PLANNING-AWARE CONTINUAL TRAVERSABILITY LEARNING FOR OFF-ROAD NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Off-road traversability estimation is particularly challenging due to diverse terrain and continuous changes in visual appearance and physical properties. To address this problem, several recent works formulate traversability estimation as a continual learning problem. These methods manage replay memory and prioritize update samples using perception-level criteria, including feature diversity and traversable-terrain feature changes. They have improved traversability prediction on newly encountered terrain while reducing forgetting of previously learned terrain. However, their perception-level criteria focus on visual changes rather than downstream planning and may prioritize samples with little influence on path selection. Moreover, these works regard untraversed terrain as non-traversable regardless of its actual traversability. This can unnecessarily constrain safe routes and degrade navigation performance in complex off-road environments. In this paper, we propose PACT, a planning-aware continual traversability framework that selects planning-relevant samples to adapt the model and counterfactually identifies obstacles and traversable terrain in untraversed regions. PACT comprises two novel modules: Planning-Aware Acquisition (PAA) and Counterfactual Asymmetric Replay (CAR). PAA prioritizes samples whose candidate paths exhibit high uncertainty and low traversability. This directly links sample selection to downstream planning rather than to perception-level criteria alone. CAR identifies obstacles and traversable terrain in untraversed regions by counterfactually evaluating how each candidate region affects planning. For asymmetric replay, identified obstacles are combined with current observations to create new training samples, while safe terrain is used for contrastive learning to maintain consistent representations across observations. On the GrandTour continual benchmark, PACT achieves the highest average planning success and lowest average prediction error with fewer samples than the baselines.

1 INTRODUCTION

Autonomous mobile robots have been increasingly deployed in diverse outdoor environments for exploration, inspection, and transportation (Mattamala et al., 2024; Gehring et al., 2021; Lee et al., 2024). Traversability estimation for these robots is essential to enable them to avoid hazardous terrain and select feasible paths (Jung et al., 2024). Off-road environments are particularly challenging because their terrain geometry, physical properties, and visual appearance vary over time. For example, there are irregular slopes and diverse obstacles, including loose gravel and mud, and illumination and weather can change terrain appearance (Jeon et al., 2024; Wellhausen et al., 2019). These challenges require traversability estimation models to continually adapt to new environments while retaining knowledge of previously learned terrain. Conventional traversability estimation models struggle to handle these challenges (Chávez-García et al., 2018).

Recent works have adopted continual learning to address the challenges in off-road environments (Lee et al., 2025; Hindel et al., 2026). The key idea behind their algorithms is to manage replay memory by selecting useful samples for adapting to new terrains and retaining previously learned knowledge. They have obtained impressive performance gains in off-road traversability estimation. However, their algorithms select samples using perception-oriented criteria, such as scene representativeness and shifts in traversable-terrain features (Ma et al., 2025). These criteria cannot directly account for downstream planning and are likely to prioritize samples that exhibit large visual

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.

changes but are less informative for planning. Moreover, these methods blindly treat untraversed regions as non-traversable, regardless of their actual traversability. This conservative labeling causes models to assign low traversability scores to safe regions, leading the planner to avoid these regions and degrading navigation performance.

In this paper, we propose PACT, a planning-aware continual traversability framework that manages its replay buffer with samples that are informative for planning and counterfactually identifies obstacles and traversable terrain in untraversed regions. Unlike previous works that use perception-oriented criteria, PACT leverages planning-oriented criteria to select samples and estimate traversability in untraversed regions. PACT comprises two novel modules: Planning-Aware Acquisition (PAA) and Counterfactual Asymmetric Replay (CAR). PAA estimates the impact of samples on downstream planning based on the uncertainty and traversability of their candidate trajectories. We hypothesize that samples are informative for planning if their candidate trajectories have high uncertainty and low traversability. PAA selects such samples to manage the replay buffer and update the traversability estimation model.

CAR regards candidate regions in untraversed area as obstacles if their impact on candidate trajectories is significant. To evaluate their impact, CAR counterfactually treats each candidate region as traversable and analyzes the changes in the uncertainty and traversability of the candidate trajectories. The detected obstacles and safe terrains are asymmetrically stored and used to train the traversability estimation model. While the detected obstacles are composited into observations to augment training samples with different backgrounds, stored safe terrains are used in contrastive learning to maintain consistent representations across observations. In this contrastive learning, safe terrains and obstacles serve as positive and negative samples, respectively, encouraging current safe-terrain representations to become more similar to safe references than to obstacle representations. CAR thus identifies obstacles and safe terrains in untraversed regions by leveraging planning-oriented criteria, rather than relying on perception-oriented criteria.

The main contribution of this work is PACT, a continual traversability framework that leverages planning-oriented criteria to select informative samples and identify obstacles in untraversed regions. We evaluate PACT on sequential GrandTour missions (Tuna et al., 2026), ranging from paved roads to forest trails and construction sites. PACT is compared with previous state-of-the-art traversability estimation methods across three aspects: 1) adaptation and forgetting, 2) the ability to distinguish visually similar hazards from traversable terrain, and 3) downstream planning performance. We observed that baselines often confuse visually similar hazards with traversable terrain, leading to erroneous traversability estimation and unsafe trajectories. PACT achieves the lowest prediction error and highest planning success with substantially fewer samples than the baselines. Ablation studies demonstrate that both of PACT's main modules play important roles, and their combination contributes substantially to the performance gains.

2 Related Work

2.1 Self-Supervised Off-Road Traversability Estimation

Self-supervised traversability estimation reduces reliance on manual annotation by deriving supervision from robot motion and terrain interaction rather than requiring human-provided terrain labels. These methods typically associate past trajectories, robot footprints, or proprioceptive responses with images and combine the resulting sparse signals with representation learning or segmentation to estimate traversability in untraversed terrain. Seo et al. (2023) label regions traversed by past vehicle trajectories as positive and estimate traversability in unlabeled regions using one-class classification and self-supervised contrastive representation learning. Jeon et al. (2024) learn platform-specific traversability through a guide-filter network that combines surface slope with semantic information and a footprint-supervision module based on pre-driving trajectories. V-STRONG (Jung et al., 2024) applies contrastive learning to vision-foundation-model features by combining human driving trajectories with instance masks generated by SAM (Kirillov et al., 2023), improving zero- and few-shot generalization to on-trail and off-trail terrain and to new environments.

In off-road environments, a robot must adapt its model to newly encountered terrain while retaining knowledge of previously learned terrain. Although the above methods reduce annotation costs and improve generalization, they do not explicitly address this continual learning requirement. PACT

addresses this requirement by combining interaction-derived self-supervision with planning-aware sample selection and bounded replay.

2.2 Continual Traversability Learning

Recent methods apply continual learning to traversability estimation by combining self-supervised labels from robot–terrain interaction with experience replay. WVN (Frey et al., 2023; Mattamala et al., 2025) enables rapid online adaptation using pretrained visual features and interaction-derived supervision. Its primary focus is adaptation to the current environment rather than explicit retention across a sequence of changing environments. IMOST (Ma et al., 2025) formulates this problem as task-free domain-incremental learning, in which visual feature distributions vary across scenes while the semantics of traversable and non-traversable outputs remain unchanged and no scene or task identifier is provided. It uses a scene-aware incremental dynamic memory to manage replay samples. Lee et al. (2025) reconstruct previous samples through generative experience recall, and COTRATE (Hindel et al., 2026) constructs a compact replay memory through diversity-aware feature selection. Together, these studies extend online adaptation toward continual retention, but manage replay memory and prioritize samples for model updates primarily using perception-oriented criteria such as predictive uncertainty, changes in traversable-terrain features, feature diversity, and scene representativeness.

Among these methods, IMOST (Ma et al., 2025) is particularly relevant to PACT because it uses scene-aware replay memory to retain traversability knowledge across changing environments. However, its replay priorities are based on scene information and reconstruction loss rather than planning influence. PACT instead uses planning-oriented criteria to select samples and counterfactual evaluation to distinguish obstacles from traversable terrain in untraversed regions. PACT stores obstacles and safe terrain asymmetrically and uses them for obstacle composition and safe-terrain contrastive learning, respectively, to support model adaptation and knowledge retention.

3 Method: PACT

PACT continually adapts its traversability estimation model to newly encountered terrain while retaining knowledge of previously learned terrain. PACT uses planning-oriented criteria to select informative samples and counterfactually identify obstacles in untraversed regions to support model adaptation and knowledge retention. Figure 1 illustrates PACT's perception–planning loop and how PAA and CAR connect planning with continual adaptation.

3.1 Problem Definition

PACT is initially trained on a source mission and then receives observations from subsequent off-road missions in temporal order without task identifiers for continual adaptation. Each mission provides a time-ordered sequence of RGB images, aligned depth maps, and vehicle states collected during robot traversal. As in prior continual traversability methods, PACT learns from robot–terrain interaction using self-supervised labels and experience replay. At time t, the observation $x_t = (I_t, D_t, q_t)$ consists of an RGB image I_t, an RGB-aligned depth map D_t, and the vehicle state q_t. We denote the initial traversability model trained on a source mission by M_{base}. Given x_t, the current model M_t combines D_t with features extracted from I_t by a frozen DINOv2 encoder (Oquab et al., 2024) to predict a per-pixel traversability map T_t. PACT projects T_t into a bird's-eye-view (BEV) traversability map and uses this map to generate and evaluate candidate trajectories that are feasible from q_t. Self-supervised annotation (SSA) constructs a traversability label map by identifying safe terrain through robot–terrain interaction and expanding the safe labels using segmentation masks to provide supervision beyond the traversed footprint. The label map Y_t provides continuous traversability targets, while the validity mask Ω_t marks pixels with valid labels so that unlabeled terrain is excluded from regression.

We propose PAA and CAR to support continual adaptation and knowledge retention through planning-oriented sample selection and asymmetric replay. At each time step, the current model predicts traversability from an observation, and the planner uses this prediction to generate and evaluate feasible candidate trajectories. PAA selects observations informative for planning using the uncertainty and traversability of candidate trajectories. Each selected observation is paired with

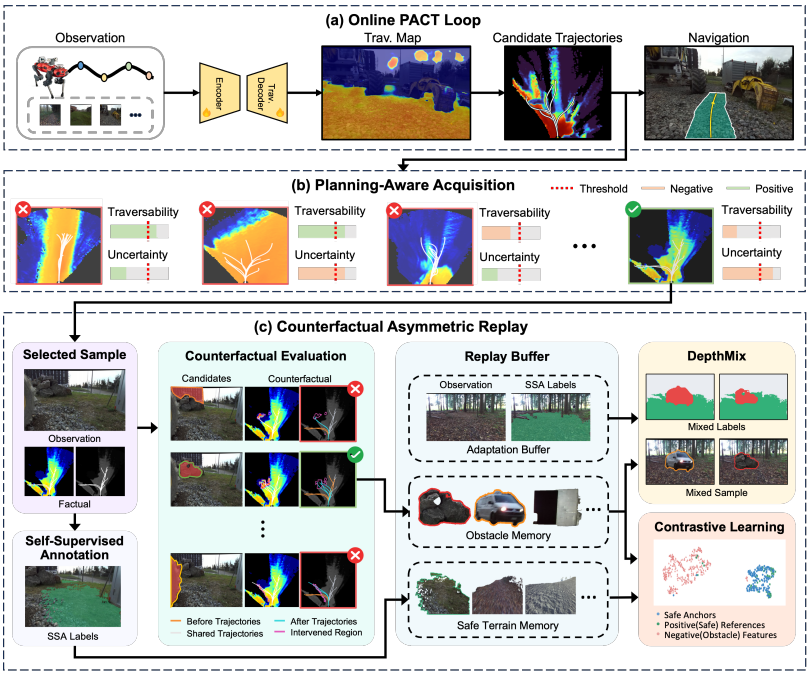

Figure 1: Overview of PACT. (a) The traversability model supports candidate-trajectory evaluation. (b) Planning-Aware Acquisition selects observations whose candidate trajectories have high uncertainty and low traversability. (c) Counterfactual Asymmetric Replay identifies planning-relevant obstacles and combines obstacle mixing with safe-terrain contrastive learning.

self-supervised labels once they become available from robot–terrain interaction. CAR identifies obstacles in untraversed regions by temporarily treating candidate regions as traversable and evaluating changes in candidate trajectories. The identified obstacles and safe terrain are asymmetrically stored and used to train the traversability estimation model. The following sections describe PAA and CAR in detail.

