RELATIVE BELLMAN CONSTRAINTS FOR VALUE LEARNING IN RESIDUAL REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Residual reinforcement learning (RL) adapts pretrained robot policies through bounded action corrections, making it appealing for real-robot learning where exploration and intervention are costly. However, value learning in residual RL typically focuses on absolute returns, whereas adaptation with a frozen base policy depends on resolving the value variations induced by these corrections. This motivates making value differences an explicit focus of supervision over the replay distribution, rather than learning them only indirectly from pointwise value targets. In this paper, we propose **ReVRA** (Relative Value Learning for Residual Adaptation), a plug-and-play value-learning module that explicitly supervises critic-induced value differences. ReVRA incorporates relative Bellman constraints alongside standard TD learning, using paired replay transitions to supervise value differences within the existing critic. Across simulated locomotion, manipulation, and VLA-based adaptation, ReVRA consistently improves residual RL performance. On real-robot manipulation tasks, it achieves higher task success with fewer interventions under the same limited interaction budget. The project code is available at https://anonymous.4open.science/r/ReVRA-DF70.

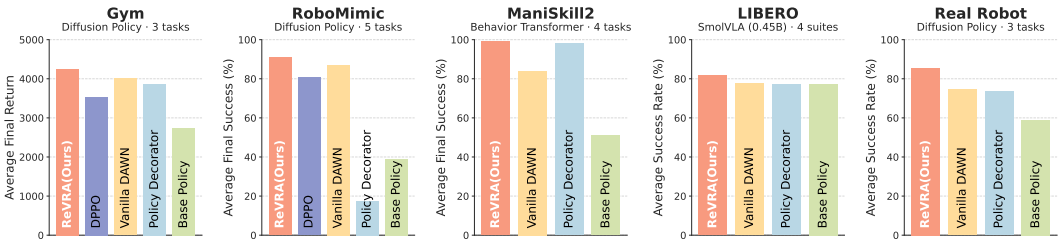


Figure 1: ReVRA delivers consistently strong policy improvement across diverse benchmarks and policy backbones. Bars show final performance averaged across tasks and seeds; higher is better.

1 INTRODUCTION

Robot policies trained on large-scale demonstration datasets have achieved substantial progress in recent years. Representative approaches range from Diffusion Policy (Chi et al., 2025) and Behavior Transformer (Shafiullah et al., 2022) to vision-language-action models (VLAs), including OpenVLA (Kim et al., 2024; 2025) and the π family (Black et al., 2024; 2025; Intelligence et al., 2025; 2026). By learning from diverse expert demonstrations, these policies acquire expressive behavioral priors and can perform complex manipulation tasks without costly online exploration. However, their performance remains closely tied to the state and action distributions covered by the training data.

Offline demonstrations cover only a limited range of conditions encountered during deployment. Changes in environments, object configurations, or contact dynamics can push the policy beyond its training distribution, with errors compounding over long-horizon execution (Ross et al., 2011). Scaling demonstrations alone cannot capture every failure case; moreover, demonstrations mainly contain successful executions and provide limited supervision for recovery states (Laskey et al., 2017). Consequently, an imitation policy may perform well under familiar conditions yet struggle to recover after deviating from demonstrated trajectories. These limitations motivate online interaction, through which a robot can encounter errors and learn corrective behaviors. This is challenging in

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.

Table 1: Summary of residual RL methods with different base policies and value-learning designs.

Research direction	Base-policy types	Algorithms	Value-related mechanism
Controller-based residual RL	Hand-crafted controllers, MPC and DMP	RPL (Silver et al., 2018), Residual RL (Johannink et al., 2019), rLfD (Davchev et al., 2022)	Inherited RL objective
Policy / exploration design	BC and generative policies (ACT, BeT and diffusion policy)	ResiP (Ankile et al., 2025b), Policy Decorator (Yuan et al., 2025), ResFiT (Ankile et al., 2025a), RFS (Su et al., 2026b)	No relative-value objective
Critic design	Diffusion Policy and BeT	DAWN (Ma et al., 2026)	Warmup and LayerNorm; Absolute value learning
Relative critic supervision	Diffusion Policy, BeT and VLA	**ReVRA(Ours)**	Explicit relative-value learning

the real world, where interaction is time-consuming, failures may require human intervention and resets, and unconstrained fine-tuning can compromise behaviors that were already reliable.

How can limited online interaction improve policy performance while preserving the capabilities acquired from demonstrations?

Residual reinforcement learning (RL) (Silver et al., 2018; Johannink et al., 2019; Ankile et al., 2025b;a; Yuan et al., 2025; Su et al., 2026a; Ma et al., 2026) provides a natural framework for adaptation. Instead of finetuning the pretrained policy, it freezes the base policy and learns a bounded residual correction. The executed action is given by $a_t = \pi_{\text{base}}(s_t) + \lambda\, \pi_{\text{res}}(s_t)$, where π_{base} denotes the pretrained policy, π_{res} is the residual policy, and λ controls the magnitude of the correction. By constraining updates to bounded adjustments, residual RL preserves the base policy as a behavioral prior and concentrates exploration around its competent actions, thereby avoiding exploration from scratch and reducing both unnecessary real-world interactions and disruptive behaviors.

However, the bounded residual formulation introduces a structural asymmetry for value learning. Given $a = a_{\text{base}} + \lambda a_{\text{res}}$ with a small residual scale λ, the base action dominates the critic input, whereas the learnable residual contributes only a small perturbation. DAWN (Ma et al., 2026) identifies this phenomenon as the *structural scale mismatch* and addresses it by applying layer normalization to the critic. By normalizing the critic's hidden activations, layer normalization stabilizes feature scales. Nevertheless, greater sensitivity does not necessarily imply more accurate value comparisons. A critic may respond strongly to a small action change without correctly predicting whether that change improves the expected return.

This distinction motivates reconsidering *how critic supervision supports residual policy improvement*. Residual actor updates depend on action-value variations, yet standard TD regression constrains differences between critic predictions only indirectly through pointwise targets. With limited interaction data and function approximation, small errors in individual predictions can still obscure the value differences relevant to policy improvement. Building on this motivation, we propose **ReVRA** (Relative Value Learning for Residual Adaptation), a plug-and-play module that complements standard TD learning with relative Bellman constraints constructed from paired replay transitions. ReVRA operates on the existing critic and replay data without introducing a separate pairwise network. We further characterize its relative Bellman formulation under fixed-policy evaluation, analyzing its relationship to standard Bellman updates and its contraction and fixed-point properties.

Empirically, we evaluate ReVRA across simulated **locomotion**, **manipulation**, **VLA-based adaptation**, and **real-robot** manipulation tasks. Our evaluation spans Diffusion Policy, Behavior Transformer, and VLA as base policies, together with diverse demonstration sources, observation modalities and network architectures. ReVRA consistently improves residual RL performance across these settings. Crucially, on real robots, it achieves higher task success with fewer interventions under a limited interaction budget, demonstrating the practical value of improved critic supervision.

2 Related Work

Residual RL. Residual RL improves a fixed base policy through additive action corrections, preserving prior competence while reducing the ineffective exploration space. Early methods combined residual policies with hand-crafted controllers, model-predictive controllers (MPC), or dynamic movement primitives (DMP) to compensate for model mismatch and contact uncertainty

(Silver et al., 2018; Johannink et al., 2019; Davchev et al., 2022). With the emergence of expressive behavior-cloning (BC) policies, recent work has primarily focused on stable and expressive policy refinement. ResiP (Ankile et al., 2025b), Policy Decorator (Yuan et al., 2025), and ResFiT Ankile et al. (2025a) improve closed-loop correction, exploration stability, or sample efficiency. RFS (Su et al., 2026b) further enlarge the adaptation interface through latent steering. Despite their different parameterizations, these methods mainly innovate on the policy side while retaining the absolute value-learning objective of the underlying RL algorithm.

Recent studies have shifted attention from policy design to value learning in residual RL. DAWN (Ma et al., 2026) stabilizes early critic learning and improves its sensitivity to residual actions through base-policy data and critic normalization, whereas Foresight Residual RL (Liu et al., 2026) uses downstream values to guide long-horizon skill transitions. Nevertheless, these methods still rely on absolute value estimation or value-based reward shaping rather than directly supervising relative values between residual actions. In contrast, ReVRA adds relative Bellman constraints over paired replay transitions to standard TD learning, directly supervising critic-predicted value differences while retaining pointwise value calibration. Detailed comparison is summarized in Table 1.

