REMEMBERING IS NOT NAVIGATING: GOAL RECURRENCE AND CLOSED-LOOP SUCCESS IN CITY-SCALE DRONE NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language navigation agents are increasingly fine-tuned on the benchmarks that rank them, and a higher closed-loop success rate is read as evidence that an agent has learned to navigate. Success can also come from memory: if the places an agent is sent to also appear in its training data, recalling where a description led is enough. EmbodiedNav-Bench, a city-scale drone benchmark, is exposed to this risk: it ships without a train/test split, so each fine-tuning study draws its own, and nothing checks what the held-out goals share with training. We measure how much success memory alone can buy there. Holding out every tenth episode, we find that held-out goals are rarely new: for 60% of them, one training episode has both a near-duplicate instruction and a goal inside the success radius. On 101 held-out episodes flown in the benchmark's simulator under our harness, a controller that flies from the start pose to the remembered endpoint of the matching training instruction succeeds on 83% of the episodes it flies from memory and on 58% overall, with a learned policy acting on the rest, against 22% for an imitation-learned policy and 32% for always flying forward; without a near-duplicate instruction or a nearby training goal, it does no better than flying forward. Gains from training fixes need the same scrutiny: an offline exposure-bias correction with geometry-verified preference optimization reduces the harm of the agent's own action history in offline next-action accuracy, yet same-budget controls without the preference signal give reductions we cannot distinguish from it, and its closed-loop effect is untested. We release goal-disjoint episode lists and recommend reporting success on them beside a retrieval baseline.

1 INTRODUCTION

Vision-language navigation (VLN) agents are ranked by closed-loop success: an agent follows an instruction in a simulator and succeeds if it ends near the goal. As agents are increasingly fine-tuned on the benchmarks that rank them, higher success is read as evidence of having learned to navigate. Success can also come from memory. If the goals an agent is sent to also appear in its training data, recalling where a description led is enough, and success rate cannot tell the two apart. Indoor VLN guards against this with held-out environments and unimodal baselines (Anderson et al., 2018; Thomason et al., 2019), and aerial benchmarks such as AerialVLN (Liu et al., 2023) and OpenFly (Gao et al., 2026) hold out whole scenes. EmbodiedNav-Bench (Zhao et al., 2026) renders one persistent city and is released as a single evaluation set, which suits the zero-shot evaluation it reports. Once part of it is used for training, each study draws its own split, and nothing checks what held-out goals share with training.

We measure how much closed-loop success memory alone can buy on this benchmark. Held-out goals are rarely new, and a controller that flies to the remembered endpoint of the matching training instruction, falling back to a learned policy when none matches, succeeds far more often than an imitation-learned policy or always flying forward; where the instruction or the goal is new, it does no better than flying forward. Gains from training fixes need the same scrutiny: for an offline exposure-bias correction, same-budget controls without its preference signal reduce the harm of self-generated history indistinguishably. Success on this benchmark should therefore be reported

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

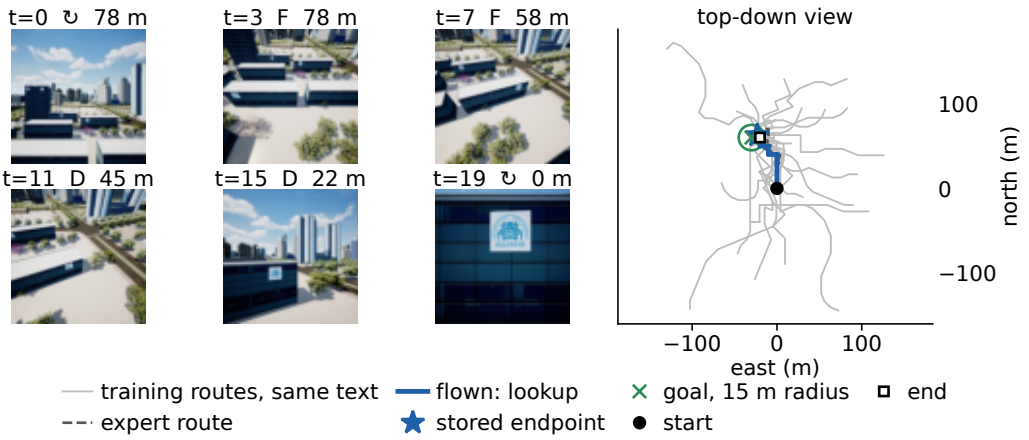

Figure 1: **The task and the shortcut.** Held-out episode 970 of EmbodiedNav-Bench, goal description "Tsinghua University Wenyang Academy": six first-person views along the expert route (labels: step, expert command, distance to the goal; F forward, D down, ↻ turn right; simulator renders released with the benchmark, CC BY 4.0, Zhao et al., 2026) and the routes seen from above. The same description occurs in 24 of the remaining episodes; 23 of their goals lie within 16 m of the group's median (routes in grey). A controller that only looks the description up and flies to that median (blue star, blue path) reaches the goal in the benchmark's simulator without using a single image.

on goal-disjoint episodes beside a retrieval baseline, and a better policy claimed only from paired comparisons on unseen goals.

Contributions. (1) A **recurrence measurement** on a 101-episode held-out grid (§4.1): in 68 episodes one training episode has a near-duplicate instruction and a goal within 15 m; 5 are new in both respects (21 of 504 held-out). (2) A **retrieval probe** (§4.2): 58.4% success, 83.0% on memory routes against 32.6% where the learned policy acts. (3) **Goal-disjoint lists** for reporting (§4.3). (4) For an **offline exposure-bias correction**: label checks, both gaps with intervals, an error decomposition and same-budget controls (§5). We do not claim a better navigator: the probe tests the benchmark, and the offline method's closed-loop effect is unknown.

2 Related work

Aerial VLN benchmarks. R2R (Anderson et al., 2018) and RxR (Ku et al., 2020) define VLN over indoor panoramas; VLN-CE moves it to continuous control (Krantz et al., 2020). AerialVLN (Liu et al., 2023) and OpenFly (Gao et al., 2026) evaluate drones on unseen scenes; EmbodiedNav-Bench (Zhao et al., 2026) evaluates goal-oriented flight in a single city and reports zero-shot results. A benchmark set in one city can still hold out goals or regions: Mirowski et al. (2018) evaluate city-scale navigation only on goals in held-out areas, and Schumann & Riezler (2022) add geographically separated splits to outdoor VLN. Our goal-disjoint lists do this for EmbodiedNav-Bench, defining held-out goals by measured instruction and endpoint recurrence rather than by a map boundary.