3.2 Planning-Aware Acquisition

Planning-aware adaptation requires sample selection to reflect downstream planning rather than visual changes alone. PAA addresses this requirement by evaluating the uncertainty and traversability of candidate trajectories. Candidate trajectories can spread widely across broad safe regions, so high uncertainty alone does not necessarily indicate a need for further learning. Low traversability alone does not necessarily indicate a need for further learning. It can reflect obstacles that the model already predicts correctly. PAA therefore prioritizes observations whose candidate trajectories have both high uncertainty and low traversability.

To compute uncertainty and traversability, PAA generates candidate trajectories using Model Predictive Path Integral (MPPI) (Williams et al., 2017). The traversability map T_t is projected into a metric BEV traversability map T_t^{BEV}, from which a terrain-cost map C_t is derived. MPPI generates trajectories by perturbing an initial control sequence, so the initialization influences which routes the candidates explore. PAA therefore initializes MPPI with multiple precomputed base trajectory priors that span diverse reachable positions and headings within the robot's motion constraints and BEV limits. Each base prior receives a quota for retained candidates according to its mean traversability on the BEV map, giving more traversable routes greater representation. MPPI generates perturbed trajectories around priors with nonzero quotas, and the lowest-cost candidates are retained within each quota. Appendix B.2 details the base-prior construction and quota allocation.

The selection criteria are computed from retained low-cost candidates to reflect routes favored by the planner. These candidates form the elite set $\mathcal{E}_t = \{\tau_i\}_{i=1}^{K}$. Each τ_i contains uniformly spaced positions p and their headings, and $|\tau_i|$ denotes the number of samples. Within this set, the mean

pairwise Fréchet distance d_t^{traj} measures trajectory divergence as an estimate of uncertainty, while $\bar{T}_t^{\mathrm{elite}}$ measures mean traversability.

$$d_t^{\mathrm{traj}} = \frac{2}{K(K-1)} \sum_{\substack{i<j \\ \tau_i,\tau_j \in \mathcal{E}_t}} d_{\mathrm{F}}(\tau_i, \tau_j), \qquad \bar{T}_t^{\mathrm{elite}} = \frac{1}{|\mathcal{E}_t|} \sum_{\tau \in \mathcal{E}_t} \frac{1}{|\tau|} \sum_{p \in \tau} T_t^{\mathrm{fill}}(p). \tag{1}$$

Here, d_{F} includes position and heading, and T_t^{fill} fills unobserved BEV cells with a fixed score (Appendix B.2).

The raw planning trigger r_t requires both high uncertainty and low traversability, using a divergence threshold δ_{div} and a traversability threshold δ_{trav}.

$$r_t = \mathbf{1}[d_t^{\mathrm{traj}} \geq \delta_{\mathrm{div}}]\, \mathbf{1}[\bar{T}_t^{\mathrm{elite}} \leq \delta_{\mathrm{trav}}]. \tag{2}$$

Temporal filtering suppresses transient selections and repeated observations. A trigger must persist over time, and successive selections must be separated by a minimum interval. Observations selected by this temporally filtered trigger undergo counterfactual obstacle identification before admission to the adaptation buffer (Section 3.3).

Complementary sampling retains safe-terrain references that the planning trigger can miss. Safe terrain can remain unselected when candidate trajectories are closely grouped or have high traversability. To include such terrain, PAA samples observations without planning triggers at spaced time intervals and stores SSA-confirmed safe regions directly in the safe-terrain memory, without counterfactual obstacle identification (Section 3.4).

3.3 Counterfactual Asymmetric Replay: Obstacle Identification

PAA selects observations informative for planning but does not identify which regions influence the candidate trajectories. CAR therefore uses counterfactual evaluation to identify obstacles for replay according to their influence on planning. This evaluation considers candidate obstacle regions selected from SAM2 masks (Ravi et al., 2025) in untraversed terrain based on geometric and visual cues.

Counterfactual evaluation isolates a candidate region's influence by changing its traversability while holding other planning conditions fixed. For a candidate mask $M_{\mathrm{obj}}^{(i)}$, CAR constructs a counterfactual map $\widetilde{T}_t^{(i)}$ by replacing the traversability values inside the mask with T_{safe}. The counterfactual map is projected to BEV, and elite trajectories are reselected from the same candidate set while the other planning conditions and self-supervised labels remain unchanged. The divergence reduction $\Delta d_i^{\mathrm{traj}}$ and mean traversability change $\Delta \bar{T}_i$ measure the planning response, with the magnitude of $\Delta \bar{T}_i$ contributing to candidate priority.

$$\begin{aligned} \widetilde{T}_t^{(i)} &= T_t\big(1 - M_{\mathrm{obj}}^{(i)}\big) + T_{\mathrm{safe}} M_{\mathrm{obj}}^{(i)}, \\ \Delta d_i^{\mathrm{traj}} &= d_t^{\mathrm{traj}} - \widetilde{d}_t^{\mathrm{traj},(i)}, \qquad \Delta \bar{T}_i = \widetilde{\bar{T}}_t^{\mathrm{elite},(i)} - \bar{T}_t^{\mathrm{elite}}. \end{aligned} \tag{3}$$

Figure 2 illustrates how candidate regions differ in their influence on trajectory divergence.

The counterfactual results determine which obstacle regions are stored for replay and which observations are retained for adaptation. Candidate regions that satisfy the depth and path-support criteria and sufficiently reduce trajectory divergence are stored in the obstacle memory; Appendix B.3 details these criteria. If a verified obstacle intervention resolves the planning trigger, the trigger can be explained by the obstacle's effect on the candidate routes. CAR therefore excludes this observation from adaptation while retaining the identified obstacles for replay. If high uncertainty and low traversability persist after all verified obstacle interventions, CAR treats the observation as requiring further terrain learning and retains it for adaptation.

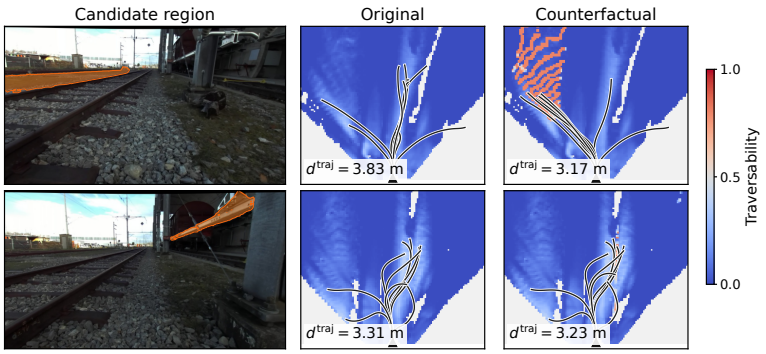

Figure 2: Counterfactual evaluation for region identification. Treating the orange region as traversable reduces trajectory divergence substantially (top) or slightly (bottom). Blue/red indicates low/high traversability; curves show elite trajectories.

3.4 Counterfactual Asymmetric Replay: Role-Specific Memory and Replay

During continual adaptation, the model must continue to recognize obstacles across different backgrounds while maintaining consistent safe-terrain representations. CAR supports these two requirements through asymmetric replay, using obstacle composition and safe-terrain contrastive learning. The obstacle memory provides stored obstacles for composition, while the safe-terrain memory provides reference regions for contrastive learning. The adaptation buffer holds observations retained for model adaptation after counterfactual obstacle identification. As the model learns from observations in the adaptation buffer, safe regions with low prediction error and high recall are extracted and transferred to the safe-terrain memory for subsequent replay. All three memories have fixed capacities; Appendix B.3 details their stored data and management rules.

To train the model to recognize stored obstacles as non-traversable across different backgrounds, CAR composites them into current observations. Depth guides the scale and placement of each obstacle. For a mixed observation, let T^{mix} and Y^{mix} denote its prediction and label, and $\mathcal{H}$ the valid pixels of the pasted obstacle. The obstacle replay loss averages prediction error within $\mathcal{H}$, giving the obstacle its own supervision term regardless of its image area.

$$L_{\text{mix}} = \frac{1}{|\mathcal{H}|} \sum_{p \in \mathcal{H}} \left(T^{\text{mix}}(p) - Y^{\text{mix}}(p)\right)^2. \tag{4}$$

L^{-}_{global} preserves non-traversable predictions on stored obstacles in their original observations, complementing the supervision on composited obstacles provided by L_{mix}.

Safe-terrain replay preserves learned safe-terrain representations while keeping them distinct from obstacle representations. Safe regions are not composited across scenes because broad terrain surfaces may not match the geometry of another scene. A copy of the model is frozen during each update to provide stable reference embeddings while the current model learns. The feature-matching term L_{distill} matches model embeddings on stored safe regions to the corresponding frozen-teacher embeddings to preserve learned representations. To also distinguish safe terrain from obstacles, contrastive learning encourages current safe embeddings to be closer to safe teacher references than to replayed obstacle embeddings. For each normalized current safe embedding $a \in \mathcal{A}$, $p^{+}(a)$ is the nearest safe teacher reference and $h^{-}(a)$ is the most similar replayed obstacle embedding. Using cosine distance d and a smooth margin penalty ℓ_m, the combined safe-feature objective is

$$L_{\text{con}} = \frac{1}{2}\left[L_{\text{distill}} + \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \ell_m\big(d(a, p^{+}(a)) - d(a, h^{-}(a))\big)\right]. \tag{5}$$

Feature-level consistency does not directly supervise traversability outputs, so L_{out} additionally supervises predictions on stored safe regions.

The overall learning objective combines adaptation to current terrain with retention of replayed knowledge. The task loss L_{task} learns from the self-supervised labels of observations retained for adaptation and is combined with the replay losses above. For retained observations, only labeled

safe regions and depth-supported obstacle regions provide regression supervision, while unlabeled terrain is excluded. The overall objective is

$$\mathcal{L} = \lambda_{\text{task}} L_{\text{task}} + \lambda_{\text{mix}} L_{\text{mix}} + \lambda_{\text{global}} L^{-}_{\text{global}} + \lambda_{\text{con}} L_{\text{con}} + \lambda_{\text{out}} L_{\text{out}}. \tag{6}$$

Each λ is the effective weight of its loss; scheduling, normalization, and detailed loss definitions are provided in Appendix B.1.

4 EXPERIMENTS

We design our experiments to investigate the following questions: (E1) Can PACT achieve robust adaptation and mitigate forgetting with fewer samples? (E2) Can PACT distinguish visually similar safe and hazardous regions? (E3) Can PACT achieve performance gains on downstream planning? (E4) How does each main component of PACT contribute to its overall performance?

We compare PACT with Complete-Memory (CM), Reservoir (Vitter, 1985), WVN, LwF (Li & Hoiem, 2018), and IMOST. To provide a fair comparison, we implemented them to take common inputs and share the same backbone. Note that while the baselines perform adaptation using every incoming observation, PACT selectively stores and utilizes only the observations that are informative for planning. The implementation details, including learning and retention principles, are described in Appendix C.1.

4.1 GRANDTOUR CONTINUAL BENCHMARK AND PROTOCOL

We construct a benchmark for continual traversability learning by sequentially arranging eight GrandTour missions (A–H), including paved roads, forest trails, and construction sites. These missions expose the model to changes in terrain materials, geometry, and illumination, testing whether it can learn new terrain without losing performance on earlier missions. The model is first trained on Mission A and then adapted to Missions B–H in this order. Each mission has separate adaptation, validation, and fixed test splits. After each adaptation stage, we evaluate the model on all missions encountered so far to measure adaptation and forgetting. Appendix A.1 provides the benchmark and protocol details, including the data splits and evaluation metrics.

4.2 OVERALL CONTINUAL LEARNING

Table 1 compares traversability prediction and forgetting after sequential adaptation through Missions B–H. Reservoir outperforms CM on most metrics, showing that storing every observed frame does not guarantee better continual-learning performance. PACT achieves the best performance across all evaluated prediction and forgetting metrics, using 81.0% fewer training samples than the baselines. This result indicates that PACT maintains prediction accuracy while preserving previously learned terrain knowledge, despite using fewer training samples.

Table 1: Overall continual learning performance after sequential adaptation through Missions B–H.

