DYNAMIC ELITE REPLAY: LEARNING BEYOND SUBOPTIMAL EXPERTS IN HUMAN-IN-THE-LOOP REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Human-in-the-loop reinforcement learning (HIL-RL) uses expert intervention and demonstrations to improve training safety and learning efficiency. Many existing methods use expert behavior as an important source of positive supervision, which is effective when reliable expert guidance is available. However, when the expert is suboptimal, continued reliance on such supervision can bias policy learning toward the expert's behavior and make it difficult for the agent to fully exploit better behaviors discovered through its own exploration. To address this problem, we propose **Dynamic Elite Replay (DER)**, an intervention learning framework that integrates elite sample selection and weighted elite learning, enabling the agent to fully exploit high-quality behaviors discovered through its own exploration. DER first identifies high-quality agent-generated experience according to task performance and safety. When complete successful trajectories are scarce, it further extracts safe progress segments that capture useful local improvements. During replay, retained elite samples are re-evaluated using current Q-value advantage, twin-critic agreement, sample freshness, and disagreement with the supervisor, reducing the influence of outdated or unreliable experience. The reweighted elite experience is then used for behavior-cloning and critic updates to reinforce validated high-quality agent behaviors. Experiments in MetaDrive show that DER can achieve performance above a fixed suboptimal supervisor while reducing safety failures. These results suggest that expert intervention can guide training without necessarily defining the final performance ceiling when high-quality behaviors discovered by the agent are effectively identified and reused.

1 INTRODUCTION

Reinforcement learning (RL) learns through trial-and-error interaction, which is difficult to use safely in tasks such as autonomous driving. Early in training, an agent may leave the road, collide with other vehicles, or spend many interactions exploring poor behaviors. Human-in-the-loop reinforcement learning (HIL-RL) addresses this problem by allowing a human or proxy supervisor to take over when the agent behaves poorly and provide corrective actions (Saunders et al., 2018; Kelly et al., 2019; Spencer et al., 2020). HACO (Li et al., 2022) and PVP (Peng et al., 2023) show that such interventions can improve training safety and policy learning, while PVP4Real (Peng et al., 2025) extends this setting to real mobile robots.

A remaining difficulty is that the supervisor need not be optimal. Prior work on imperfect demonstrations shows that demonstration quality can materially affect the learned policy (Wu et al., 2019; Jing et al., 2020; Cao & Sadigh, 2021; Brown et al., 2019; 2020). Expert correction can still prevent failures, but repeatedly treating a suboptimal supervisor as the main source of positive supervision may constrain the policy toward conservative or imperfect behavior. As the agent improves, it may discover actions or trajectories that outperform the supervisor through its own exploration. If these behaviors are not preserved, later expert-guided updates can pull the policy back toward the supervisor and weaken self-discovered improvements.

We focus on what should be learned from shared-control interaction after it has produced both expert corrections and agent-executed behavior. These two sources play different roles: expert corrections

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.

provide intervention guidance, whereas agent-executed experience may contain behaviors that go beyond the supervisor. However, replaying all agent data is also undesirable because some transitions are unsafe, some reflect only short-term progress, and others become less useful as the policy changes. The learning problem is therefore to identify useful agent-generated behavior and determine whether it should still influence the current policy when replayed.

To address this problem, we propose **Dynamic Elite Replay (DER)**. DER combines *elite sample selection* and *weighted elite learning*. The selection stage retains only agent-executed experience and identifies high-quality behavior at two complementary levels: complete episodes and shorter safe progress segments. The latter provide useful supervision before successful full episodes become common. During replay, retained transitions are re-evaluated using current Q-value support, twin-critic agreement, sample freshness, and disagreement with the supervisor. These factors determine how strongly each transition contributes to weighted self-imitation and critic TD updates. As a result, high-quality behavior discovered by the agent can remain a training target without treating takeover demonstrations as self-imitation data.

We evaluate DER in MetaDrive with a fixed suboptimal proxy supervisor and compare it with reward-driven and intervention-learning baselines under a common evaluation protocol. The experiments examine whether DER can move beyond the supervisor, whether progress-segment selection adds useful experience beyond episode-level selection, and whether replay-time re-evaluation improves the use of retained elite transitions.

Our main contributions are:

- We study HIL-RL under suboptimal supervision, where continued reliance on imperfect expert guidance can limit further policy improvement, and focus on preserving high-quality behaviors discovered by the agent itself.
- We introduce two-level elite sample selection over complete episodes and shorter safe progress segments, allowing DER to retain useful agent-generated behavior even before successful full episodes become common.
- We re-evaluate retained elite transitions during replay using current Q support, twin-critic agreement, sample freshness, and supervisor disagreement, so that stale or unreliable samples do not contribute equally to learning.
- We use the reweighted elite transitions for weighted self-imitation and critic TD learning, reinforcing validated agent-generated behaviors in both policy and value updates.

2 Related Work

Human-in-the-loop and interactive learning. DAgger (Ross et al., 2011) reduces distribution shift by querying expert actions on learner-visited states. Human-gated and risk-aware extensions such as HG-DAgger (Kelly et al., 2019), EnsembleDAgger (Menda et al., 2019), and ThriftyDAgger (Hoque et al., 2022) reduce unnecessary expert control by deciding when intervention is needed. Intervention-based methods further use takeover information as a learning signal: Trial without Error uses human intervention for safe exploration (Saunders et al., 2018), EIL interprets intervention and non-intervention as feedback (Spencer et al., 2020), HACO learns from shared human–agent control (Li et al., 2022), and PVP propagates intervention-derived proxy values through TD learning (Peng et al., 2023). Sirius reweights deployment data using estimated human trust (Liu et al., 2023), MILE models the intervention process to exploit both intervention and non-intervention feedback (Korkmaz & Biyik, 2025), PVP4Real studies data-efficient real-world intervention learning (Peng et al., 2025), IntervenGen augments limited corrective-intervention data (Hoque et al., 2024), UnaRL models multimodal human guidance (Huang et al., 2024), and AIM learns when the robot should request expert assistance (Cai et al., 2025). Related formulations also study policy-dependent human feedback (MacGlashan et al., 2017), preference-based interactive RL (Lee et al., 2021), and shared autonomy with potentially suboptimal user actions (Reddy et al., 2018). DER focuses on a different question: after shared-control interaction has produced expert and agent data, which agent-generated behaviors should be preserved and reused for further learning?

Learning with imperfect teachers and demonstrations. Imperfect-demonstration methods explicitly relax the assumption that all demonstrations are optimal. Confidence-based imitation learning

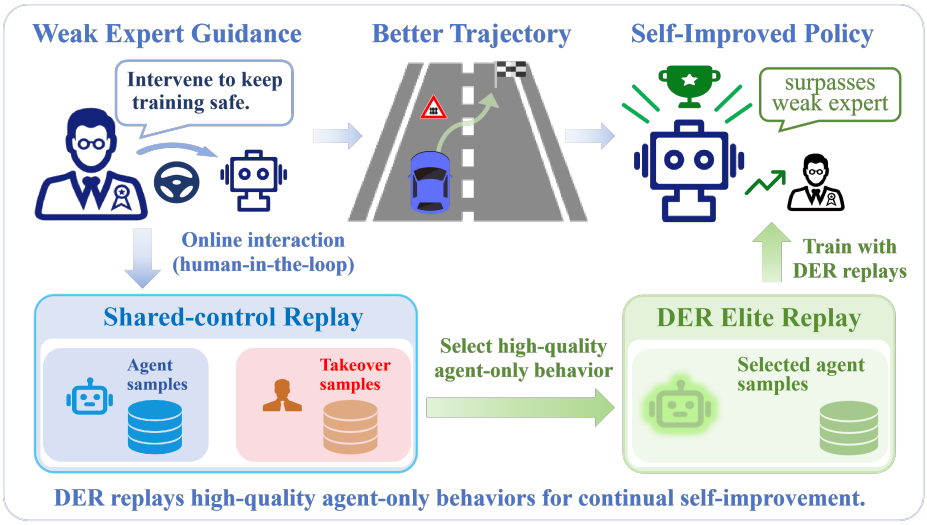


Figure 1: Motivation for DER. A suboptimal supervisor can make early exploration safer, but the agent may later discover better behavior on its own. DER preserves and replays safe, high-quality agent-generated behavior to support further policy improvement. The base shared-control replay and DER elite replay are maintained as separate data paths.

estimates demonstration reliability (Wu et al., 2019), Cao and Sadigh score demonstrations by feasibility and optimality (Cao & Sadigh, 2021), and soft expert guidance treats imperfect demonstrations as a constraint rather than an optimal target (Jing et al., 2020). T-REX and D-REX use ranked demonstrations to learn policies that can outperform the demonstrator (Brown et al., 2019; 2020). In online interactive learning, TS2C relaxes the assumption of a high-quality intervening teacher (Xue et al., 2023), while RLIF uses intervention signals as reinforcement-learning feedback and explicitly studies suboptimal intervening experts (Luo et al., 2024). DER differs in retaining the original takeover mechanism while constructing a separate agent-only elite replay path: safe, high-quality agent-executed experience is selected and later re-evaluated before reuse.

Self-imitation and experience replay. Self-Imitation Learning (Oh et al., 2018) reuses an agent's own high-return experience so that successful behavior is not immediately lost. Prioritized Experience Replay (Schaul et al., 2016) samples transitions according to learning priority, Attentive Experience Replay prioritizes states relevant to the current policy (Sun et al., 2020), and ReF-ER (Novati & Koumoutsakos, 2019) reduces the influence of replay samples that become too different from the current policy. Human-guided prioritized replay has also been studied for autonomous driving (Wu et al., 2024). More recently, E2HiL (Deng et al., 2026) selects informative HIL samples according to their estimated influence on policy entropy. DER combines agent-only selection with replay-time re-evaluation for a specific failure mode under suboptimal supervision: useful agent-discovered behavior can be weakened by continued expert-guided learning.

3 Problem Setting

We consider a discounted continuous-control MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$ with shared-control intervention during training. At state s_t, the agent proposes $a_t^A = \pi_\theta(s_t)$ and a fixed supervisor provides an action a_t^E. Let $z_t \in \{0, 1\}$ indicate whether the existing takeover rule is triggered. The action executed in the environment is

$$\tilde{a}_t = (1 - z_t)a_t^A + z_t a_t^E. \tag{1}$$

The supervisor is assumed to be useful but suboptimal: it can prevent or correct poor behavior without necessarily providing the best action at every state. During evaluation, the learned policy acts autonomously without expert rescue. The objective is therefore to improve the autonomous policy under suboptimal supervision rather than to optimize a permanently shared-control policy.