Shortcuts and train–test overlap. Models exploit whatever regularity the data offers (Geirhos et al., 2020). Unimodal ablations can rival multimodal navigation agents (Thomason et al., 2019); on EmbodiedQA, where almost every test question also occurs in training, a lookup of the identical or nearest training question rivals agents that navigate before answering (Anand et al., 2018). Overlap analyses found training paraphrases for many open-domain QA test questions (Lewis et al., 2021) and heavy test–train trajectory overlap in next-location prediction (Luca et al., 2023). We carry the question to an agent acting in closed loop: recurrence is measured jointly in the instruction and the goal position, and the lookup runs as a controller in the simulator instead of being scored offline. A published aerial agent already recalls training trajectories through a memory graph whose removal costs it success (Zhang et al., 2025); the probe isolates this channel with no learned component.

Exposure bias and history copying. Behaviour cloning degrades when the learner's errors leave the expert's state distribution; the classical fixes query an expert online (Ross et al., 2011) or anneal from teacher to model inputs (Bengio et al., 2015). When past actions are an input, a cloned policy may latch onto them as a spurious cue (de Haan et al., 2019): the copycat problem of behaviour cloning from observation histories (Wen et al., 2020), related to the inertia problem in driving (Codevilla et al., 2019). Dropping the past motion during training is a standard remedy (Bansal et al., 2019). Bundele et al. (2024) study VLN with offline reinforcement learning on R2R and RxR, conditioning on returns. Preference optimisation against a policy's own sampled mistakes has been applied offline to tool-use and web agents (Chen & Yuille, 2026) and, with simulator rollbacks, to UAV navigation (Xu et al., 2026); continuous VLN uses continued training on self-correction data built from a policy's own errors (Yu et al., 2026). The method examined here builds such pairs from logged states without a simulator, scores them by a one-step geometric distance change and trains with ORPO (Hong et al., 2024), a reference-free alternative to DPO (Rafailov et al., 2023). We add a same-budget attribution test and an error analysis. VLM agents serialise their action history into the prompt (Zhou et al., 2024); the offline analysis varies that line.

3 BENCHMARK, SPLIT AND HARNESS

EmbodiedNav-Bench releases 5,037 goal-oriented expert episodes in a digital twin of one city (Zhao et al., 2026). Each episode has a natural-language goal description, a start pose and a goal position, and an expert pose sequence without action labels. The agent has eight commands: 10 m translations forward, back, left, right, up and down, and $\pm 22.5^\circ$ yaw turns. The simulator is a Windows build of EmbodiedCity on AirSim. Our harness drives it through AirSim by applying each command as an exact pose change, and collisions are logged but do not stop the drone. There is no stop action: an episode ends as soon as the drone is within 15 m of the goal, which counts as success (an oracle criterion; the benchmark scores where the agent stops), beyond 300 m (abandoned), or after $\max(30, 2\times$ the number of expert poses$)$ commands.

Split. EmbodiedNav-Bench is distributed as a single evaluation set; the split below is ours, and "training" in this paper means the remaining episodes. Every episode whose index in the released list is divisible by 10 is held out (504 episodes); the other 4,533 form the training pool. The closed-loop grid is the held-out subset $0{:}10{:}1000$ (101 episodes, 1,950 expert transitions), and the offline evaluation of §5 uses exactly these episodes and steps (Appendix A checks that the episode data and the simulator trace agree). Five grid episodes end before any command: episode 520 starts inside the 15 m radius and episodes 30, 590, 830 and 990 start beyond 300 m (11 of the 504 held-out episodes are of these kinds; the lists mark them). The benchmark served as the navigation track of the UrbanCup 2026 challenge, where our entry ranked second; challenge scores come from the organisers' episodes and harness, so we do not compare them with ours.

4 GOAL RECURRENCE AND WHAT IT IS WORTH IN CLOSED LOOP

4.1 HOW OFTEN HELD-OUT GOALS RECUR

Held-out goals are rarely new (Fig. 2a,b). For each held-out episode we compute, against the training pool, whether its normalised instruction (lower-case, collapsed whitespace) occurs verbatim in training, the maximum `difflib` SequenceMatcher ratio to any training instruction, the distance from its goal to the nearest training goal, and whether a single training episode is both a near-duplicate (ratio ≥ 0.85) and ends within 15 m of the goal. On the grid, 23 of 101 instructions occur verbatim in training, 91 reach similarity 0.85, and 85 goals lie within 15 m of a training goal (median nearest distance 3.4 m). The strictest reading counts only episodes for which one training episode has a near-duplicate instruction *and* a goal within 15 m: 68 of 101 (303 of 504 held-out episodes). Endpoint proximity alone partly reflects goal density: 81 grid goals also lie within 15 m of a training goal whose instruction is not a near-duplicate. Instructions are short referring expressions ("the terrace on the 22nd floor of the beige-exterior building above"), and the same building parts serve as goals in many episodes: 1,080 of the 4,533 training episodes (23.8%) share their instruction with another training episode. In Fig. 2c, nearly all successes of the retrieval probe (§4.2) lie where the instruction or the goal recurs.

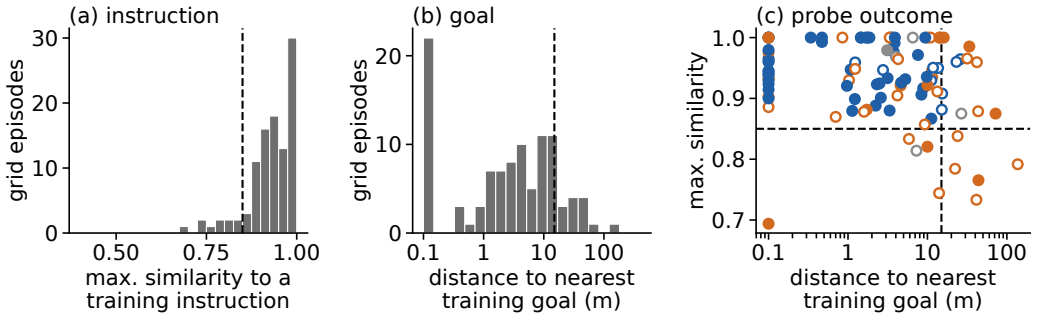

Figure 2: **Recurrence of the 101 grid goals.** (a) Maximum similarity to any of the 3,750 distinct training instructions (dashed: 0.85). (b) Distance to the nearest of the 4,533 training goals, log scale, values below 0.1 m drawn at 0.1 m (dashed: the 15 m success radius). (c) Both quantities per episode, with the outcome of the retrieval probe of §4.2: filled markers succeeded, open markers failed; colour gives the route that acted (blue lookup, orange learned policy, grey no command).

Table 1: Closed-loop success by decision route on our 101-episode held-out grid (one run; our harness on the benchmark's Windows simulator). Intervals: Clopper–Pearson 95%. "No command": five episodes end before any command because they start inside the 15 m radius (episode 520, a success) or beyond the 300 m abandonment radius (30, 590, 830, 990).

