CAN SAFETY TRANSFER ACROSS EMBODIMENTS? CORRECTIVE SUPERVISION FOR OFFLINE SAFE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline safe reinforcement learning relies on safety-relevant experience from a fixed dataset, which may be limited for a particular embodiment. Other embodiments may have demonstrated useful corrective behavior, but differences in actions and dynamics make this experience difficult to reuse. Can their observed motion provide safety supervision beyond what the target can learn from its own data? We investigate this question through cross-embodiment corrective supervision (CECS), an offline approach that transfers observed source motion into target-native training signals. The method retrieves safety-relevant source motions in locally matched task contexts, translates them into target-native actions via target inverse dynamics, filters the resulting corrections with target-side safety information, and uses them to refine the target policy. This formulation transfers corrective behavior rather than directly sharing actions or policies across embodiments. Experiments across different morphologies and task families show improved safety over pretrained target policies and competitive safety–reward trade-offs against offline safe RL baselines. Controlled comparisons at shared contexts and equal correction counts show that source supervision improves safety beyond equally trained target-only retrieval. Shuffling the corrective labels substantially degrades performance, highlighting the importance of matching corrections to the context. A matched-state analysis further shows that the refined policy reduces accumulated cost through sustained control while retaining most task progress. These results demonstrate the value of cross-embodiment motion as corrective supervision and identify context matching as an important factor in its effective use.

1 INTRODUCTION

A car and a drone traversing a corridor may face the same boundary constraint, yet their control commands cannot be transferred directly. Their trajectories can still express a common response: maintain forward progress while reducing lateral drift. Such motion may offer useful guidance when one embodiment has limited experience recovering from unsafe situations. The challenge is to turn another embodiment's observed response into a correction that fits the target's current context and dynamics.

Offline reinforcement learning (RL) makes this opportunity particularly relevant: policies must learn from fixed data, without additional exploration to discover missing behaviors (Levine et al., 2020; Fujimoto et al., 2019; Fu et al., 2020). Learning beyond the support of these data is a central difficulty (Kumar et al., 2020; Kostrikov et al., 2022), and offline safe RL additionally requires the learned policy to respect a cost budget (Le et al., 2019; Xu et al., 2022; Lee et al., 2022; Liu et al., 2024). Return- and cost-conditioned sequence policies learn from trajectories under these objectives (Liu et al., 2023; Zhang et al., 2023; Su et al., 2025), but their ability to acquire corrective behavior depends on the available target experience.

Prior work shows that observations can support offline imitation across differences in dynamics or morphology (Ma et al., 2022), while demonstrations may be infeasible under the learner's dynamics (Cao et al., 2021). We ask whether such experience can provide useful safety corrections beyond

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.

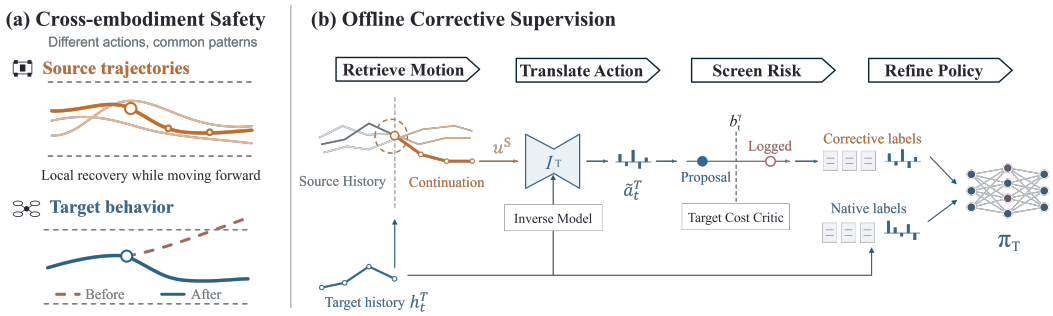


Figure 1: **Cross-embodiment safety transfer through corrective supervision.** (a) Embodiments with different actions and dynamics can share safety-relevant motion patterns, such as recovering from lateral drift while maintaining forward progress. (b) Supported source continuations are translated into target-native actions through inverse dynamics and screened by a target cost critic. Accepted corrections supplement native labels to refine the target policy offline.

what the target can learn from its own data. This requires translating source motion into target actions and assessing its risk in the receiving context.

To address this challenge, we propose *cross-embodiment corrective supervision (CECS)*, a new offline framework that converts safety-relevant motion from a source embodiment into target-native supervision (Figure 1). The method retrieves locally supported source motion, translates it through target inverse dynamics, and screens its predicted risk to construct corrective labels for offline policy refinement. We evaluate the approach across four morphologies and two task families. Comparisons with offline safe RL baselines measure the safety–reward trade-off, while matched-training controls identify the contribution of source corrections. Matched-state replay further examines how the learned supervision changes target motion. Our contributions are threefold:

- We propose *CECS*, a new offline framework that converts supported source motion into target-native corrective labels for offline policy refinement.
- We investigate cross-embodiment safety transfer across four morphologies and two task families, under both pairwise and multi-source settings. The results show that source-derived supervision can improve target-policy safety and achieve competitive safety–reward trade-offs against offline safe RL baselines.
- Through matched-training controls and matched-state replay, we isolate the value of source-derived corrections beyond target-only retrieval and show that context-matched supervision produces sustained reductions in outward drift and accumulated cost while retaining most task progress.

2 Related Work

Offline safe RL and cross-embodiment transfer. Offline safe RL learns policies from fixed data under cost constraints through cost-aware value and policy learning (Xu et al., 2022; Lee et al., 2022; Liu et al., 2025), as well as sequence and generative modeling (Liu et al., 2023; Zhang et al., 2023; Lin et al., 2023; Zheng et al., 2024; Su et al., 2025). Related risk-sensitive approaches learn conservative return distributions to support risk-averse offline policies (Ma et al., 2021). Data-sharing methods exploit experience beyond a single dataset (Yu et al., 2021; Ishfaq et al., 2024), while cross-domain offline RL selects source transitions according to their compatibility with target dynamics (Wen et al., 2024). Cross-embodiment offline RL further studies pre-training on heterogeneous robot data and conflicts between morphologies during joint learning (Abe et al., 2026).

Learning from observation and retrieval. Third-person and action-free imitation learn from observed behavior without directly reusing demonstrator actions (Stadie et al., 2017; Torabi et al., 2018), while correspondence methods relate behavior across views and embodiments (Dwibedi et al.,

2019; Zhang et al., 2021; Raychaudhuri et al., 2021; Zakka et al., 2022). Offline observation-based imitation further uses distribution matching, including in settings with mismatched dynamics or morphology (Kim et al., 2022; Ma et al., 2022). Recent work extends this direction through latent-action pretraining from unlabeled videos and transferable skill representations learned from human and robot observations (Ye et al., 2025; Kim et al., 2025). Retrieval-based policies use relevant stored experience to guide policy learning, ranging from nearest-neighbor actions to task-relevant sub-trajectories (Pari et al., 2022; Memmel et al., 2025). In contrast, we translate retrieved source motion into risk-screened, target-native corrective labels offline; the deployed policy requires neither source data nor online retrieval.

3 Problem Formulation

Constrained offline RL. For embodiment m, let $\mathcal{M}^m = (\mathcal{S}^m, \mathcal{A}^m, P^m, r^m, c^m, \gamma)$ be a constrained Markov decision process (Altman, 2021), with state space $\mathcal{S}^m$, action space $\mathcal{A}^m$, transition dynamics P^m, reward r^m, and cost c^m. For the target embodiment T, the objective is to maximize expected episodic return subject to a cost budget:

$$\max_{\pi} \; J_r(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{T_\tau} r_t \right] \quad \text{s.t.} \quad J_c(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{T_\tau} c_t \right] \leq B, \tag{1}$$

where T_τ is the final timestep of trajectory τ, and $\mathbb{E}_\pi$ denotes expectation under policy π and target dynamics P^T. We use undiscounted episodic cost $C(\tau) = \sum_t c_t$ and call an observed episode budget-feasible when $C(\tau) \leq B$. The constraint concerns expected cost; the feasible rate $F(\pi) = \Pr_{\tau \sim \pi, P^T}[C(\tau) \leq B]$ separately measures how often individual episodes satisfy the budget.

Cross-embodiment safety transfer. In the offline setting, learning uses a fixed target dataset $\mathcal{D}_T = \{\tau_i^T\}$ without additional environment interaction during training. We additionally consider a source dataset $\mathcal{D}_S = \{\tau_i^S\}$ collected by another embodiment performing the same task. Source and target share task and cost semantics but may differ in their observation spaces, action spaces, and dynamics. We assume that motion can be described in a task-defined frame shared by the two embodiments, allowing their observed behavior to be compared. Source experience is used during offline training to improve the target policy.