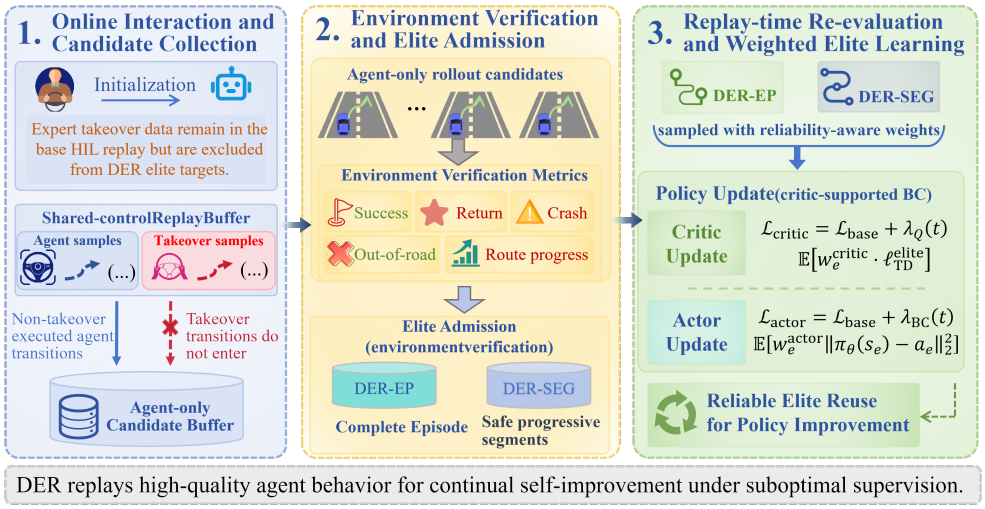


Figure 2: Overview of Dynamic Elite Replay. DER selects agent-generated behavior using episode outcomes and shorter progress segments, then re-evaluates retained transitions using current Q support, twin-critic agreement, sample freshness, and supervisor disagreement before applying weighted actor and critic updates. Takeover data remain available to the base HIL replay but are excluded from DER elite targets.

Shared-control interaction produces takeover transitions and transitions actually executed by the agent. The underlying HIL learner may use both, whereas DER constructs its elite replay path only from the latter. Specifically, DER maintains an agent-only candidate cache

$$\mathcal{C}_A = \left\{(s_t, a_t^A, r_t, s_{t+1}, d_t, \eta_t) : z_t = 0\right\}, \tag{2}$$

where d_t is the terminal indicator and η_t contains route-progress, safety, collection-time, and supervisor-disagreement metadata. Expert takeover transitions remain available to the underlying HIL learner but are excluded from DER self-imitation. DER also preserves the original environment transition structure and never constructs a synthetic TD transition across a takeover interval.

For each completed training episode τ, we record

$$m(\tau) = (y_{\text{suc}}, \rho_{\text{route}}, R, y_{\text{oor}}, y_{\text{crash}}, T), \tag{3}$$

where y_{suc} is the episode-success indicator, y_{oor} and y_{crash} are binary terminal-event indicators for out-of-road and crash outcomes, ρ_{route} is route completion, $R = \sum_t r_t$ is episode return, and T is episode length. These statistics describe the complete executed episode and may therefore include supervisor intervention. Their averages across evaluation episodes are reported as OOR and crash rates in Appendix B.5.

Episode-level outcomes are used to decide whether an episode contains useful agent behavior, but they do not certify every agent action in that episode. DER therefore uses an episode or progress segment as a *selection unit*, while individual agent-executed transitions remain the *learning unit*. Candidate transitions are filtered before entering an elite replay buffer and are re-evaluated again when replayed.

Under this formulation, DER has two coupled operations. *Elite sample selection* identifies safe and high-quality agent experience from complete episodes and local progress segments. *Weighted elite learning* re-evaluates retained transitions against the current policy and critics and determines their contribution to subsequent policy and value updates. The underlying shared-control learner and takeover rule remain fixed across DER variants; implementation-specific base losses and replay routing are given in Appendix A.1.

4 Dynamic Elite Replay

Figure 2 summarizes DER as two linked stages: *elite sample selection* and *weighted elite learning*. Conceptually, DER consists of three operations: retaining agent-executed experience that passes task-

level quality and safety checks, re-evaluating retained samples against the current learner at replay time, and using reliability weights in policy and value updates. Episode and progress segments are selection units, while individual transitions remain learning units. The quantile thresholds, collection windows, and coefficient schedules in Appendix A are a concrete MetaDrive instantiation rather than part of DER's conceptual definition.

4.1 AGENT-ONLY ELITE REPLAY BUFFERS

The underlying HIL learner retains its original shared-control replay-buffer routing, while DER augments it with two agent-only elite replay buffers: an episode-level buffer $\mathcal{B}_{\text{ep}}$ and a progress-segment buffer $\mathcal{B}_{\text{seg}}$. At each interaction step, the executed transition is first routed to the base replay buffers according to the original HIL rule. Only non-takeover transitions ($z_t = 0$), corresponding to actions actually executed by the agent, are additionally collected as DER candidates together with route-progress, safety, collection-time, and supervisor-disagreement metadata. Takeover transitions remain available to the underlying HIL learner but are excluded from DER and never used as self-imitation targets. Candidate transitions are inserted into $\mathcal{B}_{\text{ep}}$ or $\mathcal{B}_{\text{seg}}$ only after passing the corresponding episode-level or progress-segment admission criteria described below.

4.2 EPISODE-LEVEL ELITE SELECTION

At the end of each training episode, DER scores the complete executed trajectory using task performance and safety:

$$S_{\text{ep}} = \alpha_s y_{\text{suc}} + \alpha_r \rho_{\text{route}} + \alpha_R \tanh(R/R_{\text{scale}}) - \alpha_o y_{\text{oor}} - \alpha_c y_{\text{crash}}. \tag{4}$$

During an initial bootstrap phase, DER records the outcome statistics of every completed training episode. Once sufficient early-training history is available, DER estimates an episode-level admission reference from this empirical distribution. The reference consists of an episode-score threshold $\bar{S}_{\text{ep}}$, a minimum route-completion requirement $\bar{\rho}$, and bounded safety references $\bar{o}$ and $\bar{c}$. These values are then frozen and used for subsequent episode admission:

$$S_{\text{ep}} \geq \bar{S}_{\text{ep}}, \qquad \rho_{\text{route}} \geq \bar{\rho}, \qquad y_{\text{oor}} \leq \bar{o}, \qquad y_{\text{crash}} \leq \bar{c}. \tag{5}$$

Because y_{oor} and y_{crash} are binary terminal-event indicators and the implementation caps both safety references below 1, the last two conditions are operationally equivalent to requiring no out-of-road and no crash event for an admitted episode.

Episode-level admission determines whether the episode contains agent behavior worth retaining; it does not copy the mixed-control trajectory into DER. Only agent-executed candidate transitions from an admitted episode are considered, and unsafe or low-quality transitions are filtered again before insertion into the episode-level elite replay buffer $\mathcal{B}_{\text{ep}}$. When the buffer reaches capacity, higher-scoring episode units are retained preferentially. The bootstrap length, quantile definitions, transition-level filters, and buffer limits are given in Appendix A.2.

4.3 PROGRESS-SEGMENT ELITE SELECTION

Complete successful episodes are often sparse early in training, so DER also mines shorter progress segments from the ordered agent-executed candidate transitions of each completed episode. Takeover transitions have already been excluded from this sequence. The sliding window is used only for segment selection: every retained item keeps its original (s, a, r, s') transition, and DER never constructs a synthetic TD transition across a takeover gap.

Each candidate segment is scored as

$$S_{\text{seg}} = \beta_p P_{\Delta\rho} + \beta_R P_R + \beta_e \rho_{\text{end}} + \beta_i I_{\text{mid}} - \beta_u U - \beta_a P_a, \tag{6}$$

where $P_{\Delta\rho}$ measures route progress, P_R is normalized local return, ρ_{end} is the route completion reached at the end of the segment, U is the unsafe-step ratio, and P_a penalizes excessive action magnitude. I_{mid} provides a small preference for moderate disagreement with the supervisor.

Segment admission is independent of episode admission. After enough candidate segments have been observed, DER derives the segment-score threshold from the upper quantile of the accumulated

segment-score history and updates this reference as new segment scores are collected. A segment is inserted into the progress-segment elite replay buffer $\mathcal{B}_{\text{seg}}$ only if it exceeds the current score reference and satisfies the progress, safety, and action-magnitude constraints. Exact window parameters, quantiles, and filtering rules are provided in Appendix A.3.

DER uses a larger segment share while complete-task experience is limited and gradually shifts elite replay toward episode-level samples as training progresses. At each elite update, $\mathcal{B}_{\text{ep}}$ and $\mathcal{B}_{\text{seg}}$ are sampled separately using the current segment ratio and concatenated into the elite mini-batch used by Section 4.5. The exact sampling schedule is given in Appendix A.3.

4.4 Replay-time re-evaluation

Before reuse, DER re-evaluates each retained elite transition against the current actor and critics. For an elite state-action pair (s_e, a_e), the current actor proposes $a_\pi = \pi_\theta(s_e)$. Current Q support is measured by

$$\begin{aligned} \Delta Q_e &= \min_i Q_i(s_e, a_e) - \min_i Q_i(s_e, a_\pi), \\ m_Q &= \mathbf{1}[\Delta Q_e \geq -\delta_Q], \\ w_{\text{adv}} &= w_Q^{\min} + (1 - w_Q^{\min})\sigma(\Delta Q_e/\tau_Q). \end{aligned} \tag{7}$$

The hard mask m_Q prevents actor-side self-imitation when the retained action is no longer supported by the current critics, while w_{adv} provides a soft Q-advantage weight among the retained actions that remain usable.

DER then assigns three additional replay-time factors. The confidence weight w_{conf} decreases as the two critics disagree on the retained action, and the freshness weight w_{fresh} decreases with sample age. DER also uses the supervisor action log-density as a relative disagreement cue: w_{info} is largest for moderate disagreement and smaller for actions that either closely match or strongly deviate from the supervisor. This disagreement cue is applied only after task-progress and safety selection, so it reweights already-retained agent behavior rather than treating novelty as evidence of quality. Exact mappings and batch quantiles are given in Appendix A.4.