Fine-tuning Foundation Policies via RL. Post-training fine-tuning has become a common paradigm for adapting pretrained robot policies. Diffusion policies (Chi et al., 2025), behavior transformers (Shafiullah et al., 2022), and VLA (Zitkovich et al., 2023) models provide expressive control priors, but their performance often degrades under distribution shifts between offline training data and deployment environments. RL fine-tuning addresses this limitation by evaluating policy-generated trajectories with task rewards, providing feedback beyond the static demonstration distribution and directly optimizing deployment-relevant objectives (Lai et al., 2025).

Existing approaches mainly differ in whether they update the pretrained policy. DPPO (Ren et al., 2025) enables policy-gradient optimization of diffusion policies, PA-RL (Mark et al., 2025) distills critic-optimized actions into the policy, and iRe-VLA (Guo et al., 2025) alternates RL with supervised updates for stable VLA adaptation. While direct updates offer greater adaptation capacity, they incur substantial optimization costs and may disrupt pretrained behaviors. Alternatively, frozen-policy methods learn action residuals (Yuan et al., 2025; Ankile et al., 2025a), latent-space steering (Wagenmaker et al., 2025), or their combination. These methods preserve the behavioral prior but require the critic to resolve subtle value differences among corrective actions, motivating a closer examination of value learning in residual RL.

3 PRELIMINARY

Problem Formulation. We consider a standard Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma, \rho_0)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}(s' \mid s, a)$ is the transition function, $r(s, a)$ is the reward function, $\gamma \in [0, 1)$ is the discount factor, and ρ_0 is the initial-state distribution. At timestep t, an agent selects an action $a_t \sim \pi(\cdot \mid s_t)$ and transitions to $s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t)$. The objective is to find a policy that maximizes the expected discounted return:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\rho_0, \pi, \mathcal{P}} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]. \tag{1}$$

The action-value function $Q^\pi(s, a)$ denotes the expected discounted return obtained by taking action a in state s and following policy π thereafter: $Q^\pi(s, a) = \mathbb{E}_{\pi, \mathcal{P}} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \,|\, s_0 = s,\ a_0 = a \right]$.

Residual RL. Residual RL adapts a pretrained policy through additive action corrections (Silver et al., 2018; Johannink et al., 2019). The base policy π_{base} is typically learned from demonstrations and remains fixed during online training. A learnable residual policy π_{res} produces a bounded correction, yielding the executed action

$$a = a_{\text{base}} + \lambda a_{\text{res}} = \underbrace{\pi_{\text{base}}(s)}_{\text{Frozen}} + \lambda \cdot \underbrace{\pi_{\text{res}}(s)}_{\text{Learnable}}, \tag{2}$$

where $\lambda > 0$ controls the magnitude of the correction. The residual policy is typically trained using standard off-policy actor–critic algorithms, such as DDPG (Lillicrap et al., 2015) or SAC (Haarnoja et al., 2018). The critic receives the state and the combined action as input, $Q_\theta(s, a_{\text{base}} + \lambda a_{\text{res}})$.

When λ is small, residual corrections can appear as small perturbations to the base action. DAWN (Ma et al., 2026) identifies this structural scale mismatch as a challenge for value learning and applies critic normalization to improve sensitivity to residual variations. Building on this approach, we investigate a complementary relative value objective for critic learning.

4 METHODOLOGY

This section presents ReVRA, a value-learning module for residual RL that augments TD learning with relative value supervision. We first contrast absolute and relative supervision, then characterize critic-induced value differences through a relative Bellman operator. Finally, we integrate replay-based relative targets into residual SAC without changing the actor update.

4.1 ABSOLUTE AND RELATIVE VALUE LEARNING IN RESIDUAL RL

In residual RL, the critic evaluates the combined action

$$a_\phi(s) = \pi_{\text{base}}(s) + \lambda a_{\text{res},\phi}(s), \tag{3}$$

where the base policy is frozen and only the residual policy is optimized. The value-dependent component of the actor gradient is therefore

$$\nabla_\phi Q_\theta(s, a_\phi(s)) = \lambda \big[\nabla_\phi a_{\text{res},\phi}(s)\big]^\top \nabla_a Q_\theta(s,a)|_{a=a_\phi(s)}\,. \tag{4}$$

Thus, policy improvement is determined by how the critic varies over the local set of combined actions induced by the residual policy. In particular, the actor needs the critic to rank alternative residual corrections correctly, not merely to fit their absolute-return targets.

A small residual scale λ makes residual-induced action variations difficult for the critic to represent. DAWN (Ma et al., 2026) attributes this difficulty to the *structural scale mismatch* and applies critic normalization to improve sensitivity to these variations. Greater sensitivity, however, does not necessarily ensure correct value comparisons. DAWN still relies on pointwise TD regression, which provides an absolute target for each critic input but no explicit target for the difference between two predictions. Although an exact absolute critic would recover such differences, finite prediction errors may distort their ordering.

To expose this distinction, consider two critic inputs x_i and x_j, with predictions

$$Q_\theta(x_i) = Q^\pi(x_i) + \epsilon_i, \qquad Q_\theta(x_j) = Q^\pi(x_j) + \epsilon_j, \tag{5}$$

where ϵ_i and ϵ_j are their estimation errors. Their predicted value difference satisfies

$$\underbrace{Q_\theta(x_i) - Q_\theta(x_j)}_{\widehat{\Delta Q}_{ij}} = \underbrace{Q^\pi(x_i) - Q^\pi(x_j)}_{\Delta Q^\pi_{ij}} + \underbrace{\epsilon_i - \epsilon_j}_{\text{differential error}}\,. \tag{6}$$

Thus, the predicted ordering is determined by the true value gap perturbed by the differential error. A sufficient condition for preserving any nonzero true ordering is

$$|\epsilon_i - \epsilon_j| < |\Delta Q^\pi_{ij}| \quad \Longrightarrow \quad \operatorname{sign}(\widehat{\Delta Q}_{ij}) = \operatorname{sign}(\Delta Q^\pi_{ij}). \tag{7}$$

Consequently, pairs with smaller true value gaps are more susceptible to ordering errors. Such comparisons are particularly relevant in residual RL, where bounded corrections produce a local set of actions around the frozen base policy.

We next compare how pointwise and relative supervision treat this error. Considering their idealized regression errors against the true policy values gives

$$\begin{aligned} \mathcal{L}^{ij}_{\text{abs}} &= \sum_{k\in\{i,j\}} \big(Q_\theta(x_k) - Q^\pi(x_k)\big)^2 = \epsilon_i^2 + \epsilon_j^2, \qquad \mathcal{L}^{ij}_{\text{rel}} = \big(\widehat{\Delta Q}_{ij} - \Delta Q^\pi_{ij}\big)^2 = (\epsilon_i - \epsilon_j)^2, \\ &= \frac{1}{2}(\epsilon_i + \epsilon_j)^2 + \frac{1}{2}(\epsilon_i - \epsilon_j)^2. \end{aligned} \tag{8}$$

The decomposition of $\mathcal{L}^{ij}_{\text{abs}}$ is an algebraic reparameterization, rather than an explicit pairwise supervision signal. Absolute regression reduces the differential error indirectly by fitting both pointwise

targets, while relative regression uses the value difference itself as the prediction target and directly couples the two outputs. Accordingly, adding a relative constraint increases the emphasis on the differential component without removing absolute value calibration.

This distinction is central to residual policy improvement: local action-value variations guide the residual actor, while differential errors can obscure their ordering. We therefore complement standard TD learning with relative Bellman constraints, making comparison-relevant errors an explicit focus of critic supervision.

4.2 Bellman Characterization of Value Differences

We now formalize critic-induced value differences under Bellman evaluation. We define a relative Bellman operator, relate it to standard Bellman updates, and characterize its contraction, fixed point, and interaction with absolute value calibration.

Relative Bellman Operator. For two critic inputs x_i and x_j, define

$$\Delta^\pi(x_i, x_j) \coloneqq Q^\pi(x_i) - Q^\pi(x_j), \qquad \Delta_\theta(x_i, x_j) \coloneqq Q_\theta(x_i) - Q_\theta(x_j). \tag{9}$$

Here, π denotes the current executed-action policy, Δ^π is its true value-difference function, and Δ_θ is the corresponding function induced by the learned critic. Constructing Δ_θ from the existing scalar critic keeps absolute and relative predictions within the same value representation and requires no separately parameterized pairwise critic.

