PACER: Progress-Aligned Curation for Error-Resilient Imitation Learning

Shreyas Kumar and Ravi Prakash
Robert Bosch Centre for Cyber-Physical Systems
Indian Institute of Science, Bengaluru, India
{shreyaskumar, ravipr}@iisc.ac.in

Abstract: Imitation learning from human demonstrations offers a powerful framework for robotic skill acquisition, eliminating the need for explicit reward specification. However, in realistic low-data settings, where only a handful of demonstrations are available for each task, demonstrations are not only expensive to collect but can also be imperfect: operators may experience fatigue, vary in execution strategies or timing, and occasionally introduce brief corrections or deviations, such as unintended motions or hesitations, that can occur at arbitrary stages of task execution. To address these challenges, we introduce PACER, a progress-aligned framework that aligns demonstrations in latent task phase and robustly filters local corruptions before policy learning. PACER enables reliable imitation from sparse and noisy data, yielding policies that better capture intended behavior and outperform standard behavioral cloning and alignment baselines across manipulation and locomotion domains respectively by roughly 45% and improves episodic returns by over 50% on average. Code available at: anonymous.4open.science/r/PACER.

Keywords: Imitation Learning, Imperfect Demonstrations, Learning

1 Introduction

Robots that can acquire skills directly from human demonstrations promise to reduce the burden of hand-designed reward functions and enable rapid deployment in unstructured environments. Imitation learning (IL) provides a direct way to train such policies by mapping observed states to demonstrated actions. However, in realistic settings, demonstrations are both expensive to collect and often imperfect. Operators may vary in timing and execution strategies, experience fatigue, or introduce brief unintended deviations such as slip-ups or hesitations. These imperfections are especially problematic in low-data regimes, where each demonstration carries significant weight for policy learning.

Two recurring challenges arise in this setting. First, *pace variability*, demonstrations of the same task may unfold at different speeds, so the same stage of task execution can appear at different time indices across demonstrations. Second, *local corruptions*, short but notable deviations in actions, which are not random noise but biased errors, can mislead a policy if treated as ground truth.

We propose **PACER** (Progress-Aligned Curation for Error-Resilient imitation learning), a framework that addresses both challenges jointly. PACER first learns a state-dependent latent task phase that provides a shared progress variable for temporal alignment across demonstrations. It then uses robust statistics in a leave-one-out consensus scheme to detect and down-weight corrupted segments. Finally, these consensus signals are used to construct refined pseudo-labels that repair corrupted samples before policy training. The result is a simple weighted behavioral cloning procedure that remains stable and robust even when demonstrations are sparse and noisy.

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

Contributions. Our contributions are threefold: 1. We introduce a learned, state-dependent latent task phase that enables temporal alignment across demonstrations with varying paces. 2. We adapt robust statistical tools (medians, median absolute deviation, Tukey reweighting) in a leave-one-out consensus scheme to identify and filter local corruptions without letting demonstrations validate themselves. 3. We design a pseudo-label refinement procedure that leverages consensus anchors, directional alignment, and temporal smoothing to repair corrupted samples, enabling robust imitation learning with a simple weighted Huber behavioral cloning loss.

We evaluate PACER on manipulation and locomotion domains with limited and corrupted demonstrations, showing that it consistently improves policy robustness and fidelity compared to behavioral cloning and alignment-only baselines.

2 Related Work

Phase variables and temporal alignment. Movement primitives encode trajectories with an explicit phase or progress variable to synchronize executions across different paces [1, 2]. Beyond hand-crafted schedules, differentiable sequence alignment via soft-DTW has enabled learning with time-warping losses [3], and progress can be estimated directly from observations using representation learning (e.g., time-contrastive signals) [4]. Task-sketch alignment methods such as TACO jointly align weak supervision with demonstrations while learning policies [5]. Our work adopts the idea of a phase variable but *learns* a state-only monotone phase and uses it strictly as a synchronization coordinate before robust aggregation. Notably, generic time-warping objectives align sequences but do not by themselves detect or suppress short, biased action deviations; in the absence of additional robustness, corrupted segments may also be time-warped into agreement.

Robust imitation from imperfect demonstrations. A complementary line of work focuses on robustness when demonstrations are noisy or suboptimal. Approaches include symmetric-loss objectives and median-of-means formulations [6, 7], as well as confidence-based methods that reweight data using (possibly noisy) quality scores. For example, Wu et al. propose 2IWIL and IC-GAIL, which combine confidence with unlabeled data through importance weighting and occupancy-measure matching [8]. Other methods estimate demonstrator expertise more explicitly: ILEED jointly learns a policy and state-dependent expertise using demonstrator identities, motivated by crowd-sourced settings where proficiency varies widely [9]. While effective at filtering suboptimal contributors, such approaches often assume broad state coverage or large datasets (e.g., millions of games in chess) to reliably infer expertise. In contrast, PACER operates in few-demo regimes and targets short, within-trajectory corruptions, aligning demonstrations by phase and applying robust statistics locally without requiring demonstrator identities or extensive data.