4.5 Weighted policy and value learning

Each retained transition inherits the selection score of its episode or segment, which DER converts into a bounded quality factor w_{qual}. This factor preserves quality differences among admitted units, while the replay-time factors in Section 4.4 measure whether a retained transition remains reliable for the current policy. The exact mapping is given in Appendix A.4.

The actor- and critic-side sample weights are

$$w_e^{\text{actor}} = w_{\text{qual}} m_Q w_{\text{adv}} w_{\text{conf}} w_{\text{fresh}} w_{\text{info}}, \qquad w_e^{\text{critic}} = w_{\text{qual}} w_{\text{conf}} w_{\text{fresh}} w_{\text{info}}. \tag{8}$$

Let $\mathcal{B}_A$ and $\mathcal{B}_Q$ denote the elite mini-batches used for policy and value learning. DER augments the underlying actor–critic objectives with normalized weighted elite losses:

$$\mathcal{L}_{\text{actor}} = \mathcal{L}_{\text{actor}}^{\text{base}} + \lambda_{\text{BC}}(t) \frac{\sum_{j \in \mathcal{B}_A} w_{e,j}^{\text{actor}} \|\pi_\theta(s_j) - a_j\|_2^2}{\max\left(1, \sum_{j \in \mathcal{B}_A} w_{e,j}^{\text{actor}}\right)}, \tag{9}$$

$$\mathcal{L}_{\text{critic}} = \mathcal{L}_{\text{critic}}^{\text{base}} + \lambda_Q(t) \frac{\sum_{j \in \mathcal{B}_Q} w_{e,j}^{\text{critic}} \ell_{\text{TD},j}^{\text{elite}}}{\max\left(1, \sum_{j \in \mathcal{B}_Q} w_{e,j}^{\text{critic}}\right)}. \tag{10}$$

The actor term is weighted self-imitation of selected agent-executed actions. The Q-support mask and Q-advantage factor are actor-specific because they test whether a stored action is still preferable to the current policy action. The critic instead performs weighted TD learning on the retained transitions using the raw environment reward stored by DER. The exact quality references, elite TD target, replay-weight parameters, and coefficient schedules are given in Appendix A.4.

5 EXPERIMENTS

The experiments are organized around four questions that follow DER's two-stage design. RQ1 asks whether preserving and reusing high-quality agent-generated experience allows the learned policy to move beyond a fixed suboptimal supervisor and matched learning baselines. RQ2 tests whether progress-segment selection complements episode-level elite selection when complete high-quality episodes are still limited. RQ3 examines whether replay-time re-evaluation improves how retained elite transitions contribute to policy and value learning. RQ4 examines whether the resulting improvement is consistent across independent training runs and over the course of training. Safety outcomes are reported separately in Appendix B.5.

5.1 EVALUATION SETTING AND COMPARISON PROTOCOL

We evaluate DER in the MetaDrive continuous-control benchmark (Li et al., 2023). The policy is initialized from scratch and controls steering and longitudinal acceleration from vector observations. A fixed PPO policy serves as a suboptimal proxy supervisor and is used only for intervention during training; the learned policy is evaluated autonomously without expert rescue.

To isolate the effect of DER, all DER variants use the same shared-control actor–critic backbone and the same takeover rule. The components introduced by DER are exactly those described in Section 4: agent-only elite selection at the episode and progress-segment levels, replay-time re-evaluation, and weighted policy/value updates. The underlying learner and replay routing are detailed in Appendix A.1.

All matched comparisons use the same task definition, interaction budget, and evaluation protocol while preserving each baseline's own learning rule. Results are aggregated over independent training runs and reported as mean $\pm$ sample standard deviation. We compare DER with TD3 (Fujimoto et al., 2018) as a reward-driven no-intervention reference and with HACO (Li et al., 2022), PVP (Peng et al., 2023), and PVP4Real (Peng et al., 2025) as HIL-related baselines. Published results obtained under different supervisors, budgets, or evaluation protocols are not mixed into the matched comparison. Exact environment settings, run count, interaction budget, evaluation cadence, and statistical definitions are provided in Appendices B.1 and B.2.

The reported endpoint values come from the validation stream used during training for model selection. Validation trajectories never enter any replay buffer and never contribute gradients, but the same stream is used by the confirmed-best checkpoint mechanism in Appendix A.5. We therefore report these values as validation performance rather than as a sealed held-out test.

5.2 RQ1: CAN DER MOVE BEYOND A SUBOPTIMAL SUPERVISOR?

Table 1 compares DER with the fixed supervisor and matched baselines. PVP+Reward is a reward-matched PVP control that augments the original PVP configuration with environment-reward learning while leaving DER-specific elite selection and replay-time weighting disabled.

Table 1: Matched MetaDrive validation comparison under the fixed suboptimal supervisor. All learned methods use the common four-run subset and report mean $\pm$ sample standard deviation. PVP+Reward is the reward-matched PVP control; expanded DER stability is reported in Appendix B.2.

Method	Success ↑	Route ↑	Reward ↑
Suboptimal supervisor	0.280	0.709	237.6
TD3	0.050 ± 0.100	0.387 ± 0.092	122.3 ± 32.0
HACO	0.175 ± 0.126	0.670 ± 0.116	213.1 ± 29.7
PVP	0.375 ± 0.150	0.773 ± 0.042	277.9 ± 18.8
PVP4Real	0.100 ± 0.082	0.494 ± 0.116	174.6 ± 42.7
PVP+Reward (reward-matched)	0.325 ± 0.343	0.693 ± 0.234	238.96 ± 88.94
DER (ours)	$\mathbf{0.813 \pm 0.085}$	$\mathbf{0.923 \pm 0.048}$	$\mathbf{332.0 \pm 38.6}$

DER reaches 0.813 success, 0.923 route completion, and 332.0 reward, compared with 0.280, 0.709, and 237.6 for the fixed supervisor. PVP+Reward reaches 0.325 ± 0.343, 0.693 ± 0.234, and

238.96 ± 88.94; its observed means do not exceed PVP and remain well below DER. Thus, in this matched setting, adding environment-reward learning to PVP does not reproduce DER's improvement.

The comparison is limited to the fixed supervisor and matched protocols evaluated here, and the reported differences are observed means rather than claims of statistical significance. Appendix B.3 shows how success, route completion, and reward evolve during training. Appendix B.5 separately reports out-of-road and crash outcomes, while Appendix B.4 provides a representative trajectory visualization that is used only for qualitative inspection.

5.3 RQ2–RQ3: Effects of two-level selection and weighted elite learning

The ablation follows the construction of DER. *Episode-level elite replay* retains selected agent transitions only through the episode-level replay buffer $\mathcal{B}_{\mathrm{ep}}$. + *Progress-segment selection* adds the segment replay buffer $\mathcal{B}_{\mathrm{seg}}$ so that safe local progress can be retained even when the complete episode is not admitted. + *Replay-time confidence/freshness weighting* reweights retained samples according to twin-critic agreement and sample freshness. *Full DER* further applies current-Q support and supervisor-disagreement weighting, completing the replay-time re-evaluation stage. All variants keep the supervisor, underlying learner, task outcomes, and evaluation protocol fixed. The ablation uses the common run subset available for every variant so that all rows are aggregated over the same independent runs.

Table 2: Cumulative ablation of DER. Each row adds one component to the preceding configuration while keeping the underlying learner, supervisor, and evaluation protocol fixed. Bold values indicate the best mean in each column.

Configuration	Success ↑	Route ↑	Reward ↑
Episode-level elite replay	0.450 ± 0.129	0.794 ± 0.051	270.4 ± 36.3
+ Progress-segment selection	0.675 ± 0.126	0.886 ± 0.021	315.7 ± 9.0
+ Replay-time confidence/freshness weighting	0.725 ± 0.087	0.902 ± 0.037	321.6 ± 17.0
Full DER (Ours)	$\mathbf{0.813 \pm 0.085}$	$\mathbf{0.923 \pm 0.048}$	$\mathbf{332.0 \pm 38.6}$

The ablation shows a consistent improvement as the two stages of DER are progressively introduced. Starting from episode-level elite replay, adding progress-segment selection increases success from 0.450 to 0.675, route completion from 0.794 to 0.886, and episodic reward from 270.4 to 315.7. This supports the role of segment-level selection in preserving useful local agent behavior that may be missed when elite experience is identified only from complete episodes.

Adding replay-time confidence and freshness weighting further improves the three metrics to 0.725 success, 0.902 route completion, and 321.6 reward. This indicates that retained elite transitions should not contribute equally throughout training: samples with stronger critic agreement and more recent collection remain more reliable for subsequent policy and value updates.

Full DER additionally incorporates current-Q support and supervisor-disagreement weighting, achieving the best mean performance on all three metrics: 0.813 success, 0.923 route completion, and 332.0 episodic reward. The cumulative trend therefore supports the two-stage design of DER: elite sample selection determines which agent-generated behaviors are worth retaining, while replay-time re-evaluation determines how strongly those retained behaviors should influence learning.

All ablation variants use the same underlying learner, supervisor, environment-reward access, and task-outcome information. The differences therefore reflect the cumulative contribution of DER's selection and replay-weighting components. The corresponding learning dynamics are reported in Appendix B.6.

5.4 RQ4: Consistency across runs and training dynamics

Four additional DER runs expand the stability analysis to eight independent runs. Final success ranges from 0.65 to 0.90, route completion from 0.867 to 0.971, and reward from 300.1 to 382.1; every run remains above the fixed supervisor on all three metrics. The eight-run aggregate is 0.788 ± 0.092 success, 0.919 ± 0.037 route completion, and 327.7 ± 27.3 reward. Table 1, the safety comparison,

ablation, and learning curves retain the original matched four-run subset for equal aggregation across methods or variants; all run-level statistics are reported in Appendix B.2.

6 Discussion and Limitations

DER is designed for settings in which a supervisor is useful but not necessarily optimal. The current matched MetaDrive runs show higher mean success, route completion, and reward than the evaluated TD3, HACO, PVP, and PVP4Real baselines, but several limits are important for interpreting these differences.

The study uses one simulator and one fixed proxy-supervisor checkpoint, so the evidence should not be generalized to arbitrary supervisor qualities without matched evaluations. Baseline, safety, and ablation comparisons use a common four-run subset, while the separate stability analysis uses eight DER runs; we therefore report sample standard deviation without significance claims. Endpoint values come from a validation stream that also participates in confirmed-best checkpointing. These trajectories never enter replay, but the stream is not a sealed held-out test set. Moreover, no matched no-rollback control is available, so the cross-method results evaluate the reported DER system rather than isolating the gain of rollback-free DER.