4 Methodology

As shown in Figure 1, we construct corrective supervision by retrieving observed source motion, translating it into target-native actions, and selecting corrections according to predicted target risk. Accepted actions complement native target supervision during offline policy refinement. We first describe the construction for a single source, then explain its extension to multiple source motion banks. Candidate correction contexts are drawn from budget-violating target episodes before their cumulative cost exceeds the budget:

$$\mathcal{A}_T = \{(\tau, t) : \tau \in \mathcal{D}_T,\ C(\tau) > B,\ C_{<t}(\tau) \leq B\}, \tag{2}$$

where $C_{<t}(\tau) = \sum_{k<t} c_k$. These contexts provide opportunities for correction; the target-risk test below determines where a translated target-native action is retained.

4.1 Cross-Embodiment Motion Retrieval

To compare behavior across different action spaces, we retrieve motion in a shared task frame. Let h_t^T denote the target observation–action history and q_t^T the causal task-space query extracted from it. Each source-motion-bank entry pairs a query q_i^S with a subsequent observed continuation u_i^S, extracted from a budget-feasible source episode. Queries summarize the preceding 200 ms of task-space motion. Run commands describe short future planar-velocity sequences, while Circle commands specify the normalized absolute waypoint reached after 2 m of source travel. Appendix A.1 provides more details.

We compare two retrieval metrics over the same physical queries and commands. *Explicit* retrieval measures Euclidean distance between standardized histories. *Predictive* retrieval uses a pair-shared

encoder $z = E_\phi(\bar{q})$, trained on budget-feasible source and target episodes with

$$\mathcal{L}_{\text{rep}} = \mathcal{L}_{\text{future}} + \lambda_{\text{recon}}\mathcal{L}_{\text{recon}} + \lambda_{\text{align}}\mathcal{L}_{\text{align}}. \tag{3}$$

Future prediction preserves information about subsequent motion, reconstruction retains the observed history, and alignment brings together histories with similar continuations (Dwibedi et al., 2019; Zhang et al., 2021). Retrieval uses only the causal history. Explicit query coordinates and predictive embedding coordinates are standardized using target-context statistics before nearest-neighbor search. Appendix B.2 specifies the encoder objectives and training details.

For a target query, K-nearest-neighbor retrieval with $K = 16$ returns source commands u_i^S and distances d_i. Their distance-weighted continuation is

$$u_t^S = \sum_{i=1}^{K} \alpha_i u_i^S, \qquad \alpha_i \propto \exp[-d_i/\operatorname{median}_j d_j], \qquad \sum_{i=1}^{K} \alpha_i = 1. \tag{4}$$

This construction retrieves motion for target-side translation (Pari et al., 2022). To restrict retrieval to locally supported contexts, each source row is assigned a radius equal to its distance to the Kth neighbor from another episode. A target query passes the support gate when its distance to the nearest source row is within that row's radius. The supported continuation specifies the requested motion; the next step is to express it as a target action.

4.2 Target-Native Correction Construction

The same task-space motion can require different actions under different dynamics. We therefore train a target-specific inverse model I_T on target histories, their observed continuations, and the corresponding logged actions (Torabi et al., 2018). It translates retrieved motion into a pseudo-action,

$$\widetilde{a}_t^T = I_T(h_t^T, u_t^S). \tag{5}$$

Training minimizes action mean squared error. Five episode-disjoint folds ensure that each candidate is labeled by a model trained without the corresponding episode. The inverse models are frozen before correction construction; their training data and architectures are specified in Appendix A.2.

A frozen target cost critic then evaluates the translated action. Its twin action-value networks estimate negative discounted cost-to-go, which we convert into a nonnegative risk score:

$$\widehat{Q}_c(s,a) = \max\left\{-\min_{j=1,2} \bar{Q}_{c,j}^-(s,a),\, 0\right\}. \tag{6}$$

The critic is trained only on target data. To compare its discounted prediction with the remaining episodic budget, we distribute the remaining raw budget uniformly over the critic's remaining horizon $L_t = T_{\text{critic}} - t$, giving the reference allowance

$$b_t^\gamma = (B - C_{<t})_+ \frac{1-\gamma^{L_t}}{L_t(1-\gamma)}. \tag{7}$$

We retain a correction when the logged action's predicted risk exceeds this allowance and the proposal's does not:

$$\widehat{Q}_c(s_t^T, a_t^T) > b_t^\gamma, \qquad \widehat{Q}_c(s_t^T, \widetilde{a}_t^T) \le b_t^\gamma. \tag{8}$$

Thus, local support identifies relevant source experience, while the risk test determines whether its translated action qualifies as a correction at the target context. Accepted actions form a fixed correction set $\mathcal{C}_T$ and retain their retrieval confidences. If no correction survives, the method abstains from source supervision. With multiple source motion banks, each target context uses the qualified correction with the smallest ratio of retrieval distance to local source support radius. This selects one source correction per context. The resulting labels enter the same policy-refinement procedure as in the single-source setting.

4.3 Corrective Policy Refinement

We use a stochastic return- and cost-conditioned sequence policy following CDT and B2R (Liu et al., 2023; Su et al., 2025). We adopt the B2R policy architecture and pretrain it from random initialization

for 100k updates using budget-feasible target episodes. The resulting checkpoint serves as the *safe-pretrained initialization*. All refinement variants start from this checkpoint and use 2k additional updates. During refinement, native supervision comes from $\mathcal{N}_T = \{(\tau, t) : \tau \in \mathcal{D}_T,\ C(\tau) \le B\}$, while $\mathcal{C}_T$ supplies corrective actions at the accepted target contexts. Both use target-side conditioning tokens $x_t = (h_t^T, \mathrm{RTG}_t^T, \mathrm{CTG}_t^T)$. For a pseudo-action batch $\mathcal{B}_p$ sampled from $\mathcal{C}_T$, we optimize

$$\mathcal{L}_{\mathrm{nat}} = -\mathbb{E}_{(x_t,a_t)\sim\mathcal{N}_T} \log \pi_\theta(a_t \mid x_t), \quad \mathcal{L}_{\mathrm{pseudo}} = -\frac{1}{Z_p} \sum_{t\in\mathcal{B}_p} w_t \log \pi_\theta(\widetilde{a}_t^T \mid x_t),$$
$$\mathcal{L} = \mathcal{L}_{\mathrm{nat}} + \mathcal{L}_{\mathrm{pseudo}} - \alpha \mathcal{H}_{\mathrm{nat}}. \tag{9}$$

Here $w_t = \exp(-d_{\mathrm{nearest}}/\sigma_d)$ is the retrieval confidence, and $Z_p = \max\{\sum_{t\in\mathcal{B}_p} w_t, 10^{-8}\}$ normalizes the corrective loss. The scale σ_d is calibrated using support-retained queries before critic filtering. Native likelihood fits the target's budget-feasible behavior, corrective likelihood adds source-derived action supervision, and entropy $\mathcal{H}_{\mathrm{nat}}$ is evaluated on native contexts. All return and cost conditions are derived from target data and target deployment settings. We refine the policy using the fixed correction set, without further source retrieval during policy execution. Appendix A specifies initialization, optimization, confidence calibration, and RTG/CTG construction.

Algorithm 1 summarizes the single-source procedure. For each candidate target context, it retrieves supported source motion, translates it into a target-native action, and retains the proposal only if it passes the target-risk test. The accepted corrections are combined with native target supervision for policy refinement; otherwise, the initial policy is left unchanged.

Algorithm 1 Cross-Embodiment Corrective Supervision (CECS)

Input: Target and source datasets $\mathcal{D}_T, \mathcal{D}_S$; budget B; retrieval metric; frozen inverse folds I_T^{-f} and target critic $\widehat{Q}_c$; initial target policy π_{θ_0}.

```
Form native contexts N_T and candidate contexts A_T; build the source motion bank; initialize C_T ← ∅.
For each (τ, t) ∈ A_T:
    Extract q_t^T from h_t^T and retrieve K source neighbors.
    If the query fails local source support, continue.
    Aggregate u_t^S using Equation 4.
    Translate ã_t^T = I_T^{-f(τ)}(h_t^T, u_t^S) using the inverse model excluding the episode's fold.
    Compute b_t^γ using Equation 7.
    If Equation 8 holds, add (x_t, ã_t^T, w_t) to C_T.
If C_T = ∅, return π_{θ_0}.
Refine π_{θ_0} using N_T, C_T, and Equation 9; return π_θ.
```

5 Experiments

Experimental setup. We evaluate the effectiveness of the proposed framework across two benchmark task families and four morphologies, using the Run and Circle tasks from Bullet Safety Gym (Gronauer, 2022) and the corresponding DSRL offline datasets (Liu et al., 2024) for Car, Ball, Ant, and Drone. We consider both fixed-source pairs and multiple source motion banks. We report DSRL-normalized return R, normalized episodic cost C, and feasible rate F. We set $B = 10$; mean normalized cost at or below one indicates empirical satisfaction of the expected-cost constraint. Policy performance is averaged across three training seeds, each evaluated on 200 episodes. All methods share evaluation seeds within each setting; pair and multi-source experiments use separate evaluation sets. Further details are provided in Appendices A and D.