The Bellman relation for Δ^π follows by subtracting the Bellman equations associated with x_i and x_j. For $x_k = (s_k, a_k)$, let $s'_k \sim \mathcal{P}(\cdot \mid s_k, a_k)$, $a'_k \sim \pi(\cdot \mid s'_k)$, and $x'_k = (s'_k, a'_k)$ for $k \in \{i, j\}$

Definition 4.1 (Relative Bellman Operator)**.** For any bounded pairwise function Δ, the relative Bellman operator $\mathcal{T}^\pi_{\mathrm{rel}}$ is defined as

$$\begin{aligned}\left(\mathcal{T}^\pi_{\mathrm{rel}}\Delta\right)(x_i, x_j) \coloneqq\ & r(s_i, a_i) - r(s_j, a_j) \\ & + \gamma \mathbb{E}\left[\Delta(x'_i, x'_j) \,\middle|\, x_i, x_j\right].\end{aligned} \tag{10}$$

The relative operator is derived from standard Bellman evaluation rather than from an independently specified comparison target. The following result makes this relationship explicit.

Proposition 4.2 (Consistency with Standard Bellman Updates)**.** *For any bounded action-value function Q, let $\Delta_Q(x_i, x_j) \coloneqq Q(x_i) - Q(x_j)$ denote its induced value difference. Then*

$$\left(\mathcal{T}^\pi_{\mathrm{rel}}\Delta_Q\right)(x_i, x_j) = \left(\mathcal{T}^\pi Q\right)(x_i) - \left(\mathcal{T}^\pi Q\right)(x_j) = \Delta_{\mathcal{T}^\pi Q}(x_i, x_j). \tag{11}$$

Proof. See Appendix A.1 for detailed proof. □

Since Q^π is a fixed point of $\mathcal{T}^\pi$, Proposition 4.2 implies that Δ^π is a fixed point of $\mathcal{T}^\pi_{\mathrm{rel}}$. The following theorem establishes its uniqueness and the convergence of relative Bellman evaluation.

Theorem 4.3 (Contraction and Unique Fixed Point)**.** *For a fixed policy π, bounded rewards, and discount factor $\gamma \in [0, 1)$, the relative Bellman operator $\mathcal{T}^\pi_{\mathrm{rel}}$ is a γ-contraction on the space of bounded pairwise functions under the supremum norm:*

$$\left\|\mathcal{T}^\pi_{\mathrm{rel}}\Delta_1 - \mathcal{T}^\pi_{\mathrm{rel}}\Delta_2\right\|_\infty \le \gamma \left\|\Delta_1 - \Delta_2\right\|_\infty. \tag{12}$$

Consequently, $\mathcal{T}^\pi_{\mathrm{rel}}$ admits the unique fixed point Δ^π, where $\Delta^\pi(x_i, x_j) = Q^\pi(x_i) - Q^\pi(x_j)$. Moreover, repeated applications of $\mathcal{T}^\pi_{\mathrm{rel}}$ converge to Δ^π from any bounded initialization.

Proof. See Appendix A.2 for detailed proof. □

Remark 4.4 (Global Offset and Absolute Calibration)**.** Suppose that the difference function Δ_Q induced by a critic Q is a fixed point of $\mathcal{T}^{\pi}_{\mathrm{rel}}$. By Theorem 4.3, $\Delta_Q = \Delta^{\pi}$, and hence $Q(x_i)-Q(x_j) = Q^{\pi}(x_i)-Q^{\pi}(x_j)$ for every input pair. It follows that $Q(x)-Q^{\pi}(x)$ is constant across inputs, so that $Q = Q^{\pi} + c$ for some global constant c. This offset affects neither action comparisons nor, when the critic is differentiable with respect to actions, policy gradients. If Q also satisfies absolute Bellman consistency, then $Q^{\pi} + c = \mathcal{T}^{\pi}(Q^{\pi} + c) = Q^{\pi} + \gamma c$, implying $c = 0$ because $\gamma < 1$. Relative consistency therefore identifies the value-difference structure, while absolute consistency calibrates the global value level.

These analysis establish the theoretical basis of our value-difference learning formulation. Proposition 4.2 shows that the relative operator is consistent with standard Bellman evaluation. Theorem 4.3 shows that the relative Bellman operator is a γ-contraction with the unique fixed point $\Delta^{\pi}(x_i, x_j) = Q^{\pi}(x_i) - Q^{\pi}(x_j)$. Remark 4.4 further shows that relative consistency recovers the pairwise differences required for action comparison, while leaving a policy-invariant global offset unspecified. Absolute Bellman consistency fixes this constant to zero. Thus, the relative constraint preserves the solution of standard policy evaluation while directly supervising value differences.

4.3 Value-Difference Learning for Residual RL

We instantiate ReVRA within residual SAC (Haarnoja et al., 2018). The base policy remains frozen, while the residual actor and action-value critics are learned through online interaction. ReVRA modifies critic supervision by incorporating relative Bellman consistency, while leaving the residual actor update unchanged.

Relative Bellman Targets. Let $\mathcal{B} = \{\tau_i\}_{i=1}^{B}$ be a minibatch sampled from the replay buffer, where $\tau_i = (x_i, r_i, d_i, s_i')$ and d_i indicates whether the transition terminates. Although comparing alternative residual corrections from the same state would be most directly aligned with residual policy improvement, such counterfactual transitions are generally unavailable in an off-policy replay buffer. Obtaining them would require branching rollouts from an identical state or access to an accurate dynamics model.

Importantly, relative Bellman consistency is defined over arbitrary critic-input pairs and therefore does not require such counterfactual samples. We construct pairs within each randomly sampled minibatch by pairing τ_i with $\tau_{\sigma(i)}$, where σ is a cyclic shift of the batch indices. This pairs every sample with a different batch element and provides an $\mathcal{O}(B)$ stochastic estimate of the pairwise objective over the replay distribution.

For each pair (i, j), we sample $a'_{\mathrm{res},k} \sim \pi_{\mathrm{res},\phi}(\cdot \mid s'_k)$ and define

$$x'_k = \left(s'_k, \pi_{\mathrm{base}}(s'_k) + \lambda a'_{\mathrm{res},k}\right), \qquad \ell'_k = \log \pi_{\mathrm{res},\phi}(a'_{\mathrm{res},k} \mid s'_k), \quad k \in \{i, j\}. \tag{13}$$

Let $\bar{Q}$ denote the stop-gradient target-critic estimate and define $\bar{\Delta}(x_i, x_j) \coloneqq \bar{Q}(x_i) - \bar{Q}(x_j)$. The sampled relative Bellman target is

$$y^{\Delta}_{ij} = r_i - r_j + \gamma \Psi_{ij}, \qquad \Psi_{ij} = \begin{cases} \bar{\Delta}(x'_i, x'_j) - \alpha(\ell'_i - \ell'_j), & d_i = 0,\ d_j = 0, \\ \bar{\Delta}(x'_i, x_j) + r_j - \alpha \ell'_i, & d_i = 0,\ d_j = 1, \\ \bar{\Delta}(x_i, x'_j) - r_i + \alpha \ell'_j, & d_i = 1,\ d_j = 0, \\ 0, & d_i = 1,\ d_j = 1, \end{cases} \tag{14}$$

where α is the SAC entropy temperature. The first case follows by subtracting two soft Bellman backups. When exactly one transition terminates, independently masking its bootstrap would leave a one-sided absolute continuation value, which is not invariant to a global shift of the critic. Following Relative Value Learning (Höftmann et al., 2026), the mixed terminal cases instead use the terminal identity $Q(x_k) = r_k$ to express the target entirely through observed rewards and critic differences. When both transitions terminate, no bootstrap is required.

Critic and Actor Objectives. Let y_i^{Q} denote the standard SAC target. We train the critic using the coupled objective

$$\mathcal{L}_{\mathrm{critic}}(\theta) = \mathbb{E}_i\left[(Q_\theta(x_i) - y_i^{Q})^2\right] + \beta \mathbb{E}_{(i,j)}\left[(\Delta_\theta(x_i, x_j) - y^{\Delta}_{ij})^2\right], \tag{15}$$

where $\beta \geq 0$ controls the strength of relative Bellman consistency. Because $\Delta_\theta(x_i, x_j) = Q_\theta(x_i) - Q_\theta(x_j)$, both terms optimize the same critic without introducing a separate pairwise network.