DER uses environment reward and task outcomes even though the proxy-value replay in the reported PVP-style backbone discards reward. To test whether this additional signal alone explains the gain, PVP+Reward augments the original PVP configuration with environment-reward learning while disabling DER-specific elite selection and replay-time weighting. PVP+Reward reaches 0.325 ± 0.343 success, 0.693 ± 0.234 route completion, and 238.96 ± 88.94 reward, remaining far below DER in observed means. This matched control supports that reward access alone is insufficient to reproduce DER's improvement in this setting, without implying that every possible reward-augmented HIL formulation would behave identically. A second method-specific limitation is that supervisor disagreement is derived from the FakeHuman action log-density, which is useful as a relative simulation signal but is not directly available from a real human operator; a real-world implementation would require a learned human-action model or another disagreement estimator. Critic agreement, freshness, and disagreement remain practical weighting signals rather than guarantees that a retained sample is optimal.

7 Conclusion

We presented Dynamic Elite Replay (DER) for HIL-RL with a suboptimal supervisor. DER separates agent-generated experience from takeover demonstrations, identifies promising behavior using episode outcomes and shorter progress segments, and re-evaluates retained samples before they affect policy and value learning. In the matched MetaDrive evaluation, DER achieves higher mean success, route completion, and episodic reward than the evaluated reward-driven, HIL, and reward-matched PVP controls. The broader lesson is that expert intervention can remain useful for guiding interaction without requiring the learner to treat expert behavior as the only positive target: high-quality behavior discovered by the agent can be preserved, checked again, and reused throughout training.

AI Use Statement

Generative AI tools were used only as auxiliary tools for limited code refactoring and debugging, literature search and reference checking, and language polishing of the manuscript. The research ideas, methodological design, experimental setup and execution, result analysis, and scientific conclusions were developed and carried out by the authors. Generative AI was not used to generate or fabricate experimental measurements, replace the execution of experiments, or make scientific decisions on behalf of the authors. All AI-assisted code modifications and literature suggestions were independently reviewed and verified by the authors, and all reported numerical results, equations, citations, and scientific claims were checked against the implementation, experiment logs, or primary literature. The authors take full responsibility for the methodology, experiments, analysis, and final content of this work.

Ethics Statement

The experiments in this paper are conducted entirely in the MetaDrive simulator using a fixed proxy supervisor. No human-subject data are collected. The study evaluates the proposed learning framework in simulation, and conclusions about deployment with real human operators are outside the scope of the current experiments. Future studies involving human participants should follow the corresponding institutional and informed-consent requirements.

Reproducibility Statement

Matched cross-method comparisons use the same task definition, proxy-supervisor setting, interaction budget, randomization protocol, and validation schedule while preserving each baseline's own learning rule. Within DER component ablations, the underlying shared-control learner and supervision setting are held fixed. DER elite data are constructed only from agent-executed portions of training rollouts. Independent training runs are the unit of statistical aggregation, and training interactions are kept separate from model-selection validation trajectories. Full algorithmic details and hyperparameters are provided in Appendix A; evaluation protocols, statistical definitions, component configurations, fairness controls, and the scope of comparison evidence are provided in Appendix B.

References

Daniel Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 783–792, 2019.

Daniel S. Brown, Wonjoon Goo, and Scott Niekum. Better-than-demonstrator imitation learning via automatically-ranked demonstrations. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 330–359, 2020.

Haoyuan Cai, Zhenghao Peng, and Bolei Zhou. Robot-gated interactive imitation learning with adaptive intervention mechanism. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 6243–6256, 2025.

Zhangjie Cao and Dorsa Sadigh. Learning from imperfect demonstrations from agents with varying dynamics. *IEEE Robotics and Automation Letters*, 6(3):5231–5238, 2021. doi: 10.1109/LRA.2021.3068912.

Haoyuan Deng, Yudong Lin, Yuanjiang Xue, Haoyang Du, Qianzhun Wang, Boyang Zhou, Zhenyu Wu, and Ziwei Wang. E2HiL: Entropy-guided sample selection for efficient real-world human-in-the-loop reinforcement learning. *IEEE Robotics and Automation Letters*, 11(7):8084–8091, 2026. doi: 10.1109/LRA.2026.3693577.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1587–1596. PMLR, 2018.

Ryan Hoque, Ashwin Balakrishna, Ellen Novoseller, Albert Wilcox, Daniel S. Brown, and Ken Goldberg. ThriftyDAgger: Budget-aware novelty and risk gating for interactive imitation learning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 598–608, 2022.

Ryan Hoque, Ajay Mandlekar, Caelan Reed Garrett, Ken Goldberg, and Dieter Fox. IntervenGen: Interventional data generation for robust and data-efficient robot imitation learning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2840–2846, 2024. doi: 10.1109/IROS58592.2024.10801523.

Wenhui Huang, Zitong Shan, Shanhe Lou, and Chen Lv. Uncertainty-aware reinforcement learning for autonomous driving with multimodal digital driver guidance. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 18355–18361, 2024. doi: 10.1109/ICRA57147.2024.10610272.

Mingxuan Jing, Xiaojian Ma, Wenbing Huang, Fuchun Sun, Chao Yang, Bin Fang, and Huaping Liu. Reinforcement learning from imperfect demonstrations under soft expert guidance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5109–5116, 2020. doi: 10.1609/aaai.v34i04.5953.

Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J. Kochenderfer. HG-DAgger: Interactive imitation learning with human experts. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8077–8083, 2019. doi: 10.1109/ICRA.2019.8793698.

Yigit Korkmaz and Erdem Biyik. MILE: Model-based intervention learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15673–15679, 2025. doi: 10.1109/ICRA55743.2025.11128018.

Kimin Lee, Laura M. Smith, and Pieter Abbeel. PEBBLE: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6152–6163. PMLR, 2021.

Quanyi Li, Zhenghao Peng, and Bolei Zhou. Efficient learning of safe driving policy via Human-AI copilot optimization. In *International Conference on Learning Representations*, 2022.

Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. MetaDrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3461–3475, 2023. doi: 10.1109/TPAMI.2022.3190471.

Huihan Liu, Soroush Nasiriany, Lance Zhang, Zhiyao Bao, and Yuke Zhu. Robot learning on the job: Human-in-the-loop autonomy and learning during deployment. In *Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.005.

Jianlan Luo, Perry Dong, Yuexiang Zhai, Yi Ma, and Sergey Levine. RLIF: Interactive imitation learning as reinforcement learning. In *International Conference on Learning Representations*, 2024.

James MacGlashan, Mark K. Ho, Robert Loftin, Bei Peng, Guan Wang, David L. Roberts, Matthew E. Taylor, and Michael L. Littman. Interactive learning from policy-dependent human feedback. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2285–2294, 2017.

Kunal Menda, Katherine Driggs-Campbell, and Mykel J. Kochenderfer. EnsembleDAgger: A bayesian approach to safe imitation learning. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5041–5048, 2019. doi: 10.1109/IROS40897.2019.8968287.

Guido Novati and Petros Koumoutsakos. Remember and forget for experience replay. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4851–4860, 2019.

Junhyuk Oh, Yijie Guo, Satinder Singh, and Honglak Lee. Self-imitation learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3878–3887. PMLR, 2018.

Zhenghao Peng, Wenjie Mo, Chenda Duan, Quanyi Li, and Bolei Zhou. Learning from active human involvement through proxy value propagation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 20552–20563, 2023. doi: 10.52202/075280-3406.

Zhenghao Peng, Zhizheng Liu, and Bolei Zhou. Data-efficient learning from human interventions for mobile robots. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7407–7414, 2025. doi: 10.1109/ICRA55743.2025.11128012.

Siddharth Reddy, Anca D. Dragan, and Sergey Levine. Shared autonomy via deep reinforcement learning. In *Proceedings of Robotics: Science and Systems*, 2018. doi: 10.15607/RSS.2018.XIV.005.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 2011.

William Saunders, Girish Sastry, Andreas Stuhlmüller, and Owain Evans. Trial without error: Towards safe reinforcement learning via human intervention. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*, pp. 2067–2069. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *International Conference on Learning Representations*, 2016.

Jonathan Spencer, Sanjiban Choudhury, Matt Barnes, Matthew Schmittle, Mung Chiang, Peter Ramadge, and Siddhartha Srinivasa. Learning from interventions: Human-robot interaction as both explicit and implicit feedback. In *Proceedings of Robotics: Science and Systems*, 2020. doi: 10.15607/RSS.2020.XVI.055.

Peiquan Sun, Wengang Zhou, and Houqiang Li. Attentive experience replay. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5900–5907, 2020. doi: 10.1609/aaai.v34i04.6049.

Jingda Wu, Zhiyu Huang, Wenhui Huang, and Chen Lv. Prioritized experience-based reinforcement learning with human guidance for autonomous driving. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):855–869, 2024. doi: 10.1109/TNNLS.2022.3177685.

Yueh-Hua Wu, Nontawat Charoenphakdee, Han Bao, Voot Tangkaratt, and Masashi Sugiyama. Imitation learning from imperfect demonstration. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6818–6827, 2019.

Zhenghai Xue, Zhenghao Peng, Quanyi Li, Zhihan Liu, and Bolei Zhou. Guarded policy optimization with imperfect online demonstrations. In *International Conference on Learning Representations*, 2023.

A FULL DER IMPLEMENTATION DETAILS

This appendix specifies the exact experimental instantiation used in Section 5. DER itself is defined by the agent-only elite selection, replay-time re-evaluation, and weighted actor/critic updates in Section 4. For the reported experiments, these components are implemented inside an existing TD3-based HIL codebase with intervention-derived proxy-value supervision. This shared-control learner provides the underlying value-learning path, while DER determines which agent-generated experiences are retained and how they are reused. Unless stated otherwise, all values below are fixed across the reported runs.

Algorithmic summary.