Baselines. We compare against four offline safe RL methods: **CDT** (Liu et al., 2023), which conditions a sequence policy on return and cost; **B2R** (Su et al., 2025), which realigns cost conditioning to represent the remaining safety budget; **FISOR** (Zheng et al., 2024), which combines reachability-based feasibility learning with diffusion policy extraction; and **LSPC-O** (Koirala et al., 2025), which optimizes reward within a learned latent safety-constraint space. These baselines use target-only data, while our method additionally uses source trajectories. We report both *CECS–Explicit* and *CECS–Predictive*, corresponding to the retrieval metrics in Section 4.1.

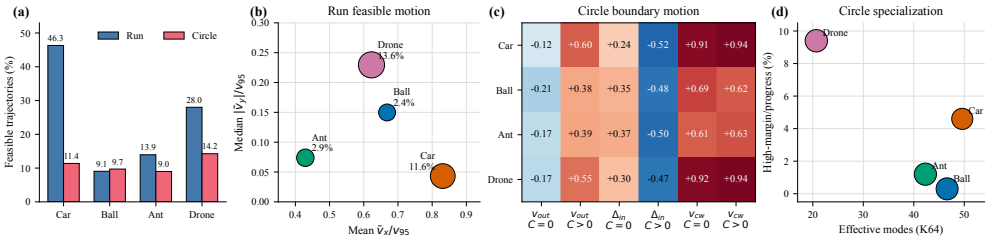


Figure 2: **Shared motion structure in offline data.** (a) Episode fractions with cost ≤ 10. (b) Run forward motion and lateral drift, using four-step mean velocities $\bar{v}_x, \bar{v}_y$; labels show efficient fractions. (c) Median outward velocity, inward displacement, and clockwise velocity near the boundary, grouped by zero or positive cost over the next 2 m. (d) Circle mode count versus high-margin, high-progress fraction. Panels (b,d) use locally feasible samples from budget-feasible episodes.

Table 1: Pair-transfer performance against target-only baselines (R/C). Both variants achieve mean normalized cost below the budget threshold in all directions; Predictive lowers average cost at the same reported reward. Bold marks the highest feasible reward in the Average column.

	Run			Circle		Average
Policy	Ball	Ant	Drone	Ball	Car	R/C
CDT	.311/.890	.717/1.268	.586/.528	.715/1.342	.714/1.078	.608/1.021
B2R	.319/3.154	.718/1.004	.508/1.595	.651/.837	.706/.977	.581/1.513
FISOR	.135/.563	.386/.193	.351/1.119	.338/.012	.363/.117	.314/.401
LSPC-O	.241/1.015	.866/11.840	.625/3.447	.670/2.031	.770/3.126	.634/4.292
CECS–Explicit	.203/.000	.718/.977	.528/.818	.500/.107	.398/.615	**.470**/**.504**
CECS–Predictive	.128/.001	.711/.802	.550/.179	.547/.132	.415/.761	**.470**/**.375**

5.1 Safety–Reward Performance

Shared motion and source selection. We examine the safety-relevant motion shared across datasets and use these observations to guide source selection for the pair study (Figure 2). In Run, Car has the highest fraction of budget-feasible episodes, with locally feasible motion concentrated around forward progress and limited lateral drift. In Circle, cost-free near-boundary segments tend to move inward while maintaining clockwise progress across embodiments. Drone has the largest fraction of locally feasible samples combining high boundary margin with clockwise progress. These observations motivate using CarRun and DroneCircle as sources in the pair-transfer study.

Pair transfer. Each target retrieves corrections from its designated same-task source. Table 1 reports the five transfer directions with qualified corrections; AntCircle has negligible correction coverage and is excluded from this comparison. The comparisons with target-only baselines show the resulting safety–reward trade-offs. On both Circle targets, our refined policies retain more reward than FISOR while keeping mean normalized cost below the budget threshold; FISOR achieves lower cost with less task progress. Across the pair-transfer settings, predictive retrieval achieves lower average cost than explicit retrieval at similar average reward, although the ranking varies by target.

Multi-source transfer. We next extend the fixed-source setting by allowing each target to retrieve from the other same-task embodiments, using the routing rule in Section 4.2. Table 2 reports results for the seven targets with qualified corrections. Both retrieval variants lower mean cost relative to the B2R policies and keep mean normalized cost below one on every reported target. They retain higher average reward than FISOR, which achieves lower average cost. These results establish the performance of corrective supervision across transfer settings.

5.2 Isolating the Value of Source Corrections

We examine whether source motion contributes beyond continued target-only training and retrieval from the target's own safe data. Using explicit retrieval, we compare CarRun→DroneRun and

Table 2: Multi-source transfer performance (R/C). Both variants achieve mean normalized cost below the budget threshold on all targets; Explicit favors lower cost and Predictive higher reward. Bold marks the highest feasible reward in the Average row.

Target	CDT	B2R	FISOR	LSPC-O	**CECS** Explicit	**CECS** Predictive
CarRun	.989/.795	.969/.105	.694/.002	.983/2.220	.944/.000	.928/.000
BallRun	.311/.886	.319/3.171	.137/.601	.244/.926	.049/.869	−.076/.531
AntRun	.717/1.284	.718/.987	.382/.130	.857/11.416	.717/.910	.710/.734
DroneRun	.587/.535	.507/1.399	.348/1.078	.625/3.353	.549/.247	.518/.622
BallCircle	.713/1.341	.645/.769	.344/.005	.670/2.023	.215/.013	.475/.089
CarCircle	.715/.960	.702/.950	.376/.107	.772/3.093	.351/.222	.373/.437
DroneCircle	.523/1.487	.462/.396	.477/.029	.593/2.513	.448/.156	.451/.200
Average	.650/1.041	.617/1.111	.394/.279	.678/3.649	.468/.345	**.483/.373**

Table 3: Matched-training attribution with explicit retrieval. Cross-intersection and Self use the same target contexts and correction counts.

	DroneRun (Car source)			BallCircle (Drone source)		
Control	$R\uparrow$	$C\downarrow$	$F\uparrow$	$R\uparrow$	$C\downarrow$	$F\uparrow$
Native	.483	2.587	43.7%	**.641**	.689	78.0%
Self	.496	2.129	49.3%	.517	.306	94.8%
Shuffled	.520	11.286	0.0%	.042	4.231	26.7%
Cross-intersection	**.551**	**.273**	**88.7%**	.510	**.101**	**99.7%**
Complete Matched	.528	.818	71.8%	.500	.107	**99.7%**

DroneCircle→BallCircle while holding target initialization, native batches, optimizer, update budget, and evaluation episodes fixed across controls.

Native continues training without pseudo-actions; **Self** retrieves budget-feasible target motion; **Shuffled** permutes qualified source pseudo-actions across fixed target candidates, preserving the action multiset and per-candidate confidence; **Cross-intersection** uses source corrections at the same target contexts and with the same correction count as Self; and **Complete Matched** uses the full source correction set. Appendix D provides the control implementation details.

At shared target contexts and equal correction counts, Cross-intersection improves safety over Self in both tasks, while increasing reward in DroneRun and largely preserving it in BallCircle (Table 3). These comparisons show that source motion supplies useful corrective content beyond target-only retrieval. Complete Matched also improves safety over Native and Self, although the shared-context subset achieves the lowest mean cost. Shuffling the same corrective actions substantially worsens safety, showing that their assignment to the receiving context is important. Together, these controls support the value of source content and context–correction matching.

5.3 Retrieval and Correction Quality

We next examine how retrieval and qualification shape the supervision received by the target policy.

Retrieval fidelity. We evaluate retrieval on episode-disjoint budget-feasible data. Command MSE measures how closely the weighted source continuation matches the target's observed motion; dispersion measures disagreement among retrieved commands; coverage measures the fraction of queries within local source support. Predictive retrieval reduces command MSE and dispersion in both task families, while increasing coverage in Circle and slightly decreasing it in Run (Table 4). These improvements accompany lower average cost in pair transfer, although the policy results in Section 5.1 show that their benefit depends on the target and source setting. Appendix B.2 provides the metric definitions and evaluation protocol.

Table 4: Held-out retrieval fidelity. Entries report Explicit/Predictive results, averaged equally across transfer directions.

Metric	Run	Circle
Command MSE ↓	.720/.579	.116/.031
Dispersion ↓	.114/.101	.040/.026
Coverage ↑	55.8/54.4%	33.1/55.7%

Table 5: Independent correction audit (rejected/retained): median motion discrepancy E and predicted cost-reduction fraction.

Target	Support E	Critic E	Critic $P(\Delta\hat{G} < 0)$ (%)
DroneRun	.439/.145	.151/.094	52.1/66.9
BallCircle	.186/.101	.091/.123	77.4/87.5
CarCircle	.267/.230	.244/.222	62.7/65.6

Table 6: Qualification ablation with explicit retrieval. Means give equal weight to the five transfer directions.

