FIRST-ORDER CONTROL USING OPTIMAL TRANSPORT SUPERVISION FOR IMITATION FROM OBSERVATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Imitation learning from observations must recover behavior from state-only demonstrations without known temporal correspondence. Differentiable simulators provide pathwise policy gradients, but differentiating through long, contact-rich trajectories can make optimization unstable. We introduce FOCUS, which combines local optimal-transport alignment with short-horizon differentiable actor–critic learning. FOCUS matches agent rollout windows to multiple sampled expert segments using a soft best-of-K objective that emphasizes locally compatible matches. The resulting transport-based costs provide differentiable imitation rewards, while a learned critic bootstraps value beyond the rollout horizon. FOCUS requires neither temporally aligned trajectories nor expert-aligned initialization or teacher forcing. Across contact-rich locomotion tasks in DFlex, including a musculoskeletal humanoid controlled by 152 muscle activations, FOCUS improves sample efficiency and training stability over adversarial imitation-from-observation methods and prior differentiable imitation baselines.

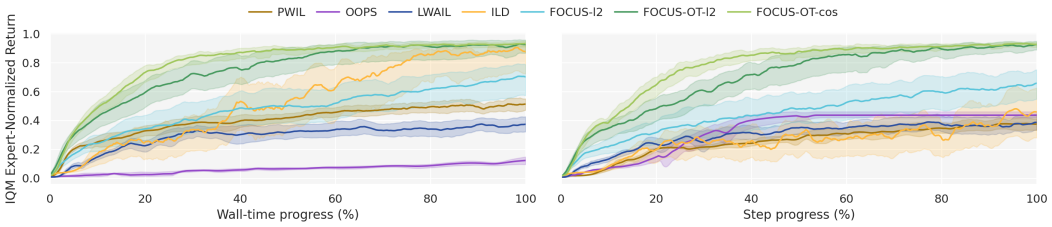


Figure 1: Aggregate performance of selected methods across four DFlex tasks. Curves show the interquartile mean (IQM) of expert-normalized return; shading indicates 95% confidence intervals. Progress is expressed as a percentage of each task's wall-clock budget (left) or interaction budget (right); the full comparison, budgets, and per-task results are reported in Appendix C.

1 INTRODUCTION

Imitation learning from observations (ILfO) must recover control from state-only demonstrations, without access to expert actions or known temporal correspondence between learner and expert trajectories. This makes supervision fundamentally ambiguous: the learner must determine both *what* behavior to imitate and *where* its current behavior corresponds to the demonstration.

Optimal-transport (OT) methods are well suited to the alignment problem because they can compare trajectories or state distributions without requiring point-wise temporal correspondence (Dadashi et al., 2021; Papagiannis & Li, 2022; Cohen et al., 2021a; Fickinger et al., 2021; Haldar et al., 2022; Chang et al., 2024). In practice, however, these methods typically convert transport costs into surrogate rewards and optimize the policy with zeroth-order RL, inheriting the sample inefficiency and high-variance updates of black-box policy optimization (Mohamed et al., 2020). Adversarial ILfO methods similarly learn from state-only demonstrations through discriminator-derived rewards (Torabi et al., 2019; Zhu et al., 2020; Candido Ramos et al., 2024), but can be sensitive to reward modeling, discriminator stability, and tuning.

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

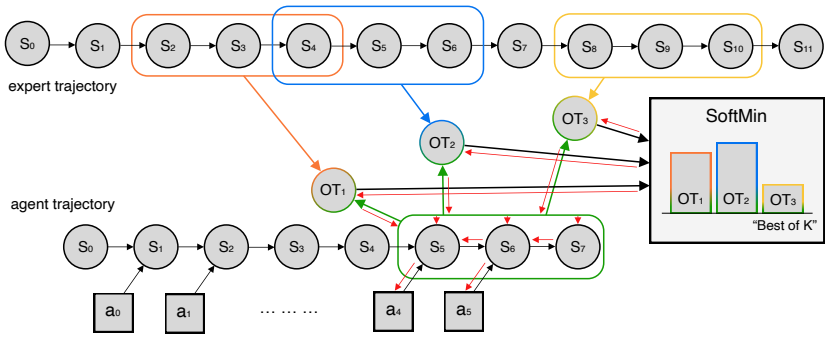


Figure 2: Computational graph of FOCUS. A short agent rollout (green) is matched to sampled expert segments (orange, blue, yellow) using OT. Soft best-of-K aggregation emphasizes well-aligned segments, and the resulting supervision is backpropagated through the simulator and policy (red arrows).

Differentiable physics simulators provide a complementary route by exposing path-wise policy gradients through the environment dynamics. Simulators such as Brax (Freeman et al., 2021), Warp (Macklin, 2022), and DFlex (Xu et al., 2022) have made this setting practical, while first-order control methods such as SHAC (Xu et al., 2022), AHAC (Georgiev et al., 2024), and SAPO (Xing et al., 2025) stabilize differentiable control through short-horizon rollouts and critic bootstrapping. These methods, however, assume access to an environment reward. Differentiable imitation already addresses some alignment challenges: GDTW-IL (Cohen et al., 2021b) matches observed trajectories through known dynamics, and ILD (Chen et al., 2023) combines unsynchronized state matching with truncated gradients, using behavior-cloning pretraining. DiffMimic (Ren et al., 2023) uses expert-aligned initialization and teacher forcing.

We focus on combining local alignment of state-only demonstrations with short differentiable rollouts and critic bootstrapping for contact-rich control. Directly differentiating a trajectory-level imitation loss through long contact-rich rollouts is particularly brittle, since contacts, numerical stiffness, and large early mismatches can produce unstable or uninformative gradients (Zhong et al., 2022; Suh et al., 2022; Holl et al., 2022; Gillen & Byl, 2022).

We introduce **FOCUS** (First-Order Control Using Optimal Transport Supervision), which turns temporally unaligned state-only demonstrations into short-horizon differentiable supervision. Rather than matching complete trajectories, FOCUS compares each agent rollout window against multiple sampled expert segments using local OT. Softmin-weighted best-of-K aggregation emphasizes locally compatible matches while retaining differentiability, reducing destructive updates from phase-shifted or unreachable segments. The resulting transport costs provide differentiable imitation rewards, and a learned critic bootstraps value beyond the local matching window. This formulation requires neither expert actions, temporally aligned trajectories, expert-aligned initialization, nor teacher forcing, and can operate even when only a single expert trajectory is available. Fig. 2 illustrates the resulting computational graph.

Across DFlex locomotion tasks (Xu et al., 2022), including the SNU Humanoid with 152 muscle actuators, FOCUS improves sample efficiency over adversarial ILfO and differentiable imitation baselines, with stable learning on the challenging humanoid tasks. These results support using simulator derivatives for high-dimensional state-only imitation.

2 Related Work

Black-box imitation from observations. ILfO learns from state-only demonstrations. BCO (Torabi et al., 2018) infers missing actions through an inverse dynamics model, while adversarial methods match state or state-transition distributions (Torabi et al., 2019). OPOLO (Zhu et al., 2020) and MAAD (Candido Ramos et al., 2024) combine distribution matching with inverse-dynamics regularization. DIFO (Huang et al., 2024) uses a conditional diffusion model as a transition discriminator to supply imitation rewards. These methods learn through environment interactions without requiring

simulator derivatives. FOCUS also learns online, but uses known differentiable dynamics to obtain path-wise policy gradients from local imitation costs.

Optimal transport for imitation. PWIL (Dadashi et al., 2021) constructs imitation rewards using greedy transport matching, while SIL (Papagiannis & Li, 2022) uses Sinkhorn-based matching. OOPS (Chang et al., 2024) matches state-transition distributions and supplies dense rewards for off-policy learning from observations. LWAIL (Yang et al., 2026) performs online adversarial Wasserstein matching in a dynamics-aware representation pretrained on random state-only data. These methods optimize policies from transport-derived rewards without differentiating through simulator dynamics. FOCUS instead couples local OT matching with short differentiable rollouts and soft best-of-K expert-segment selection. Temporal and cross-domain methods, including GWIL (Fickinger et al., 2021), TemporalOT (Fu et al., 2024), and ORCA (Huey et al., 2025), are discussed in Appendix A.

Differentiable imitation and control. GDTW-IL (Cohen et al., 2021b) differentiates a trajectory-alignment loss through known dynamics. DiffMimic (Ren et al., 2023) optimizes an ℓ_2 trajectory loss with expert-aligned initialization and teacher forcing. ILD (Chen et al., 2023) uses Chamfer-α matching without state synchronization and truncated backpropagation after behavior-cloning pretraining. FOCUS instead uses OT couplings over sampled expert segments and soft best-of-K aggregation to select locally compatible targets. It combines this supervision with short-horizon critic bootstrapping, building on SHAC (Xu et al., 2022) and SAPO (Xing et al., 2025), to propagate imitation signal beyond the differentiable rollout. Unlike these reward-based control methods, FOCUS derives its training signal from state-only demonstrations. Our experiments use neither expert-aligned initialization nor teacher forcing. Appendix A provides an extended discussion, including broader imitation-learning and learned-dynamics approaches.

3 BACKGROUND

Preliminaries: Markov Decision Process (MDP) and Imitation Learning from Observations (ILfO). We consider agents acting in a γ-discounted infinite-horizon Markov decision process (MDP), defined as $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \rho_0, r, \gamma \rangle$, where $\mathcal{S}$ and $\mathcal{A}$ denote state and action spaces, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is a transition kernel into the probability measures on $\mathcal{S}$, ρ_0 is the initial state distribution, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor.

In the imitation learning setting, an expert policy π_E generates trajectories $\xi = \{(s_0, a_0), (s_1, a_1), \ldots, (s_T, a_T)\}$ composed of state-action pairs. These trajectories form a demonstration dataset $\mathcal{D}_E = \{\xi_i\}, \xi_i \sim \pi_E$, which is used to train the imitation policy. In this work, we focus on imitation learning from observations (ILfO), where expert actions are not available. Instead, each trajectory consists only of a sequence of observed states or state transitions $\zeta = \{(s_0, s_1), (s_1, s_2), \ldots, (s_{T-1}, s_T)\}$, and the imitation policy is trained on a dataset of observed trajectories $\mathcal{D}_E = \{\zeta_i\}, \zeta_i \sim \pi_E$.

Optimal Transport for Imitation Learning. A useful lens for imitation learning is the notion of *occupancy measure* (OM), which captures the distribution of states encountered in an MDP under a policy π (Ho & Ermon, 2016). For a measurable set $B \subseteq \mathcal{S}$, we define the normalized discounted state occupancy measure as $\rho_\pi(B) = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \, \mathbb{P}(s_t \in B \mid \pi)$. The factor $1-\gamma$ gives unit total mass. The same construction applies to state-action pairs and state-state transitions, enabling comparisons between expert behavior and a learned policy at multiple levels of granularity. Framing imitation as a problem of matching such distributions removes the need for strict point-wise alignment based on classical distances (e.g., Euclidean or cosine) and instead motivates objectives that compare expert and agent trajectories in a softer manner. In particular, *optimal transport* (OT) is a popular choice for comparing agent and expert trajectories: it computes a coupling that specifies how probability mass from the agent trajectory distribution should be moved to match the expert trajectory distribution at minimum total transport cost (Dadashi et al., 2021; Papagiannis & Li, 2022; Cohen et al., 2021a; Fickinger et al., 2021; Haldar et al., 2022; Chang et al., 2024). Concretely, letting ρ_π and ρ_E denote the state OM induced by the policy π_θ and the expert respectively, one can define a trajectory-level discrepancy as the OT cost:

$$\mathrm{OT}(\rho_\pi, \rho_E) = \min_{\mu \in \Pi(\rho_\pi, \rho_E)} \int_{\mathcal{S}} \int_{\mathcal{S}} c(s, s^E) \, \mathrm{d}\mu(s, s^E), \tag{1}$$

where $c(s, s^E)$ is a state-wise cost and $\Pi(\rho_\pi, \rho_E)$ is the set of couplings between the two distributions. For finite-trajectory matching, we use uniform empirical distributions $\widehat{\rho}_\pi = H^{-1} \sum_{i=0}^{H-1} \delta_{s_i}$ and $\widehat{\rho}_E = L^{-1} \sum_{j=0}^{L-1} \delta_{s_j^E}$, where δ_s is a unit point mass at s. These distributions describe the sampled windows rather than the full discounted occupancies. The discrete coupling matrix $\mu \in \mathbb{R}_+^{H \times L}$ has row sums $1/H$ and column sums $1/L$. We use the entropy-regularized OT objective

$$\mathrm{OT}_\varepsilon(\widehat{\rho}_\pi, \widehat{\rho}_E) = \min_{\mu \in \Pi(\widehat{\rho}_\pi, \widehat{\rho}_E)} \langle C, \mu \rangle + \varepsilon \sum_{i,j} \mu_{i,j} \left(\log \mu_{i,j} - 1 \right), \tag{2}$$

where $C_{i,j} = c(s_i, s_j^E)$, and $\varepsilon > 0$ controls the smoothness of the transport plan. This regularization makes the coupling smooth and allows the optimal plan $\mu^\star$ to be efficiently approximated with the Sinkhorn algorithm (Cuturi, 2013). The indices i and j enumerate agent and expert timesteps, respectively; we write them as t and t' when defining per-step rewards. This loss can be used for traditional RL through a surrogate reward $r(s_t) = -\sum_{t'=0}^{L-1} C_{t,t'} \mu^\star_{t,t'}$, for $t = 0, \ldots, H-1$, which encourages the agent to visit states that have low-cost alignments with expert behavior.