Reward-learning and noise modeling under suboptimality. VILD tackles diverse-quality demonstrations by explicitly modeling a noise density and learning a reward alongside demonstration quality via a variational IRL/RL procedure; this addresses compounding error in naive regression-style estimators and improves data-efficiency with importance sampling [10]. In real-world crowdsourced demos, VILD demonstrates robustness but requires substantial environment interaction during RL optimization (e.g., millions of transitions), which may be impractical in some settings. Our approach differs in staying within a pure IL/BC regime: we *repair* labels using phase-local consensus and robust reweighting, then train with a weighted Huber loss, avoiding additional rollouts.

Robust statistics used in PACER. We rely on classical robust tools—medians and the median absolute deviation (MAD, with the Gaussian consistency factor 1.4826) [11], Tukey's biweight for redescending reweighting [12], and the Huber loss for training [13]. Our contribution is not these estimators themselves but their *integration* into a leave-one-out, per-phase consensus that curates short local corruptions before standard behavioral cloning.

Positioning. Where alignment-only methods typically assume clean labels once synchronized, and robustness-only methods often assume temporally aligned data, PACER combines a learned state-only phase with per-phase robust curation and pseudo-label construction. This yields a simple training objective (weighted Huber behavioral cloning) while addressing both pace variability and short corruptions observed in real demonstrations. Compared to expertise-reweighting (ILEED) and reward-learning with noise models (VILD) that can be data-hungry or require environment dynamics, PACER emphasizes low-data practicality and label repair at the phase-local level.

3 Methodology

3.1 Problem Formulation

We are given a set of N demonstrations $\{\mathcal{D}_i\}_{i=1}^{N}$, where each demonstration $\mathcal{D}_i$ consists of a time-ordered sequence of state-action pairs:

$$\mathcal{D}_i = \{(x_{i,t}, a_{i,t})\}_{t=1}^{T_i},$$

with $x_{i,t} \in \mathbb{R}^{d_x}$ representing the state and $a_{i,t} \in \mathbb{R}^{d_a}$ denoting the corresponding action at time t.

Demonstrations may include short segments of local corruption, such as unintended deviations or perturbations, but are assumed to share a consistent underlying task intent. That is, all demonstrations aim to complete the same task, even if they differ in timing or minor execution details.

We aim to robustly extract the underlying task intent so that a learned policy can reproduce the intended behavior while disregarding short corruptions. Two sources of variability make this nontrivial:

1. **Pace variability:** Demonstrations may progress at different rates; the same task phase can occur at different points in time across demonstrations.
2. **Local corruptions:** Short segments may exhibit biased deviations in action, which are not merely stochastic noise but notable shifts.

Our approach consists of two steps: (i) align all demonstrations using a shared phase variable that captures task progression, and (ii) compute robust per-phase statistics. These statistics are intended to represent typical behavior at each phase while mitigating the influence of transient corruptions.

3.2 Phase Alignment

We define a phase variable $\tau \in [0, 1]$ to represent task progression across demonstrations. Unlike absolute time, τ depends only on the state, enabling alignment even when demonstrations run at different speeds.

To estimate τ, we use a scoring function $g_\psi : \mathcal{X} \to \mathbb{R}$, parameterized by a small neural network. The requirement is monotonicity, i.e. for $t > t'$,

$$g_\psi(x_{i,t}) > g_\psi(x_{i,t'}).$$

We enforce this with a pairwise ranking loss:

$$\mathcal{L}_{\text{rank}} = \sum_{i} \sum_{t>t'} \log\Big(1 + \exp\big(m - (g_\psi(x_{i,t}) - g_\psi(x_{i,t'}))\big)\Big),$$

where $m > 0$ is a margin that enforces separation between earlier and later states, preventing collapse. Finally, scores are normalized to $[0, 1]$ to yield $\tau_{i,t}$.

3.3 Robust Consensus Statistics

We discretize the phase interval $[0, 1]$ into B equal bins. Each sample $(x_{i,t}, a_{i,t})$ is assigned to the bin corresponding to its phase value $\tau_{i,t}$. Formally, for bin $b \in \{0, \ldots, B-1\}$ we define

$$\mathcal{I}_b = \{(i,t) \mid \tau_{i,t} \in [\tfrac{b}{B}, \tfrac{b+1}{B})\},$$

which is the set of indices of all samples across demonstrations whose phase lies within the interval of bin b. Thus, bin b always represents the same fraction of task completion across all demonstrations. Longer demonstrations contribute more samples to each bin, while shorter ones contribute fewer.