Condition	R ↑	C ↓	F ↑
Safe init.	.581	1.513	59.0%
Support only	.360	.161	97.5%
Critic only	.462	.467	84.8%
Full	.470	.504	83.3%

Independent correction audit. Episode-disjoint target predictors, separate from the inverse model and qualification critic, assess standardized motion discrepancy E from the requested command and predicted cost change $\Delta\hat{G}$ relative to the logged action. They are used only for diagnosis. Support-retained candidates have lower median discrepancy in all three targets (Table 5). Among supported candidates whose logged action exceeds the allowance, critic-retained proposals more often predict cost reductions. In BallCircle, this accompanies greater motion discrepancy, distinguishing motion fidelity from predicted risk. Appendix C provides predictor validation and the audit protocol.

Effect of qualification on policy learning. With explicit retrieval and the same refinement objective, support-only achieves the lowest mean cost and highest feasibility but the lowest reward (Table 6). Critic-only and Full retain more task progress while reducing cost relative to initialization; Full has slightly higher reward and cost than critic-only. Appendix D reports per-direction results.

5.4 How Corrections Change Target Behavior

We examine how source supervision changes the target policy, from the motion supplied by retrieval to learned action changes and physical responses. All analyses use the explicit-retrieval policies.

Transferred motion. Figure 3 compares retrieved source commands with the observed target continuations at accepted correction contexts. Positive vertical differences indicate more inward motion or boundary margin, while positive horizontal differences indicate more task progress. BallRun receives mostly slower, inward motion; DroneRun exhibits context-dependent lateral corrections. In Circle, Ball receives increased margin with some loss of clockwise progress, whereas Car receives greater progress with mixed margin changes. The source commands thus express different safety–progress trade-offs across contexts and embodiments.

Policy adoption. To measure whether the policy incorporates the translated corrections, let a_0 and a_r denote the initial and refined policy means, and let $\widetilde{a}$ denote the corrective action. Directional adoption measures policy displacement along the correction direction:

$$A(h) = \frac{(a_r - a_0)^\top (\widetilde{a} - a_0)}{\max\{\|\widetilde{a} - a_0\|_2^2, 10^{-8}\}}. \tag{10}$$

Zero indicates no displacement along that direction, while one indicates that the projected displacement reaches the correction. As a comparison, we project the same policy displacement onto post-hoc permuted labels while holding the policy fixed. Adoption exceeds the permuted-label comparison in both tasks and is positive at nearly all correction contexts (Table 7). Policy displacement is also larger at correction contexts than on the safe reference set. These results show that refinement shifts target actions toward the corresponding corrective labels.

Matched-state response. We evaluate sustained physical behavior from 200 BallCircle contexts in the source–self retrieval intersection, initializing all policy branches from identical simulator

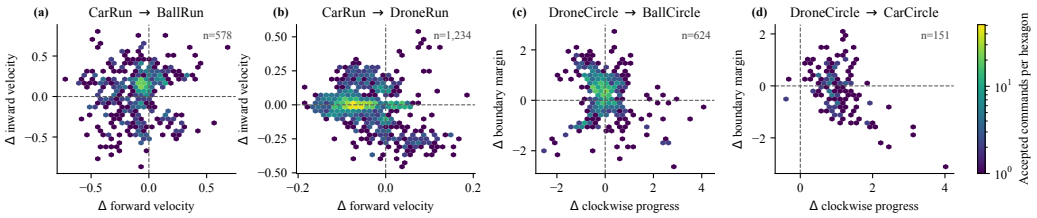


Figure 3: **Accepted source commands in the shared task frame.** Hexagons bin retrieved-minus-native motion at accepted target contexts; color indicates counts on a log scale. These commands supply the motion information used before inverse translation and policy refinement.

Table 7: Policy adoption and localization. Adoption and null entries report means ± SD of within-seed medians. Positive fractions pool correction contexts across seeds. Localization reports the displacement ratio between correction and safe reference contexts, averaged across seeds.

Transfer	Adoption	Permuted null	Positive	Localization
CarRun→DroneRun	.669 ± .013	.011 ± .002	97.5%	2.374 ± .069
DroneCircle→BallCircle	.902 ± .014	.249 ± .021	98.6%	1.859 ± .050

snapshots, policy histories, and conditioning tokens. Across three training seeds, the cost advantage of Matched over Self grows with the rollout horizon (Figure 4). At 4 s, Matched reduces mean accumulated cost from 1.462 to .242, with every seed favoring Matched at both 2 and 4 s. It also reduces outward drift from .653 m to .389 m while retaining 86.9% of Self's clockwise progress. Thus, the safety gain reflects sustained boundary correction rather than motion suppression.

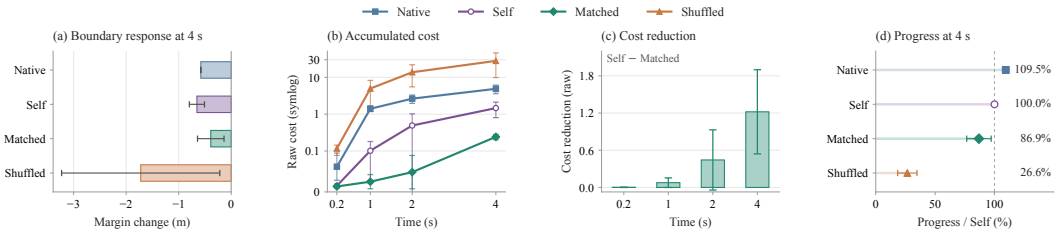


Figure 4: **BallCircle matched-state response over time.** (a) Boundary-margin change at 4 s. (b) Accumulated raw cost (symlog). (c) Self-minus-Matched cost. (d) Within-seed clockwise progress relative to Self at 4 s. Results show means ± SD across three training seeds and 200 common starts. 180 starts admit the full 4 s window.

6 Discussion and Conclusion

This work studies whether observed motion from another embodiment can provide useful safety corrections for an offline target policy. Cross-embodiment corrective supervision converts source continuations into target-native training labels for policy refinement. Experiments show safety gains beyond target-only retrieval at shared contexts, while shuffled controls demonstrate the importance of matching corrections to the receiving context. Matched-state replay shows that the refined policy reduces accumulated cost through sustained control while retaining most task progress. The results also show that motion similarity alone does not determine transfer benefit: qualification and policy learning both matter. These findings suggest that the safety value of an offline dataset extends beyond the embodiment that generated it. Several limitations remain. The study is restricted to simulated tasks and task-defined motion interfaces. Moreover, the critic provides empirical screening rather than formal safety guarantees.

AI USE STATEMENT

The authors used generative AI tools to assist in drafting and editing portions of the manuscript, including correcting grammatical issues, refining wording, and improving the clarity, conciseness, and organization of the text. All AI-assisted content was reviewed and revised by the authors to verify its technical accuracy and ensure that it faithfully conveyed the intended methods, results, and conclusions. The authors take full responsibility for the final manuscript, including all text, claims, and results.

REFERENCES

Haruki Abe, Takayuki Osa, Yusuke Mukuta, and Tatsuya Harada. Cross-embodiment offline reinforcement learning for heterogeneous robot datasets. In *The Fourteenth International Conference on Learning Representations*, 2026.

Eitan Altman. *Constrained Markov Decision Processes*. Routledge, 2021.

Zhangjie Cao, Yilun Hao, Mengxi Li, and Dorsa Sadigh. Learning feasibility to imitate demonstrators with different dynamics. *arXiv preprint arXiv:2110.15142*, 2021.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems*, 2021.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1801–1810, 2019.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pp. 2052–2062. PMLR, 2019.

Sven Gronauer. Bullet-safety-gym: A framework for constrained reinforcement learning. Technical report, Technical University of Munich, 2022.

Haque Ishfaq, Thanh Nguyen-Tang, Songtao Feng, Raman Arora, Mengdi Wang, Ming Yin, and Doina Precup. Offline multitask representation learning for reinforcement learning. *Advances in Neural Information Processing Systems*, 37:70557–70616, 2024.

Geon-Hyeong Kim, Jongmin Lee, Youngsoo Jang, Hongseok Yang, and Kee-Eung Kim. Lobsdice: Offline learning from observation via stationary distribution correction estimation. *Advances in Neural Information Processing Systems*, 35:8252–8264, 2022.

Hanjung Kim, Jaehyun Kang, Hyolim Kang, Meedeum Cho, Seon Joo Kim, and Youngwoon Lee. Uniskill: Imitating human videos via cross-embodiment skill representations. *arXiv preprint arXiv:2505.08787*, 2025.

Prajwal Koirala, Zhanhong Jiang, Soumik Sarkar, and Cody Fleming. Latent safety-constrained policy approach for safe offline reinforcement learning. In *International Conference on Learning Representations*, volume 2025, pp. 61341–61367, 2025.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020.

Hoang Le, Cameron Voloshin, and Yisong Yue. Batch policy learning under constraints. In *International Conference on Machine Learning*, pp. 3703–3712. PMLR, 2019.

Jongmin Lee, Cosmin Paduraru, Daniel J. Mankowitz, Nicolas Heess, Doina Precup, Kee-Eung Kim, and Arthur Guez. COptiDICE: Offline constrained reinforcement learning via stationary distribution correction estimation. In *International Conference on Learning Representations*, 2022.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Qian Lin, Bo Tang, Zifan Wu, Chao Yu, Shangqin Mao, Qianlong Xie, Xingxing Wang, and Dong Wang. Safe offline reinforcement learning with real-time budget constraints. In *International Conference on Machine Learning*, pp. 21127–21152. PMLR, 2023.