First-Order Model-Based Reinforcement Learning (FO-MBRL). FO-MBRL takes advantage of differentiable simulators to perform policy optimization by backpropagating gradients through the system dynamics. This allows agents to use gradient-based optimization for learning, reducing the variance associated with standard black-box approaches that rely on stochastic estimators.

One of the simplest FO-MBRL methods is Analytic Policy Gradient (APG) (Freeman et al., 2021), which optimizes a finite-horizon return of the form:

$$J(\pi_\theta) = \mathbb{E}\left[\sum_{t=0}^{T-1} \gamma^t r(s_t, a_t) \right], \tag{3}$$

where gradients $\nabla_\theta J$ are computed directly by differentiating through the unrolled simulation. This formulation corresponds to Backpropagation Through Time (BPTT) (Werbos, 1990), which has also been applied in full-horizon settings for continuous control (Degrave et al., 2019; Huang et al., 2021).

Despite its simplicity, BPTT can become unstable over long horizons due to gradient vanishing or explosion, problems which get even more severe in environments with contact-rich or discontinuous dynamics. To address this, SHAC (Xu et al., 2022) proposes a short-horizon variant, i.e. $H \ll T$ (Williams & Zipser, 1995), which augments the return with a bootstrapped terminal value estimate:

$$J_{\mathrm{SHAC}}(\pi_\theta) = \mathbb{E}\left[\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t) + \gamma^H V_\psi(s_H) \right], \tag{4}$$

where V_ψ is trained via TD(λ) (Sutton, 1988) using multi-step return targets over short rollouts. AHAC (Georgiev et al., 2024) refines this approach by dynamically adjusting the rollout horizon H to reduce gradient instabilities caused by simulation discontinuities, while SAPO (Xing et al., 2025) further improves stability by introducing maximum-entropy regularization and a soft value bootstrap:

$$J_{\mathrm{SAPO}}(\pi_\theta) = \mathbb{E}\left[\sum_{t=0}^{H-1} \gamma^t (r(s_t, a_t) + \alpha\, \mathcal{H}[\pi_\theta(\cdot \mid s_t)]) + \gamma^H V_{\mathrm{soft},\psi}(s_H) \right], \tag{5}$$

where α controls the strength of exploration via the entropy term and $V_{\mathrm{soft},\psi}$ is a learned soft value function used for bootstrapping.

4 METHOD

FOCUS requires a state-only trajectory discrepancy that handles temporal misalignment and supplies informative gradients through simulated rollouts. We combine local OT matching, best-of-K expert-segment selection, and short-horizon actor–critic learning to address these requirements.

4.1 SHORT-HORIZON OPTIMAL TRANSPORT ALIGNMENT

We present the method using state distributions; our experiments use state-state transitions. The OT cost between policy and expert distributions in Eq. 1 gives the first-order update

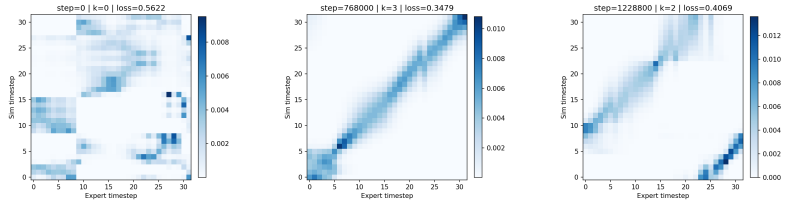
$$\theta \leftarrow \theta - \lambda \nabla_\theta \mathrm{OT}(\rho_{\pi_\theta}, \rho_E), \tag{6}$$

where λ is the learning rate. Backpropagating through full trajectories can produce unstable gradients, particularly with contact discontinuities (Degrave et al., 2019).

To mitigate this, we adopt a short-horizon strategy inspired by prior work in FO-MBRL (Xu et al., 2022; Georgiev et al., 2024; Xing et al., 2025): rather than backpropagating through the entire trajectory, we unroll a short H-step agent segment ($H \ll T$), compute OT costs against expert segments, and use these costs to construct the short-horizon actor–critic surrogate described below. Fig. 3 illustrates representative local couplings.

Local matching leaves two challenges:

1. **Mismatched targets:** A sampled expert segment can be phase-shifted or locally unreachable, directing gradients toward an unsuitable target.
2. **Myopic supervision:** Local costs provide no direct credit beyond the rollout horizon, motivating critic bootstrapping (Table 11).



(a) Initial or ambiguous match (b) Phase-aligned match (c) Phase-shifted match

Figure 3: Representative local OT couplings: (a) diffuse, (b) near-diagonal, and (c) shifted matching patterns. Rows index agent transitions and columns index expert transitions; darker entries indicate greater transport mass. Panels show selected training snapshots with different candidate expert segments. Additional examples appear in Appendix D.

4.2 Best-of-K Local Alignment

Best-of-K addresses the first challenge above: choosing an informative expert phase for the local OT comparison. Rather than computing the OT loss against the entire expert trajectory, we compare each short agent rollout $\tau_\theta^\pi = (s_0^\pi, \ldots, s_{H-1}^\pi)$ to a collection of K expert segments $\{\tau_k^E\}_{k=1}^K$. Each segment $\tau_k^E = (s_{k,0}^E, \ldots, s_{k,L-1}^E)$ is a sub-trajectory of length L sampled from the expert trajectory, as illustrated in Fig. 2. We use zero-based local timestep indices throughout. An H-step rollout also produces the terminal state s_H^π for bootstrapping. In the transition-based experiments, the H matching items are (s_t^π, s_{t+1}^π) for $t = 0, \ldots, H-1$; likewise, an expert segment contains L transitions and requires $L+1$ consecutive states.

The ideal local alignment objective would compare the rollout to the best expert segment among all admissible expert start times. Let $\mathcal{I}_E$ denote the set of valid start times across the expert demonstrations, and let τ_u^E be the length-L segment starting at $u \in \mathcal{I}_E$. A hard local objective is:

$$\mathcal{L}_{\text{local}}^\star(\theta) = \min_{u \in \mathcal{I}_E} \mathcal{L}_{\text{OT}}(\tau_\theta^\pi, \tau_u^E). \tag{7}$$

Enumerating all start times at every update is expensive. Instead, we sample K candidate start times $\{u_k\}_{k=1}^K$ and use transport-only segment scores $\ell_k(\theta) = \langle \mu^{\star(k)}, C^{(k)} \rangle$ (Eq. 10). Under temporal misalignment, a single sampled segment can be phase-shifted or locally unreachable, causing the policy to receive gradients toward a poor target. Comparing multiple candidate segments allows the weighting to favor locally compatible targets; the sampled softmin is a smooth minimum over candidates, not an unbiased estimator of the minimum over all start times (see Appendix E.4 for an empirical diagnostic). A smooth relaxation of the hard minimum motivates our segment weights:

$$\mathcal{L}_{\text{OT}}^{\text{boK}}(\theta) = -\beta \log \sum_{k=1}^{K} \exp\left(-\frac{\ell_k(\theta)}{\beta}\right), \tag{8}$$

where $\beta > 0$ is a temperature parameter, fixed at 0.5 in our experiments. As $\beta \to 0$, this log-sum-exp relaxation approaches the hard minimum. Its gradient is a Boltzmann-weighted mixture of segment gradients,

$$\nabla_\theta \mathcal{L}_{\mathrm{OT}}^{\mathrm{boK}}(\theta) = \sum_{k=1}^{K} w_k \nabla_\theta \ell_k(\theta), \quad w_k = \frac{\exp\left(-\ell_k(\theta)/\beta\right)}{\sum_{j=1}^{K} \exp\left(-\ell_j(\theta)/\beta\right)}. \tag{9}$$

These weights assign greater mass to lower-cost candidates. We use them to aggregate per-step costs in Eq. 11, with gradients through the weights. This training surrogate differs from directly minimizing the log-sum-exp relaxation above. The weighting remains local: by itself, it neither propagates credit beyond the rollout horizon nor guarantees globally optimal trajectory alignment. In a paired diagnostic on held-out expert windows at a half-gait-cycle mismatch, adding seven random alternatives yields a phase-compatible minimum-OT-cost selection in 46–72% of query–pool pairs and reduces mean phase error by approximately 3–5$\times$ across four environments (Appendix E.4).

4.3 Mitigating Myopic Supervision with Value-Based Credit Assignment

We learn a value function to propagate imitation signal beyond the short rollout (Xu et al., 2022; Georgiev et al., 2024; Xing et al., 2025). For each agent–expert segment pair, we approximate $\mu^{\star(k)}$ with a fixed number of Sinkhorn iterations and differentiate through the costs, iterations, and simulated dynamics (Appendix B.5). For each expert segment k, we define the OT-based reward for agent state s_t as:

$$r_t^{(k)} = -\mathcal{L}_t^{(k)}, \quad \text{with} \quad \mathcal{L}_t^{(k)} = \sum_{t'=0}^{L-1} C_{t,t'}^{(k)} \mu_{t,t'}^{\star(k)}, \tag{10}$$

where $t = 0, \ldots, H-1$ and $C_{t,t'}^{(k)}$ is the cost between the agent state s_t and the expert state $s_{k,t'}^{E}$ (or their corresponding transitions in the experiments). To obtain a single training signal, we aggregate the K expert-aligned per-step costs using the same softmin weights w_k from Eq. 9. The aggregated per-step cost and base OT reward are:

$$r_t^{\mathrm{OT}} = -\mathcal{L}_t^{\mathrm{OT}}, \quad \text{where} \quad \mathcal{L}_t^{\mathrm{OT}} = \sum_{k=1}^{K} w_k \, \mathcal{L}_t^{(k)}. \tag{11}$$

The weights are differentiable functions of the segment losses. Consequently, the derivative of the aggregated cost contains both a weighted cost-gradient term and a weight-gradient term. Equation 9 describes the motivating soft minimum, not the full policy update through this aggregation; Appendix B.5 gives the distinction explicitly. The critic V_ψ approximates continuation returns under expert-segment sampling. Since transport plans and weights depend on the whole window, later actions can affect earlier rewards; the critic is a surrogate, not an exact value function for a fixed Markov reward. Section 4.4 defines the shaped actor–critic objective.