Method	Frames Used ↓	Avg. RMSE ↓	Mean AUROC ↑	Safe IoU ↑	Bal. Acc. ↑	Avg. Forgetting ↓
CM	2,893	0.1183	0.9873	0.9529	0.9608	0.0174
Reservoir	2,893	0.1175	0.9932	0.9480	0.9617	0.0112
WVN	2,893	0.1229	0.9851	0.9016	0.9358	0.0206
LwF	2,893	0.1964	0.9435	0.5478	0.7605	0.0116
IMOST	2,893	0.1345	0.9835	0.8870	0.9274	0.0268
PACT	**550**	**0.1081**	**0.9950**	**0.9696**	**0.9695**	**0.0048**

4.3 VISUALLY SIMILAR SAFE–HAZARD DISCRIMINATION

We investigate whether models correctly identify safe and hazardous regions with similar visual appearances. We select 100 safe–hazard pairs per mission with similar areas and high DINO-feature similarity, yielding 800 pairs in total. All methods are evaluated on the same pairs selected independently of model predictions, with scores averaged equally across Missions A–H. Consistency@0.6

measures correct classification of both regions in a pair. False-Safe and False-Unsafe Rates measure misclassification of hazards and safe terrain, respectively.

PACT achieves 95.38% Consistency@0.6 (Table 2), exceeding the strongest baseline by 6.63 percentage points. These results demonstrate that PACT more reliably identifies visually similar safe and hazardous regions.

LwF achieves the lowest False-Safe Rate (0.25%), but misclassifies 68.50% of safe regions as hazardous. This pattern suggests that its low False-Safe Rate reflects a tendency to classify terrain as hazardous rather than reliable discrimination between safe and hazardous regions. In contrast, PACT achieves low False-Safe and False-Unsafe Rates (0.50% and 4.13%, respectively), identifying hazards while correctly recognizing safe terrain. Figure 3 compares the methods' traversability predictions for a safe floor and a hazardous pillar with similar visual appearances. CM and Reservoir underestimate the floor's traversability, whereas IMOST overestimates the pillar's traversability. PACT correctly classifies both regions despite their visual similarity.

Table 2: Safe–hazard discrimination.

Method	Consistency @0.6 ↑	False-Safe Rate ↓	False-Unsafe Rate ↓
CM	88.75%	2.75%	8.63%
Reservoir	87.50%	2.25%	10.25%
WVN	82.63%	1.88%	15.50%
LwF	31.25%	**0.25%**	68.50%
IMOST	78.25%	3.38%	18.38%
PACT	**95.38%**	0.50%	**4.13%**

Table 3: Offline global planning performance.

Method	Success Rate ↑	Collision Rate ↓	No-Path Rate ↓
CM	70.48%	27.62%	1.90%
Reservoir	71.90%	26.19%	1.90%
WVN	62.38%	22.86%	14.76%
LwF	25.40%	**0.00%**	74.60%
IMOST	61.43%	9.05%	29.52%
PACT	**94.76%**	5.24%	**0.00%**

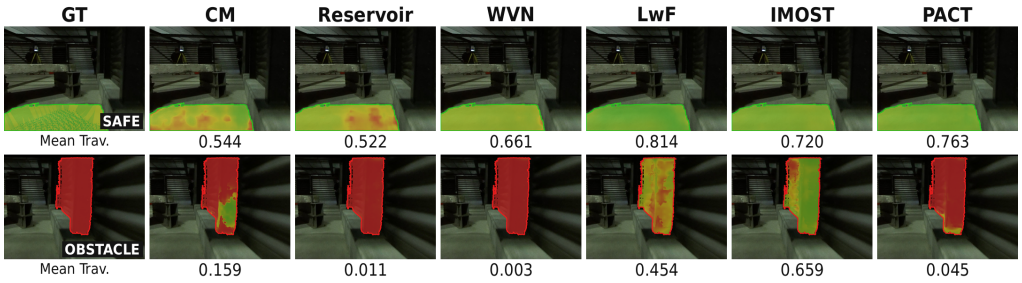

Figure 3: Traversability predictions for a visually similar safe terrain (top) and obstacle (bottom) in the same scene. Values below each prediction are mean traversability scores within the highlighted region; higher scores indicate that the model considers the region safer to traverse.

4.4 Offline Global Planning

We analyze how learned traversability maps affect planning success by using recorded navigation data. Planned paths are checked for collisions, and a query succeeds when the planner finds a collision-free path to the goal. All methods use the same logged geometry, odometry, start–goal queries, A* planner, and collision evaluator. Only the traversability cost map differs between methods. The evaluation includes 189 held-out start–goal queries from Missions B–H, with up to 30 queries per mission. Collision and no path count as failures; rates are computed over all queries per mission and averaged equally across Missions B–H.

Table 3 shows that PACT achieves a 94.76% mean planning success rate across missions, exceeding the strongest baseline, Reservoir, by 22.86 percentage points. LwF achieves a zero Collision Rate, but its No-Path Rate of 74.60% shows that it frequently fails to find a route. PACT achieves a Collision Rate of 5.24% with no path-finding failures, indicating that it can avoid collisions without frequently rejecting feasible routes. Figure 4 visualizes planning outcomes for the same start–goal query in Mission D. CM and Reservoir generate dangerous paths that collide with obstacles, and WVN, LwF, and IMOST fail to find a path. In contrast, PACT generates collision-free paths and achieves the highest mean planning success. Appendix D reports the additional results with different seeds and task orders.

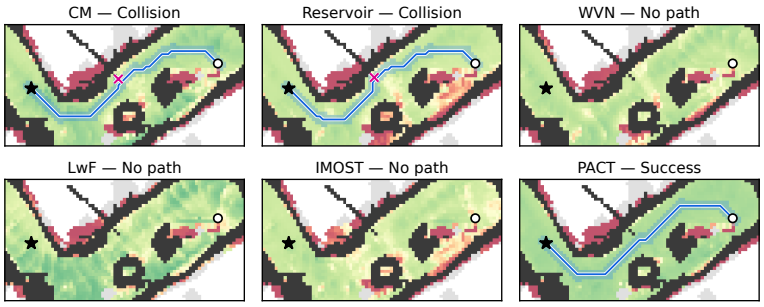

Figure 4: Mission D planning comparison. Background colors show estimated traversability, from low (red) to high (green); dark gray marks evaluation hazards, and blue lines show planned paths. Circles and stars mark the start and goal, respectively. Success requires the robot's full footprint—not only the path centerline—to avoid hazards.

4.5 Component Ablation

Ablation studies examine the individual and combined contributions of PAA and CAR under the same 550-frame budget (Table 4). The corresponding baselines use perception-oriented sample selection and standard full-frame replay.

Table 4: Component ablation with 550 frames. Baseline: perception-oriented selection and full-frame replay; *w/o*: module replacement. Prediction, forgetting, and planning metrics average over Missions A–H, A–G, and B–H, respectively.

Variant	Avg. RMSE ↓	Safe IoU ↑	Avg. Forgetting ↓	Success Rate ↑
Baseline configuration	0.1233	0.9191	0.0153	75.24%
w/o PAA	0.1164	0.9623	0.0062	84.29%
w/o CAR	0.1182	0.9426	0.0061	74.29%
Full PACT	**0.1081**	**0.9696**	**0.0048**	**94.76%**

With CAR fixed, PAA improves traversability prediction and knowledge retention and raises the planning success rate from 84.29% to 94.76%. The gain reflects the full acquisition policy, including the planning trigger and complementary safe-terrain sampling.

With PAA fixed, CAR reduces RMSE from 0.1182 to 0.1081 and raises the planning success rate from 74.29% to 94.76%. This comparison evaluates counterfactual obstacle identification and asymmetric replay jointly. Under the same frame budget, PAA improves prediction accuracy with standard replay, but its planning benefit emerges when combined with CAR. This contrast suggests that selecting samples informative for planning is more effective when paired with CAR.

5 Conclusion

We introduce PACT, a planning-aware continual traversability framework for changing off-road environments. PACT selects samples informative for planning to update the traversability model. It identifies planning-relevant obstacles through counterfactual evaluation and replays obstacles and safe terrain asymmetrically. Experimental results demonstrate that PACT achieves better traversability prediction and offline planning performance than the baselines while using approximately 81% fewer frames for adaptation and replay. PACT has several limitations that warrant further investigation. First, PAA uses fixed thresholds for trajectory divergence and traversability, which may require recalibration when the robot's motion constraints or planning horizon change. We plan to investigate adaptive threshold calibration to extend PACT across different robot and planning configurations. Second, region extraction and model updates introduce computational overhead that must be managed when adaptation and planning run concurrently. We plan to investigate lightweight region extraction and asynchronous model updates to reduce their interference with path generation.

AI USE STATEMENT

In this work, we used generative AI tools for translation during manuscript review and feedback on additional experiments. We have not used generative AI tools to generate synthetic datasets, and mathematical proof-related tasks are not applicable to this work. Additionally, we used generative AI tools to improve the readability of author-provided text, search for related literature, assist with data visualization following author-specified designs, and draft auxiliary code under the authors' instructions. We have reviewed all AI-assisted work. Specifically, we checked the technical accuracy of the revised text and visualizations and reviewed and tested the auxiliary code. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the assistance of generative AI.

REFERENCES

R. Omar Chávez-García, Jérôme Guzzi, Luca M. Gambardella, and Alessandro Giusti. Learning ground traversability from simulations. *IEEE Robotics and Automation Letters*, 3(3):1695–1702, 2018. doi: 10.1109/LRA.2018.2801794. URL https://arxiv.org/abs/1709.05368.

Jonas Frey, Matías Mattamala, Nived Chebrolu, Cesar Cadena, Maurice Fallon, and Marco Hutter. Fast traversability estimation for wild visual navigation. In *Robotics: Science and Systems (RSS)*, 2023. doi: 10.15607/RSS.2023.XIX.054. URL https://roboticsproceedings.org/rss19/p054.html.

C. Gehring, P. Fankhauser, L. Isler, R. Diethelm, S. Bachmann, M. Potz, L. Gerstenberg, and M. Hutter. ANYmal in the field: Solving industrial inspection of an offshore HVDC platform with a quadrupedal robot. In *Field and Service Robotics*, pp. 247–260. Springer, 2021. doi: 10.1007/978-981-15-9460-1_18. URL https://doi.org/10.1007/978-981-15-9460-1_18.

Julia Hindel, Simon Bultmann, Houman Masnavi, Daniele Cattaneo, and Abhinav Valada. Self-supervised online robot-agnostic traversability estimation for open-world environments. *arXiv preprint arXiv:2605.28442*, 2026. URL https://arxiv.org/abs/2605.28442.

Yurim Jeon, E In Son, and Seung-Woo Seo. Follow the footprints: Self-supervised traversability estimation for off-road vehicle navigation based on geometric and visual cues. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1774–1780, 2024. doi: 10.1109/ICRA57147.2024.10611198. URL https://arxiv.org/abs/2402.15363.

Sanghun Jung, JoonHo Lee, Xiangyun Meng, Byron Boots, and Alexander Lambert. V-STRONG: Visual self-supervised traversability learning for off-road navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1766–1773, 2024. doi: 10.1109/ICRA57147.2024.10611227. URL https://arxiv.org/abs/2312.16016.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4015–4026, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Kirillov_Segment_Anything_ICCV_2023_paper.html.

Hojin Lee, Yunho Lee, Daniel A. Duecker, and Cheolhyeon Kwon. Continual learning for traversability prediction with uncertainty-aware adaptation. *IEEE Robotics and Automation Letters*, 10(11):12109–12116, 2025. doi: 10.1109/LRA.2025.3619687. URL https://amilearning.github.io/assets/pdf/continual_ral_2025.pdf.

Joonho Lee, Marko Bjelonic, Alexander Reske, Lorenz Wellhausen, Takahiro Miki, and Marco Hutter. Learning robust autonomous navigation and locomotion for wheeled-legged robots. *Science Robotics*, 9(89):eadi9641, 2024. doi: 10.1126/scirobotics.adi9641. URL https://arxiv.org/abs/2405.01792.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018. doi: 10.1109/TPAMI.2017.2773081. URL https://arxiv.org/abs/1606.09282.

Kehui Ma, Zhen Sun, Chaoran Xiong, Qiumin Zhu, Kewei Wang, and Ling Pei. IMOST: Incremental memory mechanism with online self-supervision for continual traversability learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8788–8794, 2025. doi: 10.1109/ICRA55743.2025.11127949. URL https://arxiv.org/abs/2409.14070.

Matías Mattamala, Nived Chebrolu, Benoit Casseau, Leonard Freißmuth, Jonas Frey, Turcan Tuna, Marco Hutter, and Maurice Fallon. Autonomous forest inventory with legged robots: System design and field deployment. In *IEEE ICRA Workshop on Field Robotics*, 2024. URL https://arxiv.org/abs/2404.14157.