Zifan Liu, Xinran Li, and Jun Zhang. C2iql: Constraint-conditioned implicit q-learning for safe offline reinforcement learning. In *International Conference on Machine Learning*, pp. 38827–38841. PMLR, 2025.

Zuxin Liu, Zijian Guo, Yihang Yao, Zhepeng Cen, Wenhao Yu, Tingnan Zhang, and Ding Zhao. Constrained decision transformer for offline safe reinforcement learning. In *International Conference on Machine Learning*, pp. 21611–21630. PMLR, 2023.

Zuxin Liu, Zijian Guo, Haohong Lin, Yihang Yao, Jiacheng Zhu, Zhepeng Cen, Hanjiang Hu, Wenhao Yu, Tingnan Zhang, Jie Tan, and Ding Zhao. Datasets and benchmarks for offline safe reinforcement learning. *Journal of Data-centric Machine Learning Research*, 2024.

Yecheng Ma, Dinesh Jayaraman, and Osbert Bastani. Conservative offline distributional reinforcement learning. *Advances in neural information processing systems*, 34:19235–19247, 2021.

Yecheng Ma, Andrew Shen, Dinesh Jayaraman, and Osbert Bastani. Versatile offline imitation from observations and examples via regularized state-occupancy matching. In *International Conference on Machine Learning*, pp. 14639–14663. PMLR, 2022.

Marius Memmel, Jacob Berg, Bingqing Chen, Abhishek Gupta, and Jonathan Francis. Strap: Robot sub-trajectory retrieval for augmented policy learning. In *International Conference on Learning Representations*, volume 2025, pp. 63813–63832, 2025.

Jyothish Pari, Nur Muhammad Shafiullah, Sridhar Pandian Arunachalam, and Lerrel Pinto. The surprising effectiveness of representation learning for visual imitation. In *Robotics: Science and Systems*, 2022.

Dripta S Raychaudhuri, Sujoy Paul, Jeroen Vanbaar, and Amit K Roy-Chowdhury. Cross-domain imitation from observations. In *International Conference on Machine Learning*, pp. 8902–8912. PMLR, 2021.

Bradly C. Stadie, Pieter Abbeel, and Ilya Sutskever. Third-person imitation learning. In *International Conference on Learning Representations*, 2017.

Huikang Su, Dengyun Peng, Zifeng Zhuang, Yuhan Liu, Qiguang Chen, Donglin Wang, and Qinghe Liu. Boundary-to-region supervision for offline safe reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 57917–57946, 2025.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In *International Joint Conference on Artificial Intelligence*, pp. 4950–4957, 2018.

Xiaoyu Wen, Chenjia Bai, Kang Xu, Xudong Yu, Yang Zhang, Xuelong Li, and Zhen Wang. Contrastive representation for data filtering in cross-domain offline reinforcement learning. *arXiv preprint arXiv:2405.06192*, 2024.

Haoran Xu, Xianyuan Zhan, and Xiangyu Zhu. Constraints penalized q-learning for safe offline reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8753–8760, 2022.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. In *International Conference on Learning Representations*, volume 2025, pp. 28213–28239, 2025.

Tianhe Yu, Aviral Kumar, Yevgen Chebotar, Karol Hausman, Sergey Levine, and Chelsea Finn. Conservative data sharing for multi-task offline reinforcement learning. *Advances in Neural Information Processing Systems*, 34:11501–11516, 2021.

Kevin Zakka, Andy Zeng, Pete Florence, Jonathan Tompson, Jeannette Bohg, and Debidatta Dwibedi. Xirl: Cross-embodiment inverse reinforcement learning. In *Conference on Robot Learning*, pp. 537–546. PMLR, 2022.

Qiang Zhang, Tete Xiao, Alexei A. Efros, Lerrel Pinto, and Xiaolong Wang. Learning cross-domain correspondence for control with dynamics cycle-consistency. In *International Conference on Learning Representations*, 2021.

Qin Zhang, Linrui Zhang, Li Shen, Haoran Xu, Bowen Wang, Bo Yuan, Yongzhe Chang, and Xueqian Wang. Saformer: A conditional sequence modeling approach to offline safe reinforcement learning. In *International Conference on Learning Representations 2023 Workshop on Scene Representations for Autonomous Driving*, 2023.

Yinan Zheng, Jianxiong Li, Dongjie Yu, Yujie Yang, Shengbo Li, Xianyuan Zhan, and Jingjing Liu. Safe offline reinforcement learning with feasibility-guided diffusion model. In *International Conference on Learning Representations*, volume 2024, pp. 3838–3867, 2024.

Table 8: Task-family interfaces and training data.

Component	Run	Circle
Query	Three-sample, 200 ms history of normalized $[y, v_x, v_y]$	Four-sample, 200 ms history of normalized world coordinates $[x, y, v_x, v_y]$
Command	Short future planar-velocity sequence	Normalized absolute waypoint after 2 m of source travel
Source motion bank	Budget-feasible source episodes	Budget-feasible source episodes
Inverse data	Budget-feasible target episodes	All valid target transitions

A Implementation Details

This appendix specifies the task interfaces, target-side models, and optimization settings used in Section 4. All models are trained from the fixed offline datasets. Inverse models and cost critics are frozen before corrective labels are constructed.

A.1 Task Interfaces and Data Construction

Run and Circle queries summarize task-space position and velocity over the preceding 200 ms. Commands are extracted from the source's recorded continuation. Table 8 summarizes the task-family interfaces and training data. Source motion banks and native policy supervision use budget-feasible episodes. Run inverse models also use budget-feasible target episodes, whereas Circle inverse models use all valid target transitions. This Circle configuration follows held-out diagnostics in which safe-only inverse models were nearly command-insensitive.

Retrieval uses $K = 16$ neighbors and the metric-specific support rule in Section 4.1. For both retrieval metrics, the confidence scale σ_d is the median positive nearest-neighbor distance among support-retained queries, computed before critic filtering. When the support gate is disabled, calibration uses all candidate queries. Explicit retrieval shares one correction set per direction across policy-training seeds; predictive retrieval fits an encoder and reconstructs the correction set for each seed. Appendix B.2 gives the predictive encoder configuration and objectives.

A.2 Inverse Dynamics and Target Cost Critics

Inverse dynamics. Each target uses a deterministic MLP with three hidden layers of width 256, SiLU activations, and a tanh output of dimension $\dim(\mathcal{A}^T)$. Its input concatenates the target observation history, preceding actions when available, and the requested motion:

$$[s^T_{t-\ell+1:t},\ a^T_{t-\ell+1:t-1},\ u_t].$$

The current action a^T_t is the training label. Input features are standardized using training-fold means and standard deviations, with a standard-deviation floor of 10^{-4}. For observed target continuations, training minimizes

$$\mathcal{L}_I = \mathbb{E}_{(h_t,u_t,a^T_t)\sim\mathcal{D}^I_T}\left[\frac{\|I_T(h_t,u_t) - a^T_t\|_2^2}{\dim(\mathcal{A}^T)}\right], \tag{11}$$

where $\mathcal{D}^I_T$ follows the task-family data selection in Table 8, and u_t uses the same command definition as source retrieval.

Episodes are assigned to five folds by episode index modulo five. Teacher I_T^{-f} and its feature statistics are fitted on the other four folds and used to label contexts in fold f. We retain the final epoch checkpoint and use held-out action error for diagnosis. Optimization uses AdamW with learning rate 3×10^{-4} and weight decay 10^{-5}. Inverse-model histories are aligned by physical duration across Circle embodiments. Run uses fixed history lengths, with a longer history for Drone. These model inputs are distinct from the 200 ms task-space queries used for retrieval.

Target cost critics. Cost critics are trained with implicit Q-learning (Kostrikov et al., 2022) on the full target offline dataset, including positive-cost transitions. Each critic contains twin scalar action-value networks and a scalar state-value network. All three networks have two hidden layers of

width 512, ReLU activations, and linear outputs. Action-value networks receive $[s_t^T, a_t^T]$, while the value network receives s_t^T.

The networks estimate negative discounted cost returns. Denote them by $Q_{c,1}^-, Q_{c,2}^-, V_c^-$, with target action-value networks $\bar{Q}_{c,j}^-$. Training uses

$$\begin{aligned} \delta_t &= \min_{j=1,2} \bar{Q}_{c,j}^-(s_t, a_t) - V_c^-(s_t), \\ \mathcal{L}_V &= \mathbb{E}\left[|\eta - \mathrm{I}\{\delta_t < 0\}|\, \delta_t^2\right], \qquad \eta = 0.7, \\ y_t &= -c_t + \gamma(1 - d_t) V_c^-(s_{t+1}), \end{aligned} \tag{12}$$

$$\mathcal{L}_Q = \sum_{j=1}^{2} \mathbb{E}\left[\left(Q_{c,j}^-(s_t, a_t) - \mathrm{stopgrad}(y_t)\right)^2\right]. \tag{13}$$

Here $\mathrm{I}\{\cdot\}$ is the indicator function, and d_t marks a terminal transition or timeout. Target-Q values are detached in $\mathcal{L}_V$. Training uses unscaled raw costs, discount $\gamma = .99$, Adam with learning rate 3×10^{-4}, batch size 512, and 100k updates. Target-Q parameters follow

$$\bar{\theta}_{Q,j} \leftarrow (1 - \tau)\bar{\theta}_{Q,j} + \tau\theta_{Q,j}, \qquad \tau = .005.$$

The frozen seed-0 checkpoint supplies the risk score in Equation 6. Unlike the inverse models, the qualification critic is trained on the full target dataset without episode-disjoint cross-fitting. The independent diagnostic predictors used to audit corrections are described separately in Appendix C.