4.4 Reward Transformations for Policy and Critic Learning

In Section 4.3, we used the standard imitation reward $r = -\mathcal{L}$. While a natural baseline, its scale is not always suitable for actor-critic training: early in learning, large costs can produce high-magnitude TD targets, while near expert support the rewards concentrate near zero and provide little variation for the critic. We therefore use monotone decreasing transformations of the OT cost that preserve the alignment ordering but change the reward scale. Using the aggregated per-step cost from Eq. 11, we define:

$$r_t^{\mathrm{FOCUS}} = \phi(\mathcal{L}_t^{\mathrm{OT}}), \qquad \phi(\mathcal{L}) \in \left\{-\mathcal{L},\ \exp(-\mathcal{L}),\ -\log(1 - e^{-\mathcal{L}} + \epsilon)\right\}. \tag{12}$$

We set $\epsilon = 10^{-7}$ for numerical stability inside the logarithm. The exponential mapping requires no stabilizer. The first option recovers the raw negative-cost reward. The exponential transform bounds rewards and compresses large costs, while the logarithmic transform amplifies differences among very low-cost alignments. These transformations preserve the *local* alignment ranking; they do not in general preserve trajectory-level policy ordering after rewards are summed over time. They change the scale and curvature of the signal seen by the actor and critic. We empirically evaluated the effect of these reward parameterizations in Appendix D and Table 11. FOCUS therefore optimizes a shaped short-horizon actor-critic surrogate derived from local best-of-K OT costs, rather than directly

Algorithm 1 First-Order Control Using Optimal Transport Supervision for Imitation from Observations

Require: Expert trajectories $\mathcal{D}_E = \{\zeta_i^E\}$, where $\zeta_i^E = \{s_0^E, s_1^E, \ldots, s_T^E\}$
Require: Initial policy π_θ, value function V_ψ, number of expert segments K, segment length L, rollout horizon H, temperature β, total iterations M
Initialize π_θ, V_ψ
for 1 to M **do**
  Roll out π_θ for H steps, collecting $s_0, \ldots, s_H$; form τ^π from the H matching items and retain s_H for bootstrapping
  Sample K expert segments $\{\tau_k^E\}_{k=1}^K$ of length L from $\mathcal{D}_E$
  for $k = 1$ to K **do**
    Compute the Sinkhorn transport plan $\mu^{\star(k)}$
    Compute segment loss ℓ_k and per-step costs $\{\mathcal{L}_t^{(k)}\}_{t=0}^{H-1}$ using $\mu^{\star(k)}$ and differentiable costs (Eq. 10)
  end for
  Compute weights w_k from the segment losses and aggregate per-step costs via Eq. 11; retain gradients through w_k
  Apply reward transformation: $r_t^{\mathrm{FOCUS}} = \phi(\mathcal{L}_t^{\mathrm{OT}})$
  Update policy π_θ using actor-critic loss with short-horizon return: $J_{\mathrm{FOCUS}}(\pi_\theta) = \sum_{t=0}^{H-1} \gamma^t r_t^{\mathrm{FOCUS}} + \gamma^H V_\psi(s_H)$
  Update value function V_ψ using TD(λ) with r_t^{FOCUS}
end for

minimizing a single full-trajectory OT objective. With this transformed reward, FOCUS uses the standard bootstrapped short-horizon actor-critic surrogate

$$J_{\mathrm{FOCUS}}(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{H-1} \gamma^t r_t^{\mathrm{FOCUS}} + \gamma^H V_\psi(s_H)\right]. \tag{13}$$

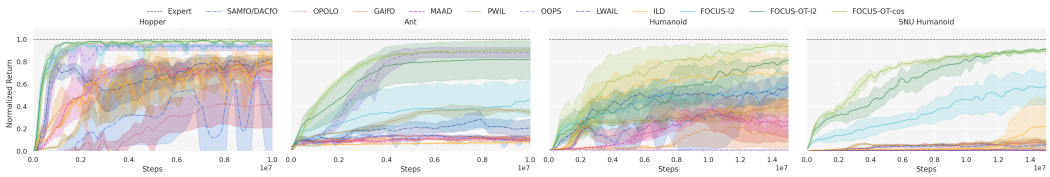
The surrogate in Algorithm 1 follows SHAC (Xu et al., 2022). Our main experiments use SAPO (Xing et al., 2025), while SHAC is evaluated in the backbone ablation. For simplicity, Algorithm 1 omits SAPO's entropy terms and twin-critic aggregation, which are described in Appendix E.1.

5 Experiments

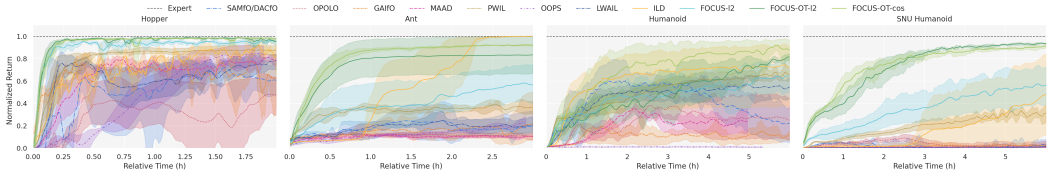
We evaluated FOCUS on a suite of continuous control environments implemented in DFlex, a differentiable physics simulator: Hopper, Ant, Humanoid, and the challenging SNU Humanoid. Unless otherwise stated, all experiments use 8 expert trajectories containing state-only observations, without expert actions or rewards, collected from policies trained using SAPO (Xing et al., 2025). Our main FOCUS experiments use SAPO and train from scratch with its standard random initialization, with no behavior-cloning pretraining, expert actions, expert-aligned initialization, or teacher forcing. We compare against the first-order and zeroth-order imitation-from-observation baselines described in Section 5.1, reporting sample efficiency and wall-clock time. We report the expert-normalized return $R_{\text{norm}} = R_\pi / R_E$, where R_π is the undiscounted episodic return of the evaluated policy under the environment's task reward (used only for evaluation, never for training) and R_E is the corresponding average return of the expert policy that generated the demonstrations.

5.1 Baselines

Among first-order approaches, our ILD baseline uses the Chamfer-α loss of Chen et al. (2023) on state trajectories in a differentiable simulator and is trained from a randomly initialized policy, with no behavior-cloning (BC) pretraining. Although originally proposed for imitation-from-demonstration, this loss operates solely on states and is compatible with our setup. We also remove the $\tanh$ transformation, which suppressed gradients in this setting. DiffMimic (Ren et al., 2023) minimizes a full-trajectory ℓ_2 loss but relies on expert-aligned initialization and teacher forcing. We therefore exclude it from this setting. Our FOCUS-L2 variant instead replaces OT with direct local ℓ_2 matching while retaining short-horizon, best-of-K training.



(a) Normalized return as a function of the number of environment interactions



(b) Normalized return as a function of wall-clock time

Figure 4: Expert-normalized return of FOCUS variants and baselines over both environment interactions and wall-clock training time. Curves show the mean across six seeds, and shaded regions show bootstrap 95% confidence intervals.

Among zeroth-order baselines, we evaluate SAM/DAC (Blondé & Kalousis, 2019; Kostrikov et al., 2018), adapted to ILfO with a state–state discriminator (SAMfO/DACfO), and OPOLO (Zhu et al., 2020), which adds inverse dynamics regularization. We include three Wasserstein-based methods: PWIL (Dadashi et al., 2021), using greedy transport matching of state-transition pairs without expert actions; OOPS (Chang et al., 2024), matching state-transition distributions; and LWAIL (Yang et al., 2026), learning an adversarial reward in a pretrained dynamics-aware representation. The on-policy baselines are GAIfO (Torabi et al., 2019) and MAAD (Candido Ramos et al., 2024), the latter also using inverse dynamics regularization.

The baselines and FOCUS variants use MINERAL (Xing et al., 2025) whenever possible. All methods except OOPS share policy and critic hidden-layer widths per environment (Appendix B.4).

5.2 FOCUS VARIANTS

We evaluated three variants of FOCUS: **FOCUS-L2** uses the same implementation as FOCUS-OT-L2, replacing OT with direct local ℓ_2 matching of state–state pairs. It retains expert-data normalization, best-of-K aggregation, exponential reward shaping, and the SAPO actor–critic, providing a control for OT within the L2 configuration. **FOCUS-OT-L2** uses an OT loss with an ℓ_2 ground cost on state–state pairs normalized using expert-data statistics. Reward shaping is $\exp(-\mathcal{L}_t^{\mathrm{OT}})$. **FOCUS-OT-Cos** uses an OT loss with a cosine ground cost on state–state pairs normalized using expert-data statistics. Reward shaping is $-\log(1 - e^{-\mathcal{L}_t^{\mathrm{OT}}} + \epsilon)$. Both transformations act on the aggregated per-step cost from Eq. 11. The two OT variants are composite configurations that differ in distance metric and reward transformation; their comparison does not isolate any one of these factors. The ablation in Appendix D.1 compares reward transformations within each ground-cost configuration: with a critic, exponential shaping gives higher mean returns than logarithmic shaping for the L2 cost on three of four tasks, while logarithmic shaping gives higher mean returns for the cosine cost on all four. For the OT variants, we batch the K Sinkhorn computations; Appendix B.5 provides implementation details. Section D.2 shows this batched Sinkhorn cost remains a modest fraction of runtime in our main experiments.

5.3 RESULTS

The aggregate results in Fig. 1 and per-task curves in Fig. 4 show that a FOCUS OT variant attains the highest mean return at the final interaction budget in every environment. Detailed endpoint statistics and confidence intervals are reported in Appendix C. On **Hopper**, all FOCUS variants and OOPS approach expert performance within the interaction budget, but FOCUS learns substantially faster in wall-clock time. On **Ant**, FOCUS-OT-Cos attains the highest mean return within the interaction budget. OOPS is competitive with the OT variants in environment interactions, with overlapping

endpoint confidence intervals, but learns much more slowly in wall-clock time. ILD attains the highest final wall-clock mean. On **Humanoid**, FOCUS-OT-Cos attains the highest mean return within both budgets, followed by FOCUS-OT-L2 and ILD. Their wall-clock confidence intervals overlap. On the **SNU Humanoid**, a contact-rich musculoskeletal model, the OT variants approach expert performance and attain substantially higher mean returns than the baselines within both budgets. All zeroth-order baselines remain far below expert performance, plausibly because the SNU Humanoid's 152 redundant muscle activations make credit assignment from state-only signals difficult without the local dynamics derivatives exploited by first-order methods. More generally, on-policy methods such as GAIfO and MAAD learn only suboptimal policies even on Hopper. At the interaction-budget endpoint, both OT variants attain higher mean returns than FOCUS-L2 in all four environments. These results support the selected OT configurations over direct local matching. The relative strengths of the two OT configurations are environment-dependent, and cannot be attributed solely to their distance metrics because reward shaping also differs.

5.4 Ablations, Runtime, and Diagnostics

We compare FOCUS with zeroth-order variants (ZO-FOCUS) that use SAC or PPO to optimize the same imitation reward without differentiating through the simulator (Appendix E.2). Both first-order OT variants attain higher mean returns than their corresponding ZO-FOCUS variants on all four tasks at matched interaction budgets. This comparison supports first-order training but changes the full actor–critic algorithm, so it does not isolate the effect of simulator derivatives. Further backbone and data ablations favor SAPO over SHAC and show that FOCUS remains effective with one expert trajectory (Appendix E).

We also evaluate the effect of the critic under several reward transformations (Appendix D.1). Including the critic improves mean return on all four tasks for FOCUS-OT-Cos and on three for FOCUS-OT-L2; the exception is Ant, where the intervals overlap. Both configurations benefit substantially on the humanoid tasks.

We explore how the number of sampled expert segments affects policy learning (Appendix E.3) and report the time spent on Sinkhorn matching as a percentage of total epoch time (Appendix D.2).

6 Conclusion

We introduced FOCUS, a first-order framework for imitation from state-only demonstrations in differentiable simulators. FOCUS combines local optimal-transport alignment, soft best-of-K segment matching, and short-horizon actor–critic learning to handle temporal misalignment and propagate imitation credit beyond the matching window, without expert actions, adversarial training, or pretraining. Across four contact-rich DFlex locomotion tasks, at least one FOCUS OT variant achieves the highest mean return among the compared methods at the final interaction budget, with near-expert performance on the complex musculoskeletal SNU Humanoid. The method also remains effective with a single expert trajectory. These results support combining local transport-based supervision with simulator gradients for learning control from temporally unaligned observations.