Relative learning modifies only the critic objective. The residual actor retains the standard SAC objective

$$\mathcal{L}_{\text{actor}}(\phi) = \mathbb{E}_{s,a_{\text{res}}}\left[\alpha \log \pi_{\text{res},\phi}(a_{\text{res}} \mid s) - Q_\theta(s,a)\right], \qquad a = \pi_{\text{base}}(s) + \lambda a_{\text{res}}. \tag{16}$$

5 Experiments

5.1 Experimental Setup

We evaluate ReVRA across simulated and real-robot tasks spanning different task types, base-policies, network architectures, demonstration sources, and observation modalities.

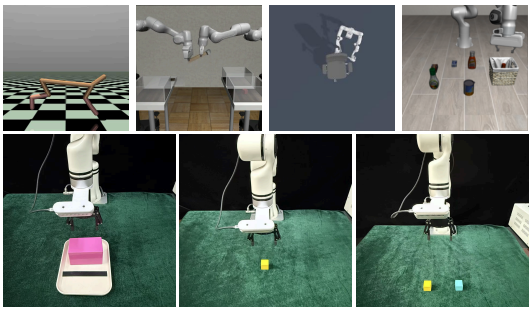


Figure 2: Visualizations of representative tasks. **Top**: Simulation benchmarks including `Gym`, `Robomimic`, `ManiSkill2` and `LIBERO`. **Bottom**: Real-robot tasks including `Push Cube`, `Grasp Cube`, and `Stack Cube`.

Base Policies. We use Diffusion Policy (Chi et al., 2025) and Behavior Transformer (Shafiullah et al., 2022) as frozen base policies for online residual adaptation. We additionally evaluate SmolVLA (0.45B) (Shukor et al., 2025) to assess applicability to VLA policy.

Task Types. Our evaluation covers locomotion in `Gym` (Brockman et al., 2016); single-arm and bimanual manipulation in `Robomimic` (Mandlekar et al., 2022); tabletop manipulation, articulated-object manipulation, mobile manipulation, and precise insertion in `ManiSkill2` (Gu et al., 2023); and language-conditioned manipulation across spatial, object, and goal variations, as well as long-horizon tasks, in `LIBERO` (Liu et al., 2023).

Demonstration Sources. Our base policies are trained on demonstrations collected through human teleoperation, RL, task and motion planning, and MPC, covering diverse levels of demonstrator expertise and data-generation procedures. More details are provided in Appendix B.2.

Real Robot Settings. We evaluate ReVRA on three sparse-reward tasks using a Realman RM75-B arm with two RGB cameras: `Push Cube`, `Grasp Cube`, and `Stack Cube` (Figure 2). A Diffusion Policy trained on real-world demonstrations serves as the frozen base policy, while the residual policy is learned through real-world interaction following HIL-SERL (Luo et al., 2025), without simulation pretraining. Further real-robot details are provided in Appendix B.3.

5.2 Baselines

We compare ReVRA against three representative baselines. **Policy Decorator** (Yuan et al., 2025) learns bounded residual actions with progressive exploration for stable online adaptation. **Vanilla DAWN** adds LayerNorm to critics trained with absolute TD learning, isolating critic normalization within the same residual RL implementation. For Diffusion Policy experiments, we additionally compare with **DPPO** (Ren et al., 2025), which directly fine-tunes the pretrained diffusion policy rather than learning a residual correction. Since Policy Decorator has been shown to substantially outperform curriculum-based JSRL (Uchendu et al., 2023), offline-to-online Cal-QL (Nakamoto et al., 2023), and optimal-transport based FISH (Haldar et al., 2023) on prior benchmarks, we focus on it as the primary residual RL baseline, alongside Vanilla DAWN and DPPO.

For Diffusion Policy, the residual methods use the released DAWN critic backbone, with a 512-dimensional input projection and two residual blocks; the normalized variants include LayerNorm. Behavior Transformer experiments use Policy Decorator's three-layer, 256-unit critic, with LayerNorm before each ReLU in the normalized variants. Within each base-policy setting, ReVRA and Vanilla DAWN share the critic backbone and training protocol, while ReVRA additionally imposes relative Bellman constraints. Architecture details are provided in Appendix B.4.

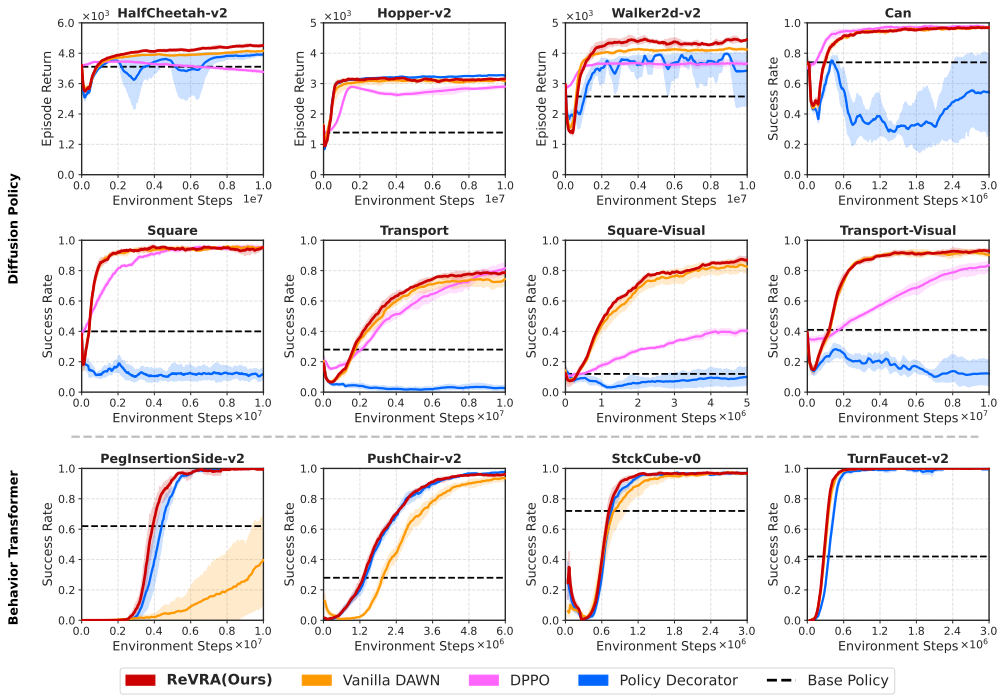


Figure 3: Learning curves across locomotion and manipulation benchmarks. The top two rows use Diffusion Policy as the base policy and the bottom row uses Behavior Transformer. Curves and shaded regions show the mean and 95% confidence intervals (CIs) over five independent runs.

Table 2: Success rates (%) on the LIBERO benchmark. All methods use SmolVLA (0.45B) as the base policy, fine-tuned for 100k gradient steps. Residual RL methods are further trained for 100k environment steps. The best result in each column is highlighted in **bold**.

Policy / Method	Spatial	Object	Goal	Long	Avg.
SmolVLA (0.45B)	88	94	82	45	77.25
+ Policy Decorator	82	95	81	50	77.00
+ Vanilla DAWN	84	94	79	53	77.50
+ ReVRA (Ours)	**89**	**97**	**86**	**55**	**81.75**

5.3 Experimental Results and Analysis

Diffusion Policy and Behavior Transformer Results. Figure 3 shows that ReVRA performs strongly with both the DAWN critic for Diffusion Policy and the Policy Decorator critic for Behavior Transformer. With Diffusion Policy, ReVRA leads on `HalfCheetah` and `Walker2d`, remains competitive on `Hopper` and state-based Robomimic tasks, and ranks among the best on the two visual tasks. Its clearest gains over the architecture-matched Vanilla DAWN baseline occur on `HalfCheetah`, `Walker2d`, and `Square-Visual`. DPPO is competitive on `Can`, `Square`, and `Transport`, but trails the base policy on `HalfCheetah` and ReVRA on visual tasks.

With Behavior Transformer, ReVRA reaches near-perfect success on all four ManiSkill2 tasks, learning faster than Vanilla DAWN on `PegInsertionSide`, `PushChair`, and `StackCube`, while remaining competitive with Policy Decorator. LayerNorm can make the critic more sensitive to small residual-induced action changes, but a stronger response does not ensure that the predicted value change has the correct direction or magnitude. Overall, ReVRA remains effective across the evaluated base policies, observation settings, and critic architectures, whereas the normalization-only baseline yields less consistent gains.

VLA-based Adaptation. To evaluate ReVRA with a VLA as the base policy, we use SmolVLA (0.45B) (Shukor et al., 2025) on the four LIBERO suites. As shown in Table 2, ReVRA achieves the

Table 3: Success and intervention rates (%) on three real-robot manipulation tasks. All methods use the same pretrained base policy and training budget (5k environment steps, ≈30 minutes of real-world training). The best results are shown in **bold**, with our method highlighted by shading.