A.3 Policy Optimization and Conditioning

Following conditional sequence modeling (Chen et al., 2021; Liu et al., 2023; Su et al., 2025), the target policy predicts actions from target history and return–cost conditions. We adopt the B2R architecture with context length three and a diagonal-Gaussian action head whose mean is bounded by tanh. The global log standard deviation is learned; corrective likelihood updates both the mean and log standard deviation.

The policy is pretrained from random initialization for 100k updates on budget-feasible target episodes, followed by 2k refinement updates. Refinement uses AdamW with native batch size $B_n = 512$, learning rate 10^{-5}, weight decay 10^{-4}, and gradient-norm clipping at .25. All task families, retrieval metrics, and source motion settings use the confidence-normalized corrective loss in Equation 9. Native continuation omits the corrective term.

Entropy is evaluated on native contexts. Starting from $\alpha = .1$, we optimize $\log \alpha$ with Adam at learning rate 10^{-4} using

$$\mathcal{L}_\alpha = \alpha\, \mathrm{stopgrad}\left(\mathcal{H}_{\mathrm{nat}} - \mathcal{H}_{\mathrm{target}}\right), \qquad \mathcal{H}_{\mathrm{target}} = -\dim(\mathcal{A}^T).$$

Conditioning uses only target rewards and costs. Native labels use the observed episode return, while corrective labels use the deployment target return. Given observed return $R(\tau)$, deployment target $R_T^\star$, and scaling divisors s_R, s_C, the tokens are

$$\mathrm{RTG}_{t,\mathrm{nat}}^T = \frac{R(\tau) - \sum_{u<t} r_u^T}{s_R}, \quad \mathrm{RTG}_{t,\mathrm{pseudo}}^T = \frac{R_T^\star - \sum_{u<t} r_u^T}{s_R}, \quad \mathrm{CTG}_t^T = \frac{(B - C_{<t})_+}{s_C}.$$

The same target-policy architecture is retained throughout refinement and deployment.

B Dataset and Retrieval Details

B.1 Dataset Composition and Descriptive Statistics

Table 9 reports the number of budget-feasible episodes in each dataset. An episode is counted as feasible when its total raw cost satisfies $C(\tau) \le B = 10$. Circle counts use complete episodes from the raw HDF5 datasets.

Figure 5 illustrates the task geometry underlying the shared motion frame. Run rewards forward progress subject to lane and speed constraints, so lane position, forward velocity, and lateral drift

Table 9: Budget-feasible episode counts and fractions in the offline datasets.

	Run		Circle	
Morphology	Feasible/total	Fraction	Feasible/total	Fraction
Car	301/650	46.3%	165/1,450	11.4%
Ball	85/939	9.1%	86/886	9.7%
Ant	253/1,815	13.9%	515/5,728	9.0%
Drone	557/1,989	28.0%	274/1,923	14.2%

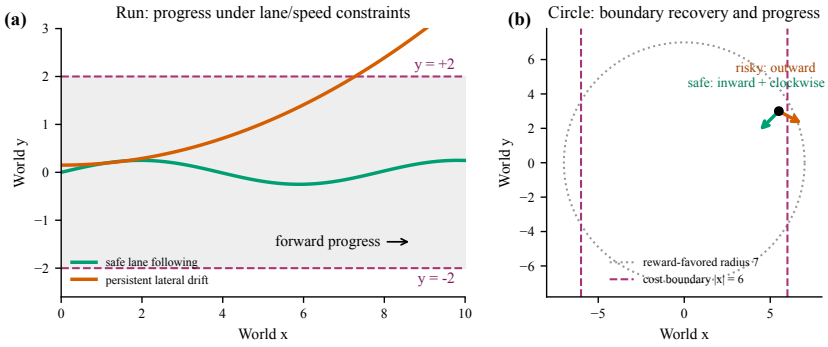


Figure 5: **Task geometry and shared motion coordinates.** Run combines forward progress with lane and speed constraints. Circle rewards clockwise progress, with cost incurred beyond $|x| = 6$. Arrows illustrate motion directions; corrective commands are extracted from observed source trajectories.

provide comparable motion quantities across embodiments. Circle rewards clockwise progress, with cost incurred beyond the vertical boundaries $|x| = 6$. Its boundary margin therefore depends on the world-coordinate position x, rather than distance to the circle center. Future waypoints describe displacement along the turning path.

The descriptive analysis in Figure 2 uses four-step mean velocities for Run. For Circle, near-boundary segments are grouped by whether they incur zero or positive cost over the following 2 m of path, and compared through outward velocity, inward displacement, and clockwise velocity. The Run motion and Circle repertoire summaries use locally feasible samples from budget-feasible episodes. These descriptive quantities characterize the available motion; the retrieval interfaces are specified separately in Appendix A.1.

B.2 Predictive Retrieval and Evaluation

Predictive metric. The predictive metric uses a pair-shared encoder trained on the same causal task-space histories as explicit retrieval. It produces a unit-normalized six-dimensional embedding $z = E_\phi(\bar{q})$. Training uses budget-feasible source and target episodes, with equal-sized batches from the two embodiments.

The objective in Equation 3 combines future-command prediction, history reconstruction, and alignment, with $\lambda_{\text{recon}} = 0.25$ and $\lambda_{\text{align}} = 0.1$. Let $\bar{q}^m$ and $\bar{u}^m$ denote histories and future commands standardized using domain-balanced training statistics, and let G_q and G_u be their decoders. With $\ell_2(v) = \|v\|_2^2/\dim(v)$, the loss terms are

$$\begin{aligned}
\mathcal{L}_{\text{future}} &= \sum_{m\in\{T,S\}} \mathbb{E}\,\ell_2(G_u(E_\phi(\bar{q}^m)) - \bar{u}^m)\,, \\
\mathcal{L}_{\text{recon}} &= \sum_{m\in\{T,S\}} \mathbb{E}\,\ell_2(G_q(E_\phi(\bar{q}^m)) - \bar{q}^m)\,, \\
\mathcal{L}_{\text{align}} &= \mathbb{E}\,\ell_2(z^T - z^S_{j(T)}) + \mathbb{E}\,\ell_2(z^S - z^T_{i(S)}).
\end{aligned} \tag{14}$$

Here $j(T)$ and $i(S)$ select the nearest opposite-embodiment future command within a minibatch. Alignment brings together histories followed by similar motion. Future commands provide training supervision, while retrieval receives only the causal history. Before nearest-neighbor search, embedding coordinates are standardized using target-context statistics.

Retrieval evaluation. We compare the two metrics using identical target queries, source rows, and observed future commands, with evaluation episodes disjoint from encoder training. Commands are standardized using the domain-balanced training statistics above. For query t, let

$$\widehat{\bar{u}}_t = \sum_{i=1}^{K} \alpha_i \bar{u}_i^S$$

be the weighted mean of the $K = 16$ retrieved source commands, with weights defined in Equation 4. Command MSE and dispersion are

$$\begin{aligned} \text{MSE} &= \mathbb{E}_t \, \ell_2\big(\widehat{\bar{u}}_t - \bar{u}_t^T\big) \,, \\ \text{Dispersion} &= \mathbb{E}_t \sum_{i=1}^{K} \alpha_i \, \ell_2\big(\bar{u}_i^S - \widehat{\bar{u}}_t\big) \,. \end{aligned} \tag{15}$$

MSE measures agreement with the target's observed continuation, while dispersion measures agreement among retrieved source commands. Both statistics include all validation queries.

Coverage is the fraction of validation queries accepted by local source support. In each metric space, a source row's radius is its distance to the Kth neighbor from another episode. A query passes when its distance to the nearest source row is within that row's radius. Radii are recomputed separately for the explicit and predictive metrics using the same cross-episode rule. The task-family summaries in Table 4 average equally across the evaluated transfer directions.

C Correction Qualification and Independent Audit

C.1 Qualification outcomes

Table 10 reports how support and target-risk filtering determine the correction set. Support and Critic-only are applied independently to the raw candidates, whereas Full applies the critic after support. Inverse translation supplies the proposed actions and is not a separate count gate.

Table 10: Correction qualification with explicit retrieval. Support/raw and Critic/raw report acceptance when each gate is applied independently. Full/support reports critic acceptance among supported candidates.