Limitations and future work. FOCUS assumes a known differentiable simulator; extending it to a learned world model, which would inject model-Jacobian error into the policy update, is left to future work. Our validation is restricted to periodic locomotion in DFlex, leaving manipulation, deformable-object control, sparse-goal and non-periodic tasks, and real-world transfer open. The best-of-K objective is also local rather than global, and can underrepresent rare phases or favor fragmentary matches when candidates are unrepresentative. Future work should broaden the task suite and explore alignment mechanisms that enforce long-horizon consistency.

AI USE STATEMENT

In this work, we used generative AI tools to aid writing, literature retrieval and discovery, and research ideation and execution, including method implementation, code review, checking mathematical derivations, and interpreting experimental results. All AI-assisted content was reviewed and verified by the authors, who take full responsibility for the final implementation, results, claims, and contents of this paper.

REPRODUCIBILITY STATEMENT

The main text specifies the objective and training procedure. Appendix B reports environment and expert-data specifications, compute resources, network architectures, optimization hyperparameters, OT marginals, Sinkhorn regularization and iteration count, transition normalization, and expert-segment sampling. Appendices C–E provide per-environment results and diagnostics. An anonymized code archive is provided as supplementary material alongside the submission. The archive includes the implementation of the proposed method as well as the configuration and launch files required to reproduce the reported experiments.

REFERENCES

Joseph Amigo, Rooholla Khorrambakht, Elliot Chane-Sane, Nicolas Mansard, and Ludovic Righetti. First order model-based rl through decoupled backpropagation. In *Proceedings of the 9th Conference on Robot Learning (CoRL)*, 2025.

Michael Bain and Claude Sammut. A framework for behavioural cloning. *Machine Intelligence 15*, 1995.

Nir Baram, Oron Anschel, and Shie Mannor. Model-based adversarial imitation learning. In *Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2016.

Lionel Blondé and Alexandros Kalousis. Sample-efficient imitation learning via generative adversarial nets. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

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

Joao Candido Ramos, Lionel Blondé, Naoya Takeishi, and Alexandros Kalousis. Mimicking better by matching the approximate action distribution. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Wei-Di Chang, Scott Fujimoto, David Meger, and Gregory Dudek. Imitation learning from observation through optimal transport. *Reinforcement Learning Journal (RLJ)*, 4:1911–1923, 2024.

Siwei Chen, Xiao Ma, and Zhongwen Xu. Imitation learning as state matching via differentiable physics. In *Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Samuel Cohen, Brandon Amos, Marc Peter Deisenroth, Mikael Henaff, Eugene Vinitsky, and Denis Yarats. Imitation learning from pixel observations for continuous control. In *Proceedings of the 35th Annual Conference on Neural Information Processing Systems (NeurIPS): Workshops*, 2021a.

Samuel Cohen, Giulia Luise, Alexander Terenin, Brandon Amos, and Marc Peter Deisenroth. Aligning time series on incomparable spaces. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021b.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2013.

Robert Dadashi, Léonard Hussenot, Matthieu Geist, and Olivier Pietquin. Primal wasserstein imitation learning. *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.

Jonas Degrave, Michiel Hermans, Joni Dambre, and Francis Wyffels. A differentiable physics engine for deep learning in robotics. *Frontiers in Neurorobotics*, 13, 2019.

Branton DeMoss, Paul Duckworth, Nick Hawes, and Ingmar Posner. Ditto: Offline imitation learning with world models. *arXiv*, abs/2302.03086, 2023.

Arnaud Fickinger, Samuel Cohen, Stuart J. Russell, and Brandon Amos. Cross-domain imitation learning via optimal transport. *arXiv*, abs/2110.03684, 2021.

C. Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax-a differentiable physics engine for large scale rigid body simulation. In *Proceedings of the 35th Annual Conference on Neural Information Processing Systems (NeurIPS): Datasets and Benchmarks Track*, 2021.

Yuwei Fu, Haichao Zhang, Di Wu, Wei Xu, and Benoit Boulet. Robot policy learning with temporal optimal transport reward. In *Proceedings of the 38th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

Levi Fussell, Kevin Bergamin, and Daniel Holden. Supertrack: motion tracking for physically simulated characters using supervised learning. *ACM Transactions on Graphics*, 40(6):1–13, 2021.

Ignat Georgiev, Krishnan Srinivasan, Jie Xu, Eric Heiden, and Animesh Garg. Adaptive horizon actor-critic for policy learning in contact-rich differentiable simulation. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Sean Gillen and Katie Byl. Leveraging reward gradients for reinforcement learning in differentiable physics simulations. *arXiv*, abs/2203.02857, 2022.

David R. Ha and Jürgen Schmidhuber. World models. *arXiv*, abs/1803.10122, 2018.

Danijar Hafner, Timothy P. Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020.

Siddhant Haldar, Vaibhav Mathur, Denis Yarats, and Lerrel Pinto. Watch and match: Supercharging imitation with regularized optimal transport. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2016.

Philipp Holl, Vladlen Koltun, and Nils Thuerey. Scale-invariant learning by physics inversion. In *Proceedings of the 36th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Bo-Ruei Huang, Chun-Kai Yang, Chun-Mao Lai, Dai-Jie Wu, and Shao-Hua Sun. Diffusion imitation from observation. In *Proceedings of the 38th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

Zhiao Huang, Yuanming Hu, Tao Du, Siyuan Zhou, Hao Su, Joshua B Tenenbaum, and Chuang Gan. Plasticinelab: A soft-body manipulation benchmark with differentiable physics. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.

William Huey, Huaxiaoyue Wang, Anne Wu, Yoav Artzi, and Sanjiban Choudhury. Imitation learning from a single temporally misaligned video. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

Victor Kolev, Rafael Rafailov, Kyle Hatch, Jiajun Wu, and Chelsea Finn. Efficient imitation learning with conservative world models. *Proceedings of Machine Learning Research*, 242:1776–1789, 2024.

Ilya Kostrikov, Kumar Krishna Agrawal, Debidatta Dwibedi, Sergey Levine, and Jonathan Tompson. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. *arXiv*, abs/1809.02925, 2018.

Seunghwan Lee, Moonseok Park, Kyoungmin Lee, and Jehee Lee. Scalable muscle-actuated human simulation and control. *ACM Transactions on Graphics*, 38(4):1–13, 2019.

Shangzhe Li, Zhiao Huang, and Hao Su. Reward-free world models for online imitation learning. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

Zechu Li, Tao Chen, Zhang-Wei Hong, Anurag Ajay, and Pulkit Agrawal. Parallel q-learning: Scaling off-policy reinforcement learning under massively parallel simulation. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

Xiao Ma, Siwei Chen, David Hsu, and Wee Sun Lee. Contrastive variational reinforcement learning for complex observations. In *Proceedings of the 4th Conference on Robot Learning (CoRL)*, 2020.

Miles Macklin. Warp: A high-performance python framework for gpu simulation and graphics. `https://github.com/nvidia/warp`, March 2022. NVIDIA GPU Technology Conference (GTC).

Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte carlo gradient estimation in machine learning. *The Journal of Machine Learning Research*, 21(1):5183–5244, 2020.

Michael C. Mozer. A focused backpropagation algorithm for temporal pattern recognition. *Backpropagation: theory, architectures, and applications*, pp. 137–169, 1995.

Rhys Newbury, Jack Collins, Kerry He, Jiahe Pan, Ingmar Posner, David Howard, and Akansel Cosgun. A review of differentiable simulators. *IEEE Access*, 12:97581–97604, 2024.

Georgios Papagiannis and Yunpeng Li. Imitation learning with sinkhorn distances. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, 2022.

Gabriel Peyré, Marco Cuturi, and Justin M. Solomon. Gromov-wasserstein averaging of kernel and distance matrices. In *Proceedings of the 33th International Conference on Machine Learning (ICML)*, 2016.

Quang Anh Pham, Janaka Brahmanage, Tien Mai, and Akshat Kumar. Iostom: Offline imitation learning from observations via state transition occupancy matching. In *Proceedings of the 39th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

Dean Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3(1):88–97, 1991.

Yongtao Qu, Shangzhe Li, and Weitong Zhang. Imitation from observations with trajectory-level generative embeddings. *arXiv*, abs/2601.00452, 2026.

Rafael Rafailov, Tianhe Yu, Aravind Rajeswaran, and Chelsea Finn. Visual adversarial imitation learning using variational models. In *Proceedings of the 35th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Jiawei Ren, Cunjun Yu, Siwei Chen, Xiao Ma, Liang Pan, and Ziwei Liu. Diffmimic: Efficient motion mimicking with differentiable physics. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023.

Thomas Rupf, Marco Bagatella, Nico Gürtler, Jonas Frey, and Georg Martius. Zero-shot offline imitation learning via optimal transport. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

Jürgen Schmidhuber. Making the world differentiable: on using self supervised fully recurrent neural networks for dynamic reinforcement learning and planning in non-stationary environments. *Forschungsberichte, TU Munich*, FKI 126 90:1–26, 1990.

Harshit Sikchi, Caleb Chuck, Amy Zhang, and Scott Niekum. A dual approach to imitation learning from observations with offline datasets. In *Proceedings of the 8th Conference on Robot Learning (CoRL)*, Proceedings of Machine Learning Research, 2024.

H. J. Terry Suh, Max Simchowitz, K. Zhang, and Russ Tedrake. Do differentiable simulators give better policy gradients? In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.

Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3: 9–44, 1988.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

Faraz Torabi, Garrett Warnell, and Peter Stone. Generative adversarial imitation from observation. *arXiv*, abs/1807.06158, 2019.

Paul J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

Ronald J. Williams and David Zipser. Gradient-based learning algorithms for recurrent networks and their computational complexity. In *Backpropagation: theory, architectures, and applications*, pp. 433–486. Psychology Press, 1995.

Eliot Xing, Vernon Luk, and Jean Oh. Stabilizing reinforcement learning in differentiable multiphysics simulation. In *Proceedings of the 13th International Conference on Learning Representations (ICLR)*, 2025.

Jie Xu, Viktor Makoviychuk, Yashraj Narang, Fabio Ramos, Wojciech Matusik, Animesh Garg, and Miles Macklin. Accelerated policy learning with parallel differentiable simulation. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, 2022.

Siqi Yang, Kai Yan, Alexander G. Schwing, and Yu-Xiong Wang. Latent wasserstein adversarial imitation learning. In *Proceedings of the 14th International Conference on Learning Representations (ICLR)*, 2026.

Miguel Angel Zamora Mora, Momchil Peychev, Sehoon Ha, Martin Vechev, and Stelian Coros. Pods: Policy optimization via differentiable simulation. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

Yaofeng Desmond Zhong, Jiequn Han, and Georgia Olympia Brikis. Differentiable physics simulations with contacts: Do they have correct gradients w.r.t. position, velocity and control? In *Proceedings of the 39th International Conference on Machine Learning (ICML): 2nd AI4Science Workshop*, 2022.

Zhuangdi Zhu, Kaixiang Lin, Bo Dai, and Jiayu Zhou. Off-policy imitation learning from observations. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

A Extended Related Work

Imitation Learning. Classical imitation learning from demonstration (ILfD) assumes expert state-action trajectories, often using behavior cloning (BC) (Pomerleau, 1991; Bain & Sammut, 1995). BC frames imitation as supervised prediction, but it suffers from compounding errors under covariate shift (Ho & Ermon, 2016). Interactive methods and inverse reinforcement learning address this by enabling policy corrections or reward inference. Generative Adversarial Imitation Learning (GAIL) (Ho & Ermon, 2016) and off-policy variants such as SAM/DAC (Blondé & Kalousis, 2019; Kostrikov et al., 2018) instead match trajectory distributions through adversarial training. Imitation learning from observations (ILfO) removes the need for expert actions and learns from state sequences alone. Early ILfO methods introduced strategies to cope with missing action information. For example, BCO (Torabi et al., 2018) first learns an inverse dynamics model to infer actions from state transitions and then applies BC to the inferred action labels. Adversarial ILfO methods match state or state-transition distributions directly from observations (Torabi et al., 2019). OPOLO (Zhu et al., 2020) and MAAD (Candido Ramos et al., 2024) strengthen this objective with regularization toward the inferred expert actions. DIFO (Huang et al., 2024) combines conditional diffusion modeling of state transitions with a classification objective to learn an imitation reward. It addresses the robustness of adversarial reward learning while retaining an expert-versus-agent discriminator. These methods operate without simulator derivatives; FOCUS uses those derivatives to optimize a local alignment-based imitation signal directly.