Decision route	Episodes	Successes	SR % [95% CI]
Verbatim training instruction	11	11	100.0 [71.5, 100.0]
Paraphrase (similarity ≥ 0.85)	42	33	78.6 [63.2, 89.7]
Learned policy (no accepted match)	43	14	32.6 [19.1, 48.5]
No command (starts $<$15 m or $>$300 m)	5	1	20.0 [0.5, 71.6]
All	101	59	58.4 [48.2, 68.1]
All, collisions counted as failures	101	53	52.5 [42.3, 62.5]

4.2 A RETRIEVAL PROBE IN THE BENCHMARK'S SIMULATOR

Controller. The probe never uses the image to choose where to go. (i) Map each training instruction whose training goals are compact (80% of them within 25 m of their median; 3,613 of 3,750 instructions, 4,084 of 4,533 episodes) to that median; (ii) look the evaluation instruction up verbatim, else take the most similar key and accept it at similarity ≥ 0.85; (iii) reject the endpoint if it is more than 120 m from the start; (iv) otherwise dead-reckon from the start pose in the episode file (exact under the harness's discrete moves), rotate the endpoint displacement into the body frame and issue the dominant-axis 10 m translation until within 8 m; (v) if nothing is accepted, a learned policy acts, prompted with its last eight executed actions and greedy decoding. That policy is an ORPO adapter of the family of §5 by name and offline behaviour (§5.5); its training record was not retained, so we cannot verify that its data exclude the grid episodes. The goal position is used only by the harness for termination and scoring, and the learned policies never receive the pose. The configuration was fixed before this run, after one earlier configuration, which also required at least three demonstrations per instruction, had been run on the same grid (39/101).

Result. The run succeeds on 59 of 101 episodes (58.4%, CI [48.2, 68.1]; mean final distance 88.2 m). Six of the 59 successes include at least one collision, which the harness logs but does not stop; counting them as failures gives 53/101 (52.5%). An imitation policy alone reached 21.78% (22/101; SPL 21.51; final distance 169.0 m) on the same grid and harness in a separate run. A controller that always commands forward would succeed on 32 of the 101 episodes (31.7%); under the harness its outcome follows from the start pose and goal alone (Appendix A). Table 1 splits the run: memory routes succeed on 44 of 53 episodes (83.0%, CI [70.2, 91.9]) and the learned policy on 14 of the 43 episodes in which it acted (32.6%, CI [19.1, 48.5]), a difference of $+50.5$ pp (episode bootstrap CI [32.7, 67.3]). Verbatim lookups never fail; all nine paraphrase failures end at the step

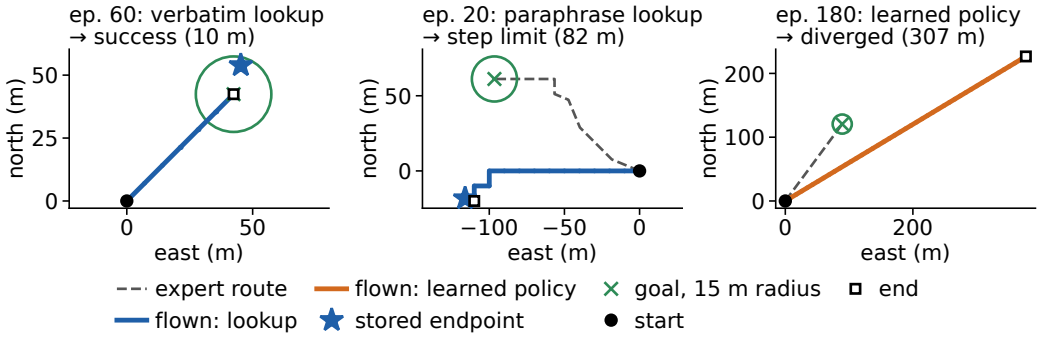

Figure 3: **Three closed-loop episodes of the 58.4% run, seen from above,** one per outcome; within each outcome the episode with the median expert path length is shown. Left: a verbatim lookup succeeds. Middle: a paraphrase lookup matches "the right side of the building built with white frames …" to "the middle of two building built with white frames …" (similarity 0.91), flies to that instruction's endpoint and stops at the step limit. Right: the endpoint of a near-duplicate is rejected by the 120 m start gate, and the learned policy acts and diverges. Positions from the simulator trace; expert routes from the recovered labels.

Table 2: Goal-disjoint subsets. n: grid episodes; held-out pool: qualifying episodes among the 504; Succ.: successes of the 58.4% run; memory routes: episodes where a stored goal was used. Intervals: Clopper–Pearson 95%.

Subset of the 101-episode grid	n	held-out pool	Succ.	memory routes	SR % [95% CI]
All grid episodes	101	504	59	53	58.4 [48.2, 68.1]
Instruction-disjoint	78	366	41	41	52.6 [40.9, 64.0]
Paraphrase-disjoint (< 0.85)	10	71	3	0	30.0 [6.7, 65.2]
Endpoint-disjoint (≥ 15 m)	16	75	4	4	25.0 [7.3, 52.4]
Goal-disjoint (both)	5	21	1	0	20.0 [0.5, 71.6]
Not goal-disjoint	96	483	58	53	60.4 [49.9, 70.3]

limit, as expected when a paraphrase maps to a different goal (Fig. 3, middle). The learned policy diverges beyond 300 m in 12 of the 43 episodes in which it acted; memory routes never do.

By recurrence. Episodes with a recurring instruction succeed at 62.2% (56 of the 90 recurring against the controller's table, CI [51.4, 72.2]) and the 11 others at 27.3% (CI [6.0, 61.0]); difference $+34.9$ pp (CI [5.5, 61.1]). Of the 43 episodes in which the learned policy acted, 33 had a recurring instruction whose endpoint the 120 m gate rejected and 10 were novel. It succeeded on 11/33 and 3/10, so we see no sign that the policy itself profits from recurrence; always-forward would have succeeded on 14 of these 43 episodes, as often as the policy. Its episodes are also longer (median expert path 190 m against 95–110 m on memory routes).

What a leaderboard number measures. A success rate mixes routes: setting aside the five episodes without a command, $\mathrm{SR} \approx \pi_{\mathrm{mem}}\mathrm{SR}_{\mathrm{mem}} + (1 - \pi_{\mathrm{mem}})\mathrm{SR}_{\mathrm{policy}}$, here with $\pi_{\mathrm{mem}} = 53/96$, $\mathrm{SR}_{\mathrm{mem}} = 83.0\%$ and $\mathrm{SR}_{\mathrm{policy}} = 32.6\%$. Any system that can map an instruction to a remembered location—explicitly (as here, or with a trajectory memory (Zhang et al., 2025)) or implicitly, as a fine-tuned model can by memorising instruction–endpoint pairs—draws on the first term, for which 91 of the 101 grid episodes are eligible, and a single SR cannot separate the terms.