Algorithm 1: Dynamic Elite Replay with an underlying HIL value learner

Require: actor–critic (π_θ, Q_1, Q_2), base shared-control replay, episode replay buffer $\mathcal{B}_{\text{ep}}$, segment replay buffer $\mathcal{B}_{\text{seg}}$

for each environment step t **do**
  propose a_t^A, apply the fixed takeover rule, and execute $\tilde{a}_t$
  route the interaction to the underlying shared-control replay system
  if $z_t = 0$ **then**
    cache the true agent transition, route/safety metadata, timestamp, and supervisor log-density
  else
    skip the executed expert action from DER candidate data
  end if
  if the episode terminates **then**
    compute episode statistics and update the bootstrap reference if it is not yet frozen
    mine progress segments and insert retained agent transitions into $\mathcal{B}_{\text{seg}}$
    apply episode admission and locally filter agent transitions for $\mathcal{B}_{\text{ep}}$
  end if
  perform the base critic/value-learning update
  if elite critic update is active **then**
    sample $\mathcal{B}_{\text{ep}}$ and $\mathcal{B}_{\text{seg}}$ using the current segment ratio
    compute quality, confidence, freshness, and supervisor-disagreement weights
    apply the weighted elite TD auxiliary term
  end if
  if this is a delayed actor-update step **then**
    perform the base actor update
    if elite actor update is active **then**
      draw a separate episode/segment elite mini-batch
      compute quality, Q-support, confidence, freshness, and disagreement weights
      apply the weighted elite BC auxiliary term
    end if
  end if
end for

A.1 UNDERLYING VALUE LEARNER AND REPLAY ROUTING

For exact reproducibility, the reported instantiation uses the existing TD3-based (Fujimoto et al., 2018) proxy-value shared-control learner in our codebase. Both human and agent proxy-value losses are active, and the actor keeps the intervention behavior-cloning term. These losses define the fixed underlying HIL value learner used in the experiments; the DER components are the separate agent-only episode/segment elite replay buffers, their admission rules, replay-time weights, and the auxiliary elite updates. Risk-constrained TD3, adaptive disengagement, and the learned risk-shield branches in the same source file are disabled for the reported DER runs.

The underlying shared-control replay system contains two buffers. When no takeover occurs, the transition is stored in the regular learner buffer. When takeover occurs, the transition is routed to the intervention buffer. The intervention buffer stores both the executed behavior action and the

novice action proposed before takeover, together with the takeover indicator and the associated supervisor-action log-density. When both buffers contain enough data, a training batch contains half regular and half intervention samples. The base batch size is 1024.

In this configuration, the base shared-control replay buffers discard environment reward. Therefore the ordinary TD target is

$$y_t^{\text{base}} = \gamma(1 - d_t) \min_i Q'_i(s_{t+1}, \pi'(s_{t+1}) + \epsilon), \tag{11}$$

with clipped TD3 target noise ϵ. On intervention steps, the underlying proxy-value learner additionally pushes the executed expert/behavior action toward $+1$ and the novice action that triggered takeover toward -1. A compact form of the critic objective is

$$\mathcal{L}_{\text{critic}}^{\text{base}} = \sum_i \mathbb{E}\left[(Q_i(s, a^{\text{beh}}) - y^{\text{base}})^2 + z(Q_i(s, a^{\text{beh}}) - 1)^2 + z(Q_i(s, a^A) + 1)^2\right], \tag{12}$$

where constants shared with the implementation are omitted for readability. The actor uses the TD3 policy objective and an intervention-masked BC term with weight 1.0:

$$\mathcal{L}_{\text{actor}}^{\text{base}} = -\mathbb{E}[Q_1(s, \pi_\theta(s))] + \lambda_H \, \mathbb{E}\left[z\|\pi_\theta(s) - a^{\text{beh}}\|_2^2\right], \qquad \lambda_H = 1. \tag{13}$$

DER leaves these backbone losses unchanged. It records the raw environment reward before the base replay discards it and uses that reward only for elite selection and the auxiliary elite TD target. This separation lets us evaluate DER as a data-selection and reuse mechanism while holding the underlying shared-control value learner fixed.

A.2 Agent-only candidates and episode-level replay buffer

Stored fields and local step quality. DER excludes takeover transitions before constructing its elite candidate data. For each agent-executed candidate transition, the implementation stores (s, a, r, s', d), route progress before and after the action, mean absolute action magnitude, a safety flag, collection time, and the supervisor-disagreement value from Eq. (23). The local step-quality value used for transition-level filtering is

$$q_t^{\text{local}} = r_t - 2\,\mathbf{1}[\text{crash} \vee \text{OOR}] - 0.25 \max(c_t, 0) - 0.02 \text{ mean } |a_t|, \tag{14}$$

where c_t is the environment cost. Expert-executed actions are never inserted into an elite replay buffer and are never used as DER self-imitation targets.

Bootstrap episode reference. Let $\mathcal{H}_0$ denote the early-training history of completed episode statistics. This history is collected independently of elite admission: it contains all completed training episodes observed during bootstrap, including episodes that are later rejected by the episode-level selector. Thus, the reference is estimated from the observed early-training episode distribution rather than from the already-filtered contents of $\mathcal{B}_{\text{ep}}$.

In the reported configuration, DER waits until at least $n_0 = 12$ completed episodes have been recorded and training has reached $t_{\text{freeze}} = 7000$ environment steps. The reward scale in Eq. (4) is set from the 75th percentile of the absolute returns in this history. The episode-score threshold and the lower score reference used later for quality weighting are

$$\bar{S}_{\text{ep}} = \text{Quantile}_{0.45}\{S_i : i \in \mathcal{H}_0\}, \qquad S_{\text{w}}^{(\text{ep})} = \text{Quantile}_{0.25}\{S_i : i \in \mathcal{H}_0\}. \tag{15}$$

The same bootstrap history determines the route and safety constraints:

$$\begin{aligned} \bar{\rho} &= \max(0.65, \text{Quantile}_{0.40}(\rho_{\text{route}}) - 0.05), \\ \bar{o} &= \min(0.75, \text{Quantile}_{0.75}(y_{\text{oor}}) + 0.10), \\ \bar{c} &= \min(0.35, \text{Quantile}_{0.75}(y_{\text{crash}}) + 0.10). \end{aligned} \tag{16}$$

Here y_{oor} and y_{crash} are the binary terminal-event indicators stored for a single training episode, not within-episode event rates. Since $\bar{o} \le 0.75 < 1$ and $\bar{c} \le 0.35 < 1$, these two implementation-level constraints are equivalent to admitting only episodes with $y_{\text{oor}} = 0$ and $y_{\text{crash}} = 0$. The continuous OOR and crash values reported during evaluation are averages of these binary outcomes across evaluation episodes. Once these statistics are available, $(\bar{S}_{\text{ep}}, \bar{\rho}, \bar{o}, \bar{c})$ are cached and remain fixed for all subsequent episode-level admission.

Why the episode reference is frozen. The reported experiments use a bootstrap-quantile reference rather than continuously recomputing the episode reference throughout training. In preliminary runs with an online quantile reference, the admission criterion became progressively harder as the training distribution improved, so fewer new episodes entered $\mathcal{B}_{\mathrm{ep}}$ late in training and the resulting learning performance was weaker. We therefore use early training only to calibrate the episode-level reference and keep the criterion fixed afterward. This also avoids a moving admission target and makes the later replay-buffer population easier to reproduce across independent runs. The online-reference behavior is a design observation from preliminary development rather than a separate ablation result reported in Section 5.

Episode admission and buffer management. A completed episode passes the main episode-level gate when $S_{\mathrm{ep}} \geq \bar{S}_{\mathrm{ep}}$, $\rho_{\mathrm{route}} \geq \bar{\rho}$, $y_{\mathrm{oor}} = 0$, and $y_{\mathrm{crash}} = 0$; the last two conditions are the operational form of the bounded safety references above. The implementation additionally requires both the full episode length and the number of agent-executed candidate transitions to be at least 100. Starting at 18k steps, after $\mathcal{B}_{\mathrm{ep}}$ contains at least 1024 flattened transitions, a stricter gate requires $y_{\mathrm{suc}} \geq 0.5$ and $\rho_{\mathrm{route}} \geq 0.84$; because success is binary, the first condition requires a successful episode.

After an episode passes these gates, transition-level filtering is applied before insertion into $\mathcal{B}_{\mathrm{ep}}$. Unsafe transitions and transitions with mean absolute action above 0.32 are removed, and only transitions at or above the 55th percentile of q_t^{local} are retained. If more than 192 transitions remain, 192 approximately evenly spaced transitions are kept. The episode-level elite replay buffer stores at most 96 episode units and 24k flattened transitions; when capacity is reached, higher-scoring episode units replace weaker ones.

New episode- and segment-level candidates stop entering the elite replay buffers at 42k environment steps. Evaluation trajectories are never used for candidate collection, bootstrap-reference estimation, elite admission, or replay.

A.3 Progress-segment mining and sampling curriculum

The implementation applies a sliding window to the ordered list of agent-executed candidate transitions from each completed training episode after takeover transitions have been removed. Segment mining is independent of episode admission, so a useful local segment may enter $\mathcal{B}_{\mathrm{seg}}$ even when the full episode fails the episode-level gate. The window is only a selection device: each retained transition keeps its true (s, a, r, s') tuple, and no synthetic TD transition is created across an omitted takeover interval.

The window length is $L = 64$ and the stride is 32. For a candidate segment, let $\Delta\rho$ be the increase from its initial route value to the largest route value reached by its retained agent transitions, R_{seg} the sum of their raw rewards, $\bar{u}_{\mathrm{seg}}$ the mean supervisor disagreement, U the unsafe-step ratio, and $\bar{a}$ the mean action magnitude. The terms in Eq. (6) are implemented as

$$\begin{aligned} P_{\Delta\rho} &= \mathrm{clip}(\Delta\rho/0.20, 0, 1), \\ P_R &= \tanh\left(\frac{R_{\mathrm{seg}}}{R_{\mathrm{scale}} L/240}\right), \\ I_{\mathrm{mid}} &= 1 - \min\left(1, \frac{|\bar{u}_{\mathrm{seg}} - 0.55|}{0.35}\right), \\ P_a &= \min(1, \bar{a}/0.28). \end{aligned} \tag{17}$$

The corresponding coefficients are $(0.55, 0.20, 0.15, 0.08, 0.35, 0.05)$ for progress, return, end-route value, disagreement bonus, unsafe ratio, and action magnitude. The disagreement bonus is active only when the supervisor-disagreement component is enabled.

After at least 32 candidate segment scores have been collected, the segment admission reference is the 75th percentile of the accumulated segment-score history. Unlike the frozen episode-level bootstrap reference, this segment reference is recomputed as additional segment scores are observed. A segment must exceed the current reference, make at least 0.08 route progress, have unsafe ratio 0, and have mean action magnitude at most 0.28. After segment-level admission, its individual transitions are filtered again by safety and the 45th percentile of q_t^{local}, and at most 64 transitions

are retained. The progress-segment elite replay buffer $\mathcal{B}_{\text{seg}}$ stores at most 96 segment units and 16k flattened transitions.