Offline imitation from observations. Offline ILfO trains from fixed datasets without collecting new environment interactions during policy learning. DILO (Sikchi et al., 2024) learns from expert observations and suboptimal offline interactions through a dual formulation. IOSTOM (Pham et al., 2025) matches state-transition occupancies using expert state trajectories and separate non-expert transitions with actions, then infers actions without training an inverse dynamics model. TGE (Qu et al., 2026) constructs rewards from temporal diffusion embeddings learned on offline trajectories. ZILOT (Rupf et al., 2025) learns a world model and a goal-conditioned value function offline for subsequent zero-shot imitation. These approaches address a different data-access setting from our online benchmark, in which the learner collects its own experience in a known differentiable simulator. We therefore discuss these methods as related work but exclude them from our empirical comparisons, which focus on online policy learning.

Differentiable Dynamics and First-Order Model-Based RL. Differentiable physics simulators have become powerful tools for RL by enabling analytical gradients through environment dynamics and reward functions. Simulators such as Brax (Freeman et al., 2021), DFlex (Xu et al., 2022), and Warp (Macklin, 2022) provide scalable differentiable environments for rigid-body systems; for a broader overview, see (Newbury et al., 2024). First-order model-based RL (FO-MBRL) algorithms use path-wise derivatives rather than high-variance, black-box estimators (Mozer, 1995; Zamora Mora et al., 2021). SHAC (Xu et al., 2022) stabilizes learning through short-horizon critic-guided rollouts, while AHAC (Georgiev et al., 2024) adapts each rollout horizon based on gradient norm, and SAPO (Xing et al., 2025) improves robustness through entropy regularization and critic shaping. DMO (Amigo et al., 2025) extends first-order optimization to non-differentiable simulators by learning a surrogate model used only during the backward pass. A related line of work learns forward or world models from data and backpropagates through them for policy updates (Schmidhuber, 1990; Ha & Schmidhuber, 2018; Hafner et al., 2020; Ma et al., 2020). These advances motivated extending first-order formulations to imitation learning, where gradients from differentiable simulators or models can be combined with trajectory-level supervision to efficiently acquire complex behaviors.

Imitation Learning via Optimal Transport. An alternative to adversarial or BC approaches frames imitation as a distribution alignment between agent and expert trajectories. Optimal transport (OT) defines such alignment costs through couplings between trajectory distributions. PWIL (Dadashi et al., 2021) derives surrogate rewards from the Wasserstein-1 distance between state-action occupancy measures, SIL (Papagiannis & Li, 2022) applies Sinkhorn distances (Cuturi, 2013), GDTW-IL (Cohen et al., 2021b) uses Gromov dynamic time warping for temporal misalignment, and GWIL (Fickinger et al., 2021) leverages Gromov-Wasserstein distances (Peyré et al., 2016) for cross-domain imitation. ROT (Haldar et al., 2022) adds a behavior cloning regularizer, and OOPS (Chang et al., 2024) computes Wasserstein distances between state-only trajectories as dense rewards for off-policy

learning. LWAIL (Yang et al., 2026) learns a dynamics-aware metric through intention-conditioned value-function pretraining on random state observations, then freezes the representation for online adversarial Wasserstein imitation with model-free policy updates. PWIL, SIL, GWIL, ROT, OOPS, and LWAIL support black-box policy learning, whereas GDTW-IL differentiates its trajectory-alignment objective through a known transition model. FOCUS combines local OT matching with short differentiable rollouts and critic bootstrapping.

Temporal alignment and ordering. TemporalOT (Fu et al., 2024) uses temporally contextualized costs and a temporal mask to constrain transport-based reward assignment. ORCA (Huey et al., 2025) formulates imitation rewards around covering demonstration frames in order, addressing failures of frame-level matching under temporal misalignment. These objectives emphasize an aspect distinct from local phase selection: matching a compatible short segment does not ensure that the policy covers all demonstration subgoals in the correct order. Best-of-K selection in FOCUS therefore does not resolve the global ordering and coverage problems studied by these works, which remain relevant limitations for non-periodic tasks.

Imitation Learning via Differentiable Dynamics/Physics. Combining imitation learning with differentiable simulation enables direct optimization of trajectory similarity between agent and expert. DiffMimic (Ren et al., 2023) minimizes a ℓ_2 loss between agent and expert state trajectories using a differentiable simulator, but assumes expert-aligned initialization and teacher forcing. ILD (Chen et al., 2023) uses Chamfer-α matching without state synchronization, combines deviation and coverage terms, and trains with truncated backpropagation after behavior-cloning pretraining. Similarly to FO-MBRL, IL methods can also leverage learned forward models to optimize policies with gradient-based objectives when a differentiable simulator is unavailable (Baram et al., 2016; Fussell et al., 2021; Rafailov et al., 2021; DeMoss et al., 2023; Kolev et al., 2024; Li et al., 2025).

FOCUS. FOCUS targets ILfO from state-only demonstrations in differentiable simulators, using path-wise gradients through short rollouts rather than discriminator-based reward learning or zeroth-order policy updates. Relative to trajectory-level GDTW or Chamfer matching, FOCUS combines OT alignment over sampled expert segments with soft best-of-K aggregation and critic bootstrapping. Soft best-of-K aggregation provides differentiable segment selection, making the imitation objective robust to temporal misalignment while keeping backpropagation local enough for contact-rich dynamics. This yields a practical first-order ILfO framework for differentiable simulators.

B EXPERIMENTAL SPECIFICATIONS

B.1 ENVIRONMENTS

We evaluated our method in four continuous control environments implemented in DFLEX (Xu et al., 2022), each designed to test different aspects of learning from state-only expert demonstrations.

Hopper: A planar, single-legged robot is tasked with hopping forward as quickly as possible.

Ant: A quadrupedal agent must learn to walk forward efficiently using four independently actuated legs.

Humanoid: A full-body humanoid robot is trained to run forward. This is the most complex OpenAI Gym-style (Brockman et al., 2016) task in our suite, featuring high-dimensional observations of 76 values and 21 control inputs.

SNU Humanoid: A musculoskeletal model of a humanoid lower body, originally introduced by Lee et al. in 2019, actuated by 152 muscle-tendon units (MTUs). Each MTU provides a single control signal that regulates muscle contraction forces. This environment poses the most challenging dynamics and control structure in our benchmark.

Details about the dimensionalities of observation and action spaces, as well as expert data used for training, are summarized in Table 1.

Environment	Observation Space $\mathcal{S}$	Action Space $\mathcal{A}$	Expert Trajectories # Trajectories $\times$ Length
Hopper	$\mathbb{R}^{11}$	$\mathbb{R}^{3}$	8×1000
Ant	$\mathbb{R}^{37}$	$\mathbb{R}^{8}$	8×1000
Humanoid	$\mathbb{R}^{76}$	$\mathbb{R}^{21}$	8×1000
SNU Humanoid	$\mathbb{R}^{53}$	$\mathbb{R}^{152}$	8×1000

Table 1: Specifications of DFLEX environments and expert data used in the reported experiments.

Table 2 reports the expert return statistics for each environment, providing an absolute performance reference alongside the expert-normalized results.

Environment	Mean return	Standard deviation
Hopper	4812.97	7.32
Ant	9329.51	37.59
Humanoid	8225.22	81.05
SNU Humanoid	6498.92	49.13

Table 2: Expert returns in the original environment reward units. Values are rounded to two decimal places; standard deviations describe return variability, not confidence intervals.

B.2 IMPLEMENTATION DETAILS

We implement FOCUS on top of MINERAL[1], a minimal yet high-performance reinforcement learning library that provides reliable implementations of modern model-based algorithms such as SHAC (Xu et al., 2022) and SAPO (Xing et al., 2025). MINERAL also includes efficient implementations of standard model-free RL algorithms, validated for parallel simulation (Li et al., 2023).

We implement all baselines within MINERAL, using their original implementations as references. For PPO, we implement GAIL/GAIfO (Ho & Ermon, 2016; Torabi et al., 2019) and MAAD (Candido Ramos et al., 2024); for SAC, we implement SAM/DAC (Blondé & Kalousis, 2019; Kostrikov et al., 2018), and OPOLO (Zhu et al., 2020). We additionally incorporate ILD (Chen et al., 2023) and our own FOCUS framework, using both SHAC (Xu et al., 2022) and SAPO (Xing et al., 2025) as the

[1] https://github.com/etaoxing/mineral

underlying reinforcement learning algorithms. In all reported experiments, ILD denotes the state-only Chamfer-α loss trained from a randomly initialized policy, without BC pretraining, as described in Section 5.1. PWIL also uses SAC, with greedy matching over state-transition pairs rather than expert actions. LWAIL and OOPS use DDPG backbones with TD3-style updates; LWAIL matches $(s, s' - s)$ features, while OOPS matches (s, s') pairs. Tables 7 and 8 report their settings.

B.3 Compute Resources

All reported experiments were run on a shared compute cluster. Each training job corresponds to one configuration, specified by method, environment and random seed, and requests one GPU with at least 8 GB of VRAM, one CPU core, and 8 GB of system RAM. The cluster pool is heterogeneous and includes NVIDIA RTX 3090 24 GB, RTX A5000 24 GB, RTX A6000 48 GB, and A100 PCIe 40/80 GB GPUs; the reported experiments used RTX 3090 GPUs because they were the most available workers. Each main result is averaged over six such independent seed jobs. Per-run wall-clock requirements are reported by the time-based learning curves in Fig. 4 and Fig. 7, and ablations use the same one-job-per-configuration execution model.

B.4 Hyperparameters

We aim to keep the hyperparameters consistent across methods whenever possible. All methods except OOPS share policy and critic hidden-layer widths per environment. OOPS uses its reference-implementation widths; activation functions, normalization, critic counts, and optimization settings can also differ by algorithm (Tables 3 and 7). For RL-specific parameters, we closely follow those used in the SAPO paper (Xing et al., 2025), with adaptations for each underlying algorithm as necessary.

Table 3 summarizes the shared and algorithm-specific hyperparameters used across SHAC, SAPO, PPO, and SAC. Table 4 provides the multilayer perceptron (MLP) architectures used per environment. Tables 5 and 6 report imitation objectives, architectures, and selected per-environment settings. Tables 7 and 8 give the policy-learning and imitation settings for PWIL, LWAIL, and OOPS.

For discriminator-based methods, we subsample expert transitions to reduce temporal redundancy, following the expert-data subsampling practice used in GAIL (Ho & Ermon, 2016). With factor q and a random initial offset u, we retain one-step pairs (s_t, s_{t+1}) at indices $t = u + nq$, provided both states lie within the same trajectory. This reduces the number of training pairs while preserving their one-step temporal spacing.

B.5 OT Implementation Details

We report the following implementation details for the OT variants of FOCUS, complementing Section 4 and Section 5.1:

- Transport marginals assign mass $1/H$ to each agent state transition and $1/L$ to each expert state transition.
- We use entropy-regularized Sinkhorn with regularization $\varepsilon = 0.05$, run for 100 Sinkhorn iterations.
- Candidate weights use a fixed temperature $\beta = 0.5$, with no schedule.
- The segment score is the transport-only cost $\ell_k = \langle \mu^{\star(k)}, C^{(k)} \rangle = \sum_{t=0}^{H-1} \mathcal{L}_t^{(k)}$, without an entropy term or further averaging over timesteps.
- Agent and expert state-transition features use the same coordinatewise normalization, $\widetilde{s}_j = (s_j - \mu_{E,j})/\sqrt{\sigma_{E,j}^2 + \epsilon}$, where $\mu_{E,j}$ and $\sigma_{E,j}^2$ are the mean and variance of feature j in the expert dataset, and $\epsilon > 0$ is a numerical stabilizer.
- For agent and expert state-transition feature vectors s and s^E, both L2 variants use mean squared error (MSE): $c_{\mathrm{MSE}}(s, s^E) = \frac{1}{d} \sum_{j=1}^{d} (\widetilde{s}_j - \widetilde{s}_j^E)^2$, where d is the feature dimension. This is the ground cost for FOCUS-OT-L2 and the direct matching cost for FOCUS-L2.
- Expert-segment start indices are sampled independently and uniformly, with replacement, over all valid start positions across the demonstration trajectories.