4.3 A GOAL-DISJOINT EVALUATION PROTOCOL

Which held-out goals count as new decides what the probe can do: *instruction-disjoint* (no verbatim training match), *paraphrase-disjoint* (no training instruction at similarity ≥ 0.85), *endpoint-disjoint* (no training goal within 15 m), and *goal-disjoint* (paraphrase- and endpoint-disjoint); Table 2 gives their sizes and the probe's success on each, and lists for the grid and the full held-out pool are

released. Removing verbatim matches barely matters (78 episodes, 52.6%), because paraphrases carry most of the lookup. Removing paraphrases leaves 10 episodes and 3 successes, all by the learned policy; removing goals within 15 m of a training goal leaves 16 and 4 (the four endpoint-disjoint episodes on which a paraphrase was accepted all failed); the goal-disjoint subset has 5 episodes and 1 success. On all three the probe succeeds exactly as often as always-forward would (3/10, 4/16, 1/5). One paraphrase-disjoint episode (830) and one endpoint-disjoint episode (990) start beyond 300 m and cannot be solved under the harness. The 96 episodes that have a near-duplicate instruction or a nearby goal in training carry 58 of the 59 successes. We recommend (i) reporting SR on the goal-disjoint lists next to the grid, with n; (ii) reporting the retrieval probe as a baseline; and (iii) adding new goals, since 21 goal-disjoint held-out episodes cannot give tight estimates. Endpoint-disjointness at the success radius is the strictest of our single criteria; lexical novelty alone leaves the lookup nearly intact.

5 What an offline exposure-bias correction buys

A training fix deserves the same scrutiny as a success rate. A VLM policy that reads its own action history is trained on expert histories but deployed on self-generated ones; the method examined here corrects this offline, recovering action labels from poses and training against the policy's own sampled errors with a geometric preference signal. We ask what its offline gain can be credited to.

5.1 Action labels from poses and their identifiability

Translation labels can be recovered from the poses almost exactly; turn directions cannot. Inverse dynamics gives one label per transition: a translation is the body-frame displacement between consecutive poses, a transition with no displacement is a turn. This yields 102,312 labelled transitions from 5,037 episodes, each checked against the poses (Table 5, Appendix C). The released poses carry the yaw, so horizontal translations are identifiable: all 54,341 checkable horizontal translations point where the label says given that yaw, and 99.9% are exactly one 10 m step. Overall 75,232 of 76,002 checkable translations (99.0%) are exact; the main exception is 714 transitions labelled *up* that do not move. Turns are different. All 21,273 checkable turns are stationary, but their direction can be checked only against the heading change between the translations that bracket a run of turns, and the labelled net rotation matches it in only 3,556 of 5,033 runs (70.7%). Turns are 21.3% of the labels, so part of the turn error below is label noise.

5.2 Model, objectives and decoding

The policy is Qwen2.5-VL-7B (Bai et al., 2025) with a rank-32 LoRA adapter (Hu et al., 2022) ($\alpha = 64$) on all linear projections of the language model; the vision encoder is frozen. The prompt holds the goal text, the camera pitch, a history line of up to eight past commands by name (or the literal `none`), and the eight command names; the current image is attached. The target is the command name as plain text, with no added vocabulary. **Imitation** minimises the negative log-likelihood of the response tokens, $\mathcal{L}_{\text{SFT}} = -\sum_j \log p_\theta(y_j \mid x, y_{<j})$, on 68,331 training tuples from 4,526 training-pool episodes, for one epoch. Forward commands are capped at 20,000 (29% of targets); the history line is empty or the expert's (Table 4). **Preference pairs** are mined on training-pool states: $K = 6$ commands are sampled at temperature 1; a sample $a \neq a^\star$ qualifies as rejected if its one-step decrease in distance to the goal, $r(s,a) = \|\mathbf{p}_s - \mathbf{g}\| - \|T(\mathbf{p}_s, a) - \mathbf{g}\|$, is worse than the expert's by more than 1 m, or if it is the opposite turn of an expert turn, or a translation with $r < -1$ m when the expert turned ($r = 0$ for turns). The worst qualifying sample is kept, and the expert command is the chosen response. The mining prompt's history is empty, a corrupted expert history or the clean expert history, never the policy's rollout (Table 4); 6,472 pairs from 885 episodes pass a verification gate, and 51% of their targets are forward. **ORPO** (Hong et al., 2024) then optimises, starting from the imitation adapter,

$$\mathcal{L}_{\text{ORPO}} = \mathcal{L}_{\text{SFT}}(y_w) - \lambda \log \sigma\Big(\log \tfrac{\text{odds}_\theta(y_w|x)}{\text{odds}_\theta(y_l|x)}\Big), \qquad \text{odds}_\theta(y \mid x) = \tfrac{p_\theta(y|x)}{1-p_\theta(y|x)}, \tag{1}$$

with length-normalised likelihood p_θ, $\lambda = 0.1$, learning rate 3×10^{-5}, one epoch (405 steps), six seeds. **Decoding** is greedy with at most 12 new tokens; the first command name in the output is the action. No held-out episode occurs in the imitation data or in the preference pairs. Full hyperparameters are in Appendix A.

Table 3: Offline next-action accuracy (%) of the imitation policy and six ORPO seeds (seed mean), with both gaps and 95% intervals. Per-seed history-harm gaps: 1.2–2.9.

Policy	teacher	no-history	self	teacher − self	no-history − self
Imitation (1 model)	71.4	60.6	50.7	20.6 [14.2, 27.4]	9.9 [4.5, 15.7]
+ ORPO (6 seeds)	70.6	61.0	59.0	11.7 [6.6, 17.2]	2.0 [−0.9, 5.4]
Self gain / gap reduction			+8.2 [3.3, 13.7]	9.0 [4.1, 14.3]	7.9 [3.1, 13.4]

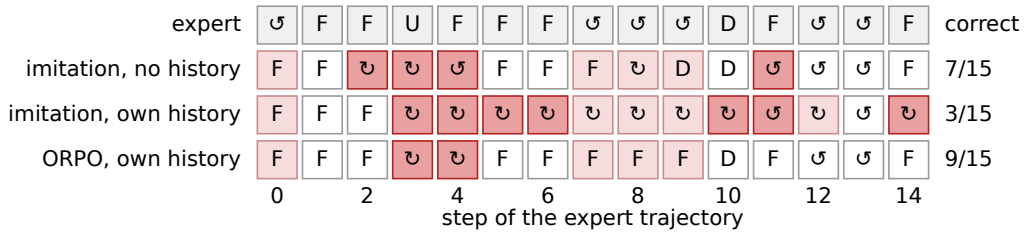

Figure 4: **Copied turns on one grid episode.** Commands along the logged expert trajectory of episode 890: the expert's, and the predictions of the imitation policy without history and with its own history and of ORPO (seed 1) with its own history, first run of each (F/B/L/R/U/D translations; ↺ left and ↻ right turns; dark red: a turn where the expert translates, light red: another error; right: correct steps). Selection rule: of the 24 grid episodes in which the imitation policy with its own history makes at least one such turn error, the one with the lower-median count (7).