Matías Mattamala, Jonas Frey, Piotr Libera, Nived Chebrolu, Georg Martius, Cesar Cadena, Marco Hutter, and Maurice Fallon. Wild visual navigation: Fast traversability learning via pretrained models and online self-supervision. *Autonomous Robots*, 49(3):19, 2025. doi: 10.1007/s10514-025-10202-x. URL https://doi.org/10.1007/s10514-025-10202-x.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Ha6RTeWMd0.

Junwon Seo, Sungdae Sim, and Inwook Shim. Learning off-road terrain traversability with self-supervisions only. *IEEE Robotics and Automation Letters*, 8(8):4617–4624, 2023. doi: 10.1109/LRA.2023.3284356. URL https://arxiv.org/abs/2305.18896.

Turcan Tuna, Jonas Frey, Frank Fu, Katharine Patterson, Tianao Xu, Maurice Fallon, Cesar Cadena, and Marco Hutter. GrandTour: A legged robotics dataset in the wild for multi-modal perception and state estimation. *arXiv preprint arXiv:2602.18164*, 2026. URL https://arxiv.org/abs/2602.18164.

Jeffrey S. Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software*, 11(1):37–57, 1985. doi: 10.1145/3147.3165. URL https://doi.org/10.1145/3147.3165.

Lorenz Wellhausen, Alexey Dosovitskiy, René Ranftl, Krzysztof Walas, Cesar Cadena, and Marco Hutter. Where should I walk? predicting terrain properties from images via self-supervised learning. *IEEE Robotics and Automation Letters*, 4(2):1509–1516, 2019. doi: 10.1109/LRA.2019.2895390. URL https://cesarcadena.ethz.ch/files/RAL2019_lwellhausen.pdf.

Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M. Rehg, Byron Boots, and Evangelos A. Theodorou. Information theoretic MPC for model-based reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1714–1721, 2017. doi: 10.1109/ICRA.2017.7989202. URL https://doi.org/10.1109/ICRA.2017.7989202.

A Dataset and Continual Protocol

A.1 GrandTour Split and Continual Protocol

We form eight sequential missions from the GrandTour data and train the base model on Mission A before adapting it to Missions B–H in chronological order. Each mission is divided temporally into adaptation, validation, and test subsets using a 70/10/20 target ratio. A 30-frame guard interval is removed at each split boundary to reduce near-duplicate observations across subsets. The learner uses the adaptation subset for continual updates and the validation subset for checkpoint selection; the test subset is reserved for final evaluation. Table A1 reports the resulting frame counts after applying the guard intervals.

Figure A1 shows representative observations from the eight missions. Mission A is used for base-model training, whereas Missions B–H form the chronological continual-adaptation stream.



Figure A1: Representative RGB observations from GrandTour Missions A–H. The examples illustrate the changes in terrain material, geometry, illumination, and scene structure encountered across the continual stream. Mission A is used for base-model training, and Missions B–H are encountered sequentially during continual adaptation.

Table A1: GrandTour temporal split used in all experiments.

Mission	Adaptation	Validation	Test
A	5,885	785	1,661
B	4,036	521	1,133
C	4,431	577	1,246
D	2,442	293	677
E	1,991	228	549
F	3,862	496	1,083
G	4,261	553	1,197
H	2,256	266	624
Total	29,164	3,719	8,170

Table A1 reports the full temporal splits. We construct each BEV map from observations accumulated over the preceding 3 s and update the map and planning at the first observation at least 0.5 s after the previous update. At the maximum speed of 1.3 m/s configured for the planning model, the nominal 0.5-s update interval corresponds to 0.65 m of travel. PACT evaluates its sample-selection criteria at these update times. To provide the same candidate set to all methods, we generate fixed training labels for the corresponding frames within the temporal adaptation splits and use them as a common adaptation pool. Intersecting the recorded BEV-update frame IDs with the adaptation splits yields 2,893 eligible frames from the 23,279 adaptation frames in Missions B–H. This pool is constructed without applying PACT's acquisition trigger. The numbers of eligible frames for B–H are 505, 549, 304, 244, 480, 530, and 281, respectively. All-valid baselines train on every frame in this common pool. All methods receive the same frozen DINOv2 features, RGB-aligned metric depth, continuous traversability labels generated by self-supervised annotation (SSA), and valid masks. The evaluation ground truth assigns continuous values of 0.8–1.0 to the traversed footprint

and 0.6–0.8 to its SAM-expanded traversable region, while geometry-verified hazards are treated as non-traversable. Pixels without valid evaluation labels are excluded from the perception metrics. We use 0.6 as the common safe–hazard decision threshold and first compute every metric per mission before averaging equally across missions. SSA uses the same reference values for all missions: 4.0 m/s^2 for acceleration, 120.0 m/s^3 for jerk, and a roughness penalty of 0.2. Across all missions, trajectory scores span 0.8000–0.9992 and dense positive scores, including both footprint and SAM-expanded support, span 0.6001–0.9985.

Training schedule and checkpoint selection. The prediction and acquisition rules do not take mission identities as inputs, but the experimental runner processes the missions sequentially and uses their validation partitions for checkpoint selection. The recorded protocol requires a trained checkpoint at each mission transition; it is therefore a mission-structured evaluation rather than a fully boundary-free deployment test. Validation and early stopping are performed within training updates, with the optimization settings in Appendix B.1 and checkpoint-selection criteria in Appendix C.1.

Forgetting. For each of Missions A–G, forgetting is the final RMSE minus the lowest recorded RMSE among the available evaluation checkpoints from that mission's introduction onward, clipped below at zero. We average these differences over A–G; Mission H is excluded because no subsequent mission follows it.

Safe–hazard discrimination metrics. For each mission m, we evaluate $N_m = 100$ safe–hazard pairs selected independently of model predictions. Each pair contains safe and hazardous regions from different test frames in the same mission. The safe-to-hazard area ratio is restricted to $[2/3, 1.5]$. Pairs are selected in descending order of frozen DINOv2 feature similarity without reusing either region. Let $s_{m,i}$ and $h_{m,i}$ denote the mean predicted traversability of the safe and hazardous regions in pair i. At the common threshold of 0.6, the mission-level metrics are

$$\begin{aligned}
\text{Consistency}_m &= \frac{1}{N_m}\sum_{i=1}^{N_m}\mathbf{1}[s_{m,i} \geq 0.6 \wedge h_{m,i} < 0.6], \\
\text{Margin}_m &= \frac{1}{N_m}\sum_{i=1}^{N_m}(s_{m,i} - h_{m,i}), \\
\text{FalseSafe}_m &= \frac{1}{N_m}\sum_{i=1}^{N_m}\mathbf{1}[h_{m,i} \geq 0.6], \\
\text{FalseUnsafe}_m &= \frac{1}{N_m}\sum_{i=1}^{N_m}\mathbf{1}[s_{m,i} < 0.6].
\end{aligned} \tag{A1}$$

Here, $\mathbf{1}[\cdot]$ is the indicator function. Each reported score is the arithmetic mean of the eight mission-level scores; Consistency@0.6 and the two error rates are expressed as percentages.

B PACT Implementation Details

B.1 Model, Optimization, and Continual Objective

The predictor combines frozen DINOv2 ViT-S/14 patch features with aligned metric depth. RGB observations have a resolution of 540×960, and DINOv2 receives resized inputs of 504×896 and returns 384-channel patch features. The geometry branch concatenates normalized depth, a local surface-normal cue, and a depth-validity mask, while the semantic branch projects the 384-channel DINOv2 feature map. Both branches form four-level pyramids with channel widths 16, 32, 64, and 128; a learned spatial gate fuses the two modalities at every level, followed by low-resolution context refinement and a skip-connected decoder. The final decoder feature is used by both the sigmoid traversability head and the contrastive objective. Excluding the frozen DINOv2 encoder, the traversability network contains 1,680,065 parameters. The Mission-A base model is trained with Adam at a learning rate of 10^{-3}, while continual updates use 10^{-4}; both use a batch size of one and

at most 15 epochs. Training runs for at least three epochs and uses early stopping with patience two and a minimum improvement of 10^{-3}. The main-paper run uses seed 1337.

The continual objective is

$$\begin{aligned}\mathcal{L} = \lambda_{\text{task}} L_{\text{task}} + \lambda_{\text{mix}} L_{\text{mix}} + \lambda_{\text{global}} L^{-}_{\text{global}} \\ + \lambda_{\text{con}} L_{\text{con}} + \lambda_{\text{out}} L_{\text{out}}.\end{aligned} \tag{B1}$$

The target weights $(\lambda_{\text{task}}, \lambda_{\text{mix}}, \lambda_{\text{global}}, \lambda_{\text{con}}, \lambda_{\text{out}})$ are $(1, 1, 1, 0.2, 1)$. The replay, contrastive, and safe-output weights increase linearly to their target values over five epochs, while the weight of L^{-}_{global} remains fixed. The reported experiments use the current-task loss weight without additional batch-count normalization. All regression losses are evaluated only on valid supervision, and the mixed-sample obstacle loss is normalized within the composited mask so that a small obstacle is not overwhelmed by the surrounding image area.

Masked regression. Let Ω contain only pixels with valid labels and finite predictions. We use the weighted masked error

$$\ell_{\Omega,w}(T, Y) = \frac{\sum_{p\in\Omega} w_p \big(T(p) - Y(p)\big)^2}{\max\{1, \sum_{p\in\Omega} w_p\}}. \tag{B2}$$

In the reported experiments, $w_p = 1$ for all valid pixels in both current-task and replay regression. Let $\Omega^{\text{mix}}_{\text{cur}}$ contain valid current-scene pixels outside the pasted obstacle, and let $\mathcal{H}$ contain valid pasted-obstacle pixels. The regression terms are

$$\begin{aligned} L_{\text{task}} &= \tfrac{1}{2}\big[\ell_{\Omega,w}(T, Y) + \ell_{\Omega^{\text{mix}}_{\text{cur}},w}(T^{\text{mix}}, Y^{\text{mix}})\big], \\ L_{\text{mix}} &= \ell_{\mathcal{H},1}(T^{\text{mix}}, Y^{\text{mix}}), \\ L^{-}_{\text{global}} &= \ell_{\Omega^-,1}(T^-, Y^-), \\ L_{\text{out}} &= \ell_{\Omega^+,1}(T^+, R^+). \end{aligned} \tag{B3}$$

Here (T^-, Y^-, Ω^-) describes a stored obstacle sample, and Ω^+ is a stored safe-region mask. R^+ uses the stored SSA label when available, otherwise the stored reference prediction, and finally the frozen teacher prediction as a fallback. Reference values are detached from the computation graph. Without a mixed sample, $L_{\text{task}} = \ell_{\Omega,w}(T, Y)$; an empty replay mask contributes zero.

Safe-feature replay. Decoder embeddings are pooled within valid regions on a 4×8 patch grid and L2-normalized. Patches with at least 25% valid coverage are used; if none meet this threshold, all patches with nonzero coverage are used. $\mathcal{A}$ contains current safe embeddings, $\mathcal{P}$ contains teacher embeddings from a selected safe-memory reference, and $\mathcal{N}$ contains embeddings of composited obstacles. For feature retention, (u_j, v_j) pairs the student and frozen-teacher embeddings at the same patch of a stored safe sample. At the start of each continual update, a copy of the current model is frozen as the teacher. The teacher is fixed for the entire update, and gradients do not propagate through its embeddings. We use cosine distance and a smooth margin with $m = 0.4$ and $\eta = 0.1$:

$$\begin{aligned} d(u, v) &= 1 - u^\top v, \qquad \ell_m(z) = \eta \log\left(1 + \exp\left(\frac{z + m}{\eta}\right)\right), \\ p^+(a) &= \arg\min_{p\in\mathcal{P}} d(a, p), \qquad h^-(a) = \arg\min_{h\in\mathcal{N}} d(a, h), \\ L_{\text{distill}} &= \frac{1}{J}\sum_{j=1}^{J} d(u_j, v_j), \\ L_{\text{sep}} &= \frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}} \ell_m\big(d(a, p^+(a)) - d(a, h^-(a))\big), \\ L_{\text{con}} &= \tfrac{1}{2}(L_{\text{distill}} + L_{\text{sep}}). \end{aligned} \tag{B4}$$

The safe reference for separation is selected by its similarity to the current safe region; the retention reference cycles through coverage and hard examples. The displayed average applies when both terms are available; otherwise only available terms are averaged. If safe teacher references are missing but current safe and obstacle patches exist, separation uses $\ell_m(-d(a, h^-(a)))$. If no term is available, $L_{\text{con}} = 0$.