	Shared	PPO	SAC	SHAC	SAPO
Num envs N	64				
Batch size	2048				
Horizon H	32				
Mini-epochs K	–	5	8	16	16
Discount γ	0.99				
TD/GAE λ	–	0.95	–	0.95	0.95
Actor LR η	–	5e-4	5e-4	2e-3	2e-3
Critic LR η	–	5e-4	5e-4	2e-3	5e-4
Entropy LR η	–	–	5e-3	–	5e-3
Learning-rate schedule	–	KL$(0.008)^*$	–	linear	linear
Optimizer type	AdamW			Adam	
Optimizer (β_1, β_2)	(0.9, 0.999)			(0.7, 0.95)	(0.7, 0.95)
Grad clip	0.5			1.0	
Norm type	LayerNorm				
Activation type	SiLU			ELU	
State-dependent σ	yes			no	
Log std range	–	$\log[0.1, 1.0]$	$[-5, 2]$	–	$[-5, 2]$
Num critics C	–		2	1	2
Target update rate τ	–		0.005	–	
SHAC target coefficient	–			0.2	
Replay buffer size	–		10^6	–	
Target entropy	–		$-\dim(\mathcal{A})/2$	–	same
Initial temperature	–		1.0	–	1.0

*PPO uses an adaptive learning-rate schedule targeting a KL divergence of 0.008: the learning rate is divided by 1.5 when the measured KL exceeds twice the target and multiplied by 1.5 when it falls below half the target, within $[10^{-6}, 10^{-2}]$.

Table 3: Hyperparameters for underlying reinforcement learning algorithms. Shared values are used across all methods unless specified. Adapted from SAPO (Xing et al., 2025), with the SHAC column reporting the original-style configuration used in our backbone ablation. SAC's τ is the Polyak update rate (`sac.tau`); the SHAC target coefficient is the configuration value `target_critic_alpha`, listed separately. Settings for PWIL and the DDPG/TD3-based baselines LWAIL and OOPS are in Table 7.

	Hopper	Ant	Humanoid	SNU Humanoid
Actor	(128, 64, 32)	(128, 64, 32)	(256, 128)	(512, 256)
Critic	(64, 64)	(64, 64)	(128, 128)	(256, 256)

Table 4: Actor and critic MLP hidden-layer widths for each DFLEX environment. All algorithms share these widths per network, except OOPS, which uses the reference-implementation sizes (actor $(256, 256)$, critic $(1024, 1024)$, ReLU, no normalization) in every environment (Table 7). Algorithm-specific settings for PPO, SAC, SHAC, and SAPO, including the number of critics, are reported in Table 3.

	SAMfO/ DACfO	OPOLO	GAIfO	MAAD	ILD	FOCUS-L2	FOCUS-OT
Expert seg. length						32	32
Imitation loss	Disc	Disc	Disc	Disc	Chamfer-α	L2	OT-{L2, Cos}
Matching	(s, s')	(s, s')	(s, s')	(s, s')	s	(s, s')	(s, s')
Expert seg. count K						8	8
Discriminator	Critic†	Critic	Critic	Critic			
Inverse model	–	Critic	–	Critic			

Table 5: Imitation-learning objectives and architectures; selected per-environment settings are in Table 6. "Critic" denotes the critic architecture; †except SAMfO/DACfO on SNU Humanoid, whose discriminator uses $(128, 128)$ against a $(256, 256)$ critic. MAAD augments GAIfO with inverse-model regularization. Baseline settings were selected from a small grid over expert batch size, subsampling factor, discriminator update frequency and iterations, discriminator width, and inverse-model regularization. We use the same six random seeds (100, 110, 120, 1000, 1100, 1200) across all training experiments, including hyperparameter selection and ablations.

Method	Setting	Hopper	Ant	Humanoid	SNU Humanoid
SAMfO/DACfO	Disc. update freq.	5	5	10	10
	Expert batch size	512	512	512	512
	Subsampling factor	20	1	1	20
OPOLO	Disc. update freq.	5	5	5	5
	Expert batch size	512	512	512	512
	Subsampling factor	20	20	20	20
	Inverse-model reg.	10	1	1	10
	α	0.1	0.1	1	0.1
GAIfO	Disc. update freq.	10	5	10	10
	Expert batch size	2048	512	512	1024
	Subsampling factor	1	1	1	1
MAAD	Disc. update freq.	5	5	5	5
	Expert batch size	2048	2048	2048	2048
	Subsampling factor	20	20	20	20
	Inverse-model reg.	10	10	10	10

Table 6: Selected hyperparameters for the discriminator-based imitation baselines in each environment. For these methods, subsampling expert trajectories often improved performance. SAMfO/DACfO and GAIfO have no inverse-model regularizer. Table 5 reports the architectures and tuning-seed protocol.

	PWIL	LWAIL	OOPS
RL backbone	SAC	DDPG (TD3-style)	DDPG (TD3-style)
Actor widths	Table 4	Table 4	$(256, 256)$
Critic widths	Table 4	Table 4	$(1024, 1024)$
Activation / norm	SiLU / LayerNorm	SiLU / LayerNorm	ReLU / none
Actor LR	5e-4	3e-4	3e-4
Critic LR	5e-4	1e-3	3e-4
Batch size	2048	100	100
Target update rate τ	0.005	0.005	0.003
Horizon	32	1	1000 (full episode)
Exploration noise	–	0.1	0.2
Target noise / clip	–	0.2 / 0.5	0.1 / 0.5

Table 7: Policy-learning settings for PWIL, LWAIL, and OOPS. All use 64 environments, $\gamma = 0.99$, and the same 8 expert trajectories as the other baselines. Their imitation-specific settings are in Table 8.

	PWIL	LWAIL	OOPS
Agent–expert matching	(s, s') atoms	$(s, s' - s)$	(s, s') atoms
Reward	$5\exp(-\beta\,\text{cost})$, $\beta = 1/2/1/1$	Wasserstein critic	Sinkhorn, scale 5
Expert subsampling	20	1	1
Discriminator widths	–	Critic widths (Table 4)	–
Disc. optimizer / LR	–	Adam / 1e-3	–
Gradient penalty	–	10	–
Expert / policy batch	–	4000 / 4000	–
Disc. update frequency	–	63 epochs (≈ 4032 steps)	–
Disc. update iterations	–	40	–
Disc. pretraining	–	2500 iterations	–
ICVF encoder widths	–	$(256, 256)$	–
ICVF data / training	–	$\approx$10k random transitions / 20k training steps	–
Sinkhorn regularization	–	–	$\varepsilon = 0.05$
Sinkhorn iterations	–	–	1000

Table 8: Imitation and pretraining settings for PWIL, LWAIL, and OOPS. PWIL's β is given as Hopper/Ant/Humanoid/SNU Humanoid, selected from $\{1, 2, 3\}$ on the evaluation seeds. LWAIL's random-rollout ICVF data (approximately 9,984 transitions) are included in the plotted interaction budget. The plotted wall-clock time includes random-data collection, ICVF pretraining, and discriminator pretraining.

Segment weights and the actor surrogate. For fixed β, let $\bar{\ell} = \sum_k w_k \ell_k$, with w_k defined in Eq. 9. Differentiating this weighted cost through its weights gives

$$\nabla_\theta \bar{\ell} = \sum_k w_k \left[1 - \frac{\ell_k - \bar{\ell}}{\beta}\right] \nabla_\theta \ell_k. \tag{14}$$

It differs from the log-sum-exp gradient $\sum_k w_k \nabla_\theta \ell_k$: the latter is recovered if the weights in the weighted cost are detached. A high-cost candidate can have a negative coefficient in the former, since increasing its loss can reduce its weight. This identity concerns the scalar weighted segment cost; the actor further applies per-step reward shaping, discounting, and critic bootstrapping, with gradients propagated through the Sinkhorn iterations. SAPO additionally includes its entropy terms.

C PER-ENVIRONMENT BENCHMARK RESULTS

Figure 5 reports the aggregate comparison for all methods, complementing the selected methods shown in Fig. 1.

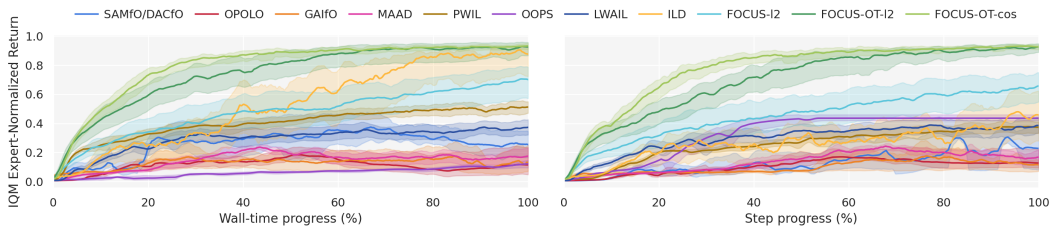


Figure 5: Aggregate performance of all methods across four DFlex tasks. Curves show the interquartile mean (IQM) of expert-normalized return; shading indicates 95% confidence intervals. Progress is expressed as a percentage of each task's wall-clock budget (left) or interaction budget (right), as specified in Tables 10 and 9.

Figures 6 and 7 provide enlarged per-environment views of the learning curves in Fig. 4 for environment interactions and wall-clock time. Tables 9 and 10 report the corresponding endpoint values. All entries are means with bootstrap 95% confidence intervals across six seeds. LWAIL interaction counts include initial random-data collection (approximately 9,984 transitions), and its wall-clock time includes data collection, ICVF pretraining, and discriminator pretraining (Table 8).