5.3 Offline results with both gaps

Offline evaluation asks, on the logged expert trajectory, whether the predicted next command equals the expert's. Three arms differ only in the history line: expert (*teacher*), `none`, or the policy's own rolled-forward predictions (*self*). Two gaps follow: the *teacher-forcing gap* (teacher − self), the full train/deploy mismatch, and the *history-harm gap* (no-history − self), the cost of reading one's own history instead of none. All seven policies were served by one process on the 1,950-step grid. The self arm was run five times and the others twice per policy, because batched greedy decoding changes some predictions between identical runs and self-history amplifies this. Intervals resample episodes and, per arm, runs (20,000 draws); they condition on the trained models.

Reading. ORPO lowers the history-harm gap by 7.9 points (CI [3.1, 13.4]) and the teacher-forcing gap by 9.0 (CI [4.1, 14.3]); self-history accuracy rises by 8.2 points, while teacher and no-history accuracy do not move beyond noise (Table 3). Three statements follow and nothing stronger. (i) After ORPO, reading self-generated history is nearly harmless relative to reading none (residual 2.0, CI [−0.9, 5.4]). (ii) It is not useful: self-history stays below no-history, and ORPO with self-history is not better than the imitation policy with no history (−1.6, CI [−5.7, 1.8]). (iii) None of this concerns reaching goals. The reduction does not depend on recurring goals: on the instruction-disjoint and endpoint-disjoint subsets of the same grid the history-harm gap falls from 10.4 to 2.8 and from 10.5 to 2.7 points (point estimates; 78 and 16 episodes). The baseline is a single imitation seed; §5.4 shows how much the gap varies across seeds.

Where the gain comes from. The gain comes from suppressing copied turns (Table 7, Appendix D). Under self-history the imitation policy predicts a turn where the expert translates on 13.9% of all steps, against 4.7% with no history: its own past turns in the history line induce further turns, the copycat pattern of Wen et al. (2020). Fig. 4 shows the pattern on one episode chosen by a fixed rule. ORPO removes almost all of it (1.6%): this one type falls by 12.3 points of all steps, more than the net self-history gain of 8.2, because other types grow. Horizontal–vertical confusions stay at 21–22% of steps despite 2,103 training pairs of that type, and turn-to-translation errors grow (7.1 → 9.7%). By expert action, forward accuracy under self-history rises from 55.8 to 79.8% while descend, left-turn and right-turn accuracy fall (Appendix D). ORPO thus mainly suppresses history-driven turns and shifts mass toward the majority command, which is 48% of grid steps.

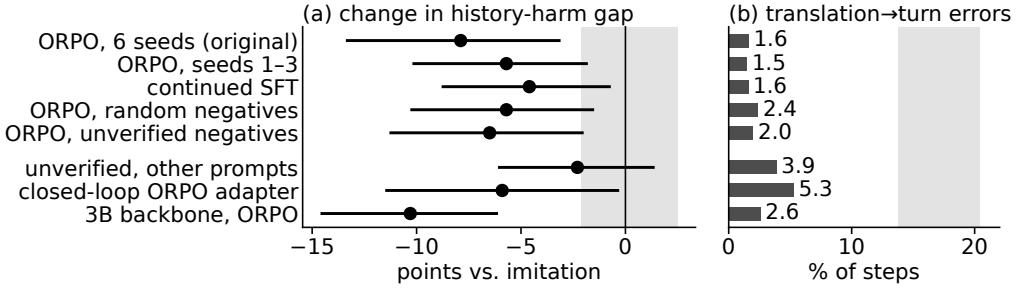

Figure 5: **Attribution controls.** (a) Change in the history-harm gap relative to the imitation policy served in the same session (points; 95% hierarchical-bootstrap intervals; three seeds per arm, six in the original session, one closed-loop adapter); grey band: the same change for two further imitation seeds relative to the released one. (b) Translation-to-turn errors under self-history (% of steps); grey band: the three imitation seeds. The 3B row is relative to a 3B imitation policy.

5.4 Attribution controls, seeds and relabelled turns

Is the reduction due to the preference signal or to further training on the mined prompts? To separate the two, we trained controls from the imitation adapter with ORPO's optimiser settings (three seeds each): *continued SFT* on the 6,472 chosen responses and ORPO with *random negatives* (a uniformly random other command), both on the verified pairs' prompts; and ORPO with *unverified negatives* (a wrong command sampled from the imitation policy, without the geometric test), first from a separate mining pass with different prompts and then on the verified pairs' prompts (6,364 pairs; 398 steps against 405). Each evaluation session serves the imitation adapter next to the adapters it tests, and every result is a change relative to that session's imitation policy (Fig. 5; all accuracies in Table 6). The released imitation adapter and ORPO seeds 1–3 were re-evaluated likewise, with the same software on different hardware; the controls were trained on images re-extracted from the public release (Appendix A). Between sessions the imitation policy's accuracies move by up to about one point and its history-harm gap by up to two (here 70.8/59.9/51.8 for teacher/no-history/self), and for ORPO seeds 1–3 the original session's 9.9 $\rightarrow$ 2.5 becomes 8.1 $\rightarrow$ 2.4.

The preference signal adds little we can detect. Relative to imitation, the three ORPO seeds change the history-harm gap by -5.7 points (CI $[-10.2, -1.8]$) and continued SFT by -4.6 (CI $[-8.8, -0.7]$). Continued SFT thus recovers about four fifths of the reduction; the difference between the two changes, 1.2 points (CI $[-1.5, 4.0]$, paired by episode), is not significant but allows the preference term up to about 4 of the 5.7 points. It also removes the dominant failure of §5.3: translation-to-turn errors under self-history fall to 1.6% of steps, against 1.5% for ORPO. The mined prompts differ from the imitation data in two ways that could each suppress copied turns, and these controls do not separate them: 57% of them have no history (imitation data: 48%), and 51% have a forward target (imitation data: 29%). ORPO with random negatives (-5.7) and ORPO with unverified negatives on the same prompts (-6.5) are indistinguishable from verified ORPO (differences 0.0, CI $[-3.1, 2.3]$, and -0.8, CI $[-3.2, 1.7]$), although 32% of the unverified negatives coincide with the verified ones. Separately mined unverified pairs, whose prompts differ (42% forward targets), gave a smaller change (-2.3; 3.5 points less than verified ORPO, CI [0.4, 6.6]); the difference between the two unverified runs, -4.2 (CI $[-7.9, -0.1]$), is borderline, so this deficit is at most suggestive of a prompt effect. None of the five cross-session contrasts is corrected for multiplicity.

Seed variation of the baseline. The imitation gap varies across training seeds by about as much as ORPO or its controls reduce it (Fig. 5a, grey band). Two further imitation seeds, trained with the same recipe on the re-extracted images, have history-harm gaps of 10.6 (CI [6.0, 15.2]) and 6.0 (CI [1.7, 10.3]), against 8.1 (CI [2.9, 13.5]) for the released seed in the same session, so the reduction of §5.3 should be read as the reduction for the released seed. The ORPO seeds and every control trained on the verified pairs' prompts nevertheless end below the lowest imitation seed's gap.