Depth-aware obstacle composition. A stored obstacle is resized using the ratio of its representative depth to the depth at the target placement, with the scale clipped to $[0.85, 1.45]$. The same transformation aligns its appearance, depth, and non-traversable label. Placement candidates are scored using image-boundary coverage and estimated visibility, with target ratios of 0.85 and 0.35, respectively. Composites with fewer than 64 pasted pixels or less than 70% of the expected resized mask area are discarded. The recorded training composites use depth-supported front-view placement; the implementation preserves the object mask rather than applying dense per-pixel occlusion clipping from sparse depth. Appendix B.5 illustrates obstacle composition and the inputs to contrastive learning.

B.2 Base Trajectory-Prior Construction

PACT constructs ten state-conditioned base trajectory priors independently of the predicted traversability map. For each discretized motion state, a dynamics lattice generates control sequences within the ANYmal motion and BEV limits. Sequences that violate the heading envelope are removed, and nearby terminal states retain only their lowest-effort control sequence. The remaining terminal states are normalized by forward progress, lateral position, and relative heading and divided into ten maximin coverage groups. A central, low-effort representative from each group is cached as a base prior for that motion state. These base priors initialize the subsequent MPPI rollouts; they are distinct from the cost-ranked elite trajectories used by Planning-Aware Acquisition (PAA).

Table B1 reports the full planning configuration. GrandTour uses a forward-only unicycle abstraction of ANYmal-D and an 8-m spatial horizon, rather than a fixed temporal horizon. Each control trajectory is integrated at 0.1-s intervals and resampled to 64 equally spaced points along path length before cost evaluation and Fréchet-distance computation. For the discrete Fréchet distance, each trajectory sample is represented as $(x, y, 0.5\cos\psi, 0.5\sin\psi)$, where (x, y) is its metric position and ψ its heading. Distances are computed in this augmented space, so position and heading differences contribute in compatible metric units. The pairwise mean is defined as zero for fewer than two trajectories; the standard elite set contains ten trajectories. The ten elite slots are allocated across the base priors using a softmax of their mean traversability scores. The initial integer quotas are the floors of ten times these normalized weights, and remaining slots are assigned in descending order of the fractional remainders. Only priors with a nonzero quota generate rollouts; each active prior generates 50 perturbed rollouts, yielding at most 500 candidates.

Table B1: Base-prior lattice and online MPPI configuration.

Component	Setting
Vehicle model	Forward-only ANYmal-D unicycle
BEV grid	121×121 at 0.1 m/cell
Spatial horizon / path samples	8.0 m / 64
Integration timestep	0.1 s
Maximum speed	1.3 m/s
Maximum acceleration / deceleration	1.5 / 1.5 m/s^2
Maximum yaw rate / yaw acceleration	1.0 rad/s / 2.0 rad/s^2
Primitive length / maximum duration	1.0 m / 2.5 s
State bins (position, yaw)	0.25 m, 5°
State bins (speed, yaw rate)	0.5 m/s, 0.04 rad/s
Terminal-state bins (position, yaw)	0.5 m, 10°
Maximum heading deviation	1.25 rad
Base priors / rollouts per active prior	10 / 50
Elite trajectories / Fréchet heading radius	10 / 0.5 m

The BEV map used for scoring is denoted by T^{BEV} after small enclosed holes are interpolated; T^{fill} assigns a default traversability value to its remaining unobserved cells. Here, unobserved cells lack valid BEV predictions; they do not refer to all terrain the robot has not traversed. The default score is used for planning in these cells, not as a training label for untraversed terrain. The same filled scores are used for mean traversability of elite trajectories in the main paper; the timestep subscript is omitted here for readability. For a trajectory τ with L sampled positions, the planning cost is

$$
\begin{aligned}
C(\tau) = \frac{1}{L}\sum_{\ell=1}^{L}\left(1 - T^{\mathrm{fill}}(p_\ell)\right) \\
+ 0.25\,\frac{1}{L}\sum_{\ell=1}^{L}\mathbb{I}[p_\ell \text{ is unobserved}]\,, \qquad \text{(B5)} \\
T^{\mathrm{fill}}(p) = \begin{cases} T^{\mathrm{BEV}}(p), & T^{\mathrm{BEV}}(p) \text{ is observed}, \\ 0.35, & \text{otherwise}. \end{cases}
\end{aligned}
$$

Candidate trajectories are ranked by traversability cost and exposure to unobserved cells, with the goal-cost weight set to zero.

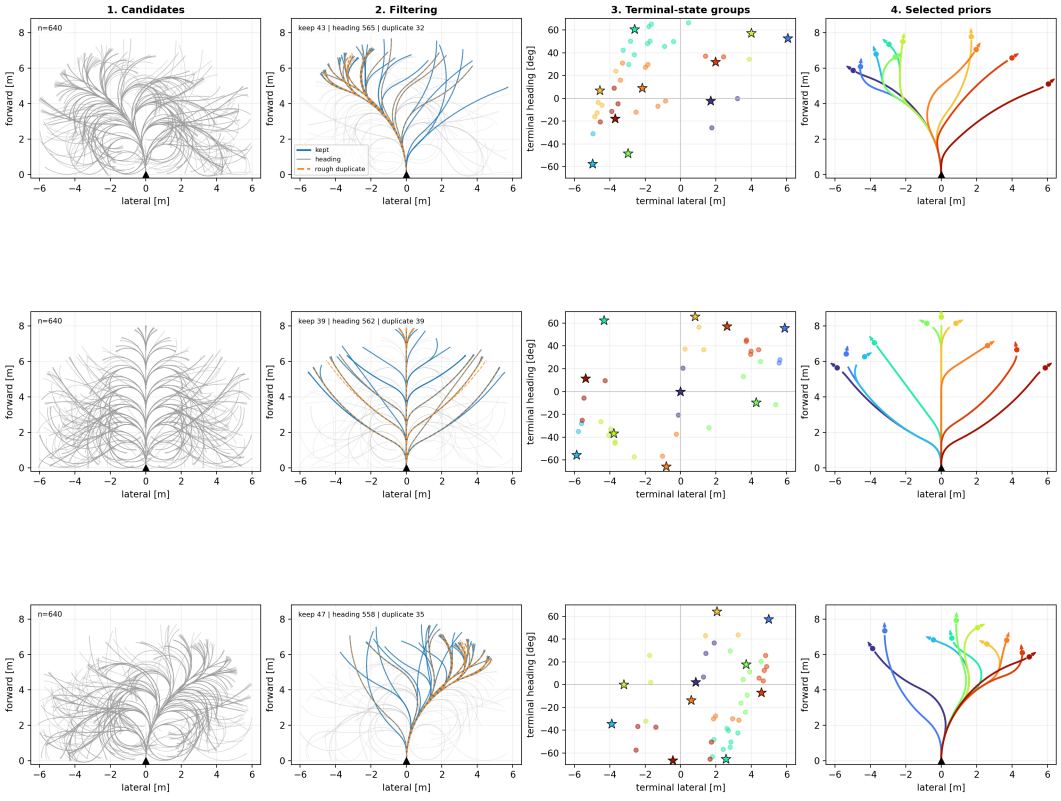

Figure B1: Construction of the ten base trajectory priors for representative initial yaw-rate states. Rows from top to bottom use initial yaw rates of -0.2, 0, and 0.2 rad/s at 1 m/s. From left to right, the procedure expands dynamically feasible candidates, filters heading-envelope violations and near-duplicate terminal states, groups the remaining terminal states to cover forward progress, lateral position, and relative heading, and retains one low-effort representative per group. The resulting base priors are independent of traversability predictions and should not be confused with the online cost-ranked elite trajectory set.

B.3 Acquisition, Region Diagnosis, and Memory Settings

Table B2 lists the settings used by PAA and Counterfactual Asymmetric Replay (CAR). PAA triggers when the elite-trajectory Fréchet divergence and the mean traversability along the elite trajectories pass their respective criteria. A short temporal filter suppresses isolated events and a minimum time gap prevents near-duplicate acquisitions. Additional sampling of safe terrain provides SSA-labeled regions from observations for which the planning trigger is inactive. The reported run considers all temporally filtered, eligible non-trigger observations rather than capping this branch at a fraction of the trigger count. No additional minimum safe-area fraction is imposed at this frame-selection stage; valid safe labels are still required for safe-region supervision.

Table B2: PACT acquisition, diagnosis, and clustering settings.

Component	Setting
Elite-trajectory Fréchet threshold	3.2 m
Elite-trajectory mean-traversability threshold	0.70
Temporal persistence	2 hits in a 3-frame window
Minimum acquisition spacing	3.0 s
Non-trigger sampling	All eligible, temporally filtered frames
Scene cosine-distance threshold	0.20
Strong scene-novelty distance	0.55
Scene persistence count	2
Maximum pending interval	10 s
Obstacle appearance cosine-distance threshold	0.50
Obstacle geometry JS-divergence threshold	0.40
Within-cluster duplicate similarity	0.98
Maximum fraction occupied by one cluster	0.30

SAM2 masks are first classified as object, terrain, or unresolved regions using depth-derived geometry. Ambiguous regions are further classified using visual similarity and image-position cues, while sky regions are discarded. Only regions classified as objects can enter the obstacle memory; a favorable counterfactual planning response alone does not assign an obstacle label. Unlabeled terrain is not automatically assigned a safe or non-traversable label. For each candidate region, counterfactual diagnosis changes only the region-specific traversability cost and keeps the vehicle state, candidate controls, remaining costs, and planner parameters fixed. A strong candidate has sufficient depth and path support and reduces elite-trajectory divergence below the trigger threshold. If no candidate reduces divergence below the threshold, PACT retains at most two path-supported candidates that meet the minimum absolute and relative divergence-reduction criteria. Table B3 gives the corresponding thresholds.

Table B3: Counterfactual region-routing settings.

Criterion	Value
Minimum valid-depth ratio	0.50
Minimum changed BEV cells	100
BEV change tolerance	0.05
Minimum path-support ratio	0.03
Minimum absolute divergence reduction	0.10 m
Minimum relative divergence reduction	0.02
Maximum retained partial-support objects per frame	2

For a binary front-view mask, the counterfactual map is formed by element-wise replacement with $T_{\text{safe}} = 0.8$ and projected using the same depth and camera geometry as the original map. Let $\mathcal{C}_i$ contain BEV cells with finite scores in both maps whose absolute change exceeds 0.05, and let $\widetilde{\mathcal{E}}_t^{(i)}$ be the reselected elite set. Path support counts sampled trajectory points rather than distinct visited

cells:

$$s_i(\mathcal{E}) = \frac{\sum_{\tau\in\mathcal{E}} \sum_{p\in\tau} \mathbf{1}[\mathrm{cell}(p) \in \mathcal{C}_i]}{\sum_{\tau\in\mathcal{E}} |\tau|},$$
$$s_i^{\mathrm{path}} = \max\{s_i(\mathcal{E}_t), s_i(\widetilde{\mathcal{E}}_t^{(i)})\}. \tag{B6}$$

Repeated visits to a cell are counted separately. After the depth, path-support, and divergence criteria determine admission, the candidate priority uses the divergence reduction and the magnitude of the traversability change:

$$q_i = \begin{cases} \min\{1,\ 0.75 + 0.08[\Delta d_i^{\mathrm{traj}}]_+ + 0.15|\Delta\bar{T}_i|\}, & \text{strong,} \\ \min\{0.70,\ 0.35 + 0.06[\Delta d_i^{\mathrm{traj}}]_+ + 0.20|\Delta\bar{T}_i|\}, & \text{partial.} \end{cases} \tag{B7}$$

Here $[x]_+ = \max(0, x)$. A positive traversability change is not an additional admission requirement. This candidate priority score contributes to storage priority and provides a fallback replay priority; later replay priorities can reflect prediction errors on stored obstacles.

The 4% object-area threshold applies to object regions used for adaptation supervision, not to counterfactual diagnosis or obstacle-memory admission. Obstacles selected for replay are exempt from this area threshold.

Before adaptation-buffer admission, PACT reevaluates the planning trigger for each verified obstacle intervention using the original divergence and traversability thresholds. If any such intervention resolves the trigger, the frame is excluded from adaptation while obstacle-memory admission is preserved; frames with unresolved triggers remain eligible.

PACT allocates 64 entries to the adaptation buffer, which holds selected samples that still require current-terrain learning. The obstacle memory holds 64 entries, while the safe-terrain memory reserves 64 entries for safe regions from triggered observations and 64 for those from complementary non-trigger observations, for 192 long-term entries in total. The main run uses 550 of the 2,893 eligible frames for adaptation or reference construction. This is approximately 81.0% fewer frames used for adaptation and replay. Selection still requires processing the input stream; the count does not measure observed frames, inference cost, simultaneous memory occupancy, or the number of gradient updates. We report the adaptation-buffer capacity separately from the long-term replay capacity because the buffer holds samples for current updates rather than long-term replay.