For each bin b, we collect all samples indexed by $\mathcal{I}_b$ and compute robust summaries that capture the typical behavior at that phase of the task while resisting corruption. The median action and median state

$$\alpha_a[b] = \mathrm{median}\{\, a_{i,t} : (i,t) \in \mathcal{I}_b \,\}, \qquad \alpha_s[b] = \mathrm{median}\{\, x_{i,t} : (i,t) \in \mathcal{I}_b \,\},$$

serve as stable anchors, representing what demonstrations usually do at that phase. The median magnitudes

$$\beta_a[b] = \mathrm{median}\{\, \|a_{i,t}\| : (i,t) \in \mathcal{I}_b \,\}, \qquad \beta_s[b] = \mathrm{median}\{\, \|\dot{x}_{i,t}\| : (i,t) \in \mathcal{I}_b \,\}.$$

capture the typical strength of actions and the typical rate of state change, giving a sense of the *pace* of execution. To characterize local task dynamics, we compute approximate tangents by differencing the per-bin medians, yielding an action tangent $t_a[b]$ from $\alpha_a[b]$ and a state tangent $t_s[b]$ from $\alpha_s[b]$.

3.4 Trust Estimation via LOO Residuals

To ensure that corrupted demonstrations do not *validate* themselves, all per-sample trust computations are performed in a leave-one-out (LOO) manner. When evaluating a sample from demonstration i, the bin-level consensus statistics are recomputed without including that same demonstration. Formally, for bin b,

$$\mathcal{I}_b^{(-i)} = \{(j,u) \in \mathcal{I}_b : j \neq i\}.$$

To identify which samples are reliable and which should be down-weighted, we define $r_{i,t}$ as the action residual in bin b.

$$r_{i,t} = \|a_{i,t} - \alpha_a^{(-i)}[b]\|.$$

This is the distance between the actual action and the bin's median action. Large residuals signal the samples that disagree with the phase consensus.

Residuals must be normalized because some bins are naturally more variable than others. We use the median absolute deviation (MAD), a robust measure of spread:

$$\mathrm{MAD}_a^{(-i)}[b] = 1.4826 \times \mathrm{median}\big\{\, |r_{j,u}^{(-i)} - \mathrm{median}\{r_{m,v}^{(-i)} : (m,v) \in \mathcal{I}_b^{(-i)}\}| : (j,u) \in \mathcal{I}_b^{(-i)} \big\}.$$

The constant 1.4826 ensures that for Gaussian data, MAD matches the standard deviation. Unlike variance, MAD is stable even if a minority of points are extreme outliers.

By dividing each residual by the robust scale, we obtain a robust z-score:

$$z_{i,t}^{(-i)} = \frac{r_{i,t}^{(-i)}}{\mathrm{MAD}_a^{(-i)}[b] + \varepsilon},$$

where $\varepsilon > 0$ prevents division by zero. This dimensionless score says how many *robust standard deviations* away the sample lies from the median. Values near zero mean strong agreement; large values indicate deviation.

To translate these scores into trust values, we employ the Tukey biweight function

$$w_{i,t}^{(-i)} = \begin{cases} \big(1 - (z_{i,t}^{(-i)}/c)^2\big)^2, & z_{i,t}^{(-i)} \le c, \\ 0, & z_{i,t}^{(-i)} > c, \end{cases} \qquad w_{i,t}^{(-i)} \leftarrow \max(w_{i,t}^{(-i)}, w_{\min}),$$

with cutoff $c \in [3,5]$. Samples close to the median receive weights near one, moderate deviations are smoothly down-weighted, and extreme outliers are entirely discarded. A small floor, $w_{i,t}$ prevents vanishing gradients. This acts like an *attention filter*, retaining the bulk of consistent demonstrations while rejecting local corruptions.

3.5 Pseudo-Labels and Policy Learning

For each phase bin b, we consolidate the robust summaries into a structured descriptor, termed a *ribbon token*:

$$z_b = \Big(\alpha_a[b],\, \beta_a[b],\, \alpha_s[b],\, \beta_s[b],\, t_a[b],\, t_s[b],\, \mathrm{MAD}[b]\Big).$$

Each token thus encodes both consensus behavior and the degree of variability present at that phase.