The episode-level and progress-segment elite replay buffers are physically separate and both use score-based unit replacement. At update time, the requested elite mini-batch is split according to the current segment ratio, sampled separately from $\mathcal{B}_{\text{ep}}$ and $\mathcal{B}_{\text{seg}}$, and concatenated along the batch dimension. If one buffer cannot provide its nominal share, the unused quota is reassigned to the other buffer.

The segment sampling ratio follows

$$\rho_{\text{seg}}(t) = \begin{cases} \rho_0, & t < t_d, \\ (1-\zeta_t)\rho_0 + \zeta_t \rho_{\min}, & t \geq t_d, \end{cases} \qquad \zeta_t = \text{clip}\left(\frac{t - t_d}{T_d}, 0, 1\right). \tag{18}$$

The reported configuration uses $\rho_0 = 0.25$, $\rho_{\min} = 0.03$, $t_d = 28$k, and $T_d = 10$k. If the most recent 32 training episodes already have mean success at least 0.75 and mean $\max(y_{\text{oor}}, y_{\text{crash}}) \leq 0.22$, the segment fraction is immediately capped at 0.03. Sampling within each flattened replay buffer is uniform over stored transitions; selection quality and replay-time reliability are applied through loss weights rather than prioritized-sampling probabilities.

A.4 Per-sample weighting and auxiliary updates

Selection-quality weight. Each retained transition inherits the score S_u of its episode or segment unit. DER converts it to

$$w_{\text{qual}} = \min\left(w_{\max},\, 1 + \frac{\max(0, S_u - S_{\text{w}}^{(u)})}{\tau_w}\right), \qquad S_u \in \{S_{\text{ep}}, S_{\text{seg}}\}, \tag{19}$$

where $S_{\text{w}}^{(u)}$ is the corresponding quality reference. For episode-level samples, the reference is $S_{\text{w}}^{(\text{ep})}$ from Eq. (15); segment-level samples use the reference maintained by the segment selector. The reported configuration uses $\tau_w = 0.35$ and $w_{\max} = 1.5$.

Current Q support. For the actor-side elite BC term, the implementation uses

$$\Delta Q_e = \min_i Q_i(s_e, a_e) - \min_i Q_i(s_e, \pi_\theta(s_e)). \tag{20}$$

The reported configuration uses a critic margin $\delta_Q = 0$, so the keep mask is one only when $\Delta Q_e \geq 0$. The soft advantage factor in Eq. (7) uses $\tau_Q = 0.05$ and $w_Q^{\min} = 0.05$. This factor affects the actor-side self-imitation term only.

Critic agreement and freshness. For two TD3 critics,

$$d_Q = |Q_1(s_e, a_e) - Q_2(s_e, a_e)|, \qquad w_{\text{conf}} = w_c^{\min} + (1 - w_c^{\min}) e^{-d_Q/\tau_c}, \tag{21}$$

and

$$w_{\text{fresh}} = w_f^{\min} + (1 - w_f^{\min}) \exp\left[-\frac{t - t_{\text{stored}}}{\tau_f}\right]. \tag{22}$$

The reported configuration uses $\tau_c = 0.10$, $\tau_f = 20{,}000$ environment steps, and a floor of 0.35 for both factors.

Supervisor-disagreement weight. The supervisor log-density is converted to the relative disagreement signal

$$u_t = 1 - \text{clip}\big(\exp[\log p_E(a_t^A \mid s_t)], 0, 1\big). \tag{23}$$

For each elite mini-batch, let q_l, q_m, and q_h denote the low, target, and high disagreement quantiles. The information weight is

$$w_{\text{info}} = w_{\min} + (1 - w_{\min}) \begin{cases} \text{clip}\left(\dfrac{u - q_l}{q_m - q_l}, 0, 1\right), & u \leq q_m, \\ \text{clip}\left(\dfrac{q_h - u}{q_h - q_m}, 0, 1\right), & u > q_m. \end{cases} \tag{24}$$

The reported quantiles are $(0.20, 0.60, 0.90)$ with $w_{\min} = 0.30$. If the batch has fewer than four samples or its disagreement values are essentially constant, this factor is set to one to avoid unstable quantiles. Although the implementation retains the historical configuration name "entropy weight," the quantity is not policy entropy.

Elite critic target and update timing. The critic auxiliary term uses the raw environment reward stored in the elite replay buffers:

$$y_e^{\text{elite}} = r_e + \gamma(1 - d_e) \min_i Q'_i(s'_e, \pi'(s'_e) + \epsilon), \qquad \ell_{\text{TD}}^{\text{elite}} = \sum_i (Q_i(s_e, a_e) - y_e^{\text{elite}})^2. \tag{25}$$

The critic weight is $w_{\text{qual}} w_{\text{conf}} w_{\text{fresh}} w_{\text{info}}$; it deliberately does not use the actor-specific Q keep mask or w_{adv}. Both auxiliary losses are normalized by the sum of their effective sample weights, as shown in Eqs. (9)–(10), which keeps their scale stable when many samples are downweighted. The critic draws an elite mini-batch on every critic update once its start and buffer-size conditions are met. The actor draws a separate elite mini-batch only on the delayed TD3 policy-update steps, so the actor and critic auxiliary terms generally do not use exactly the same transitions.

Loss schedules. Elite BC starts once DER is active and the combined elite replay buffers contain at least 96 transitions. Its coefficient warms up over 12k steps from the 7k DER start toward 0.016, starts decaying at 34k, and moves toward 0.005 over 20k steps. Elite critic calibration starts at 9k, warms up over 12k steps toward 0.08, and decays from 34k over 12k steps toward 0.025.

Starting at 36k steps, an adaptive BC floor checks the most recent recorded 32 training-episode metrics. If mean success is below 0.75 or mean $\max(y_{\text{oor}}, y_{\text{crash}})$ is above 0.30, the elite BC coefficient is raised to at least 0.008. This uses training outcomes, not validation rollouts.

Replay-weight summary. The replay logic separates *admission* from *influence*. Episode- and progress-segment criteria first decide whether an agent transition is eligible to enter an elite replay buffer. For a retained sample, the actor-side and critic-side effective weights are

$$w_e^{\text{actor}} = w_{\text{qual}}\, m_Q\, w_{\text{adv}}\, w_{\text{conf}}\, w_{\text{fresh}}\, w_{\text{info}}, \qquad w_e^{\text{critic}} = w_{\text{qual}}\, w_{\text{conf}}\, w_{\text{fresh}}\, w_{\text{info}}.$$

Here, w_{qual} reflects the quality of the admitted episode or segment; (m_Q, w_{adv}) measure whether the stored action is still supported by the current critics; w_{conf} reduces the influence of samples with large twin-critic disagreement; w_{fresh} downweights stale samples; and w_{info} gives the largest weight to moderate, non-extreme disagreement with the supervisor. The segment ratio $\rho_{\text{seg}}(t)$ in Eq. (18) controls how the two elite replay buffers are mixed when an elite mini-batch is formed. The normalized losses in Eqs. (9)–(10) then determine the final contribution of these weighted samples to the auxiliary actor and critic updates.

A.5 Confirmed-best actor rollback

Rollback starts at 30k environment steps and is driven by validation metrics. In the reported launcher, no reward reference is configured, so the implementation uses route completion in the reward-score slot. The effective validation score is therefore

$$S_{\text{val}} = 0.55\,\text{Success} + 0.35\,\text{Route} - 0.25\,\text{OOR} - 0.25\,\text{Crash}. \tag{26}$$

A candidate must satisfy $\text{OOR} \le 0.35$, $\text{Crash} \le 0.20$, $\text{Success} \ge 0.60$, and $\text{Route} \ge 0.792$. Normally its score must exceed the confirmed score by at least 0.005. If the confirmed actor has not been refreshed for 8k steps, a candidate may also qualify when its score is within 0.12 of the current confirmed score.

A qualifying validation point first creates a candidate. A later evaluation within at most 6k steps counts as a confirmation if its score is within 0.08 of the candidate score. Two confirmations are required before the actor is stored as the confirmed best. If a later score is more than 0.08 below the confirmed score, or violates the safety/completion constraints, patience $= 1$ triggers rollback.

The rollback restores the actor, actor target, and actor optimizer state; it does not restore the critic. After restoration, actor optimization is paused for 5k steps, while critic training, including the elite critic term, continues. Validation rollouts never enter the base or elite replay buffers. Because rollback changes the subsequent training trajectory, we treat it as a training-stabilization detail rather than as a separate DER contribution. Without a matched no-rollback run, no performance gain is attributed specifically to this mechanism.

A.6 Implementation summary

Table 3 summarizes the active settings used in the reported DER runs.

Table 3: Key settings used by the reported DER runs. Values come from the active DER launcher rather than class defaults in inactive code branches.

Component	Setting	Role
Base learner	TD3 MLP $[256, 256]$; lr 10^{-4}; $\gamma = 0.99$; $\tau = 0.005$	Fixed shared-control actor–critic backbone.
Base replay/update	capacity 50k; learning starts at 10 steps; batch 1024; train every environment step; no added action noise	Underlying online actor–critic optimization used in all reported DER runs.
Intervention supervision	proxy values ± 1; intervention BC weight 1.0; base replay reward discarded	Fixed shared-control supervision in the reported instantiation.
FakeHuman gate	PPO 4.8M checkpoint; deterministic expert action; density threshold 0.05	Uses the fixed supervisor action-density rule for takeover.
Bootstrap reference	min. 12 episodes; score $q = 0.45$; weight $q = 0.25$; freeze at 7k	Sets admission from early learner history rather than supervisor metrics.
Episode collection	7k–42k; min. full/agent-candidate length 100; route floor 0.65	Defines when episode candidates may enter elite replay buffer.
Episode-level buffer	24k transitions; 96 units; max. 192/episode; local $q = 0.55$	Stores filtered agent-only transitions from strong episodes.
Success gate	from 18k after 1024 episode-buffer transitions; success ≥ 0.5; route ≥ 0.84	Tightens episode admission once enough elite data have been collected.
Quality weight	$\tau_w = 0.35$; cap 1.5	Bounds the effect of high-scoring admitted units.
Progress-segment buffer	length 64; stride 32; 16k transitions; 96 units; local $q = 0.45$	Recovers useful local progress when complete success is sparse.
Segment curriculum	$0.25 \rightarrow 0.03$ after 28k over 10k	Reduces segment use as full-task behavior improves.
Elite mini-batches	actor 192; critic 192; episode/segment sampled separately then concatenated	Uses the current segment ratio while keeping both elite replay buffers physically separate.
Q support	margin 0; $\tau_Q = 0.05$; floor 0.05	Prevents actor imitation of elite actions no longer supported by current Q values.
Critic confidence	$\tau_c = 0.10$, floor 0.35	Downweights elite samples with high twin-critic disagreement.
Freshness	$\tau_f = 20$k, floor 0.35	Reduces the effect of stale elite behavior.
Supervisor disagreement	batch quantiles $0.20/0.60/0.90$, floor 0.30	Gives highest weight to moderate, non-extreme deviation.
Elite BC	base 0.016; late floor 0.005; recovery floor 0.008	Adds bounded self-imitation of agent-generated actions.
Elite TD	base 0.08; late floor 0.025	Uses raw-reward TD calibration on selected agent transitions.
Rollback	start 30k; two confirmations; 5k actor pause	Stabilizes late training; not part of elite admission.