A smaller backbone. The reduction also appears on a smaller backbone. With Qwen2.5-VL-3B and the same procedure (imitation on the same tuples; pairs mined from the 3B imitation policy; three ORPO seeds), imitation reaches 69.8/60.7/48.7 (history-harm gap 12.1) and ORPO changes the gap by -10.3 (CI $[-14.6, -6.1]$). The signature is the same: translation-to-turn errors under self-history fall from 18.5% to 2.6% of steps. This backbone has no continued-SFT control.

Relabelled turns. Turn-label noise does not explain the reduction. Relabelling every stationary transition between two horizontal translations from the heading change they certify (§5.1) changes 544 labels (395 stationary translations become turns, 149 turn directions flip) and marks 3,616 transitions as unidentifiable. Imitation retrained without them takes 4,067 instead of 4,271 optimiser steps. On relabelled targets, with the 38 unidentifiable grid steps masked, it scores 75.0/61.8/50.5 against 71.9/60.4/51.7 for the original imitation policy: cleaner labels raise teacher-forced accuracy but leave the history harm in place (gap 11.3 against 8.8). ORPO on top (pairs mined afresh, three seeds) changes the gap by -8.5 (CI $[-13.1, -3.9]$), with translation-to-turn errors falling from 17.3% to 4.7% of steps (-7.5 on the original targets).

5.5 What the closed-loop runs say about it

Little. The closed-loop adapters are not the offline models: their weight files differ from the offline imitation adapter and all six seeds. Offline, in one session with the offline imitation adapter, they score 68.4/45.7/38.5 (imitation, the 21.78% run) and 71.0/60.4/57.8 (ORPO, the fallback of the 58.4% run), and the ORPO adapter's gap change is indistinguishable from the released seeds' (difference -0.1, CI $[-4.9, 4.4]$). The only closed-loop number for a preference-trained policy is the fallback route of §4.2, 14 of the 43 gate-selected episodes in which it acted; the imitation 21.78% comes from a separate run whose per-episode outcomes were not retained, so the two cannot be paired. Three observations hold regardless: the ORPO adapter's offline accuracy near 60% coexists with 32.6% closed-loop success on its route; the learned policy diverged beyond 300 m in 12 of those 43 episodes; and it issued *turn left* on 671 of its 1,561 closed-loop steps (43%) and *turn right* never, although left turns are 11.5% of the grid's expert steps. The offline proxy, which always observes the expert's next state, can see none of this. Settling the question needs both policies run on identical episodes, overall and on the goal-disjoint lists, in the Windows-only simulator, which was not available to us.

6 Limitations and conclusion

Limitations. **Split and harness** are ours: the harness teleports the drone, which can pass through buildings, and ends an episode on entering the 15 m radius, so our success rates are not comparable with other protocols' numbers. **Closed loop**: one run of the probe, whose learned fallback is not deterministic under batched serving, and no paired test of the offline method. **Scope**: one benchmark; we make no claim that the offline method transfers to R2R, RxR or VLN-CE. **Attribution**: three seeds per control, intervals conditional on the trained models, five uncorrected cross-session contrasts, and a history and class mix of the mined prompts that the controls do not separate. **Exact-match accuracy** penalises equally good alternative commands (e.g. descending before or after moving forward). **Lexical similarity** misses semantic paraphrases and also pairs different goals (Fig. 3, middle); the joint criterion (68/101) is the most conservative count. **Shared frame**: the probe relies on the single coordinate frame shared by all episodes, which is the property measured.

Conclusion. In this work, we measured goal recurrence in EmbodiedNav-Bench and re-examined an offline exposure-bias correction trained on the same release. Almost every held-out goal recurs among the remaining episodes, so remembering where goals were is worth more closed-loop success than the learned policies we have, which do not beat flying forward. Success on this benchmark should therefore be reported on goal-disjoint episodes and next to a retrieval baseline. The offline correction makes self-generated history nearly harmless on logged states, mostly by suppressing history-driven turns. Same-budget controls on the same prompts, without the preference signal or without its geometric check, are indistinguishable from it, so the gain cannot be attributed to the preference signal. The method rests on near-exact translation labels but noisy turn labels, and it has yet to be tested where it matters, in a paired, goal-disjoint closed-loop comparison.

REPRODUCIBILITY STATEMENT

The supplementary material contains the code and data that produce every number, table and figure in the paper, and its README maps each of them to the script that produces it. It includes the frozen trace, per-episode outcomes and lookup table of the closed-loop run (with SHA-256 hashes); the recovered-label data, the recurrence measurement and the goal-disjoint lists for the grid and the full held-out pool; the label identifiability check; the 63 per-step prediction files of the original offline evaluation, from which Tables 3 and 7 are re-derived exactly; and, for §5.4, the data-construction, training and evaluation scripts with the training arguments, preference pairs, relabelled data and per-step predictions of every control. Single proportions carry Clopper–Pearson intervals; differences and accuracies carry percentile bootstraps over episodes (and evaluation runs, where repeated) with fixed seeds, conditioning on the trained models. The simulator and base models are third-party, and adapter weights are not included.

ETHICS STATEMENT

The work uses a public benchmark and simulator and involves no human subjects or personal data. Urban drone navigation is dual-use; this paper contributes an evaluation study and a caution against reading leaderboard success as capability, and releases no deployable controller beyond a lookup over public benchmark data. We report the challenge ranking only as context and draw no comparison with other entries, whose numbers come from a different protocol.

AI USE STATEMENT

In this work, we used generative AI tools (Claude, Anthropic, run as several agents under the author's direction) to design and give feedback on the control experiments and analyses, to implement them (data construction, training, evaluation and statistics code), to reformat the released episode data for the analyses, to refine hypotheses about the source of the offline gain, and to interpret results; the agents also launched and monitored the training and evaluation runs on rented GPUs. We have not used generative AI tools to generate synthetic data sets or for translation, and theoretical models, mathematical claims and proofs, and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to create and modify the figures, to create and edit code, to draft and revise the text, to search for and summarise related literature, to verify and format references, and to propose keywords. We have reviewed all AI-assisted work: every number in the paper is produced by scripts from the result files in the supplementary material, an independent AI-assisted check re-derived the numbers from the raw records, every reference was checked against its publisher, arXiv or OpenReview record, and the author decided which experiments to run and which claims to make and checked the results and the text. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

REFERENCES

Ankesh Anand, Eugene Belilovsky, Kyle Kastner, Hugo Larochelle, and Aaron Courville. Blindfold baselines for embodied QA, 2018. NIPS 2018 Visually-Grounded Interaction and Language (ViGIL) Workshop; arXiv:1811.05013.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Shuai Bai, Keqin Chen, Xuejing Liu, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. ChauffeurNet: Learning to drive by imitating the best and synthesizing the worst. In *Proceedings of Robotics: Science and Systems (RSS)*, 2019.