Given a sample $(x_{i,t}, a_{i,t})$ from demonstration i, assigned to bin b, we refine its raw label into a pseudo-label $y^*_{i,t}$. This pseudo-label integrates information from the demonstration itself with the consensus ribbon token, while respecting the trust score $w_{i,t}$. The procedure unfolds in five steps:

1. **Debiasing toward the anchor:** Raw actions are softly corrected toward the leave-one-out median action $\alpha_a^{(-i)}[b]$:

   $$y^{(1)}_{i,t} = \gamma_{i,t}\, a_{i,t} + (1-\gamma_{i,t})\, \alpha_a^{(-i)}[b],$$

   with

   $$\gamma_{i,t} = 1 - \lambda_{\text{debias}}\,(1 - w_{i,t}), \quad \gamma_{i,t} \in [0,1].$$

   When a sample is consistent ($w_{i,t} \approx 1$), the debiased label remains close to the original action. When trust is low, the target is shifted toward the per-bin consensus. This ensures we *repair* the label toward what most demos do at that phase.

2. **Alignment with the ribbon tangent:** Each bin provides a canonical forward direction. When state geometry is informative, say for manipulation tasks, the state tangent $t_s[b]$ is chosen; otherwise the action tangent $t_a[b]$ serves as a fallback:

   $$t_{\mathrm{dir}}[b] = \begin{cases} t_s[b], & \text{if available,} \\ t_a[b], & \text{otherwise.} \end{cases}$$

   This choice ensures that progress is consistently expressed relative to the ribbon, independent of individual demonstration deviations.

3. **Sideways attenuation:** The debiased label is decomposed relative to $t_{\mathrm{dir}}[b]$:

   $$y^{(1)}_{\parallel} = \big(y^{(1)}_{i,t} \cdot t_{\mathrm{dir}}[b]\big)\, t_{\mathrm{dir}}[b], \qquad y^{(1)}_{\perp} = y^{(1)}_{i,t} - y^{(1)}_{\parallel}.$$

   Perpendicular components are attenuated in proportion to corruption:

   $$y^{(2)}_{i,t} = y^{(1)}_{\parallel} + (1-\rho_{i,t})\, y^{(1)}_{\perp}, \qquad \rho_{i,t} = \rho_0(1-w_{i,t})\, \mathbf{1}_{\{\text{state tangent available}\}}.$$

   Here $\rho_0 \in [0,1]$ sets the maximum shrinkage applied when trust is minimal. Thus, in tasks where there is a clear geometric path ($\mathbf{1} = 1$), low-trust samples are pulled toward the ribbon, while in tasks without such a path ($\mathbf{1} = 0$) the adjustment is disabled.

4. **Speed regularization:** The action magnitude is softly adjusted toward the robust bin speed $\beta_a[b]$:

   $$s_{i,t} = (1-\eta_{i,t})\, \|y^{(2)}_{i,t}\| + \eta_{i,t}\, \beta_a[b], \qquad \eta_{i,t} = \eta_0(1-w_{i,t}).$$

   Here $\eta_0 \in [0,1]$ controls the maximum influence of the consensus speed when trust is minimal. The final scaled target is

   $$y^{(3)}_{i,t} = s_{i,t}\, \frac{y^{(2)}_{i,t}}{\|y^{(2)}_{i,t}\| + \varepsilon}.$$

   Thus, high-trust samples ($w_{i,t} \approx 1$) retain their original magnitude, while low-trust samples are gradually rescaled toward the bin-level consensus pace.

5. **Temporal smoothing:** To encourage coherence over time, we apply an exponential moving average (EMA) across consecutive targets in the same demonstration:

   $$y^*_{i,t} = (1-\kappa_{i,t})\, y^{(3)}_{i,t} + \kappa_{i,t}\, y^*_{i,t-1},$$

   where $\kappa_{i,t}$ increases when both the current and previous samples have low trust, thereby smoothing uncertain regions more strongly.

Training then reduces to a standard weighted behavioral cloning loss:

$$\mathcal{L} = \mathbb{E}_{i,t}\left[\, w_{i,t}\, \mathrm{Huber}(f_\theta(x_{i,t}) - y^*_{i,t}) \,\right].$$

All handling of corrupted demonstrations is encoded in the pseudo-label construction, while the learning stage itself remains simple and stable.

4 Experiments

We evaluate **PACER** in two domains: (i) manipulation with a **Franka wiping** task, and (ii) locomotion with **Hopper-v4**. In both settings, demonstrations are intentionally constructed to include short segments of unintended behavior, capturing realistic operator slip-ups.