Method	Push Cube		Grasp Cube		Stack Cube	
	Success ↑	Intervention ↓	Success ↑	Intervention ↓	Success ↑	Intervention ↓
Diffusion Policy	90.0	–	56.7	–	30.0	–
+ Policy Decorator	**100.0**	7.9	73.3	15.2	46.7	21.9
+ Vanilla DAWN	**100.0**	6.4	70.0	11.3	53.3	20.3
+ ReVRA (Ours)	**100.0**	**5.3**	**86.7**	**8.5**	**70.0**	**17.2**

highest success rate in every suite and improves the average from 77.25% to 81.75%, outperforming SmolVLA, Policy Decorator, and Vanilla DAWN by 4.50, 4.75, and 4.25 percentage points, respectively. The largest gain occurs on `LIBERO-Long`, where success increases from 45% to 55%. Unlike the compared residual methods, which reduce base-policy performance on some suites, ReVRA improves it across all four, demonstrating its applicability to VLA-based residual adaptation.

Real Robot Results. Table 3 shows that ReVRA improves task success while reducing intervention rates under the same training budget. On `Push Cube`, all residual methods reach 100% success, but ReVRA requires the fewest interventions. Its advantage becomes more pronounced on the harder tasks: compared with the strongest baseline, ReVRA improves success by 13.4% on `Grasp Cube` and 16.7% on `Stack Cube`, while also achieving lower intervention rates. These results indicate that ReVRA learns more effective residual corrections from limited real-world interaction, extending its simulated improvements to real robot adaptation.

Effect of Absolute and Relative Supervision. Figure 4 reports ablations of the absolute and relative critic objectives. On `Walker2d`, relative-only learning remains competitive with the coupled variants, showing that useful policy updates can be obtained without explicit absolute-value calibration. On the more challenging `PegInsertionSide`, however, both absolute-only and relative-only learning converge more slowly and exhibit greater variability, whereas combining the two objectives reaches near-perfect success substantially earlier. This contrast supports their complementary roles: relative supervision emphasizes value differences, while absolute TD learning anchors the critic's value scale during finite-sample training. See Appendix C.1 for more ablations.

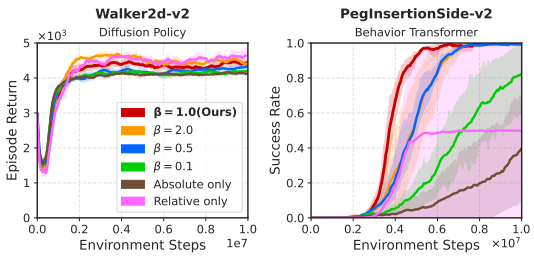


Figure 4: Effects of absolute and relative supervision. Learning curves with different relative-loss weights and individual critic objectives. "Absolute only" denotes Vanilla DAWN ($\beta = 0$), while "Relative only" removes $\mathcal{L}_{\text{abs}}$.

The results are also reasonably robust to the relative-loss weight. On `Walker2d`, $\beta \in \{0.5, 1, 2\}$ yields comparable performance. On `PegInsertionSide`, $\beta = 0.1$ provides insufficient relative supervision, while $\beta \in \{0.5, 1.0, 2.0\}$ eventually reaches high success. Among them, $\beta = 1$ converges fastest and provides the best overall balance across the two tasks.

6 Conclusion

In this paper, we presented ReVRA, a plug-and-play value-learning module for residual RL that makes critic-induced value differences an explicit focus of supervision. ReVRA complements standard TD learning with relative Bellman constraints from paired replay transitions, without introducing a separate network or modifying the residual actor. We further characterized its relationship between relative and standard Bellman evaluation. Across simulated locomotion, manipulation, VLA-based adaptation, and real-robot tasks, ReVRA consistently improved residual RL with Diffusion Policy, Behavior Transformer, and SmolVLA as base policies. These results demonstrate the practical value of value-difference supervision for residual adaptation.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on the presentation of our methodology and mathematical claims. Additionally, we used them to draft and edit manuscript text, create or revise scientific figures, suggest experimental parameters, and write or edit software code. The authors reviewed all AI-assisted work, checked technical claims and figures against the underlying derivations and experimental records, and tested AI-assisted code before use. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Lars Ankile, Zhenyu Jiang, Rocky Duan, Guanya Shi, Pieter Abbeel, and Anusha Nagabandi. Residual off-policy rl for finetuning behavior cloning policies. *arXiv preprint arXiv:2509.19301*, 2025a.

Lars Ankile, Anthony Simeonov, Idan Shenfeld, Marcel Torne, and Pulkit Agrawal. From imitation to refinement - residual rl for precise assembly. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 01–08, 2025b. doi: 10.1109/ICRA55743.2025.11127442.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/black25a.html.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025. doi: 10.1177/02783649241273668. URL https://doi.org/10.1177/02783649241273668.

Todor Davchev, Kevin Sebastian Luck, Michael Burke, Franziska Meier, Stefan Schaal, and Subramanian Ramamoorthy. Residual learning from demonstration: Adapting dmps for contact-rich manipulation. *IEEE Robotics and Automation Letters*, 7(2):4488–4495, 2022. doi: 10.1109/LRA.2022.3150024.

Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. Maniskill2: A unified benchmark for generalizable manipulation skills. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=b_CQDy9vrD1.

Yanjiang Guo, Jianke Zhang, Xiaoyu Chen, Xiang Ji, Yen-Jen Wang, Yucheng Hu, and Jianyu Chen. Improving vision-language-action model with online reinforcement learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15665–15672, 2025. doi: 10.1109/ICRA55743.2025.11127299.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*,

volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

Siddhant Haldar, Jyothish Pari, Anant Rai, and Lerrel Pinto. Teach a robot to fish: Versatile imitation from one minute of demonstrations. *arXiv preprint arXiv:2303.01497*, 2023.

Marc Höftmann, Jan Robine, and Stefan Harmeling. Relative value learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ulTRUwrzt9.

Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, et al. $\pi^*_{0.6}$: A vla that learns from experience. *arXiv preprint arXiv:2511.14759*, 2025.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, et al. $\pi_{0.7}$: A steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026.

Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 6023–6029, 2019. doi: 10.1109/ICRA.2019.8794127.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=ZMnD6QZAE6.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Hanyu Lai, Xiao Liu, Junjie Gao, Jiale Cheng, Zehan Qi, Yifan Xu, Shuntian Yao, Dan Zhang, Jinhua Du, Zhenyu Hou, Xin Lv, Minlie Huang, Yuxiao Dong, and Jie Tang. A survey of post-training scaling in large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2771–2791, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.140. URL https://aclanthology.org/2025.acl-long.140/.

Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. Dart: Noise injection for robust imitation learning. In Sergey Levine, Vincent Vanhoucke, and Ken Goldberg (eds.), *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pp. 143–156. PMLR, 13–15 Nov 2017. URL https://proceedings.mlr.press/v78/laskey17a.html.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 44776–44791. Curran Associates, Inc., 2023. doi: 10.52202/075280-1939. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/8c3c666820ea055a77726d66fc7d447f-Paper-Datasets_and_Benchmarks.pdf.

Yuhan Liu, Xinyu Zhang, Litao Liu, and Abdeslam Boularias. Foresight residual rl for long-horizon robot manipulation with vision-language-action models. *arXiv preprint arXiv:2607.16506*, 2026.

Jianlan Luo, Charles Xu, Jeffrey Wu, and Sergey Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *Science Robotics*, 10(105):eads5033, 2025. doi: 10.1126/scirobotics.ads5033. URL https://www.science.org/doi/abs/10.1126/scirobotics.ads5033.

Guozheng Ma, Lu Li, Haoyu Wang, Zixuan Liu, Pierre-Luc Bacon, and Dacheng Tao. What makes value learning efficient in residual reinforcement learning? In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=tSMZHXtJwZ.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In Aleksandra Faust, David Hsu, and Gerhard Neumann (eds.), *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1678–1690. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/mandlekar22a.html.

Max Sobol Mark, Tian Gao, Georgia Gabriela Sampaio, Mohan Kumar Srirama, Archit Sharma, Chelsea Finn, and Aviral Kumar. Policy-agnostic RL: Offline RL and online RL fine-tuning of any class and backbone. In *Scaling Self-Improving Foundation Models without Human Supervision*, 2025. URL https://openreview.net/forum?id=Mevh2AmlPc.