C.1 ENVIRONMENT-INTERACTION RESULTS

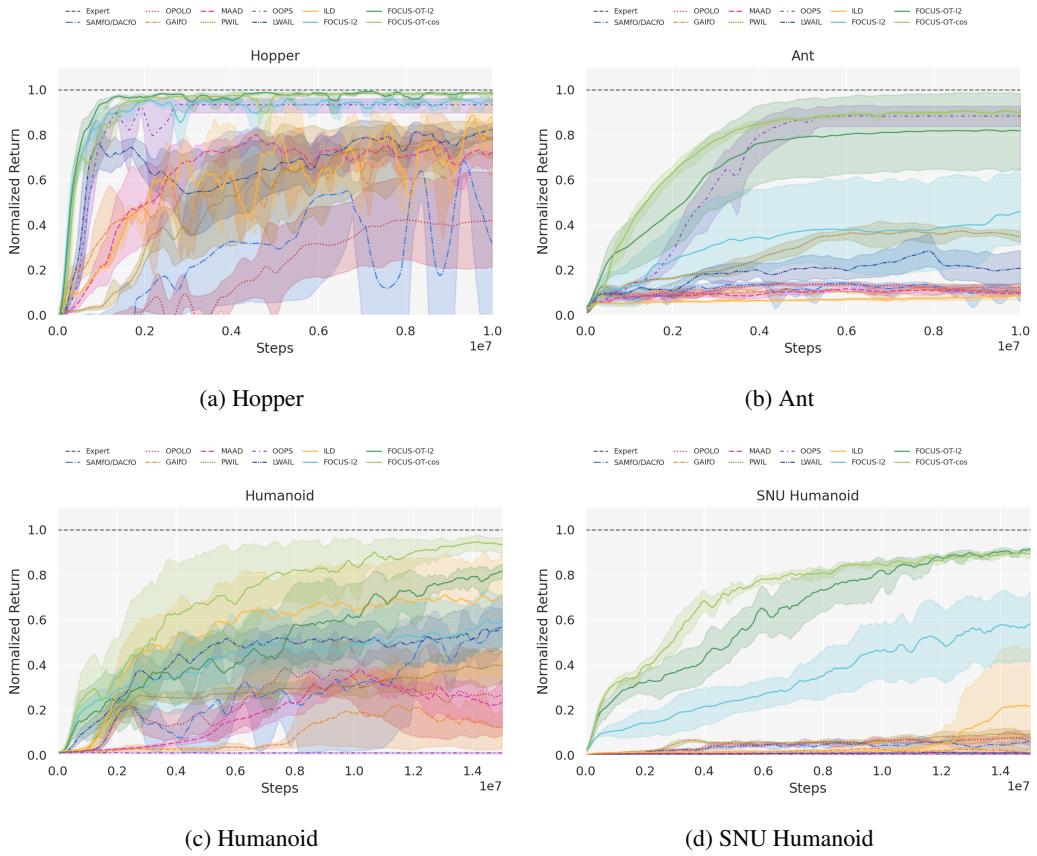
Method	Hopper	Ant	Humanoid	SNU Humanoid
SAMfO/DACfO	$0.320\ [-0.465, 0.720]$	$0.093\ [0.061, 0.120]$	$0.572\ [0.456, 0.711]$	$0.057\ [0.024, 0.089]$
OPOLO	$0.419\ [0.213, 0.634]$	$0.123\ [0.104, 0.140]$	$0.265\ [0.087, 0.459]$	$0.064\ [0.042, 0.081]$
GAIfO	$0.792\ [0.624, 0.881]$	$0.108\ [0.094, 0.133]$	$0.122\ [0.030, 0.191]$	$0.012\ [-0.006, 0.033]$
MAAD	$0.723\ [0.664, 0.796]$	$0.100\ [0.088, 0.109]$	$0.234\ [0.128, 0.332]$	$0.010\ [0.007, 0.013]$
PWIL	$0.778\ [0.697, 0.833]$	$0.349\ [0.323, 0.375]$	$0.398\ [0.317, 0.468]$	$0.093\ [0.073, 0.112]$
OOPS	$0.934\ [0.899, 0.961]$	$0.883\ [0.838, 0.927]$	$0.010\ [0.007, 0.013]$	$0.002\ [0.000, 0.004]$
LWAIL	$0.823\ [0.783, 0.851]$	$0.208\ [0.140, 0.288]$	$0.566\ [0.486, 0.649]$	$0.008\ [0.005, 0.012]$
ILD	$0.852\ [0.778, 0.983]$	$0.088\ [0.070, 0.106]$	$0.708\ [0.360, 0.894]$	$0.217\ [0.016, 0.478]$
FOCUS-L2	$0.951\ [0.938, 0.966]$	$0.459\ [0.315, 0.653]$	$0.596\ [0.479, 0.714]$	$0.582\ [0.419, 0.723]$
FOCUS-OT-L2	$\mathbf{0.988}\ [\mathbf{0.981}, \mathbf{0.994}]$	$0.818\ [0.647, 0.984]$	$0.816\ [0.784, 0.847]$	$\mathbf{0.912}\ [\mathbf{0.902}, \mathbf{0.920}]$
FOCUS-OT-Cos	$0.983\ [0.979, 0.986]$	$\mathbf{0.899}\ [\mathbf{0.884}, \mathbf{0.914}]$	$\mathbf{0.934}\ [\mathbf{0.895}, \mathbf{0.965}]$	$0.894\ [0.876, 0.912]$

Table 9: Expert-normalized return at the final interaction budget, reported as mean [bootstrap 95% confidence interval] across six seeds. Budgets are 10M interactions for Hopper and Ant and 15M for Humanoid and SNU Humanoid. The highest mean in each environment is bold.

At fixed interaction budgets, an OT variant attains the highest mean in every environment: FOCUS-OT-L2 on Hopper and SNU Humanoid, and FOCUS-OT-Cos on Ant and Humanoid. On Hopper, both OT variants approach expert return (0.988 and 0.983), followed by FOCUS-L2 (0.951) and OOPS (0.934). On Ant, FOCUS-OT-Cos reaches 0.899, followed by OOPS (0.883) and FOCUS-OT-L2 (0.818); their confidence intervals overlap. These three methods have higher means than FOCUS-L2 (0.459) and PWIL (0.349). On Humanoid, FOCUS-OT-Cos leads with 0.934, followed by FOCUS-OT-L2 (0.816) and ILD (0.708); LWAIL reaches 0.566, close to SAMfO/DACfO (0.572). On SNU Humanoid, the OT variants reach 0.912 and 0.894, compared with 0.582 for FOCUS-L2 and 0.217 for ILD. The three additional Wasserstein baselines show task-dependent performance: OOPS is strongest among them on Hopper and Ant, LWAIL on Humanoid, and PWIL on SNU Humanoid. OOPS remains near zero on both humanoid tasks.

The confidence intervals qualify these rankings. In particular, FOCUS-OT-L2 has a wide interval on Ant ($[0.647, 0.984]$), whereas FOCUS-OT-Cos has a higher mean and a tighter interval ($[0.884, 0.914]$). ILD's intervals are wide on Humanoid ($[0.360, 0.894]$) and SNU Humanoid

([0.016, 0.478]), indicating substantial uncertainty in its endpoint means on those tasks. These six-seed intervals should therefore be considered together with the means; they are uncertainty estimates rather than formal pairwise tests of superiority.



(a) Hopper

(b) Ant

(c) Humanoid

(d) SNU Humanoid

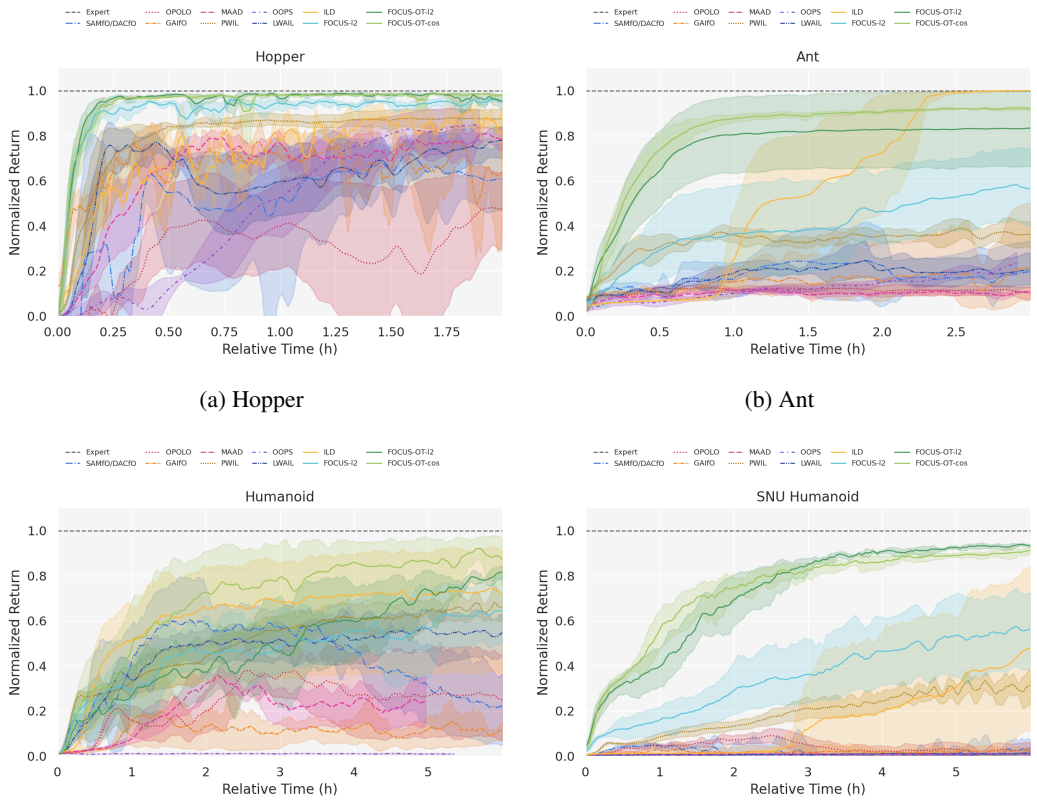
Figure 6: Expert-normalized return of FOCUS variants and baselines as a function of environment interactions. Curves show the mean across six seeds, and shaded regions show bootstrap 95% confidence intervals.

C.2 Wall-Clock Results

Method	Hopper	Ant	Humanoid	SNU Humanoid
SAMfO/DACfO	0.608 [0.470, 0.763]	0.198 [0.126, 0.333]	0.221 [0.047, 0.361]	0.034 [−0.002, 0.077]
OPOLO	0.475 [0.292, 0.634]	0.104 [0.075, 0.131]	0.251 [0.071, 0.448]	0.029 [0.002, 0.063]
GAIfO	0.639 [0.346, 0.845]	0.215 [0.071, 0.500]	0.119 [0.067, 0.154]	0.025 [0.007, 0.040]
MAAD	0.788 [0.740, 0.832]	0.107 [0.103, 0.111]	0.242 [0.131, 0.353]	0.010 [0.007, 0.013]
PWIL	0.876 [0.834, 0.918]	0.362 [0.305, 0.420]	0.661 [0.603, 0.720]	0.310 [0.226, 0.384]
OOPS	0.851 [0.780, 0.916]	0.234 [0.180, 0.284]	0.010 [0.008, 0.011]	0.002 [0.001, 0.004]
LWAIL	0.780 [0.700, 0.841]	0.206 [0.141, 0.281]	0.547 [0.448, 0.631]	0.014 [0.005, 0.024]
ILD	0.807 [0.657, 0.919]	**1.001** [**0.999**, **1.003**]	0.722 [0.366, 0.911]	0.479 [0.114, 0.844]
FOCUS-L2	0.956 [0.951, 0.961]	0.567 [0.401, 0.746]	0.646 [0.504, 0.806]	0.563 [0.389, 0.725]
FOCUS-OT-L2	0.952 [0.921, 0.981]	0.835 [0.667, 1.000]	0.817 [0.784, 0.845]	**0.935** [**0.928**, **0.941**]
FOCUS-OT-Cos	**0.985** [**0.982**, **0.988**]	0.920 [0.913, 0.927]	**0.872** [**0.754**, **0.971**]	0.914 [0.895, 0.930]

Table 10: Expert-normalized return at the final wall-clock budget, reported as mean [bootstrap 95% confidence interval] across six seeds. Budgets are 2 hours for Hopper, 3 hours for Ant, and 6 hours for Humanoid and SNU Humanoid. The highest mean in each environment is bold.

The wall-clock comparison is more mixed. On Hopper, FOCUS-OT-Cos attains the highest mean (0.985), followed by FOCUS-L2 (0.956) and FOCUS-OT-L2 (0.952). PWIL is the strongest baseline (0.876), followed by OOPS (0.851) and ILD (0.807). ILD leads on Ant with 1.001 $[0.999, 1.003]$, followed by FOCUS-OT-Cos (0.920) and FOCUS-OT-L2 (0.835); the latter has a substantially wider interval. Despite its competitive performance per environment interaction, OOPS learns much more slowly in wall-clock time on Hopper and Ant. On Ant, its mean is only 0.234 at three hours, compared with 0.883 at 10M interactions. On Humanoid, FOCUS-OT-Cos and FOCUS-OT-L2 attain higher means (0.872 and 0.817) than ILD (0.722), although their intervals overlap. On SNU Humanoid, both OT variants exceed all baseline means, with FOCUS-OT-L2 leading at 0.935 $[0.928, 0.941]$. PWIL attains the highest zeroth-order baseline means on Humanoid (0.661) and SNU Humanoid (0.310), while OOPS remains near zero on both. No method has a wall-clock endpoint mean above 0.9 in every environment.



(a) Hopper

(b) Ant

(c) Humanoid

(d) SNU Humanoid

Figure 7: Expert-normalized return of FOCUS variants and baselines as a function of wall-clock training time. Curves show the mean across six seeds, and shaded regions show bootstrap 95% confidence intervals.

D METHOD DIAGNOSTICS AND COMPUTATIONAL COST

This section examines the imitation objective and its computational behavior. We report reward-shaping and critic ablations, Sinkhorn runtime scaling, and qualitative OT transport-plan diagnostics.