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

Valay Bundele, Mahesh Bhupati, Biplab Banerjee, and Aditya Grover. Scaling vision-and-language navigation with offline RL. *Transactions on Machine Learning Research (TMLR)*, 2024. arXiv:2403.18454.

Yixiong Chen and Alan Yuille. Agentic-DPO: From imitation to agentic policy optimization on expert trajectories, 2026. arXiv:2607.10601.

Felipe Codevilla, Eder Santana, Antonio M. López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Yunpeng Gao, Chenhui Li, Zhongrui You, Junli Liu, et al. OpenFly: A comprehensive platform for aerial vision-language navigation. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2502.18041.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2:665–673, 2020.

Jiwoo Hong, Noah Lee, and James Thorne. ORPO: Monolithic preference optimization without reference model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. arXiv:2106.09685.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2021. arXiv:2008.02637.

Shubo Liu, Hongsheng Zhang, Yuankai Qi, Peng Wang, Yaning Zhang, and Qi Wu. AerialVLN: Vision-and-language navigation for UAVs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Massimiliano Luca, Luca Pappalardo, Bruno Lepri, and Gianni Barlacchi. Trajectory test-train overlap in next-location prediction datasets. *Machine Learning*, 112(11):4597–4634, 2023. doi: 10.1007/s10994-023-06386-x.

Piotr Mirowski, Matt Grimes, Mateusz Malinowski, Karl Moritz Hermann, Keith Anderson, Denis Teplyashin, Karen Simonyan, Koray Kavukcuoglu, Andrew Zisserman, and Raia Hadsell. Learning to navigate in cities without a map. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 627–635, 2011.

Raphael Schumann and Stefan Riezler. Analyzing generalization of vision and language navigation to unseen outdoor areas. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 7519–7532, 2022. doi: 10.18653/v1/2022.acl-long.518.

Jesse Thomason, Daniel Gordon, and Yonatan Bisk. Shifting the baseline: Single modality performance on visual navigation & QA. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

Chuan Wen, Jierui Lin, Trevor Darrell, Dinesh Jayaraman, and Yang Gao. Fighting copycat agents in behavioral cloning from observation histories. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Peng Xu, Chengcheng Wang, and Shaohua Wan. AeroDPO: Unleashing lightweight UAV navigation with high-fidelity perception and automated preference optimization, 2026. arXiv:2608.07557.

Zhuoyuan Yu, Yuxing Long, Zihan Yang, Chengyan Zeng, Hongwei Fan, Jiyao Zhang, and Hao Dong. CorrectNav: Self-correction flywheel empowers vision-language-action navigation model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18737–18745, 2026. doi: 10.1609/aaai.v40i22.38942.

Weichen Zhang, Chen Gao, Shiquan Yu, Ruiying Peng, Baining Zhao, Qian Zhang, Jinqiang Cui, Xinlei Chen, and Yong Li. CityNavAgent: Aerial vision-and-language navigation with hierarchical semantic planning and global memory. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 31292–31309, 2025. doi: 10.18653/v1/2025.acl-long.1511.

Baining Zhao, Ziyou Wang, Jianjie Fang, Zile Zhou, Yanggang Xu, Yatai Ji, Jiacheng Xu, Qian Zhang, Weichen Zhang, Chen Gao, and Xinlei Chen. How far are large multimodal models from human-level spatial action? A benchmark for goal-oriented embodied navigation in urban airspace. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 10326–10337, 2026. doi: 10.1145/3770855.3817501.

Gengze Zhou, Yicong Hong, and Qi Wu. NavGPT: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

A Implementation details

Table 4: History condition at each stage (mining samples empty/corrupted/clean histories with probabilities 40/30/30; the shares shown are those of the kept pairs).

Stage	history line in the prompt
Imitation training (68,331 tuples)	expert (52.2%) or `none` (47.8%)
Mining and ORPO (6,472 pairs)	`none` 57%, corrupted or truncated 15%, expert 27%
Offline evaluation	expert / `none` / self (rolled-forward predictions)
Closed-loop fallback in the 58.4% run	self (last eight executed actions)

Data alignment: for all 101 grid episodes the instruction in our recovered-label data equals the one in the simulator trace, and for the 96 with a logged step the logged distance matches the data's goal to within 0.01 m. **Always-forward baseline:** the harness applies each command as the exact pose change of Section 3; replaying all 2,013 logged discrete commands of the 58.4% run from their logged poses reproduces every next logged position to within 2.5 cm. The harness takes the start heading from the episode file, which differs from the heading of the first recovered pose by one 22.5° turn in 15 grid episodes; we recover it from the first logged translation where one exists (86 episodes) and use the recovered-label heading otherwise. Always-forward then succeeds on 32 of the 101 grid

episodes (33 with the recovered-label headings throughout) and on 144 of the 504 held-out episodes. By route of the 58.4% run it would succeed on 4 of the 11 verbatim, 13 of the 42 paraphrase and 14 of the 43 learned-policy episodes, and on 21 of the 78 instruction-disjoint episodes (the probe: 41). **Imitation:** ms-swift LoRA, rank 32, $\alpha = 64$, all linear layers of the language model, vision tower frozen, bf16, learning rate 10^{-4}, cosine schedule with 3% warmup, batch 4 $\times$ gradient accumulation 4, one epoch, seed 42, maximum length 4,096. **ORPO:** same LoRA configuration continued from the imitation adapter, $\beta = 0.1$ (the odds-ratio weight λ), learning rate 3×10^{-5}, cosine schedule without warmup, batch 1 $\times$ accumulation 16, one epoch = 405 steps, loss on the last assistant turn, seeds 1–6. The corruption applied to 30% of mining histories replaces each past command with a random one with probability 0.3 (in 40% of cases), truncates the history (30%), or empties it (30%). Mining samples states from training-pool episodes with images, shuffled with a fixed seed. Serving used vLLM 0.9.2 with transformers 4.53.3; an older vLLM silently mis-loads these adapters (accuracy near 8%, unparseable outputs), which the evaluation guards against by refusing any arm below 30% accuracy or with more than five unparseable predictions. The original imitation model and ORPO seeds were trained and evaluated on RTX 4090 GPUs with the image files of the original pipeline. The controls of §5.4 were trained and evaluated with the same software versions on L40S and A100 GPUs, with images re-extracted from the public release (JPEG, quality 95).

B Additional tables

Table 5: Identifiability check of the 102,312 recovered labels. The final transition of each episode (5,037) has no successor pose and cannot be checked.