Adaptation-buffer entries are optimized in batches of at most 32, with the oldest pending entry always included and the remaining entries selected by cluster-balanced priority. After an update, a selected frame remains in the adaptation buffer when its safe-region RMSE exceeds 0.16 or its safe recall is below 0.80. If safe-region RMSE improves by less than 2% for two successive evaluations, the entry is terminated as a learning plateau rather than being retried indefinitely. An entry that satisfies both safe-region criteria contributes only its SSA-verified safe region to the corresponding partition of the safe-terrain memory; its original frame-level entry is then removed from the adaptation buffer.

For admission to the safe-terrain and obstacle memories, candidates with descriptor cosine similarity at least 0.98 within a cluster are treated as duplicates and only the higher-valued entry is retained. Scene clusters for the adaptation buffer and safe-terrain memory use cosine distance between masked DINOv2 descriptors. Obstacle clusters require both appearance cosine distance at most 0.50 and geometry-distribution Jensen–Shannon divergence at most 0.40, preventing visually similar objects with substantially different depth and extent from collapsing into one replay group. When a bank is full, replacement is drawn preferentially from an overrepresented cluster while preserving at least two entries per cluster whenever capacity permits; the most redundant lower-priority entry is removed only if the incoming candidate has a higher value. Safe-terrain replay combines nearest appearance matches, underused cluster coverage, and entries with high prediction drift, whereas obstacle replay cycles through clusters and prioritizes entries with false-safe predictions.

B.4 Counterfactual Diagnostic Examples

Figure 2 in the main paper visualizes two stored diagnostic cases from Mission B. The selected region is assigned traversability 0.8 while the candidate trajectory set is held fixed; costs and elite

selection are then recomputed. In the first case, elite-trajectory divergence decreases from 3.83 m to 3.17 m, crossing the 3.2 m threshold. In the second, it decreases only from 3.31 m to 3.23 m and remains above the threshold. These examples illustrate different planner responses to region-specific changes; threshold crossing alone is not the complete memory-admission test, which also checks depth and path support.

Orange highlights the candidate region, and blue and red indicate low and high BEV traversability on a shared 0–1 scale. Dark curves with white outlines show the original and reselected elite trajectories; unobserved cells are light gray.

B.5 Asymmetric Replay Examples

Figure B2 illustrates the different uses of obstacle and safe-terrain memories with a reconstruction from saved observations and region masks.

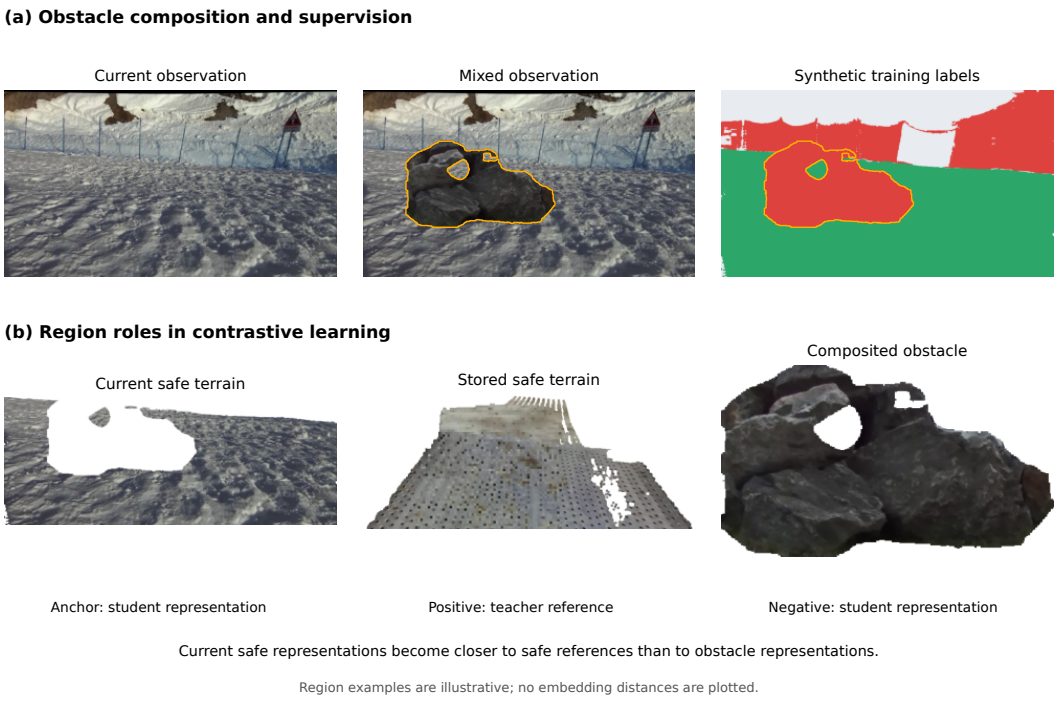

Figure B2: Illustrative asymmetric replay reconstructed from saved regions. Top: current observation, mixed observation, and synthetic labels; orange outlines mark the pasted obstacle, green denotes safe supervision, red denotes non-traversable supervision, and gray denotes unlabeled pixels. Bottom: current safe, stored safe, and composited obstacle regions illustrate anchor, positive, and negative roles. The panels illustrate sample construction and loss inputs, not a recorded training batch or an actual nearest-reference retrieval.

Depth determines the obstacle scale and placement; in this example, a source depth of 4.294 m and a target depth of 3.018 m give a scale factor of 1.423. Pasted pixels receive non-traversable labels, while the RGB values, labels, and validity mask outside the pasted region remain unchanged. Gray pixels have no supervision and are excluded from regression.

The lower row shows the region roles in the contrastive term: current safe embeddings are anchors, teacher embeddings from safe memory provide positive references, and embeddings of composited obstacles provide negatives. Thus, an obstacle contributes both to regression on the mixed observation and to contrastive separation from safe terrain. Safe-memory regions are not pasted into the current scene; their teacher representations provide references across observations, while feature matching additionally retains the stored safe-region representations.

B.6 CONTINUAL UPDATE PROCEDURE

Algorithm B1 summarizes the interaction between acquisition, diagnosis, memory admission, and learning. Region diagnosis determines whether a triggered observation remains eligible for adaptation; it does not replace the geometric criteria for obstacle labels or the SSA labels for safe terrain. Admissions that require SSA are deferred until those labels become available. The procedure uses the filtering, memory, and stopping rules in Appendix B.3 and the loss definitions in Appendix B.1.

Algorithm B1 PACT acquisition, memory admission, and continual updates.

Input: source-trained model M_{base}, observation stream, delayed SSA labels, and fixed-capacity adaptation and replay memories.

1. Initialize the current model from M_{base} and initialize the adaptation buffer, obstacle memory, and safe-terrain memory.
2. At each eligible BEV update, predict traversability, score candidate trajectories, and compute elite-trajectory divergence and mean traversability.
3. Apply the planning trigger, temporal persistence, and minimum acquisition interval.
4. **For a triggered observation:**
   (a) Generate candidate obstacle regions using SAM2 masks and geometric and visual cues.
   (b) Treat each candidate as traversable, reselect elite trajectories under otherwise fixed planning conditions, and measure the changes in divergence and traversability.
   (c) Admit verified obstacles that satisfy the depth, path-support, and divergence criteria to obstacle memory, subject to capacity and duplicate rules.
   (d) Reevaluate the full planning trigger under the verified obstacle interventions. If any intervention resolves the trigger, exclude the observation from adaptation; otherwise retain it for adaptation once SSA labels are available.
5. **For an eligible non-trigger observation:** retain its SSA-confirmed safe regions as complementary safe references when the labels become available, without region diagnosis.
6. **When a continual update is scheduled:**
   (a) Freeze a copy of the current model as the teacher, and select adaptation samples and available obstacle and safe-memory references.
   (b) Composite replayed obstacles into adaptation observations using depth-consistent placement, with aligned labels and validity masks.
   (c) Optimize the continual objective using valid current labels, obstacle regression in mixed and original observations, safe-feature matching and contrastive separation, and safe-output supervision. Apply only available loss terms.
   (d) Apply the stopping criteria. Transfer learned safe regions from the adaptation buffer to safe memory; retain entries that still need learning, and remove entries terminated by the learning-plateau rule.
7. Maintain fixed capacities with duplicate removal and cluster-based replacement; repeat as new observations and labels arrive.

C EVALUATION PROTOCOLS AND ADDITIONAL QUALITATIVE RESULTS

C.1 BASELINE IMPLEMENTATIONS

We implement the baselines on a shared feature-extraction backbone to compare their sample-selection and knowledge-retention policies (Table C1). All baselines use the same base checkpoint, task order, inputs, supervision, learning rate, and training seed (1337) as PACT. Shared supervision refers to the supplied SSA labels and validity masks, while each method retains its own learning and replay objectives. We match long-term replay capacity at 192 entries for PACT, Reservoir, and

IMOST. Appendix B.3 specifies PACT's memory configuration, and Table C2 reports the measured replay storage.

Table C1: Baseline adaptation and memory configurations.

Method	Adaptation mechanism	Memory setting
CM	Learns from every valid frame observed so far	Unbounded full-memory reference
Reservoir	Uniform stream sampling and replay	Global reservoir, 192 frames
WVN	Fast adaptation on the current mission	No replay memory
LwF	Current-mission learning with output distillation	No frame replay
IMOST	Frozen-feature scene clustering and replay	192 entries; cluster capacity 20
PACT	Planning-aware acquisition and asymmetric replay	192 long-term entries; see Appendix B.3

The recorded baseline runs use Adam with learning rate 10^{-4}, batch size 1, and at most 15 epochs per stage, with a minimum of 3 epochs, patience 2, and minimum improvement 10^{-3}. IMOST uses an expansion threshold of 0.10. LwF uses binary-KL output distillation with weight 1. WVN selects a trained checkpoint using current-mission validation RMSE, whereas the other baselines use validation across the missions seen so far. The shared policy first checks that relative RMSE increases on prior missions do not exceed 10% and hazard-recall drops on any seen mission do not exceed 0.03, measured against the stage-input checkpoint. Among eligible trained checkpoints, it minimizes mean validation RMSE; candidates within 0.001 are ranked by worst-mission RMSE and then class-averaged F1. When a trained checkpoint is required but none passes these checks, the same ranking is applied to the trained candidates without the checks.

WVN initializes each stage from the preceding checkpoint; the LwF distillation teacher is a frozen copy of that checkpoint. These baselines train on every valid current adaptation frame, irrespective of the subset retained for future replay.

Replay storage. Table C2 reports serialized storage after Mission H for the three methods with bounded replay memories. For PACT, we count files belonging to active indexed entries; for Reservoir and IMOST, we include the feature, depth, and label files referenced by retained frames rather than counting only their path records.

Replay storage includes the model inputs, supervision, reference data, and memory-management metadata required for replay. Auxiliary storage includes accompanying source and intermediate files that are retained or loaded but not used directly in replay.

Table C2: Final replay storage in MiB ($1\,\mathrm{MiB} = 2^{20}$ bytes).

Method	Entries	Replay	Auxiliary	Total
PACT	192	861.62	495.85	1357.47
Reservoir	192	1264.48	198.33	1462.81
IMOST	192	1268.47	196.13	1464.60

At the same number of long-term entries, PACT requires less serialized replay storage than Reservoir and IMOST, both for replay data alone and when auxiliary files are included.

Matched-budget component ablation. For the 2×2 acquisition–replay ablation in the main paper, all four variants use the same 550-frame budget. The acquisition comparison replaces the IMOST-style feature-distribution criterion with PACT's planning trigger and complementary sampling of safe terrain from non-trigger observations. The replay comparison uses standard full-frame replay or CAR. The per-mission allocation for B–H is 105, 94, 64, 43, 86, 101, and 57 frames, respectively. This allocation is shared by both acquisition policies and both replay strategies; it counts used samples rather than simultaneous replay-memory occupancy.

C.2 Additional Qualitative Traversability Results

Figure C1 compares thresholded safe-region predictions on four held-out observations from Missions A, C, E, and H. The examples complement the aggregate metrics by showing how much of the self-supervised traversable region is recovered after sequential adaptation.

