Richard Schwarzkopf, Nils Alexander Rack, Kevin Rösch, Kaiwen Wang, Jan-Hendrik Pauls, Martin Lauer, Igor Gilitschenski, Holger Caesar, and Christoph Stiller. Reasoning models do not yet follow their reasoning in autonomous driving: The kitscenes longtail dataset, 2026. URL https://arxiv.org/abs/2603.23607.

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.

Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, et al. Alpamayo-r1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail. *arXiv preprint arXiv:2511.00088*, 2025.

Runsheng Xu, Hubert Lin, Wonseok Jeon, Hao Feng, Yuliang Zou, Liting Sun, John Gorman, Kate Tolstaya, Sarah Tang, Brandyn White, Ben Sapp, Mingxing Tan, Jyh-Jing Hwang, and Dragomir Anguelov. Wod-e2e: Waymo open dataset for end-to-end driving in challenging long-tail scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3709–3718, June 2026.

Kaiwen Zhang, Zhenyu Tang, Xiaotao Hu, Xingang Pan, Xiaoyang Guo, Yuan Liu, Jingwei Huang, Li Yuan, Qian Zhang, Xiao-Xiao Long, et al. Epona: Autoregressive diffusion world model for autonomous driving. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27220–27230. IEEE, 2025.

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. *Advances in Neural Information Processing Systems*, 38: 27920–27956, 2025.

APPENDIX

Table 4: **Test-time self-distillation gains are consistent across runs.** We repeat the experiment in Table 2 with 3 random seeds and report mean±std. Our method improves all metrics over Alpamayo 1.5 on both KITScenes LongTail and WOD-E2E, with final evaluation outperforming online evaluation.

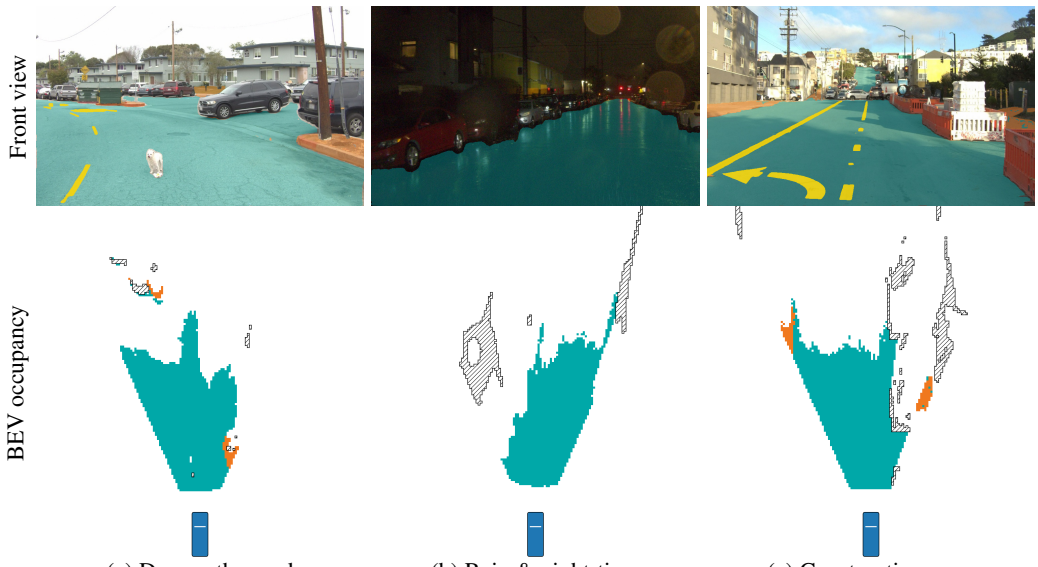
Dataset	Policy	Test-time learning	Eval.	MMS ↑	RFS ↑	ADE ↓
KITScenes LongTail	Alpamayo 1.5 (Wang et al., 2025)	–	–	4.69	–	3.30
		TTLR (ours)	online	5.10 ±0.09	–	2.36 ±0.11
			final	5.26 ±0.22	–	2.07 ±0.13
WOD-E2E	Alpamayo 1.5 (Wang et al., 2025)	–	–	–	5.68	4.26
		TTLR (ours)	online	–	6.75 ±0.10	3.39 ±0.04
			final	–	6.89 ±0.12	3.24 ±0.06

Table 5: **Test-time learning from scratch improves performance consistently.** We repeat the experiment in Table 3 with 3 random seeds and report mean±std. Our method outperforms the uncalibrated policies and Centaur (Sima et al., 2025) on all metrics for all three architectures.

Dataset	Policy	Test-time learning	MMS ↑	ADE ↓
KITScenes LongTail	VADv2 (Jiang et al., 2026)	–	3.21 ±0.12	29.51 ±3.04
		Centaur (Sima et al., 2025)	2.98 ±0.68	24.88 ±2.5
		TTLR (ours)	3.65 ±0.06	9.22 ±0.23
	CILRS (Codevilla et al., 2019)	–	3.32 ±0.05	29.20 ±1.39
		Centaur (Sima et al., 2025)	3.32 ±0.06	29.18 ±1.34
		TTLR (ours)	3.52 ±0.08	21.35 ±1.27
	CoverNet (Phan-Minh et al., 2020)	–	3.28 ±0.07	9.12 ±0.83
		Centaur (Sima et al., 2025)	3.29 ±0.07	9.30 ±0.76
		TTLR (ours)	3.86 ±0.18	4.17 ±0.40

AI ASSISTANCE

We used recent LLMs (Claude Sonnet 5, GPT-5, Qwen, and DeepL) to polish writing and to draft paragraphs and figures based on the authors' content and ideas. Additionally, we used Claude Code for writing helper functions and training loops.



(a) Dog on the road (b) Rain & night-time (c) Construction zone

Figure 6: **Qualitative examples on WOD-E2E**. Each column shows a different scenario. **Top:** front-view camera images overlaid with segmentation masks predicted by SAM 3 (Carion et al., 2026) (we follow the same color scheme as in Figure 3). **Bottom:** the corresponding bird's-eye-view (BEV) occupancy maps, obtained by lifting the front-view masks into 3D using metric depth maps from UniDepthV2 (Piccinelli et al., 2025) and projecting them onto the ground plane. **(a)** Small obstacles, such as the dog on the road, are reliably detected. **(b)** Our method remains robust at night-time and on wet asphalt. **(c)** Dense obstacle layouts, such as closely spaced construction barriers, are accurately captured.