Mitsuhiko Nakamoto, Simon Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 62244–62269. Curran Associates, Inc., 2023. doi: 10.52202/075280-2719. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/c44a04289beaf0a7d968a94066a1d696-Paper-Conference.pdf.

Allen Ren, Justin Lidard, Lars Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 77288–77329, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/c0749c39aaff9e9e4c91f7118bf21b1e-Paper-Conference.pdf.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/ross11a.html.

Nur Muhammad Shafiullah, Zichen Cui, Ariuntuya (Arty) Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 22955–22968. Curran Associates, Inc., 2022. doi: 10.52202/068431-1668. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/90d17e882adbdda42349db6f50123817-Paper-Conference.pdf.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Tom Silver, Kelsey Allen, Josh Tenenbaum, and Leslie Kaelbling. Residual policy learning. *arXiv preprint arXiv:1812.06298*, 2018.

Entong Su, Tyler Westenbroek, Anusha Nagabandi, and Abhishek Gupta. RFS: Reinforcement learning with residual flow steering for dexterous manipulation. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=Kt9tJeOwjy.

Entong Su, Tyler Westenbroek, Anusha Nagabandi, and Abhishek Gupta. RFS: Reinforcement learning with residual flow steering for dexterous manipulation. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=Kt9tJeOwjy.

Ikechukwu Uchendu, Ted Xiao, Yao Lu, Banghua Zhu, Mengyuan Yan, Joséphine Simon, Matthew Bennice, Chuyuan Fu, Cong Ma, Jiantao Jiao, Sergey Levine, and Karol Hausman. Jump-start reinforcement learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 34556–34583. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/uchendu23a.html.

Andrew Wagenmaker, Yunchu Zhang, Mitsuhiko Nakamoto, Seohong Park, Waleed Yagoub, Anusha Nagabandi, Abhishek Gupta, and Sergey Levine. Steering your diffusion policy with latent space reinforcement learning. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 258–282. PMLR, 27–30 Sep 2025.

Xiu Yuan, Tongzhou Mu, Stone Tao, Yunhao Fang, Zhang Zhang, and Hao Su. Policy decorator: Model-agnostic online refinement for large policy model. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 28129–28164, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/45c361d4117d598d4bb6568b407e9ac9-Paper-Conference.pdf.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A PROOFS

A.1 PROOF OF PROPOSITION 4.2

Proof. Recall that the standard Bellman operator under policy π is given by

$$(\mathcal{T}^\pi Q)(x) = r(s,a) + \gamma \mathbb{E}\left[Q(x') \,|\, x\right], \qquad x = (s,a), \tag{17}$$

where the successor input $x' = (s', a')$ is generated according to $s' \sim \mathcal{P}(\cdot \mid s, a)$ and $a' \sim \pi(\cdot \mid s')$.

For any bounded action-value function Q, its induced pairwise value difference is $\Delta_Q(x_i, x_j) = Q(x_i) - Q(x_j)$. Applying the relative Bellman operator to Δ_Q gives

$$\begin{aligned}
(\mathcal{T}^\pi_{\mathrm{rel}} \Delta_Q)(x_i, x_j) &= r(s_i, a_i) - r(s_j, a_j) + \gamma \mathbb{E}\left[\Delta_Q(x_i', x_j') \,\middle|\, x_i, x_j\right] \\
&= r(s_i, a_i) - r(s_j, a_j) + \gamma \mathbb{E}\left[Q(x_i') - Q(x_j') \,\middle|\, x_i, x_j\right] \\
&= r(s_i, a_i) + \gamma \mathbb{E}\left[Q(x_i') \,|\, x_i\right] \\
&\quad - \left(r(s_j, a_j) + \gamma \mathbb{E}\left[Q(x_j') \,\middle|\, x_j\right]\right) \\
&= (\mathcal{T}^\pi Q)(x_i) - (\mathcal{T}^\pi Q)(x_j) \\
&= \Delta_{\mathcal{T}^\pi Q}(x_i, x_j).
\end{aligned} \tag{18}$$

The third equality follows from the linearity of expectation and the fact that the marginal successor distribution of each component is the one induced by π. Therefore, applying the relative Bellman operator to an induced value difference is equivalent to taking the difference between the corresponding standard Bellman updates. □

A.2 PROOF OF THEOREM 4.3

Proof. Let $\mathcal{B}(\mathcal{X} \times \mathcal{X})$ denote the space of bounded real-valued pairwise functions equipped with the supremum norm

$$\|\Delta\|_\infty := \sup_{x_i, x_j \in \mathcal{X}} |\Delta(x_i, x_j)|. \tag{19}$$

This is a complete metric space.

We first verify that $\mathcal{T}^\pi_{\mathrm{rel}}$ maps bounded pairwise functions to bounded pairwise functions. Since the reward function is bounded, there exists $R_{\max} < \infty$ such that $|r(s,a)| \le R_{\max}$ for every (s,a). Hence, for any bounded Δ,

$$\begin{aligned}
|(\mathcal{T}^\pi_{\mathrm{rel}} \Delta)(x_i, x_j)| &\le |r(s_i, a_i)| + |r(s_j, a_j)| + \gamma \mathbb{E}\left[|\Delta(x_i', x_j')| \,\middle|\, x_i, x_j\right] \\
&\le 2R_{\max} + \gamma \|\Delta\|_\infty.
\end{aligned} \tag{20}$$

Therefore, $\mathcal{T}^\pi_{\mathrm{rel}} \Delta$ is bounded.

Now consider any $\Delta_1, \Delta_2 \in \mathcal{B}(\mathcal{X} \times \mathcal{X})$. For every pair (x_i, x_j), the reward-difference terms cancel, yielding

$$\begin{aligned}
&|(\mathcal{T}^\pi_{\mathrm{rel}} \Delta_1)(x_i, x_j) - (\mathcal{T}^\pi_{\mathrm{rel}} \Delta_2)(x_i, x_j)| \\
&\quad = \gamma \left|\mathbb{E}\left[\Delta_1(x_i', x_j') - \Delta_2(x_i', x_j') \,\middle|\, x_i, x_j\right]\right| \\
&\quad \le \gamma \mathbb{E}\left[\left|\Delta_1(x_i', x_j') - \Delta_2(x_i', x_j')\right| \,\middle|\, x_i, x_j\right] \\
&\quad \le \gamma \|\Delta_1 - \Delta_2\|_\infty.
\end{aligned} \tag{21}$$

Taking the supremum over all (x_i, x_j) gives

$$\|\mathcal{T}^\pi_{\mathrm{rel}} \Delta_1 - \mathcal{T}^\pi_{\mathrm{rel}} \Delta_2\|_\infty \le \gamma \|\Delta_1 - \Delta_2\|_\infty. \tag{22}$$

Because $\gamma < 1$, the relative Bellman operator is a γ-contraction. The Banach fixed-point theorem therefore guarantees that $\mathcal{T}^\pi_{\mathrm{rel}}$ admits a unique fixed point.

It remains to identify this fixed point. The action-value function Q^π satisfies the standard Bellman equation $\mathcal{T}^\pi Q^\pi = Q^\pi$. By Proposition 4.2,

$$\mathcal{T}^\pi_{\mathrm{rel}} \Delta_{Q^\pi} = \Delta_{\mathcal{T}^\pi Q^\pi} = \Delta_{Q^\pi}. \tag{23}$$

Thus, Δ_{Q^π} is a fixed point of the relative Bellman operator. By uniqueness, it is the only fixed point, and it is given by

$$\Delta_{Q^\pi}(x_i, x_j) = Q^\pi(x_i) - Q^\pi(x_j). \tag{24}$$

Moreover, for any bounded initialization Δ_0, the iteration $\Delta_{k+1} = \mathcal{T}^\pi_{\mathrm{rel}} \Delta_k$ satisfies

$$\|\Delta_k - \Delta_{Q^\pi}\|_\infty \leq \gamma^k \|\Delta_0 - \Delta_{Q^\pi}\|_\infty , \tag{25}$$

and therefore converges to Δ_{Q^π}. □

B Implementation Details

B.1 Hyperparameters

Our implementation builds on the DPPO (Ren et al., 2025) and Policy Decorator (Yuan et al., 2025) codebases. Unless otherwise specified, we retain their original hyperparameters and use identical settings across compared methods. The two key hyperparameters of ReVRA are reported in Table 4.

Table 4: Key hyperparameters used in each experimental setting.