Label class	checkable	consistent	check
Horizontal translation	54,341	54,341 (100%)	direction given the logged yaw
		54,303 (99.9%)	exactly one 10 m horizontal step
Down	14,332	14,315 (99.9%)	10 m vertical
Up	7,329	6,614 (90.2%)	10 m vertical; 714 do not move
Turn (left/right)	21,273	21,273 (100%)	pose does not move
Turn direction (runs)	5,033	3,556 (70.7%)	net rotation = heading change

C Label identifiability check

The heading convention is fixed by the translations (yaw grows clockwise; a left turn is $-22.5°$). Of the 1,477 inconsistent turn runs, the heading changed by less than 22.5° per labelled turn in 636 (not at all in 491), by more in 772, and by exactly that amount but in the other direction in 69, which indicates both direction errors and mislabelled transitions. Camera pitch never changes within an episode, so no pitch command is needed. On the 111 grid turn steps whose direction is checkable, both policies are rarely correct (imitation teacher-forced 17.1% on the 76 consistent and 10.0% on the 35 inconsistent labels; ORPO 10.5% and 6.2%), so mid-trajectory turns are a weak point regardless of label quality.

D Error analysis

Tables 7 and 8 split the errors by type and by expert action. Forward is 48.0% of the grid's expert steps, and every confusion cell that occurs in evaluation also occurs among the training pairs, so the pattern is not explained by missing pair types.

E Memory-representation diagnostic

We also asked whether re-encoding history as coarse spatial memory (dead-reckoned position quantised into regions found by structural-entropy or k-means partitioning of the expert transition graph) reduces the harm. On an earlier 40-episode, 817-step grid it did not: under self-history, region memory scored 40.7%, an action list 39.3% and raw coordinates 36.5%, all below a no-history control

Table 6: Controls, seeds, relabelled turns and the closed-loop adapters, offline (next-action accuracy, %; seed means; 95% intervals from the hierarchical bootstrap). Each session serves an imitation adapter (IL) next to the adapters it tests; Δ gap is the change in the no-history − self gap relative to that session's first IL row. [a]Relabelled targets, unidentifiable steps masked (1,912 of 1,950 steps). [b]Original targets.

Session / policy	teacher	no-hist.	self	no-hist.−self	Δ gap vs IL
Re-evaluation					
imitation (IL)	70.8	59.9	51.8	8.1 [3.3, 13.2]	–
ORPO (released, 3 seeds)	70.6	60.7	58.2	2.4 [−0.7, 6.2]	−5.7 [−10.2, −1.8]
Continued SFT					
imitation (IL)	70.9	60.0	51.6	8.5 [3.6, 13.6]	–
Continued SFT (3 seeds)	71.1	61.0	57.1	3.9 [0.9, 7.7]	−4.6 [−8.8, −0.7]
Random negatives					
imitation (IL)	70.8	60.1	51.3	8.8 [4.0, 13.6]	–
ORPO, random negatives (3 seeds)	71.7	61.0	58.0	3.1 [−0.2, 7.1]	−5.7 [−10.3, −1.5]
Unverified, own prompts					
imitation (IL)	71.1	59.6	51.2	8.4 [3.5, 13.4]	–
ORPO, unverified negatives (3 seeds)	71.5	60.4	54.2	6.2 [2.6, 10.1]	−2.3 [−6.1, 1.4]
Unverified, same prompts					
imitation (IL)	70.9	59.8	51.8	8.0 [2.9, 13.4]	–
ORPO, unverified negatives (3 seeds)	70.4	60.4	58.9	1.5 [−1.7, 5.5]	−6.5 [−11.3, −2.0]
Closed-loop adapters					
imitation (IL)	70.9	59.7	51.2	8.5 [3.6, 13.4]	–
closed-loop imitation adapter	68.4	45.7	38.5	7.2 [1.9, 12.5]	−1.3 [−7.2, 4.8]
closed-loop ORPO adapter	71.0	60.4	57.8	2.6 [−0.3, 5.7]	−5.9 [−11.5, −0.3]
3B backbone					
imitation (IL)	69.8	60.7	48.7	12.1 [7.1, 16.8]	–
ORPO 3B (3 seeds)	70.3	61.2	59.4	1.8 [−0.6, 4.0]	−10.3 [−14.6, −6.1]
Relabelled turns[a]					
IL, relabelled turns	75.0	61.8	50.5	11.3 [5.5, 17.2]	–
ORPO on relabelled IL (3 seeds)	72.9	62.4	59.5	2.8 [−0.2, 6.2]	−8.5 [−13.1, −3.9]
IL, original labels	71.9	60.4	51.7	8.8 [3.7, 14.0]	−2.6 [−7.0, 2.2]
Relabelled turns[b]					
IL, relabelled turns	73.6	60.6	50.4	10.2 [4.8, 15.8]	–
ORPO on relabelled IL (3 seeds)	71.7	61.4	58.7	2.7 [−0.2, 6.0]	−7.5 [−12.3, −3.0]
Imitation seeds					
imitation (IL)	70.8	59.7	51.6	8.1 [2.9, 13.5]	–
IL seed 43	71.4	60.2	49.6	10.6 [6.0, 15.2]	2.5 [−1.9, 6.8]
IL seed 44	71.2	59.2	53.2	6.0 [1.7, 10.3]	−2.1 [−7.1, 2.9]

Table 7: Where the errors are: error rate by type as % of all 1,950 steps (runs and, for ORPO, seeds averaged), with the number of preference pairs of each (chosen → rejected) type.

Error type (expert → predicted)	pairs	IL no-hist.	IL self	ORPO no-hist.	ORPO self
turn → opposite turn	682	2.4	3.0	2.5	2.4
turn → translation	649	8.6	7.1	9.1	9.7
translation → turn	1,488	4.7	13.9	3.6	1.6
horizontal ↔ vertical	2,103	19.1	21.2	18.0	21.5
translation → other translation (same plane)	1,550	4.5	4.1	5.7	5.7
unparseable	0	0.0	0.0	0.0	0.0
all errors	6,472	39.4	49.3	39.0	41.0

scored on the same steps and targets (60.3%; region − control = −19.7, CI [−28.7, −10.7]; the control may be a different checkpoint) and, in point estimate, below a majority-class baseline (45.4%).

Table 8: Accuracy (%) by expert action on the 1,950-step grid (imitation; ORPO seed mean).

Expert action	n	IL teacher	IL no-hist.	IL self	ORPO teacher	ORPO no-hist.	ORPO self
forward	936	77.9	74.9	55.8	86.7	83.4	79.8
down	372	83.7	79.8	74.5	79.6	69.5	69.5
turn left	224	63.8	45.3	51.3	53.1	40.6	36.9
up	124	63.7	41.5	40.6	49.5	34.0	39.1
left	113	45.1	11.5	7.1	26.5	4.3	3.0
turn right	103	49.0	10.7	15.5	33.5	8.8	7.7
right	45	36.7	13.3	0.0	32.0	2.4	0.0
back	33	33.3	3.0	1.2	33.3	4.0	5.9

In a separate check on the 101-episode grid, a one-step dead-reckoning error left the region label unchanged at 21.1% of the steps checked, and at 0.8% when the error was in the first translation. Structural entropy did not outperform k-means (40.7 vs 41.1%). We keep this only as evidence that the history channel, in every encoding tested, hurt the imitation policy.