Direction	Raw candidates	Support/raw	Critic/raw	Full/support	Full
CarRun→BallRun	41,344	53.35%	2.73%	2.62%	578
CarRun→AntRun	138,226	23.61%	.03%	.08%	26
CarRun→DroneRun	163,076	65.04%	.90%	1.16%	1,234
DroneCircle→BallCircle	50,947	45.21%	2.29%	2.71%	624
DroneCircle→CarCircle	137,198	10.00%	1.82%	1.10%	151

Among supported candidates, the critic distinguishes three outcomes: the logged action is already within the allowance, both the logged action and proposal exceed it, or the proposal satisfies the allowance where the logged action does not. Table 11 separates these outcomes. CarCircle rejects candidates mainly because the logged action requires no intervention, whereas DroneRun rejects them mainly because the proposal remains above the allowance.

C.2 Independent audit protocol and validation

We assess candidate quality using target-only predictors separate from the inverse models and qualification critic. Each predictor is a three-layer MLP that receives three observations, the preceding two actions, and the action being assessed, and forecasts subsequent motion and discounted cost.

Table 11: Risk-filter outcomes among supported candidates. No intervention: the logged action is within the allowance. Proposal rejected: both actions exceed it. Retained: only the proposal satisfies it. Percentages use supported candidates as the denominator.

Target	Supported	No intervention (%)	Proposal rejected (%)	Retained (%)
BallRun	22,055	7.9	89.5	2.62
AntRun	32,641	.2	99.7	.08
DroneRun	106,065	10.1	88.7	1.16
BallCircle	23,031	22.4	74.9	2.71
CarCircle	13,725	80.3	18.6	1.10

For each of five episode folds, one fold is evaluated, the next fold calibrates the diagnostics, and the remaining three train a two-member ensemble. Training uses observed target transitions; source commands and pseudo-actions are used only during evaluation.

We audit every raw candidate in DroneRun, BallCircle, and CarCircle. The DroneRun motion target is an eight-step velocity sequence; the Circle target is the absolute waypoint reached after 2 m of path. Forecast units and horizons match the retrieved commands. BallRun is excluded because its available predictor forecasts one step, whereas its corrective commands span four steps.

Let $F_T(h, \widetilde{a})$ denote the ensemble-mean motion forecast. Motion discrepancy and predicted cost change are

$$E(h, \widetilde{a}, u^S) = \frac{\left\| [F_T(h, \widetilde{a}) - u^S]/\sigma_u \right\|_2}{\sqrt{\dim(u)}}, \tag{16}$$

$$\Delta\widehat{G} = \widehat{G}(h, \widetilde{a}) - \widehat{G}(h, a), \tag{17}$$

where σ_u contains coordinate scales fitted on the training folds. Lower E indicates closer predicted agreement with the requested motion, and $\Delta\widehat{G} < 0$ indicates lower predicted cost than the logged action. These diagnostics do not alter correction selection.

Table 12 reports factual held-out prediction accuracy. Absolute-waypoint R^2 includes predictability from the current position. Motion and cost forecasts follow logged continuations after the assessed action; the audit therefore measures predicted compatibility, while Section 5.4 examines realized physical responses.

Table 12: Factual held-out accuracy of the independent predictors. Command R^2 uses the forecast motion coordinates, E uses training-fold standardization, and cost MAE is measured in raw discounted-cost units.

Target	Held-out rows	Command R^2	Median E	Cost MAE
DroneRun	250,000	.997	.023	7.49
BallCircle	168,818	.997	.023	7.09
CarCircle	250,000	.997	.016	7.92

Group comparisons. For support, we compare retained and rejected raw candidates. For the critic, we restrict the comparison to supported candidates whose logged action exceeds the allowance. Table 13 supplements the main-text median errors with uncertainty in mean-error differences and cost-reduction fractions. We jointly resample source episodes for both groups and recompute row-weighted statistics, keeping the fitted predictors fixed.

Support-retained candidates have lower mean motion error in all three targets. Critic-retained candidates more often predict a cost reduction, with positive intervals in DroneRun and BallCircle; the CarCircle interval overlaps zero.

Paired-action comparison. We also hold the target context and source command fixed and compare the logged action with its translated replacement. Table 14 reports their respective errors, the mean paired gain $E_{\text{native}} - E_{\text{pseudo}}$, and the fraction of candidates with positive gain. Circle translations

Table 13: Retained-minus-rejected differences with episode-cluster bootstrap 95% intervals. The support column reports the interval for mean E. Critic columns report the difference in $P(\Delta\widehat{G} < 0)$ in percentage points.

Target	Support: mean-E interval	Critic: difference (pp)	Critic: interval (pp)
DroneRun	$[-.376, -.350]$	$+14.9$	$[12.2, 17.6]$
BallCircle	$[-.092, -.080]$	$+10.1$	$[7.0, 13.1]$
CarCircle	$[-.030, -.020]$	$+2.9$	$[-5.6, 11.5]$

Table 14: Paired command-matching audit on retained corrections. Both actions are assessed against the same source command at the same target context. Positive mean gain indicates lower error after translation.

Target	Candidates	Median E_{native}	Median E_{pseudo}	Mean gain	Improved (%)
DroneRun	1,234	.0940	.0942	$-.0001$	47.7
BallCircle	624	.1462	.1232	$+.0194$	81.7
CarCircle	151	.2328	.2216	$+.0117$	68.9

improve predicted command matching, whereas DroneRun's mean paired change is near zero. Thus, lower error across support-selected groups and improved matching from changing the action are separate effects.

D Additional Policy Results

D.1 Complete policy results

Within each setting, all methods use training seeds $0, 1, 2$ and the same 200 environment seeds: 70000–70199 for pair transfer and 71000–71199 for multi-source transfer. The baseline checkpoints are unchanged between settings; their reported results differ because the evaluation episodes differ. Table 15 provides complete baseline results, and Table 16 supplements the main-text reward–cost comparisons with feasible rates.

D.2 Control implementation and qualification ablations

The attribution controls share target initialization, native batches, optimizer, update budget, and evaluation episodes. Native uses no corrective labels; Self retrieves budget-feasible target motion; Cross-intersection uses source corrections at the same candidate contexts and with the same correction count as Self; and Complete Matched uses the full qualified source set. Shuffled permutes qualified pseudo-actions across fixed candidate contexts while preserving the action multiset and each context's confidence. This permutation is applied after qualification, without rerunning the gates.

In the multi-source setting, routing selects the qualified correction with the smallest ratio of retrieval distance to local source support radius. Each selected correction remains target-native; routing does not average actions across source embodiments.

Qualification ablations retain the same inverse translation and policy optimization while changing the selection gates. Support-only omits target-risk filtering, Critic-only omits support, Full uses both, and Raw uses neither. Table 17 gives direction-level results underlying the main-text aggregate comparison.

Figure 6 additionally includes Raw refinement for DroneRun and CarCircle. The gates change the safety–reward balance: for example, target-risk filtering restores substantial DroneRun progress relative to Support-only, while retaining lower cost than the safe initialization.

Table 15: Complete target-only baseline results. Values are means ± one standard deviation across three training seeds. R and C denote normalized reward and cost; F is the percentage of budget-feasible episodes. Dashes indicate targets outside the reported pair-transfer setting.