A.7 Implementation organization

The reported DER implementation is built on the same TD3-based shared-control learner described in Appendix A.1. The implementation keeps the base HIL replay path separate from DER's agent-only elite replay path. The DER-specific path contains the episode- and progress-segment buffers, their admission rules, and the replay-time weighted actor and critic updates described in Sections 4 and Appendix A.4.

The experimental codebase also contains unrelated or historical branches that are disabled in the reported configuration and do not contribute to the results in this paper. All active algorithmic compo-

nents and hyperparameters needed to reproduce the reported DER configuration are summarized in Appendix A.6.

B Experimental and Reproducibility Details

This appendix provides the exact protocol and run-level statistics for Section 5. The main text focuses on the experimental questions; reproduction and aggregation details are collected here.

B.1 Training, intervention, and evaluation protocol

The policy receives vector observations and outputs a two-dimensional continuous action for steering and longitudinal acceleration/braking. MetaDrive uses 50 training scenarios beginning at scenario index 100 and 50 evaluation scenarios beginning at index 1000, with traffic density 0.06 and a maximum episode length of 1500 steps. Training uses the FakeHuman environment, while evaluation disables corrective control so that the learned policy acts autonomously.

The proxy supervisor is a PPO policy pretrained for 4.8M steps. It is fixed throughout learner training and is used only after the existing takeover rule is triggered. Deterministic supervisor execution uses the mode of the supervisor action distribution. Takeover is triggered when the density of the learner-proposed action under the supervisor distribution falls below 0.05.

Table 1, the safety analysis, the learning-curve comparison, and the cumulative ablation use the same four-run matched subset for all corresponding methods or variants. Four additional independent DER runs are used only for the expanded stability analysis in Section 5.4 and Appendix B.2, giving eight DER runs in total. Each run uses an independent random seed and independently initializes the policy, optimizer, base replay buffers, and DER elite replay buffers; no replay data or checkpoint state is shared across runs.

Each training run contains $50{,}000$ executed environment interactions. The current policy is evaluated every 1000 interactions for 20 autonomous episodes. Validation trajectories never enter the base or elite replay buffers, never contribute to elite-reference estimation, and never produce gradient updates. Because the confirmed-best checkpoint mechanism in Appendix A.5 uses the same validation metrics for model selection, the reported endpoint values are validation results rather than measurements from a sealed held-out test set.

B.2 Statistical definitions and run-level results

For independent run-level measurements $x_1, \ldots, x_n$, we report

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \qquad s_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}. \tag{27}$$

The uncertainty reported in the tables is sample standard deviation across independent training runs, not standard error or a confidence interval.

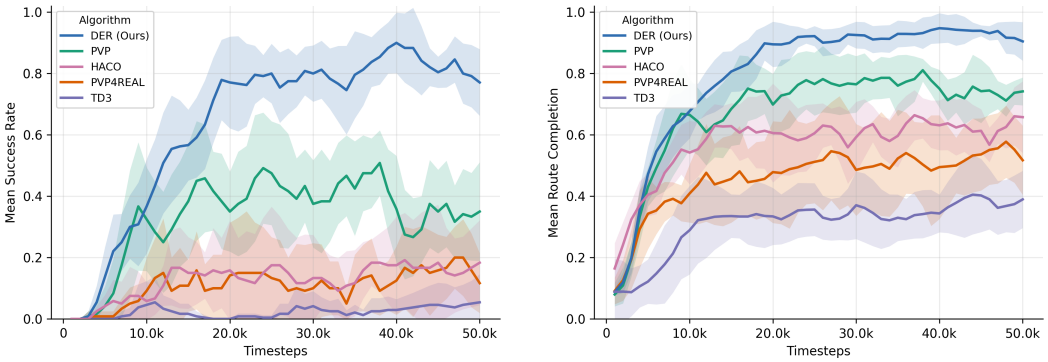
Table 4: Run-level DER validation results. Runs 1–4 are the original matched subset used in Table 1; Runs 5–8 are additional independent DER runs used only for the expanded stability analysis.

Run	Final Success	Last-10 Success	Peak Success	Route	Reward
Run 1	0.900	0.850	1.000	0.898	300.1
Run 2	0.700	0.820	1.000	0.867	303.1
Run 3	0.800	0.785	0.950	0.954	342.4
Run 4	0.850	0.855	1.000	0.971	382.1
Matched 4 mean ± SD	0.813 ± 0.085	0.828 ± 0.032	0.988 ± 0.025	0.923 ± 0.048	332.0 ± 38.6
Run 5	0.850	0.695	0.850	0.953	343.6
Run 6	0.850	0.760	0.900	0.917	311.0
Run 7	0.700	0.780	0.900	0.885	315.7
Run 8	0.650	0.620	0.800	0.905	323.9
All 8 mean ± SD	0.788 ± 0.092	0.771 ± 0.080	0.925 ± 0.076	0.919 ± 0.037	327.7 ± 27.3

The DER row in Table 1 uses the matched four-run aggregate, preserving the same aggregation scope as the comparison methods. The eight-run aggregate is used only to assess DER's run-to-run stability. Last-10 Success averages the final ten validation points within each run, while Peak Success records the maximum validation success reached during training; both are descriptive training diagnostics rather than endpoint comparison metrics.

B.3 Training dynamics for the matched comparison

Figures 3 and 4 report validation dynamics for the common matched run subset. Solid curves show the across-run mean at each validation point, and shaded regions show one sample standard deviation across that subset. The numerical values in Table 1 come from the final validation point; the curves provide temporal context and do not replace the run-level endpoint comparison.



(a) Success rate **(b) Route completion**

Figure 3: Task-completion dynamics for the matched comparison over the full interaction budget. DER is compared with TD3, HACO, PVP, and PVP4Real under the same validation schedule as Table 1.

The task-level curves show how the matched methods evolve during training. DER generally maintains stronger late-stage success and route completion within this subset, while the endpoint comparison in Table 1 remains the primary quantitative result.

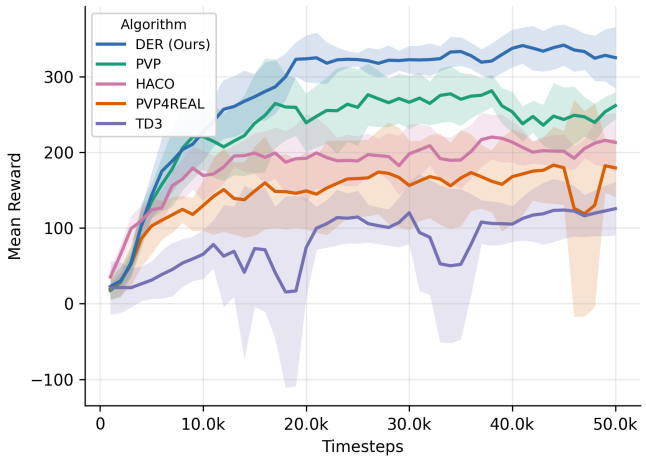
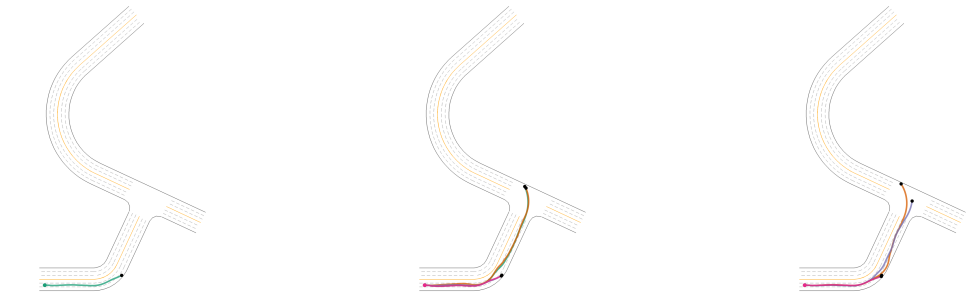


Figure 4: Episodic-reward dynamics for the matched comparison. The reward curve is shown separately from the task-completion metrics to keep the panels readable; its endpoint ordering is consistent with Table 1.



(a) Suboptimal Expert **(b) HACO** **(c) PVP**



(d) PVP4Real **(e) DER (Ours)** **(f) All Methods**

Figure 5: Qualitative trajectory comparison in a common MetaDrive scene. Panels (a)–(e) show the fixed suboptimal supervisor, HACO, PVP, PVP4Real, and DER, and panel (f) overlays the trajectories. The visualization is intended only to illustrate representative behavioral differences; all quantitative claims are based on aggregate validation statistics across independent runs.

B.4 Qualitative trajectory visualization

Figure 5 provides a qualitative visualization of one common MetaDrive scene using the fixed suboptimal supervisor and the learned comparison policies. The purpose of this figure is to make the behavioral differences visible in a concrete rollout, not to estimate average performance. In this scene, DER follows the drivable branch through the intersection and continues along the route, whereas several comparison trajectories terminate earlier or deviate from the successful path. Because a single rollout can be sensitive to the sampled scene and policy realization, we do not use this example to rank methods or to support statistical claims; the quantitative conclusions remain those reported from the independent-run validation results in Table 1, Table 5, and Appendix B.3.