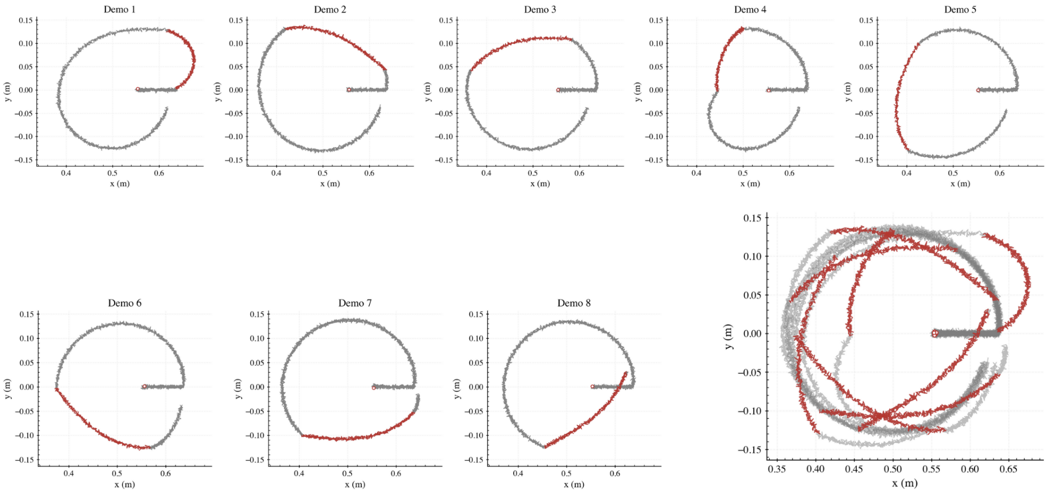

Figure 1: **Franka wiping demonstrations.** Each panel depicts an end-effector trajectory during the wiping task. Red segments highlight portions of the motion influenced by injected perturbations in the action space, representing unintended deviations from the intended circular path.

Franka wiping demonstrations. The task requires the robot to follow a circular wiping trajectory on a planar surface. To simulate realistic operator behavior, demonstrations are generated in MuJoCo with natural variability in execution speed and with controlled perturbations (highlighted in red in Fig. 1) that introduce localized deviations in end-effector velocity. The system state is defined by the end-effector (EE) Cartesian position (x, y), and the control input corresponds to the EE velocity command $(\dot{x}, \dot{y})$.

Hopper demonstrations. Demonstrations are generated in MuJoCo by collecting expert rollouts from a PPO policy and augmenting them with short segments of perturbed actions, which locally alter joint behavior while preserving the overall task structure. Early terminations are retained to capture variability in demonstration quality, producing a dataset that includes both successful and imperfect rollouts.

Baselines. We compare PACER against standard behavioral cloning variants, **BC**, **Weighted-BC**, and **BC-RNN**. In addition, we include **ILEED** [9], which introduces a state and demonstration dependent expertise variable $\rho(s, i)$ that adjusts the likelihood of observed actions. In the continuous-action setting, as considered here, ρ scales the covariance of the Gaussian policy ($\Sigma \mapsto \Sigma/\rho$). Details are deferred to the Appendix.

Evaluation protocol. For **Hopper-v4** we report episodic return under two evaluation settings: *clean*, corresponding to runs without test-time noise, and *perturbed*, where controlled action perturbations are introduced at test time. For **Franka wiping** we evaluate trajectory accuracy with respect to the reference circle using two metrics: (i) root mean squared error (RMSE, in meters) measuring average deviation, and (ii) maximum deviation (MaxDev, in meters) measuring the worst-case error. All results are averaged over three seeds; we report mean ± standard deviation together with the 95% confidence interval (CI) computed across per-seed means.

Method	Clean Return ↑		Perturbed Return ↑	
BC	1231.1 ± 153.8	[174.0]	895.10 ± 21.70	[24.60]
BC-RNN	451.10 ± 143.3	[162.1]	372.40 ± 115.7	[130.9]
Weighted-BC	1671.9 ± 723.2	[818.4]	1040.5 ± 10.60	[12.00]
ILEED	492.50 ± 116.0	[131.3]	487.20 ± 117.1	[132.5]
PACER	**2545.1** ± 647.6	[732.9]	**1876.4** ± 653.7	[739.7]

Table 1: **Hopper-v4.** Episodic returns (higher is better). Values are mean ± std with 95% CI in brackets, n=3 seeds.

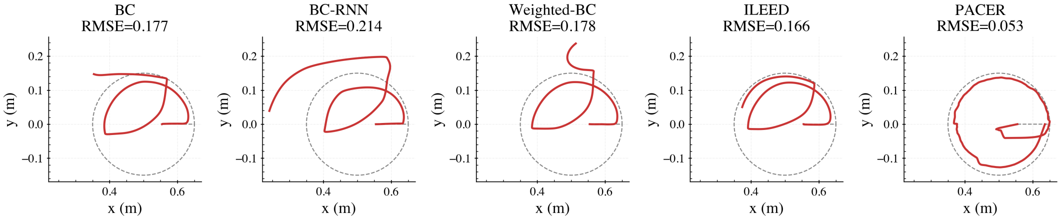

Figure 2: **Franka wiping rollouts.** Rollouts of learned policies on the circular wiping task. PACER adheres most closely to the reference path with smaller radial deviation and reduced phase drift, consistent with the quantitative gains reported in Table 2.