Base Policy	**Benchmark / Task**	**Residual scale λ**	**Relative weight β**
Diffusion Policy	Gym (all tasks)	0.30	1.00
	Robomimic (all tasks)	0.10	0.25
Behavior Transformer	ManiSkill2: StackCube	0.03	1.00
	ManiSkill2: PegInsertionSide	0.30	1.00
	ManiSkill2: TurnFaucet	0.20	1.00
	ManiSkill2: PushChair	0.20	1.00
VLA (SmolVLA 0.45B)	LIBERO (all suites)	0.05	0.10
Diffusion Policy	Real robot	0.10	1.00

B.2 Environment Details

We evaluate our method on four simulation benchmark suites: `Gym`, `Robomimic`, `ManiSkill2`, and `LIBERO`. These benchmarks collectively cover low-dimensional continuous control, single- and dual-arm robot manipulation, pixel-based imitation learning, and long-horizon language-conditioned tasks.

Table 5: Simulation benchmark configurations.

Benchmark / input	**Task**	**Observation dim.**	**Action dim.**	**Max. Episode Step**	**No. of demo trajectories**	**Generation method**
Gym	Hopper-v2	11	3	1000	N/A[†]	D4RL medium-policy rollouts
	Walker2d-v2	17	6	1000		
	HalfCheetah-v2	17	6	1000		
Robomimic	Can	23	7	500	300/task	Multi-human teleoperation
	Square	23	7	500		
	Transport	59	14	1100		
ManiSkill2	StackCube	55	4	200	1000/task	Task and motion planning
	PegInsertionSide	50	7	200		Task and motion planning
	TurnFaucet	43	7	200		Model predictive control
	PushChair	131	20	200		Reinforcement learning
LIBERO[‡]	Spatial	9	7	600	50/task	Human teleoperation
	Object	9	7	600		
	Goal	9	7	600		
	Long (LIBERO-10)	9	7	600		
Real Robot	Push Cube	128×128	4	200	10	Human teleoperation
	Grasp Cube	128×128	4	200	20	
	Stack Cube	128×128	4	200	20	

[†] Gym itself does not prescribe an offline demonstration dataset. For our Gym experiments, we use the D4RL medium dataset, which contains approximately 10^6 transitions per task rather than a fixed number of demonstration trajectories.

[‡] LIBERO uses a 7-DoF Franka Panda arm and a two-dimensional gripper state, yielding a 9-dimensional proprioceptive input. Policies additionally receive workspace and wrist RGB observations as well as a language instruction. All Libero tasks use a 7-dimensional continuous end-effector action.

B.3 Real-Robot Implementation Details

We adopt the HIL-SERL (Luo et al., 2025) paradigm for real-world training, implemented based on the official LeRobot codebase [1].

Robot Platform and Observations. Our real-robot experiments are conducted on a RealMan RM75-B robotic arm equipped with a parallel gripper. As illustrated in Figure 5, the workspace is observed by two RGB cameras: a wrist-mounted camera provides a close-up view of the interaction region, while a fixed side-view camera captures the global scene. Each image is cropped and resized to 128×128. The policy observation at timestep t is

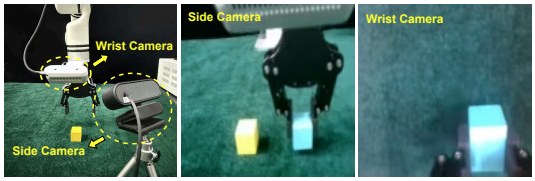


Figure 5: Real-robot setup.

$$o_t = \left\{ I_t^{\text{wrist}}, I_t^{\text{side}}, x_t^{\text{ee}} \right\},$$

where $x_t^{\text{ee}} \in \mathbb{R}^7$ denotes the robot end-effector state. The same observation definition and image preprocessing pipeline are used for demonstration collection, Diffusion Policy training, residual RL fine-tuning, and evaluation.

Action Representation and Low-level Execution. The learning-space action is four-dimensional:

$$a_t = [\Delta x_t, \Delta y_t, \Delta z_t, g_t],$$

where the first three dimensions specify the relative Cartesian translation and g_t controls the gripper.

Before execution, each translational component is clipped to $[-1, 1]$ and filtered using exponential smoothing with coefficient 0.4. The filtered translation is then multiplied by 0.01 and passed to the RealMan controller as a Cartesian displacement command. This scaling limits the magnitude of each robot motion and is applied only in the execution layer; actions stored in the dataset and replay buffer remain in the normalized learning space. The robot is controlled at $10\,\text{Hz}$.

The gripper command is processed independently of the Cartesian translation and is neither smoothed nor multiplied by the positional scale. We use a hysteresis-style decoder: the gripper closes when $g_t > 0.2$, opens when $g_t < -0.2$, and retains its previous state otherwise. At the beginning of every episode, the robot returns to a predefined joint configuration, after which we wait five seconds before enabling policy control.

Human Intervention. A 3Dconnexion SpaceMouse is used for demonstration collection and optional intervention during online training and evaluation. The operator controls the Cartesian translation and the gripper in the same normalized action space used by the learned policies. Human commands pass through the same clipping, smoothing, scaling, and gripper-decoding pipeline as autonomous commands.

During residual RL, an intervention replaces the autonomous action for the current timestep. The replay buffer records the action that was actually executed on the robot, together with an intervention indicator. We store transitions in the temporally aligned form

$$(o_t, a_t, r_t, o_{t+1}, d_t),$$

where o_t is captured before executing a_t, and o_{t+1} is collected after the resulting robot motion.

Episode and Reward Design. Each episode is limited to 20 seconds, corresponding to a maximum of 200 steps at a control frequency of $10\,\text{Hz}$. A successful completion produces a terminal task reward of $+1$, whereas failure and timeout terminate the episode with zero task reward.

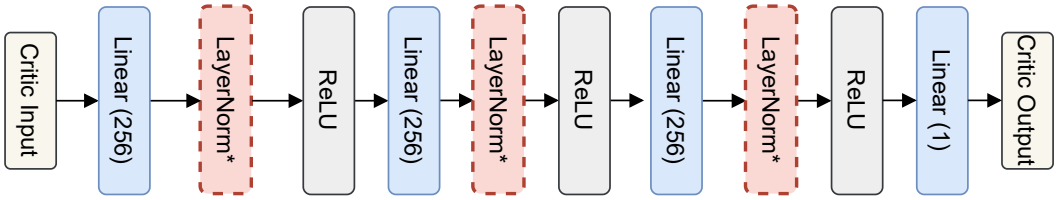
[1] https://huggingface.co/docs/lerobot/hilserl

B.4 Critic Backbones and Normalization Placement

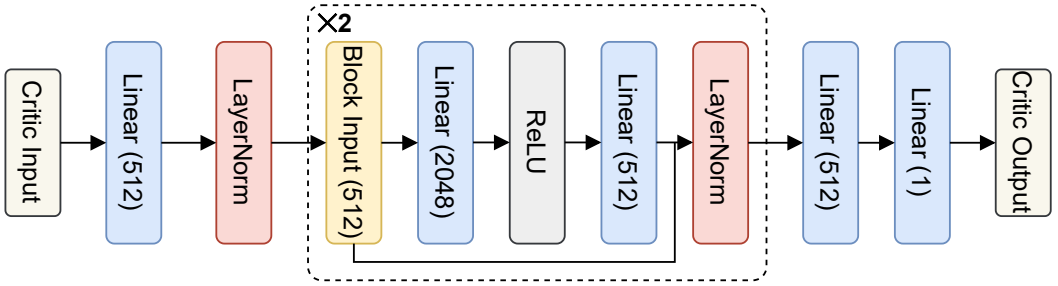
Our experiments cover two critic backbones taken from the corresponding public implementations. The Behavior Transformer experiments follow the official Policy Decorator codebase,[2] whereas the DAWN-based Diffusion Policy experiments follow the released DAWN implementation.[3] Figure 6 illustrates the resulting architectural differences.

The Policy Decorator critic is a sequential MLP with three 256-unit hidden layers. In the normalization-only BeT variant, LayerNorm is inserted after each hidden linear transformation and before the ReLU activation. Enabling normalization therefore does not change the width, depth, or connectivity of the original Policy Decorator critic.

The released DAWN critic uses a different backbone: a 512-dimensional input projection followed by two wide residual blocks, each containing a $512 \to 2048 \to 512$ transformation. LayerNorm is applied after the input projection and after each residual addition. We refer to this architecture specifically as the *released DAWN critic*, since the residual-block topology is taken from the public implementation.