B.5 Safety outcomes and safety dynamics

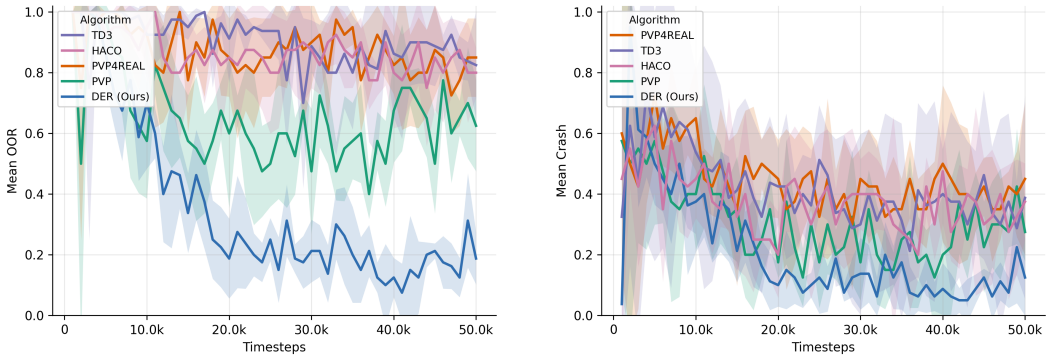
Out-of-road (OOR) and collision rates measure failure modes rather than task completion and are therefore reported separately. OOR and collision are not mutually exclusive. For direct comparability across methods, Table 5 uses the common matched run subset described in Appendix B.1.

Table 5: Safety outcomes at the final validation point for the common matched run subset. Entries are mean $\pm$ sample standard deviation.

Method	OOR ↓	Crash ↓
Suboptimal supervisor	0.720	0.310
TD3	0.825 ± 0.176	0.388 ± 0.333
HACO	0.800 ± 0.082	0.375 ± 0.126
PVP	0.625 ± 0.150	0.275 ± 0.096
PVP4Real	0.850 ± 0.129	0.450 ± 0.252
DER (Ours)	$\mathbf{0.188 \pm 0.085}$	$\mathbf{0.125 \pm 0.126}$

DER has the lowest mean OOR and crash rates among the evaluated methods at the final validation point. Relative to the fixed suboptimal supervisor, the mean OOR rate decreases from 0.720 to 0.188 and the mean crash rate decreases from 0.310 to 0.125. These are observed validation means rather than claims of statistical significance or formal safety guarantees.

Figure 6 shows how the two failure rates evolve during training. The safety advantage is therefore not evaluated only at the final checkpoint: DER trends toward lower OOR and crash rates over the later training stage, while the comparison methods remain substantially higher on at least one of the two failure metrics.

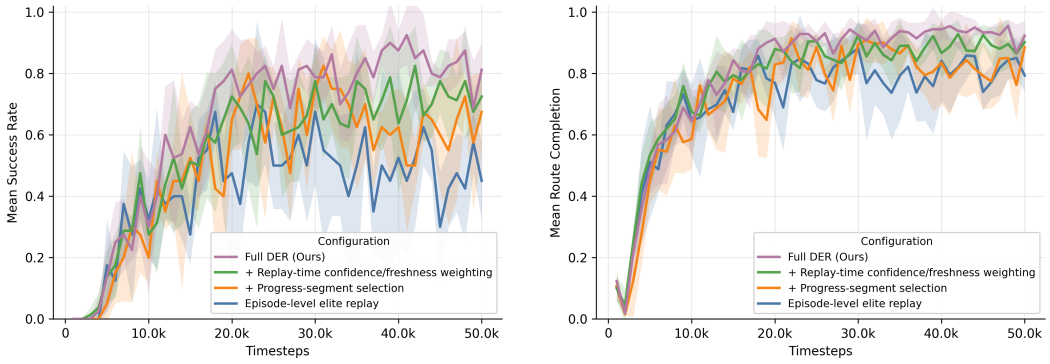


(a) Out-of-road rate **(b) Crash rate**

Figure 6: Safety validation dynamics over the full matched interaction budget. Lower is better. Panel (a) reports out-of-road rate and panel (b) reports crash rate; Table 5 provides the endpoint statistics.

B.6 Ablation learning dynamics

Table 2 reports the endpoint measurements for the cumulative DER ablation. Figure 7 focuses on success and route completion, where the temporal separation between configurations is most informative. The episodic-reward trajectory is retained separately in Figure 8 for completeness because the stronger variants become close in reward later in training.



(a) Success rate **(b) Route completion**

Figure 7: Task-completion dynamics for the cumulative DER ablation. Each configuration adds one component to the preceding variant: episode-level elite replay, progress-segment selection, replay-time confidence/freshness weighting, and the complete DER weighting rule. Final values are reported in Table 2.

The curves support two observations. First, progress-segment selection produces a clear improvement over episode-only replay in both success and route completion, showing that safe local progress provides useful training signal before complete successful episodes are consistently available. Second, replay-time weighting further improves late-stage task completion, with Full DER achieving the

strongest final success and route completion in Table 2. These curves therefore support the role of the weighting stage as controlling which retained samples have the greatest influence on subsequent learning.

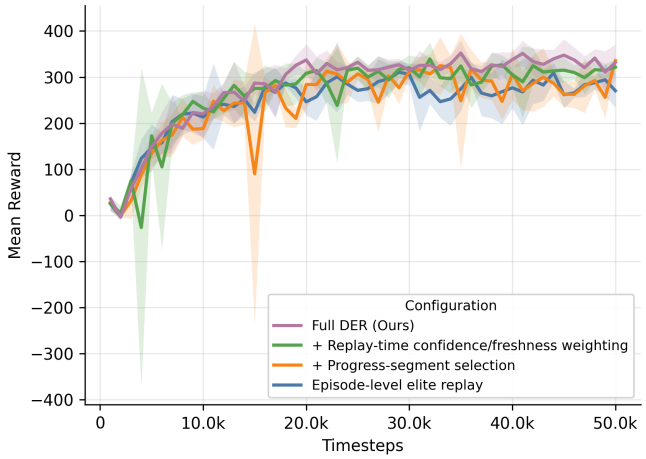


Figure 8: Episodic-reward dynamics for the cumulative DER ablation. The stronger variants remain close in reward during part of late training, while the endpoint means follow the cumulative improvement reported in Table 2.

The reward curves remain relatively close during late training, but their endpoint means follow the same cumulative trend as Table 2: progress-segment selection improves reward over episode-level elite replay, the confidence/freshness stage increases it further, and Full DER achieves the highest final mean reward. Because all ablation variants use the same task outcomes, training budget, supervisor, and underlying learner, this comparison is designed to isolate DER's internal selection and replay-weighting choices rather than differences in reward information.

B.7 Component definitions and process diagnostics

The rows in Table 2 are cumulative and correspond exactly to the configurations visualized in Figures 7 and 8. All rows use the same matched run subset. The expanded eight-run DER aggregate is reported only in Section 5.4 and Appendix B.2 as a stability analysis, not as a replacement for the matched comparison in Table 1. *Episode-level elite replay* enables only the episode replay buffer and uses the elite actor/critic losses without progress-segment mining or dynamic replay-time reliability factors. + *Progress-segment selection* enables the second elite replay buffer and route-progress segment admission while keeping the replay-time confidence, freshness, Q-support, and supervisor-disagreement factors disabled. + *Replay-time confidence/freshness weighting* activates w_{conf} and w_{fresh} from Eqs. (21)–(22), so uncertain or old retained samples receive less influence. *Full DER* additionally activates the actor-side Q-support terms (m_Q, w_{adv}) in Eq. (7) and the supervisor-disagreement factor w_{info} in Eq. (24); the disagreement component also enables the small I_{mid} term in segment scoring. Other hyperparameters are held fixed rather than retuned separately for each row.

The confidence temperature is 0.10 with a minimum weight of 0.35; the freshness time scale is 20k steps with the same floor. Supervisor-disagreement weighting uses the $0.20/0.60/0.90$ batch quantiles and a 0.30 minimum weight. These settings are identical to Appendix A.4. Because all four rows use the same reward and task-outcome access and the same matched protocol, the ablation is designed to isolate the cumulative contribution of the DER components above.

During implementation, replay-buffer occupancy, segment sampling, critic-disagreement, freshness, current-Q support, and supervisor-disagreement statistics are monitored as sanity checks to verify that the corresponding DER components are active. These diagnostics are not used as empirical evidence or as substitutes for the matched component ablations in Table 2; the quantitative component analysis is based on the task-performance results and learning curves reported above.

B.8 COMPARISON PROTOCOL AND SCOPE OF EVIDENCE

For a fair comparison, the matched runs use the same MetaDrive task definition, observation/action spaces, interaction budget, randomization protocol, and evaluation schedule while preserving each method's own learning rule. TD3 is used as a no-intervention reward-driven reference. HACO (Li et al., 2022), PVP (Peng et al., 2023), and PVP4Real (Peng et al., 2025) provide HIL-related comparisons with distinct intervention-learning objectives, whereas DER focuses on selecting, re-evaluating, and reusing high-quality agent-generated experience after shared-control interaction. We do not rewrite the baselines as DER variants. Published numbers from different MetaDrive versions, supervisor checkpoints, budgets, or evaluation protocols are treated only as literature context rather than as direct entries in Table 1.

Table 6: Implementation and matching summary for the compared methods. "Same proxy" denotes the fixed proxy-supervisor setting used by the matched HIL comparisons; each baseline otherwise preserves its own learning rule. PVP+Reward is the reward-matched PVP control used to test whether environment-reward access alone accounts for DER's improvement.

Method	Learner/backbone	Intervention	Learning signal	Budget	Matching / tuning
TD3	TD3	None	Environment reward	50k	Same task/evaluation
HACO	Method-specific HIL	Same proxy	Method-specific	50k	Original learning rule
PVP	Proxy-value HIL	Same proxy	Proxy values; base reward discarded	50k	Original learning rule
PVP4Real	Method-specific HIL	Same proxy	Method-specific	50k	Original learning rule
PVP+Reward	PVP-style HIL	Same proxy	Proxy values + environment reward	50k	Reward-matched PVP control
DER	DER HIL backbone	Same proxy	Base proxy + elite raw reward	50k	DER selection/reweighting

PVP+Reward is implemented from the original PVP configuration by enabling environment-reward learning while keeping DER-specific episode/segment elite selection and replay-time reliability weighting disabled. It is used only as a matched control for reward access, not as an additional proposed method.

In the reported instantiation, the underlying shared-control learner follows the proxy-value formulation used in PVP (Peng et al., 2023) and keeps its reward-discarding base replay, while DER separately uses raw environment return and task outcomes for elite admission and for the auxiliary elite TD target. DER should therefore not be described as reward-free. This implementation detail is kept explicit in comparisons, but it is separate from DER's core contribution: identifying and reusing validated agent-generated behavior.

The confirmed-best mechanism uses the same evaluation stream reported for endpoint validation, although those trajectories never enter replay or gradient updates. This creates a model-selection dependence that should not be confused with data leakage into training. We therefore report these values as validation performance and do not attribute a separate performance gain to rollback. Claims about generalization to unseen scenarios would require a disjoint frozen-policy test and are outside the evidence reported here.