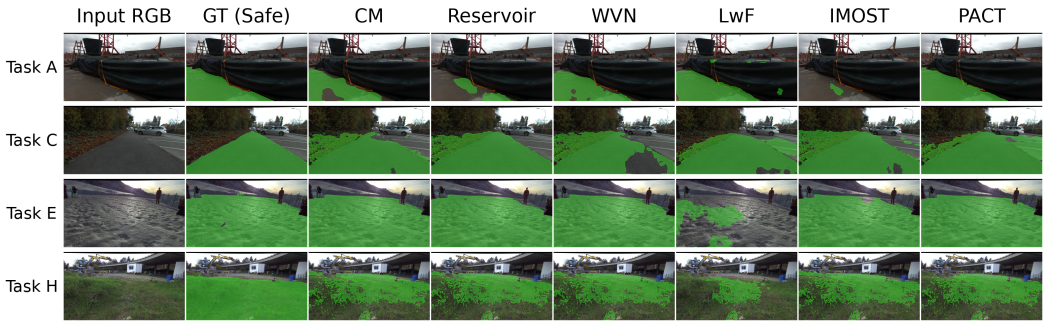

Figure C1: Qualitative safe-region coverage on held-out observations from Missions A, C, E, and H. Green denotes traversability scores of at least 0.6. These examples illustrate safe-region coverage, not safe–hazard discrimination.

C.3 Offline Global-Planning Protocol

Table C3: Offline global-planning protocol.

Component	Setting
Evaluation missions	B–H held-out test subsets
Queries per mission (B/C/D/E/F/G/H)	30/30/9/30/30/30/30
Start–goal distance	10–15 m
Query policy	Route containment with geometric diversity
Endpoint candidate spacing	1 m
Grid resolution	0.2 m
Planner	8-connected A*
Lethal / hysteresis thresholds	0.60 / 0.57
Traversability / clearance weights	4 / 4
Clearance decay	0.6 m
Robot footprint lateral half-width	0.28 m
Geometric-obstacle height and support	0.25 m above ground; at least 2 hits
Prediction-fusion residual and support	0.25 m; at least 2 elevated observations

The global-planning evaluation uses held-out logged geometry and odometry from Missions B–H. We generate start–goal queries from the logged route and geometry without consulting model predictions or evaluation ground truth. Every method receives exactly the same 189 queries and the same fixed A* planner; only the predicted traversability cost map changes across methods. The fixed evaluation ground truth is used only to score the returned path, not to construct a query or guide search. A query succeeds when A* returns a path whose endpoint falls within the goal tolerance and whose full robot footprint avoids both geometric obstacles and fixed-ground-truth hazards. No-path and collision outcomes are both counted as failures in the main paper.

C.4 Planning Success and Failure Examples

Figure C2 adds two full-scene query comparisons.

Figures 4 and C3 compare all six methods on held-out Mission D and F queries, respectively. On Mission D, CM's footprint overlaps a hazard despite a clear centerline, Reservoir's centerline intersects a hazard, and PACT maintains body clearance. On Mission F, PACT fails while CM and

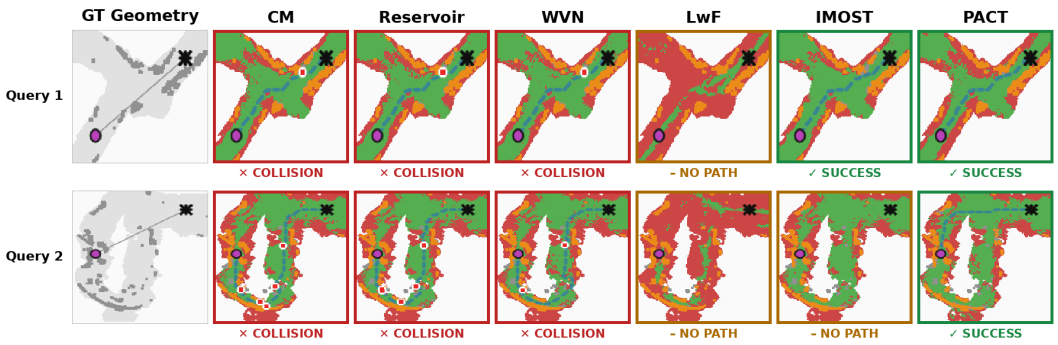

Figure C2: Additional offline planning results on two held-out queries using the same geometry and planner across methods.

Reservoir succeed. Collision checks use a heading-aligned $0.84\,\mathrm{m} \times 0.56\,\mathrm{m}$ footprint, so a clear centerline alone does not ensure success. Magenta cells show footprint–hazard intersections, crosses mark contacts, and light-blue bands indicate swept footprints. The bottom row of Figure C3 enlarges the dashed-box area; blue rectangles show footprints at the highlighted poses. Red/green backgrounds indicate low/high traversability; dark-gray evaluation hazards are overlaid only for interpretation. Blue curves are paths, circles are starts, and stars are goals.

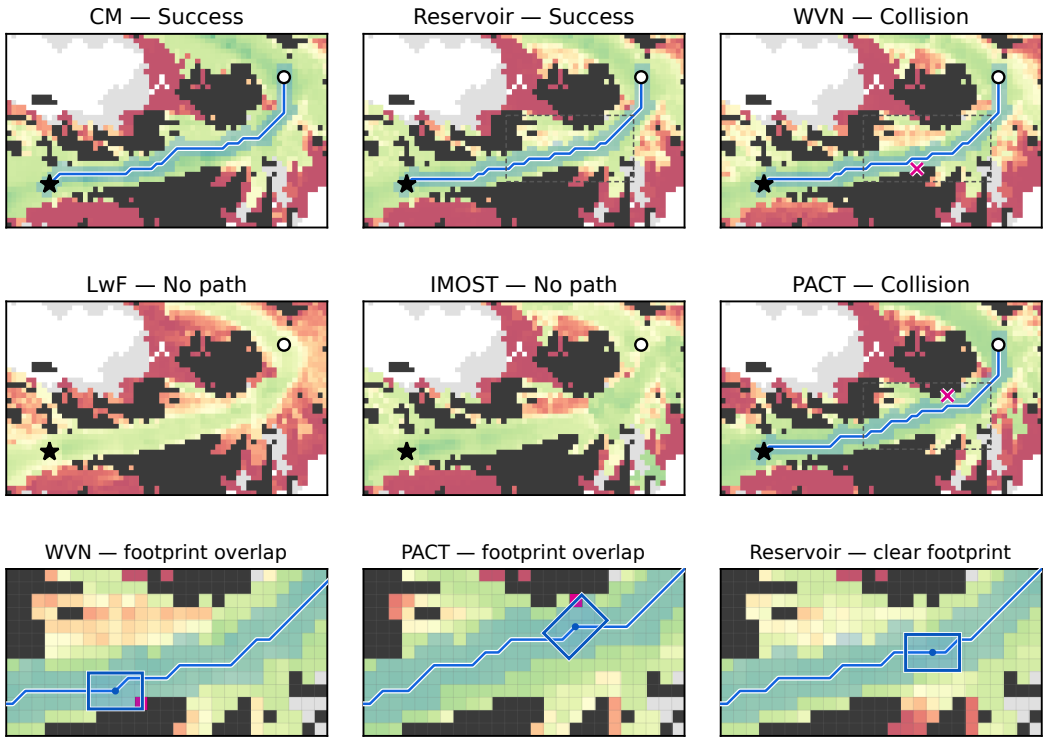

Figure C3: Mission F failure case. WVN and PACT have clear centerlines but collide under footprint evaluation; PACT's rotated body corner overlaps a geometric-hazard cell. CM and Reservoir succeed, with Reservoir shown as the close-up comparator in the same area. LwF and IMOST return no path. These examples complement the full-query evaluation.

D Robustness and Sensitivity Analyses

D.1 Seed Robustness

We repeat model training with seeds 1337, 2027, and 3407. The values below are the mean and population standard deviation over the three seeds; seed 1337 is the main run reported in the main tables. Table D1 shows that PACT achieves the best mean result for all five traversability prediction and forgetting metrics. Its RMSE and Safe IoU vary little across the tested seeds, indicating consistent prediction accuracy and safe-region coverage rather than a gain confined to one training run.

Table D1: Traversability prediction and forgetting over three training seeds (mean ± population standard deviation).

Method	RMSE ↓	AUROC ↑	Safe IoU ↑	Bal. Acc. ↑	Forgetting ↓
CM	0.1168 ±0.0016	0.9887 ±0.0014	0.9520 ±0.0043	0.9608 ±0.0017	0.0155 ±0.0028
Reservoir	0.1185 ±0.0023	0.9921 ±0.0016	0.9346 ±0.0099	0.9553 ±0.0046	0.0158 ±0.0034
WVN	0.1235 ±0.0039	0.9872 ±0.0015	0.9088 ±0.0138	0.9404 ±0.0062	0.0256 ±0.0057
LwF	0.1968 ±0.0039	0.9522 ±0.0180	0.5413 ±0.0198	0.7567 ±0.0038	0.0137 ±0.0059
IMOST	0.1346 ±0.0023	0.9850 ±0.0011	0.8832 ±0.0124	0.9260 ±0.0053	0.0272 ±0.0053
PACT	**0.1086 ±0.0007**	**0.9932 ±0.0015**	**0.9694 ±0.0008**	**0.9687 ±0.0005**	**0.0038 ±0.0016**

Table D2 reports the corresponding safe–hazard discrimination results and the main-run margins for seed 1337. PACT's main-run margin is 0.6654, exceeding CM, the strongest baseline on this metric, by 0.0687. PACT achieves the largest mean safe–hazard margin and the highest paired consistency across training seeds. The high consistency and small variation show that PACT reliably identifies both regions in visually similar safe–hazard pairs across the tested training runs.

Table D2: Safe–hazard discrimination for the main run and across three training seeds.

	Main run (1337)	Three seeds (mean ± std.)	
Method	Margin ↑	Consistency@0.6 ↑	Margin ↑
CM	0.5967	$88.50 \pm 0.54\%$	0.6155 ± 0.0139
Reservoir	0.5927	$84.96 \pm 1.83\%$	0.5987 ± 0.0043
WVN	0.5759	$84.29 \pm 1.43\%$	0.5936 ± 0.0126
LwF	0.4801	$31.83 \pm 4.71\%$	0.4833 ± 0.0076
IMOST	0.5664	$79.12 \pm 1.99\%$	0.5736 ± 0.0071
PACT	**0.6654**	$\mathbf{95.17 \pm 0.39\%}$	$\mathbf{0.6597 \pm 0.0059}$

For planning robustness, we separately vary the model-training seed while fixing the query bank and vary the query-bank seed while fixing the seed-1337 model (Figure D1).

PACT achieves the highest or joint-highest planning success for every training seed, with query-seed results ranging from 93.81% to 94.76%. Mean planning success is $91.11 \pm 2.92\%$ under training-seed variation and $94.13 \pm 0.45\%$ under query-seed variation. PACT therefore retains its planning advantage when either the trained model or the query set changes, although training randomness produces greater variation in success than query selection in these tests.

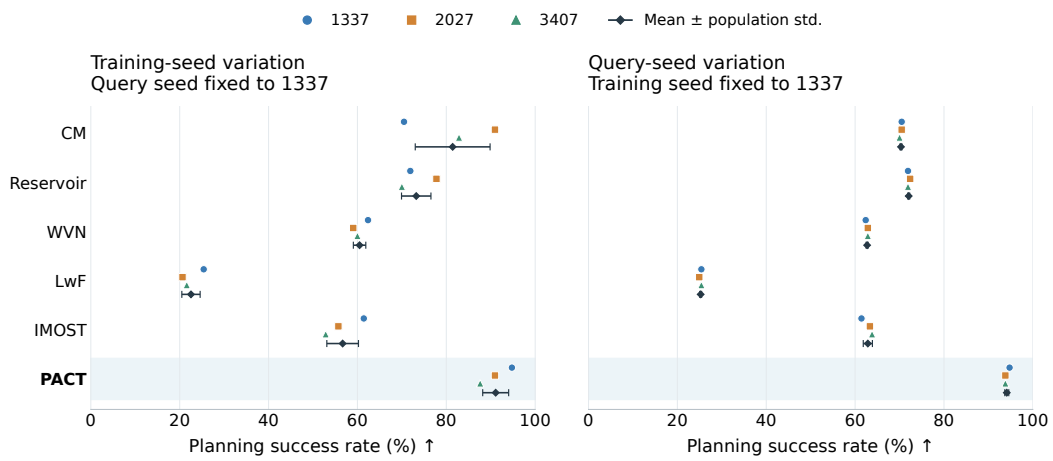

Figure D1: Planning success under training-seed and query-seed variation. Colored markers show individual seeds; diamonds and horizontal bars show the mean and population standard deviation. The other seed is fixed to 1337 in each panel; both panels use the same horizontal scale.