(a) Policy Decorator critic. Dashed LayerNorm layers are enabled only in the normalized BeT variants.



(b) Released DAWN critic with an initial normalized projection and two post-normalized residual blocks.

Figure 6: Critic backbones used in our experiments. The Behavior Transformer experiments use the Policy Decorator MLP, while the DAWN-based Diffusion Policy experiments use the residual-block critic from the released DAWN implementation.

For the normalized Policy Decorator MLP, a hidden layer is computed as

$$h_{\ell+1}^{\mathrm{MLP+LN}} = \mathrm{ReLU}\left(\mathrm{LN}\left(W_\ell h_\ell + b_\ell\right)\right). \tag{26}$$

For the released DAWN critic, the input projection and each residual block are given by

$$\begin{aligned} h_0^{\mathrm{DAWN}} &= \mathrm{LN}\left(W_{\mathrm{in}} x + b_{\mathrm{in}}\right), \\ h_{\ell+1}^{\mathrm{DAWN}} &= \mathrm{LN}\left(h_\ell + F_\ell(h_\ell)\right). \end{aligned} \tag{27}$$

Thus, the DAWN skip connection is added *before* LayerNorm and does not form a normalization-free forward or gradient bypass. The two backbones differ not only in normalization placement, but also in width, parameter count, and residual connectivity.

To separate the contribution of relative value learning from these architectural differences, Figure 3 evaluates ReVRA under both backbones. In the BeT setting, ReVRA and its absolute-value baseline share the same Policy Decorator MLP. In the Diffusion Policy setting, DAWN+ReVRA and DAWN share the same released DAWN critic and training configuration.

[2] https://github.com/tongzhoumu/policy_decorator

[3] https://github.com/Guozheng-Ma/DAWN

C Additional Experimental Results

C.1 Additional Loss Ablations

Figure 7 extends the loss ablation in Figure 4. All three objectives perform similarly on `Can` and `Hopper`, where their learning curves quickly converge. The distinction is clearer on `PegInsertionSide`: the coupled objective reaches near-perfect success much earlier, while absolute-only learning improves slowly and relative-only learning shows substantial variability across seeds. On `StackCube`, combining the losses also improves learning speed and final performance, although the gap is smaller. `Walker2d` shows that relative-only learning can match or slightly exceed the coupled objective. Overall, the coupled objective provides a balanced performance profile across the evaluated tasks.

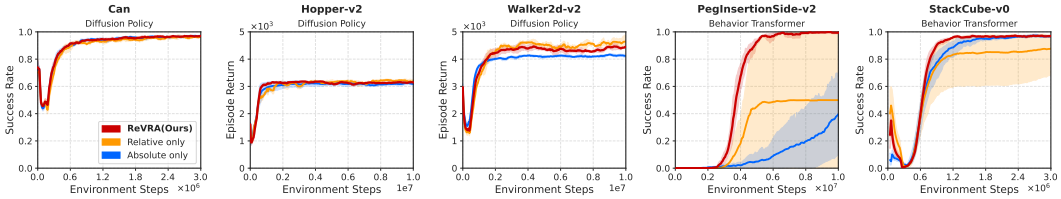


Figure 7: Additional ablations of critic supervision. ReVRA's coupled objective is compared with relative-only learning (without absolute TD loss) and absolute-only learning (without relative loss) across five tasks.

C.2 Effect of Critic LayerNorm

Figure 8 compares ReVRA with and without critic LayerNorm. Both variants retain the same absolute and relative critic objectives. With Diffusion Policy and the DAWN critic architecture, LayerNorm accelerates early learning on `Walker2d`, but the two variants achieve similar final returns on both locomotion tasks. With Behavior Transformer and the Policy Decorator critic, removing LayerNorm slightly delays early learning on `StackCube` but accelerates learning on `PushChair` and `PegInsertionSide`; final success remains similar across all three tasks. Thus, LayerNorm can affect learning speed, but ReVRA maintains strong final performance without it across both critic architectures.

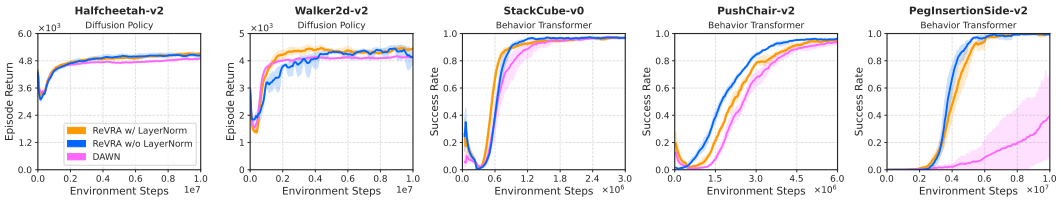


Figure 8: Effect of critic LayerNorm. ReVRA with and without LayerNorm is compared with DAWN. The first two tasks use Diffusion Policy as the base policy; the remaining three use Behavior Transformer.

LayerNorm acts at different locations in the two architectures: after the input projection and residual blocks in the DAWN critic, but at each hidden layer in the Policy Decorator MLP. It changes feature scaling and sensitivity without directly supervising the accuracy of value differences, and its benefits are not uniform across these settings. We therefore recommend relative Bellman supervision as the core modification, with LayerNorm treated as an optional, architecture-dependent choice. ReVRA without LayerNorm remains competitive with its normalized counterpart and achieves final performance comparable to or better than DAWN across all five tasks, showing that its effectiveness does not depend on normalization.

C.3 REVRA WITH FULL DAWN

We implement this experiment in the publicly released DAWN codebase, using Diffusion Policy as the frozen base policy. Figure 9 compares full DAWN (Ma et al., 2026), including implicit data warmup and critic LayerNorm, with the same method augmented by ReVRA on two Gym locomotion tasks. On `HalfCheetah`, the methods perform similarly early in training, after which DAWN + ReVRA achieves higher returns. On `Walker2d`, DAWN + ReVRA maintains higher returns for most of training. These results show that relative value supervision provides further gains when combined with DAWN's warmup and normalization.

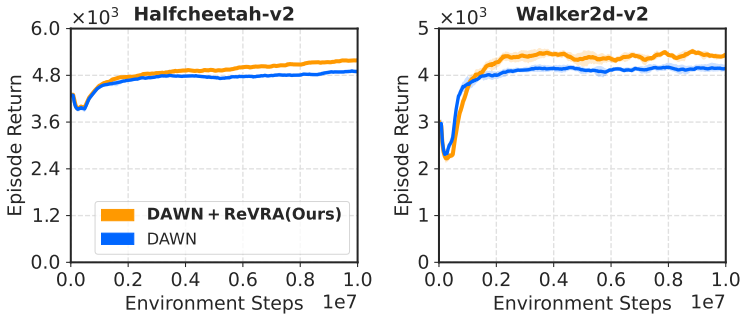


Figure 9: Episode returns on two Gym locomotion tasks with Diffusion Policy as the frozen base policy. Full DAWN includes implicit data warmup and critic LayerNorm; DAWN + ReVRA additionally applies relative value supervision.

D ALGORITHM

Algorithm 1 summarizes ReVRA within residual SAC. At each critic update, ReVRA pairs replay transitions and applies relative Bellman supervision to the existing action-value critics.

Algorithm 1 ReVRA: Relative Value Learning for Residual Adaptation

Require: Frozen base policy π_{base}, residual scale λ, relative weight β

Initialize residual actor $\pi_{\text{res},\phi}$, critic Q_θ, target critic $\bar{Q}$, and replay buffer $\mathcal{D}$
for each environment step **do**
  Sample $a_{\text{res},t} \sim \pi_{\text{res},\phi}(\cdot \mid s_t)$
  Execute $a_t = \pi_{\text{base}}(s_t) + \lambda a_{\text{res},t}$
  Observe (r_t, s_{t+1}, d_t) and store $((s_t, a_t), r_t, d_t, s_{t+1})$ in $\mathcal{D}$
  for each gradient step **do**
    Sample $\mathcal{B} = \{\tau_i\}_{i=1}^{B}$ from $\mathcal{D}$; pair i with $j = \sigma(i)$ by a cyclic shift
    Form successor inputs x_i' and log probabilities ℓ_i' using the current residual policy
    Compute SAC targets y_i^Q and relative targets $y_{i,\sigma(i)}^{\Delta}$ using Eq. 14
    Update Q_θ by minimizing Eq. 15
    Update $\pi_{\text{res},\phi}$ using Eq. 16
    Update the entropy temperature and target critic as in SAC
  end for
end for