Method	RMSE ↓		MaxDev ↓	
BC	0.1733 ± 0.0166	[0.0188]	0.3239 ± 0.0297	[0.0336]
BC-RNN	0.2445 ± 0.0265	[0.0300]	0.4791 ± 0.0484	[0.0548]
Weighted-BC	0.1492 ± 0.0273	[0.0309]	0.2804 ± 0.0337	[0.0381]
ILEED	0.1909 ± 0.0180	[0.0204]	0.3590 ± 0.0492	[0.0557]
PACER	**0.0818** ± 0.0180	[0.0204]	**0.1356** ± 0.0264	[0.0299]

Table 2: **Franka wiping (MuJoCo).** Geometric metrics in meters (lower is better). Values are mean ± std with 95% CI in brackets, n=3 seeds.

Results: Franka wiping. PACER achieves the lowest trajectory errors (Table 2). Compared to the strongest baseline (Weighted-BC), PACER reduces RMSE by roughly 45% (from 0.1492m to 0.0818m) and more than halves MaxDev (from 0.2804m to 0.1356m), indicating tighter adherence to the target circle. Qualitative overlays in Figure 2 mirror these trends.

Results: Hopper-v4. PACER attains the highest returns in both evaluation conditions (Table 1). Relative to the best-performing BC variant (Weighted-BC), PACER improves mean return by over 50% in the clean setting (from 1671.9 to 2545.1) and shows a similar gain under induced test-time perturbations. The demo-weighted variant, Weighted-BC, narrows the gap but remains below PACER.

5 Conclusion

We studied imitation learning from small collections of demonstrations that contain short, unintended deviations. PACER addresses this setting by aligning trajectories in task progress and forming robust per-phase targets before policy learning. Concretely, PACER aggregates actions within phase bins using median statistics, down-weights outliers with Tukey biweights, and applies a leave-one-out consensus so that no demonstration can validate its own deviation. A single MLP is then trained with a weighted Huber objective on these repaired targets. The intuition is that progress-aligned consensus recovers the intended behavior at each stage of the task while suppressing localized errors that arise in individual rollouts.

Under the reported protocols on locomotion (Hopper-v4) and manipulation (Franka wiping), PACER achieved higher episodic returns and lower trajectory error than behavioral cloning variants (BC, Weighted-BC, BC-RNN) and ILEED. All methods used the same executed actions for supervision and comparable model/optimization settings; the principal difference is how supervision targets are constructed. In our experiments, ILEED, designed to model demonstrator-level expertise, was less effective when deviations were brief and occurred within trajectories. PACER's phase-wise consensus is directly aligned with this failure mode, which likely explains the observed gains.

Beyond empirical performance, PACER is simple to deploy: after computing progress and robust per-phase summaries, training reduces to standard behavioral cloning. The intermediate quantities (per-phase medians, scales, weights) provide interpretable diagnostics that expose where demonstrations agree or conflict, which can aid dataset inspection and future curation.

This work has limitations. PACER assumes a meaningful progress coordinate and benefits most when deviations are short relative to the trajectory; performance may degrade under long-horizon or globally biased corruptions, or when progress is poorly estimated. The method introduces a small set of robustness hyperparameters (number of bins, smoothing window, Tukey cutoff) that mediate bias-variance trade-offs. Promising directions include learning progress directly from observations in richer sensory settings, adapting the consensus step to capture uncertainty beyond medians, integrating PACER with sequence or diffusion policies, combining label repair with limited online interaction for iterative refinement, and expanding evaluation to broader real-world tasks and operators. These steps aim to retain PACER's simplicity while increasing its coverage beyond the localized-deviation regime evaluated here.

A Implementation Details

A.1 Baselines

BC uses the same MLP and optimization as PACER but trains on executed actions a with unit weights. **Weighted-BC** (Traj-BC) applies a per-*demo* scalar weight proportional to an empirical corruption mass (down-weighting more corrupted demonstrations). **BC-RNN** uses a GRU over short sequences with MSE on action prediction.