D.2 Task-Order Sensitivity

We compare the original target-mission order (A→B→C→D→E→F→G→H) with two alternative orders: reverse (A→H→G→F→E→D→C→B) and a permutation fixed before observing outcomes (A→D→H→E→B→F→C→G). All three orders use training seeds 1337, 2027, and 3407. The planning query bank is fixed to seed 1337 so that only training randomness and task order vary. For each order, forgetting is averaged over all missions except the final mission. Figure D2 compares the mean and population standard deviation over the three seeds for each order.

PACT attains the highest mean planning success in all three orders; under the shuffled order, it also obtains the lowest RMSE and forgetting. Under the reverse order, Reservoir has lower RMSE but lower planning success than PACT, showing that lower average prediction error does not necessarily translate into better planning performance.

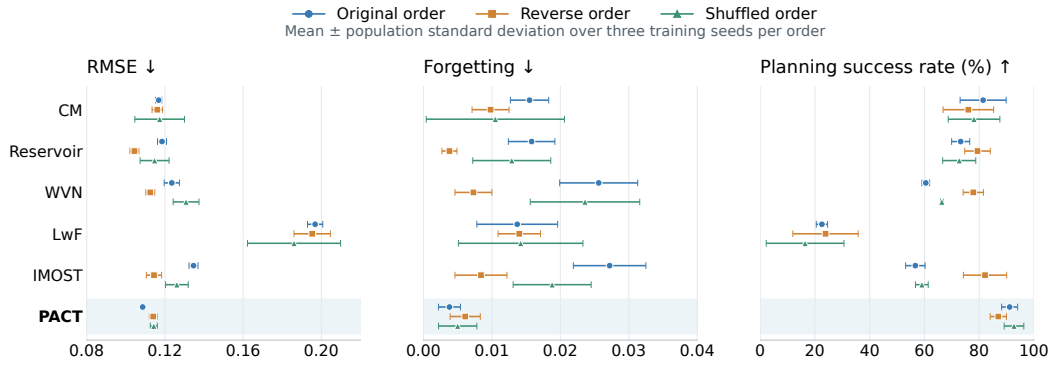

Figure D2: Performance under original, reverse, and shuffled target-mission orders. Markers and horizontal bars show the mean and population standard deviation over three training seeds per order, with the query seed fixed to 1337. Lower RMSE and forgetting, and higher planning success, are better.

Aggregating the six reverse- and shuffled-order training runs, Reservoir obtains the lowest final RMSE, while PACT obtains the lowest forgetting and the highest planning success (Table D3). PACT also has the smallest RMSE standard deviation, indicating the most stable final prediction error across the evaluated order and seed changes. These two alternative orders provide evidence of robustness but do not establish complete task-order invariance.

Table D3: Aggregate performance over the reverse and shuffled orders, with three training seeds per order.

Method	RMSE ↓	Forgetting ↓	Planning Success (%) ↑
CM	0.1167 ± 0.0092	0.0102 ± 0.0074	77.06 ± 9.40
Reservoir	$\mathbf{0.1096 \pm 0.0075}$	0.0083 ± 0.0061	76.03 ± 6.35
WVN	0.1216 ± 0.0104	0.0154 ± 0.0101	72.09 ± 6.32
LwF	0.1907 ± 0.0186	0.0141 ± 0.0068	20.13 ± 13.64
IMOST	0.1203 ± 0.0076	0.0136 ± 0.0071	70.58 ± 12.92
PACT	0.1142 ± 0.0019	$\mathbf{0.0056 \pm 0.0025}$	$\mathbf{89.84 \pm 4.36}$

D.3 Mission-Level Learning Dynamics

Figure D3 reports mission-level test RMSE after each adaptation stage for the seed-1337 runs used in Table 1. Each column is a checkpoint after adaptation through the indicated mission, and each row is a fixed test mission. Reading a row from left to right shows whether prediction error on that mission rises or falls during later adaptation. Gray cells denote missions not yet encountered; the first recorded evaluation is after Mission B, so both A and B start in that column. All methods share one color scale, and the final columns reproduce the mean RMSE values in the main table. The same records also reproduce its forgetting values using the definition in Appendix A.1.

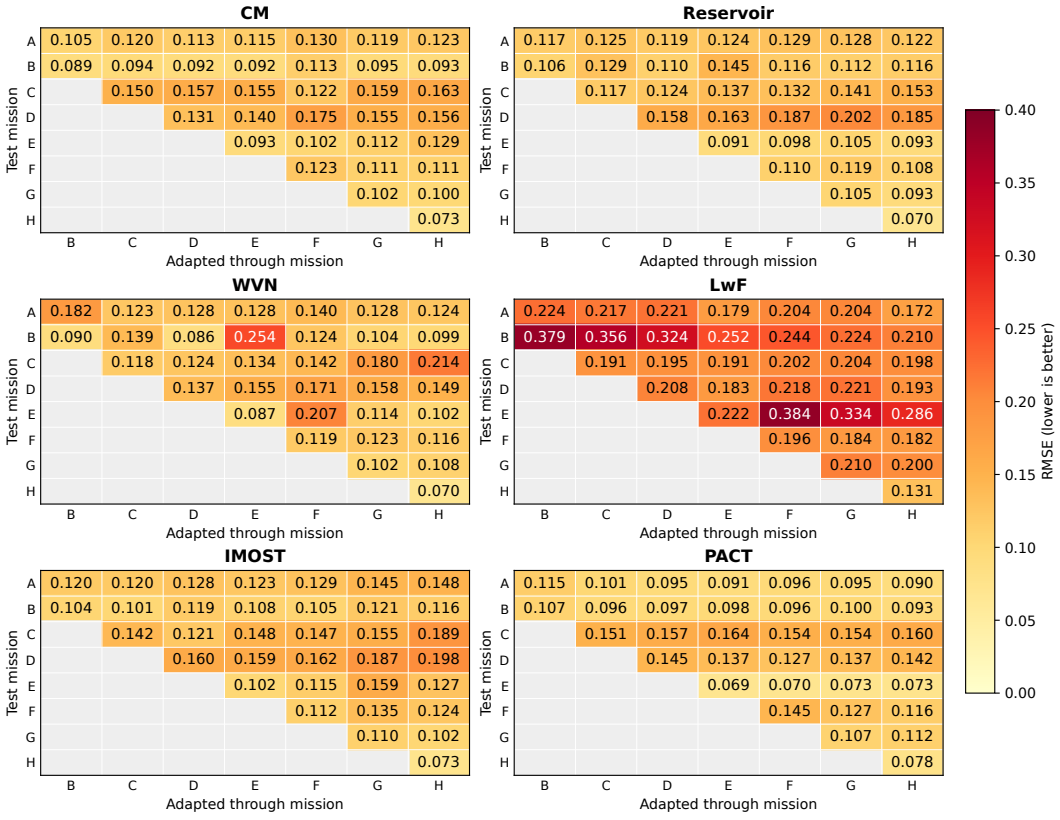

Figure D3: Mission-level test RMSE throughout sequential adaptation (seed 1337). Columns indicate the most recently learned mission; rows indicate the evaluated mission. Lower values are better, and all methods use the same color scale. Gray cells correspond to missions not yet encountered. Missions A and B are first evaluated after B.

The matrices show how final performance arises from changes across the stream rather than from a single endpoint comparison. For PACT, Mission-A RMSE decreases from 0.115 after Mission B to 0.090 after Mission H, and Mission-F RMSE decreases from 0.145 at its introduction to 0.116

after Mission H. These examples show that performance on an earlier mission can improve during subsequent adaptation, although error does not decrease at every stage or on every mission. The LwF matrix also shows why retention should be interpreted together with absolute prediction error: modest final forgetting can coexist with high error throughout the sequence. PACT combines the lowest final mean RMSE with the lowest average forgetting in these runs.

E Computational Efficiency

E.1 Planning-Backend Runtime

We benchmark the planning backend on an NVIDIA GeForce RTX 5080 using 100 BEV updates from the Mission C adaptation stream. CUDA provides a 37.53× mean speedup while preserving elite indices and both raw and temporally filtered trigger decisions across all 100 updates (Table E1). This benchmark measures the planning backend only; it excludes RGB preprocessing, DINOv2 feature extraction, traversability inference, SSA generation, counterfactual diagnosis, and model optimization, and therefore must not be interpreted as end-to-end system latency.

Table E1: Planning-backend runtime over 100 BEV updates.

Statistic	CPU	CUDA
Mean latency	2.3856 s	63.56 ms
Median latency	2.3485 s	62.76 ms
p95 latency	2.8431 s	78.62 ms
p99 latency	2.8617 s	85.98 ms
Peak allocated GPU memory	–	524 MiB
Peak reserved GPU memory	–	584 MiB
Elite-index mismatch	–	0/100
Trigger mismatch	–	0/100

E.2 Recorded RGB-D Processing Latency

We additionally measure the serial processing pipeline from recorded RGB-D input through fresh DINOv2 extraction, traversability prediction, temporal BEV accumulation, and CUDA planning on the RTX 5080. The timing environment uses Python 3.10.18, PyTorch 2.7.1 with CUDA 12.8, NumPy 1.26.4, and OpenCV 4.7.0, with one PyTorch CPU thread. Ten initial frames and the first planning update are excluded as warmup, leaving 734 ordinary frames and 100 frames that also refresh the BEV and run planning. Table E2 separates these cases because planning is invoked at BEV-update intervals rather than on every input frame.

Table E2: Serial recorded-input latency. Times are in milliseconds. Component rows are not mutually exclusive with complete-frame rows.

Measured scope	N	Mean	p95
RGB preprocessing + fresh DINOv2	834	31.35	31.54
Predictor + depth preparation	834	13.11	13.41
Prior retrieval and construction	100	43.21	65.10
CUDA MPPI / Fréchet divergence	100	61.25	76.25
Complete ordinary frame	734	56.77	59.46
Complete BEV/planning frame	100	468.33	610.30

Complete-frame measurements include recorded-input loading and projection/BEV processing. BEV/planning frames have a median latency of 451.46 ms and a maximum of 666.22 ms; 22 of 100 exceed the configured 500 ms BEV-update interval. This recorded-input benchmark excludes sensor acquisition, stereo-depth generation, synchronization, SSA, event processing, and concurrent learning.

E.3 Region Extraction and Counterfactual Evaluation Time for Trigger Frames

We measure SAM2 mask extraction, subsequent mask partitioning, and counterfactual evaluation separately on three preserved trigger frames from Missions B, C, and H. SAM2 uses 48 points per side, prediction-IoU and stability thresholds of 0.62, and the small Hiera checkpoint. The measurements exclude model initialization, one SAM2 warmup, and one counterfactual warmup per frame. Timing is collected without xFormers or the optional SAM2 CUDA post-processing extension; the latter post-processing step is skipped in this environment.

Table E3: Region extraction and counterfactual evaluation time for trigger frames, in seconds. These three illustrative frames are not an estimate of tail latency over the full stream.

Mission	SAM2	Partitioning	Objects	Diagnosis total
B	3.164	0.812	11	0.410
C	4.126	0.467	11	0.374
H	3.707	0.442	10	0.332

Across the 32 stored object candidates, counterfactual evaluation averages 34.88 ms per object, including region-specific BEV modification, candidate rescoring, elite reselection, and divergence calculation. For all three measured frames, SAM2 extraction takes substantially longer than counterfactual evaluation and dominates the measured region-extraction and diagnosis time. Rebuilding a temporal context from recorded data averages 2.04 s and is reported separately from per-object evaluation. SAM2 extraction is timed anew, whereas diagnosis uses preserved object masks and a context rebuilt with the final predictor. SSA generation, object filtering and artifact I/O, memory routing, and optimization are outside this component benchmark. These event costs apply to trigger frames, not every input frame.

E.4 Optimizer-Step Latency

We reconstruct a representative adaptation batch from a Mission H observation and the safe-terrain and obstacle memories, using the depth-supported placement and obstacle-compositing routines. The timing batch uses the default depth-supported placement without an additional planning-corridor restriction. With batch size 1 per branch and full objective weights, each measured step evaluates all five losses through four student forwards, two frozen-teacher forwards, backpropagation, and Adam. After five warmup steps, 30 repetitions of this fixed batch have mean, median, and p95 latency of 223.85, 223.77, and 224.67 ms, respectively; peak allocated GPU memory is 7,098 MiB. This measures the optimizer core, not a complete continual-learning update: sample retrieval, synthesis and DINOv2 preparation, disk loading, validation, early stopping, memory updates, and checkpoint publication are excluded.