		Pair transfer			Multi-source transfer		
Target	Method	R	C	F (%)	R	C	F (%)
CarRun	CDT	—	—	—	$.989 \pm .000$	$.795 \pm .231$	75.3 ± 15.6
	B2R	—	—	—	$.969 \pm .001$	$.105 \pm .180$	$99.8 \pm .3$
	FISOR	—	—	—	$.694 \pm .016$	$.002 \pm .004$	100.0 ± 0.0
	LSPC-O	—	—	—	$.983 \pm .016$	2.220 ± 3.285	63.0 ± 54.0
BallRun	CDT	$.311 \pm .003$	$.890 \pm .090$	98.7 ± 2.3	$.311 \pm .002$	$.886 \pm .087$	99.3 ± 1.2
	B2R	$.319 \pm .011$	3.154 ± 3.529	33.3 ± 57.7	$.319 \pm .011$	3.171 ± 3.529	33.3 ± 57.7
	FISOR	$.135 \pm .097$	$.563 \pm .894$	76.8 ± 37.1	$.137 \pm .099$	$.601 \pm .981$	74.7 ± 42.2
	LSPC-O	$.241 \pm .100$	1.015 ± 1.757	72.3 ± 47.9	$.244 \pm .105$	$.926 \pm 1.604$	72.5 ± 47.6
AntRun	CDT	$.717 \pm .005$	$1.268 \pm .093$	29.5 ± 9.5	$.717 \pm .002$	$1.284 \pm .062$	28.8 ± 4.6
	B2R	$.718 \pm .009$	$1.004 \pm .359$	63.0 ± 27.4	$.718 \pm .010$	$.987 \pm .352$	63.0 ± 31.6
	FISOR	$.386 \pm .034$	$.193 \pm .119$	97.7 ± 2.3	$.382 \pm .038$	$.130 \pm .085$	97.8 ± 2.3
	LSPC-O	$.866 \pm .027$	11.840 ± 1.230	0.0 ± 0.0	$.857 \pm .024$	11.416 ± 1.182	0.0 ± 0.0
DroneRun	CDT	$.586 \pm .007$	$.528 \pm .385$	77.5 ± 22.3	$.587 \pm .006$	$.535 \pm .403$	76.5 ± 22.1
	B2R	$.508 \pm .103$	1.595 ± 1.656	60.2 ± 49.3	$.507 \pm .103$	1.399 ± 1.619	62.8 ± 51.1
	FISOR	$.351 \pm .058$	$1.119 \pm .446$	73.5 ± 13.0	$.348 \pm .061$	$1.078 \pm .437$	73.3 ± 11.8
	LSPC-O	$.625 \pm .125$	3.447 ± 3.194	20.3 ± 27.4	$.625 \pm .124$	3.353 ± 3.133	22.8 ± 30.1
BallCircle	CDT	$.715 \pm .005$	$1.342 \pm .034$	27.7 ± 5.2	$.713 \pm .003$	$1.341 \pm .082$	28.0 ± 7.9
	B2R	$.651 \pm .005$	$.837 \pm .343$	68.0 ± 22.6	$.645 \pm .009$	$.769 \pm .270$	70.5 ± 17.3
	FISOR	$.338 \pm .016$	$.012 \pm .020$	$99.5 \pm .9$	$.344 \pm .015$	$.005 \pm .009$	$99.7 \pm .6$
	LSPC-O	$.670 \pm .245$	2.031 ± 3.376	66.3 ± 57.4	$.670 \pm .242$	2.023 ± 3.378	66.7 ± 57.7
CarCircle	CDT	$.714 \pm .004$	$1.078 \pm .073$	61.3 ± 6.8	$.715 \pm .004$	$.960 \pm .049$	65.3 ± 2.4
	B2R	$.706 \pm .012$	$.977 \pm .350$	70.3 ± 8.3	$.702 \pm .015$	$.950 \pm .303$	71.8 ± 5.8
	FISOR	$.363 \pm .031$	$.117 \pm .104$	$99.2 \pm .6$	$.376 \pm .042$	$.107 \pm .035$	$99.0 \pm .5$
	LSPC-O	$.770 \pm .134$	3.126 ± 5.350	65.8 ± 57.0	$.772 \pm .132$	3.093 ± 5.298	66.2 ± 57.3
DroneCircle	CDT	—	—	—	$.523 \pm .019$	$1.487 \pm .076$	18.3 ± 3.3
	B2R	—	—	—	$.462 \pm .008$	$.396 \pm .095$	90.3 ± 3.3
	FISOR	—	—	—	$.477 \pm .015$	$.029 \pm .025$	98.8 ± 1.0
	LSPC-O	—	—	—	$.593 \pm .025$	$2.513 \pm .783$	5.0 ± 8.7

Table 16: Mean feasible rates (%) across three training seeds. Safe initialization is the B2R checkpoint used before refinement. Explicit and Predictive denote the two retrieval metrics.

Setting	Target	Safe initialization	Explicit	Predictive
Pair	BallRun	33.3	100.0	100.0
	AntRun	63.0	65.0	76.7
	DroneRun	60.2	71.8	93.2
	BallCircle	68.0	99.7	99.7
	CarCircle	70.3	80.0	81.7
Multi-source	CarRun	99.8	100.0	100.0
	BallRun	33.3	66.7	66.7
	AntRun	63.0	73.2	83.5
	DroneRun	62.8	95.0	68.3
	BallCircle	70.5	99.8	99.7
	CarCircle	71.8	92.5	86.8
	DroneCircle	90.3	97.8	95.5

Table 17: Qualification ablations with explicit retrieval. Entries report mean normalized reward/cost across three training seeds.

Direction	Safe initialization	Support-only	Critic-only	Full
CarRun→BallRun	.319/3.154	.256/.000	.208/.000	.203/.000
CarRun→AntRun	.718/1.004	.668/.317	.715/.947	.718/.977
CarRun→DroneRun	.508/1.595	−.017/.000	.546/.844	.528/.818
DroneCircle→BallCircle	.651/.837	.574/.174	.495/.059	.500/.107
DroneCircle→CarCircle	.706/.977	.320/.317	.345/.486	.398/.615

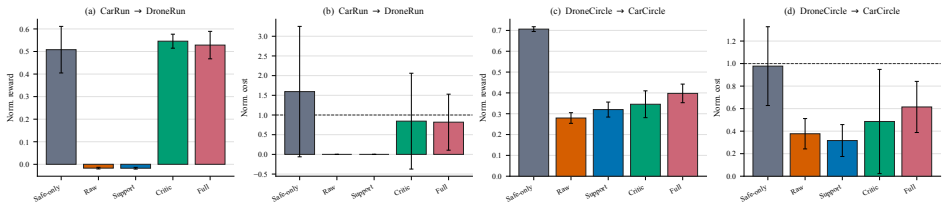


Figure 6: Qualification ablations on DroneRun and CarCircle with explicit retrieval, including Raw refinement without either gate. Results summarize three training seeds; error bars show one standard deviation.

E Additional Behavioral Analysis

E.1 Adoption measurement and command geometry

Directional adoption is defined in Equation 10. Table 7 reports the mean and standard deviation of within-seed median adoption across the frozen explicit-metric policies. The positive fraction pools accepted correction contexts across training seeds. Localization is computed within each seed as policy displacement at correction contexts divided by displacement on a deterministic safe-context reference set; values above one indicate that refinement changes actions more strongly at correction contexts.

The permuted-label comparison holds the trained policy fixed and projects its displacement onto a fixed permutation of pseudo-actions. It therefore measures sensitivity to context–correction correspondence and differs from the separately trained Shuffled control in Section 5.2. BallCircle's positive permuted projection also reflects a component shared across correction directions.

Figure 7 shows how the Circle commands relate to task geometry. BallCircle receives commands that increase boundary margin on average, with some loss of clockwise progress. The smaller CarCircle correction set emphasizes clockwise progress and includes both positive and negative margin changes.

E.2 Matched-state replay protocol

We evaluate the frozen BallCircle policies at 200 candidate contexts from the Cross/Self intersection, selected across initial boundary-margin quantiles. Each branch starts from the same reconstructed observed agent state, Bullet and RNG snapshot, policy history, RTG, and remaining-cost token. The branches share their starting state, while unobserved state from the original data collection remains unrecovered. Matched denotes the Complete Matched policy from Section 5.2.

The analysis uses policy seeds $0, 1, 2$, diagnostic seed 74000, and the same ordered contexts for every condition. One continuous rollout per policy and context supplies the .2, 1, 2, and 4 s prefixes. Episode time and conditioning are restored at the starting context and updated throughout the rollout. Branches stop at episode end; complete windows are available for 200, 200, 199, and 180 contexts, respectively. At each starting context, all policies have the same observed step count.

Table 18 retains all 200 contexts, accumulating cost until the requested horizon or episode end. Matched has lower cost than Self in every training seed at both 2 and 4 s, while retaining most clockwise progress.

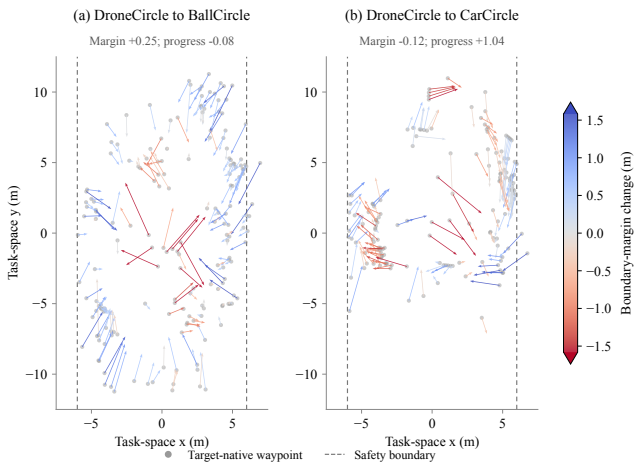


Figure 7: Circle corrections in world coordinates. Gray points mark target-native waypoints, and arrows terminate at retrieved DroneCircle waypoints. Color indicates the change in x-boundary margin.

Table 18: BallCircle temporal response across three training seeds and all 200 starting contexts. Costs are raw accumulated costs up to the horizon or episode end. Progress retention is the ratio of pooled means relative to Self.

Horizon	Self cost	Matched cost	Cost reduction	Progress retained
.2 s	.0150	.0133	11.1%	99.1%
1 s	.1017	.0250	75.4%	97.2%
2 s	.4917	.0483	90.2%	91.3%
4 s	1.4617	.2417	83.5%	87.0%

Restricting the comparison to contexts with a complete requested window gives Matched/Self costs of $.0486/.4941$ at 2 s and $.2296/1.4167$ at 4 s. The cost reduction persists in every seed. This restriction uses only the initial episode timestep and requested duration.

Figure 4 averages within-seed progress ratios, giving 86.9% retention at 4 s; Table 18 instead reports the ratio of pooled means, giving 87.0%. Figure error bars are sample standard deviations across three training seeds.