ILEED (continuous actions). We follow the continuous-action variant in Beliaev et al. [9]. The policy is a mixture of M diagonal Gaussians; a backbone network outputs $\{\pi(s), \mu(s), \sigma(s)\}$. A state- and demonstration-dependent expertise variable $\rho(s,i) = \sigma\big(\langle f_\phi(s), \omega_i\rangle\big)$ with $\rho \in [\rho_{\min}, 1]$ modulates the demonstrator likelihood by scaling the covariance: $\Sigma \mapsto \Sigma/\rho^2$. For a single Gaussian component, the log-likelihood is

$$\log \mathcal{N}\Big(a; \mu, \tfrac{\Sigma}{\rho^2}\Big) = -\tfrac{1}{2}\big[d \log(2\pi) + \log|\Sigma| - 2d \log \rho + \rho^2 (a-\mu)^\top \Sigma^{-1} (a-\mu)\big] \, .$$

and the mixture likelihood marginalizes across components using the softmax weights of $\alpha(s)$. The auxiliary latent-dynamics objective predicts z_{t+1} from (z_t, a_t) where $z = f_\phi(s)$, trained with a Smooth-ℓ_1 loss. Optimization proceeds in two stages: (i) warm-up of the GMM policy with $\rho \equiv 1$, and (ii) joint training of the policy, expertise, and auxiliary dynamics. Each demonstration is assigned a distinct identity i. At test time we use the mixture mean as the deterministic action.

B Hyperparameters

Category / Parameter	**Value**
BC / PACER / Weighted-BC / ILEED (Shared optimization)	
Network width (MLP hidden size)	128
Optimizer / LR	Adam / 1×10^{-3}
Batch size	8192
Epochs	240
Gradient clip (global norm)	1.0
BC-RNN	
GRU hidden size	128
Sequence length / stride	32 / 16
Batch size	64
Learning rate / Epochs	1×10^{-3} / 240
PACER	
Phase bins B	96
Tukey cutoff c	4.685
Min sample weight $w_{\min}$	0.02
LOO debias λ_{debias}	0.5
Magnitude blend η_0	0.5
EMA coefficient κ	0.0
ILEED	
Mixture components M	5
State embed dim k	16
Expertise floor $\rho_{\min}$	0.05
Aux dynamics weight λ_{aux}	1×10^{-2}
Warm-up: epochs / LR	80 / 1×10^{-3}
Joint: epochs / LR	160 / 6×10^{-4}

Table 3: **Hyperparameters.** Values used in all reported experiments unless otherwise noted.

C Pseudocode

Algorithm 1 PACER: Progress-Aligned Curation for Error-Resilient Imitation Learning

```
Require: Demos $\mathcal{D} = \{(x_{i,t}, a_{i,t})\}$; bins $B$; Tukey cutoff $c$; floor $w_{\min}$; hyperparams $\lambda_{\text{debias}}, \rho_0, \eta_0, \kappa$.
$\tau \leftarrow \textsc{PhaseEstimate}(\mathcal{D})$                                   ▷ state-only ranking → $[0,1]$
$\{I_b\}_{b=0}^{B-1} \leftarrow \textsc{PhaseBins}(\tau, B)$
$\mathcal{S} \leftarrow \textsc{RobustBinStats}(\mathcal{D}, \tau, B)$                  ▷ medians/magnitudes/tangents

for all $(i,t)$ with $(i,t) \in I_b$ do
    $w \leftarrow \textsc{LOOWeight}(i, t, b, \mathcal{S}, c, w_{\min})$
    $y^* \leftarrow \textsc{PseudoLabel}(i, t, b, \mathcal{S}, w, \lambda_{\text{debias}}, \rho_0, \eta_0, \kappa)$
    Accumulate $\mathcal{L} \mathrel{+}= w \cdot \mathrm{Huber}(f_\theta(x_{i,t}) - y^*)$
end for
$\theta^\star \leftarrow \arg\min_\theta \mathcal{L}$; return $f_{\theta^\star}$

function $\textsc{PhaseEstimate}(\mathcal{D})$
    Train scorer $g_\psi$ with pairwise ranking loss
        $\sum_i \sum_{t>t'} \log\big(1 + \exp\big(m - (g_\psi(x_{i,t}) - g_\psi(x_{i,t'}))\big)\big)$
    Normalize scores to $\tau \in [0,1]$; return $\tau$
end function

function $\textsc{RobustBinStats}(\mathcal{D}, \tau, B)$
    For each bin $b$:
        $\alpha_a[b] = \mathrm{median}\{a_{i,t}\}, \quad \beta_a[b] = \mathrm{median}\{\|a_{i,t}\|\}, \quad \alpha_s[b] = \mathrm{median}\{x_{i,t}\}$
    Tangents: $t_a[b] \leftarrow \Delta\alpha_a[b]$; $t_s[b] \leftarrow \Delta\alpha_s[b]$                ▷ finite-diff across bins
    return $\{\alpha_a, \beta_a, \alpha_s, t_a, t_s\}$
end function

function $\textsc{LOOWeight}(i, t, b, \mathcal{S}, c, w_{\min})$
    $\alpha_a^{(-i)}[b] \leftarrow$ median action in bin $b$ excluding demo $i$             ▷ leave-one-out
    $r^{(-i)} \leftarrow \|a_{i,t} - \alpha_a^{(-i)}[b]\|$
    $\mathrm{MAD}^{(-i)} \leftarrow 1.4826 \cdot \mathrm{median}\big|\, r^{(-i)} - \mathrm{median}(r^{(-i)})\,\big|$
    $z \leftarrow \dfrac{r^{(-i)}}{\mathrm{MAD}^{(-i)} + \varepsilon}; \quad w \leftarrow \max\big(\mathbf{1}[z \le c]\,(1 - (z/c)^2)^2,\ w_{\min}\big)$
    return $w$
end function

function $\textsc{PseudoLabel}(i, t, b, \mathcal{S}, w, \lambda_{\text{debias}}, \rho_0, \eta_0, \kappa)$
    Debias toward anchor: $\gamma = 1 - \lambda_{\text{debias}}(1 - w); \quad y^{(1)} = \gamma a_{i,t} + (1-\gamma)\alpha_a^{(-i)}[b]$
    Direction choice: $t_{\text{dir}} \in \{t_s[b], t_a[b]\}$ (prefer $t_s[b]$ if available)
    Sideways attenuation: project $y^{(1)}$ onto $t_{\text{dir}}$:
        $y_\parallel = (y^{(1)} \cdot \hat{t})\hat{t}, \quad y_\perp = y^{(1)} - y_\parallel, \quad y^{(2)} = y_\parallel + (1-\rho)\, y_\perp,\ \rho = \rho_0(1-w)\mathbf{1}\{t_s[b]\}$
    Speed blend: $\eta = \eta_0(1 - w)$; $s = (1-\eta)\|y^{(2)}\| + \eta\, \beta_a[b]$; $y^{(3)} = s\, \dfrac{y^{(2)}}{\|y^{(2)}\| + \varepsilon}$
    EMA smoothing: $y^* = (1-\kappa)\, y^{(3)} + \kappa\, y^*_{\text{prev}}$ with $\kappa \uparrow$ when $w, w_{\text{prev}}$ are low
    return $y^*$
end function
```

References

[1] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):328–373, 2013.

[2] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann. Probabilistic movement primitives. *Advances in neural information processing systems*, 26, 2013.

[3] M. Cuturi and M. Blondel. Soft-dtw: a differentiable loss function for time-series. In *International conference on machine learning*, pages 894–903. PMLR, 2017.

[4] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, and G. Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1134–1141, 2018. doi:10.1109/ICRA.2018.8462891.

[5] K. Shiarlis, M. Wulfmeier, S. Salter, S. Whiteson, and I. Posner. Taco: Learning task decomposition via temporal alignment for control, 2018. URL https://arxiv.org/abs/1803.01840.

[6] V. Tangkaratt, N. Charoenphakdee, and M. Sugiyama. Robust imitation learning from noisy demonstrations. *arXiv preprint arXiv:2010.10181*, 2020.

[7] L. Liu, Z. Tang, L. Li, and D. Luo. Robust imitation learning from corrupted demonstrations, 2022. URL https://arxiv.org/abs/2201.12594.

[8] Y.-H. Wu, N. Charoenphakdee, H. Bao, V. Tangkaratt, and M. Sugiyama. Imitation learning from imperfect demonstration, 2019. URL https://arxiv.org/abs/1901.09387.

[9] M. Beliaev, A. Shih, S. Ermon, D. Sadigh, and R. Pedarsani. Imitation learning by estimating expertise of demonstrators, 2022. URL https://arxiv.org/abs/2202.01288.

[10] V. Tangkaratt, B. Han, M. E. Khan, and M. Sugiyama. Vild: Variational imitation learning with diverse-quality demonstrations, 2019. URL https://arxiv.org/abs/1909.06769.

[11] P. J. Rousseeuw and C. Croux. Alternatives to the median absolute deviation. *Journal of the American Statistical Association*, 88(424):1273–1283, 1993.

[12] K. Kafadar. The efficiency of the biweight as a robust estimator of location. *Journal of Research of the National Bureau of Standards*, 88(2):105, 1983.

[13] P. J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73 – 101, 1964. doi:10.1214/aoms/1177703732. URL https://doi.org/10.1214/aoms/1177703732.