D.1 REWARD SHAPING AND CRITIC ABLATION

We ablate the reward transformation in Eq. 12 and the critic to assess the design choices underlying FOCUS-OT-L2 and FOCUS-OT-Cos. Table 11 reports final expert-normalized returns, and Figure 8 shows the corresponding learning curves. We first compare reward transformations with the critic enabled, then remove the critic for each configuration. The suffixes `l2` and `cos` identify the OT ground cost; `exp`, `logexp`, and `neg` denote exponential, logarithmic, and untransformed negative-cost rewards, respectively.

For the L2 cost, exponential shaping (`exp-l2`) achieves higher mean returns than logarithmic shaping (`logexp-l2`) on three of four tasks: Ant (0.818 vs. 0.662), Humanoid (0.816 vs. 0.432), and SNU Humanoid (0.912 vs. 0.458). Its confidence intervals are also narrower in these three environments. The logarithmic transformation has a slightly higher mean on Hopper (0.993 vs. 0.988), where both are close to expert performance and their intervals overlap. With a critic, the untransformed negative-L2 reward yields near-zero returns on Ant, Humanoid, and SNU Humanoid. This comparison motivates the exponential reward used by FOCUS-OT-L2.

For the cosine cost, the logarithmic transformation (`logexp-cos`) outperforms exponential shaping in mean return on every task. The largest gains occur on Humanoid (0.934 vs. 0.363) and SNU Humanoid (0.894 vs. 0.273), where the confidence intervals are also substantially narrower. These results motivate the logarithmic reward used by FOCUS-OT-Cos and show that the choice of reward transformation should be made jointly with the cost.

Finally, removing the critic reduces mean return in all four environments for `logexp-cos` and in three of four for `exp-l2`. The exception is Ant, where `exp-l2` has a slightly higher mean without the critic (0.850 vs. 0.818), with overlapping confidence intervals. On Humanoid, `exp-l2` drops from 0.816 to 0.015 and `logexp-cos` from 0.934 to 0.329; on SNU Humanoid, they drop from 0.912 to 0.265 and from 0.894 to 0.583, respectively. These results support retaining the critic in both selected OT configurations, particularly on the humanoid tasks, while showing that its benefit is not universal. The effect also depends on reward parameterization: removing the critic increases mean return for `exp-cos` on Ant and SNU Humanoid and for `neg-l2` on all four tasks.

Method	Hopper	Ant	Humanoid	SNU Humanoid
critic-exp-l2	0.988 $[0.981, 0.994]$	0.818 $[0.647, 0.984]$	0.816 $[0.784, 0.847]$	0.912 $[0.902, 0.920]$
no-critic-exp-l2	0.887 $[0.834, 0.940]$	0.850 $[0.764, 0.923]$	0.015 $[0.012, 0.017]$	0.265 $[0.065, 0.515]$
critic-logexp-l2	0.993 $[0.989, 0.996]$	0.662 $[0.341, 0.906]$	0.432 $[0.150, 0.721]$	0.458 $[0.158, 0.769]$
no-critic-logexp-l2	0.799 $[0.587, 0.950]$	0.610 $[0.341, 0.854]$	0.013 $[0.010, 0.016]$	0.281 $[0.063, 0.535]$
critic-neg-l2	0.816 $[0.499, 0.983]$	0.008 $[0.005, 0.011]$	0.007 $[0.005, 0.009]$	0.008 $[0.007, 0.010]$
no-critic-neg-l2	0.870 $[0.768, 0.942]$	0.602 $[0.354, 0.782]$	0.012 $[0.008, 0.016]$	0.155 $[0.008, 0.339]$
critic-exp-cos	0.864 $[0.772, 0.938]$	0.529 $[0.493, 0.556]$	0.363 $[0.283, 0.431]$	0.273 $[0.189, 0.352]$
no-critic-exp-cos	0.678 $[0.555, 0.766]$	0.794 $[0.759, 0.825]$	0.100 $[0.043, 0.168]$	0.610 $[0.557, 0.656]$
critic-logexp-cos	0.983 $[0.979, 0.986]$	0.899 $[0.884, 0.914]$	0.934 $[0.895, 0.965]$	0.894 $[0.876, 0.912]$
no-critic-logexp-cos	0.861 $[0.831, 0.886]$	0.771 $[0.759, 0.783]$	0.329 $[0.209, 0.446]$	0.583 $[0.548, 0.620]$

Table 11: Reward-shaping and critic ablation. Values are final returns normalized by each environment's expert return, reported as mean with a 95% bootstrap confidence interval across six seeds, shown as [lower, upper]. The configurations `critic-exp-l2` and `critic-logexp-cos` correspond to FOCUS-OT-L2 and FOCUS-OT-Cos, respectively.

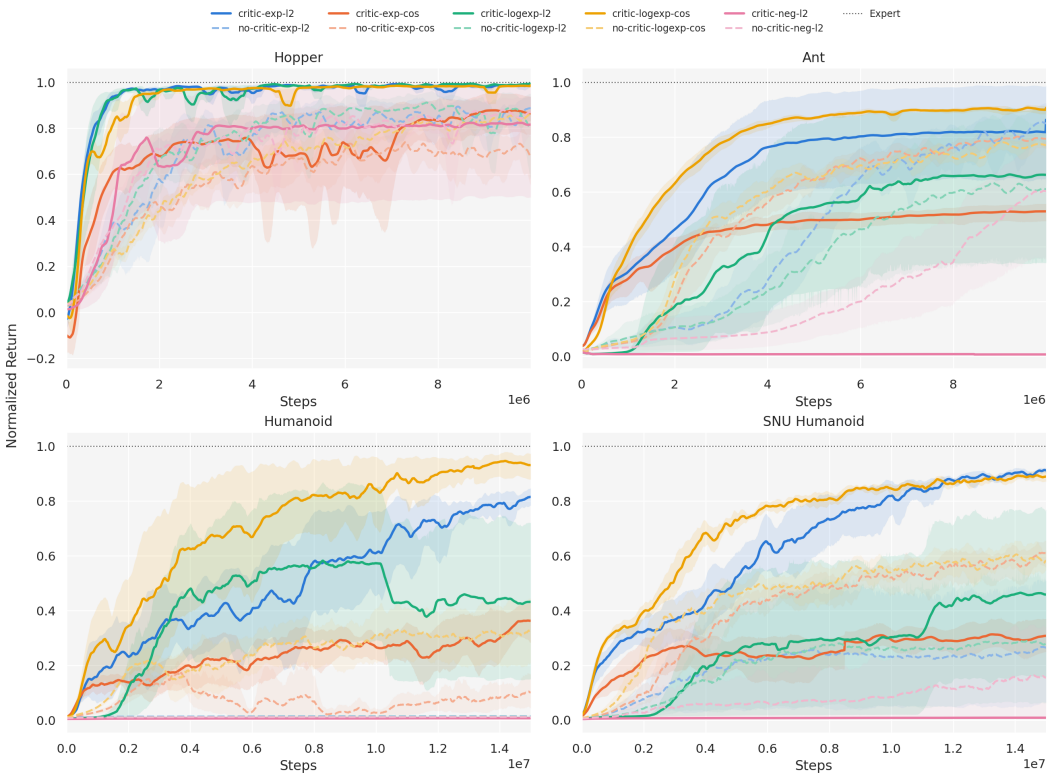


Figure 8: Learning curves for the reward-shaping and critic ablation. Returns are normalized by each environment's expert return (dotted line at 1). Solid and dashed curves denote runs with and without a critic, respectively. Lines show the mean across six seeds, and shaded bands show 95% bootstrap confidence intervals.

D.2 SINKHORN RUNTIME SCALING

Table 12 measures the matching step on SNU Humanoid as a percentage of total epoch time. The no-OT L2 variant is fastest, but the batched Sinkhorn overhead is modest for the regime used in the main experiments ($K = 8$) and becomes more visible only for much larger K.

Setting	L2	OT $K = 1$	OT $K = 4$	OT $K = 8$	OT $K = 16$	OT $K = 32$	OT $K = 64$
Matching step time	0.162%	0.668%	0.689%	0.691%	0.724%	1.025%	1.850%

Table 12: Runtime scaling of the matching step as a percentage of total epoch time on SNU Humanoid. All settings other than K are held fixed (horizon $H = 32$, segment length $L = 32$, parallel environments, batch size, actor/critic updates, hardware), so only the matching step varies.

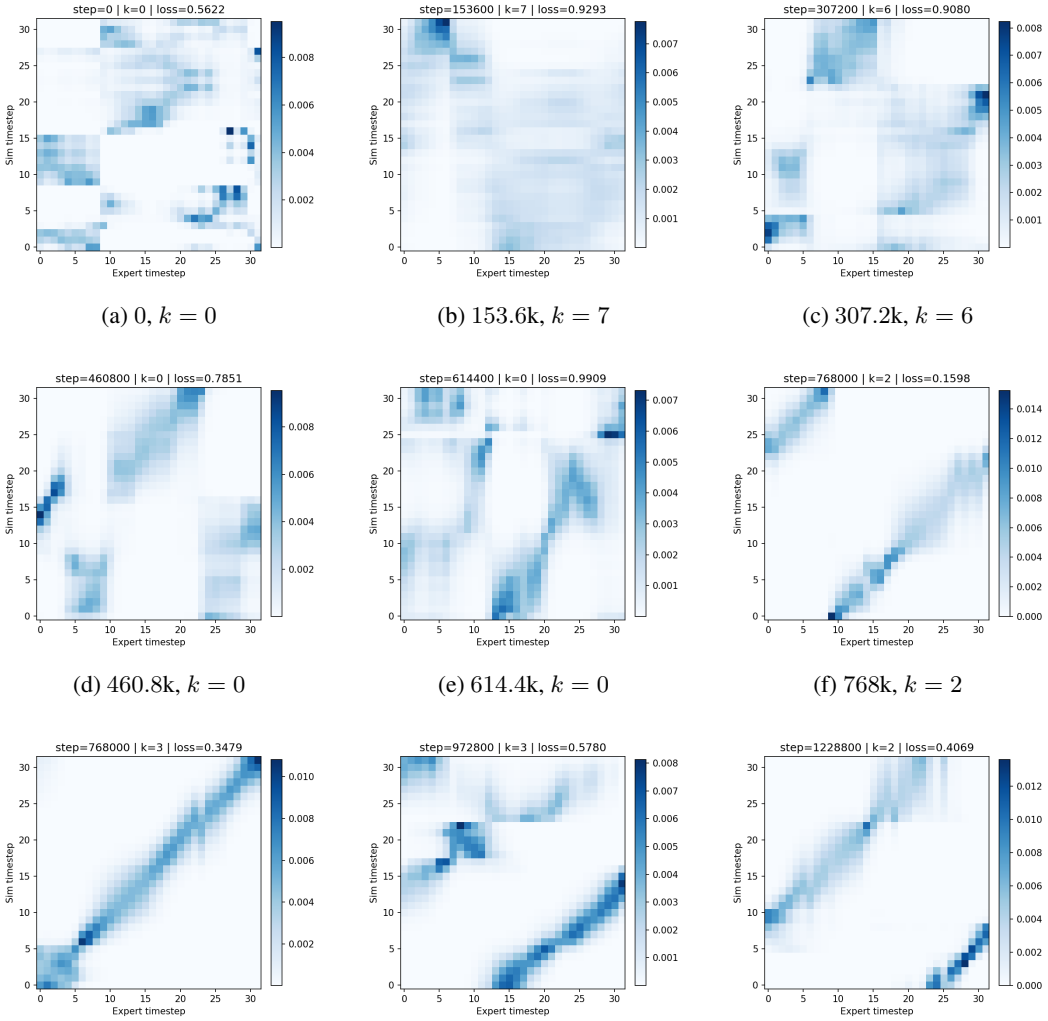
D.3 Visualization of Optimal Transport Plans

For each candidate expert segment τ_k^E, Sinkhorn returns a transport plan $\mu^{\star(k)} \in \mathbb{R}_+^{H \times L}$ between the H states in the agent rollout and the L states in the sampled expert segment. We visualize $\mu^{\star(k)}$ as a heatmap with agent steps on the vertical axis and expert segment steps on the horizontal axis. Darker entries correspond to larger transport mass and indicate which expert states contribute most to the OT cost and reward for each agent state. With uniform temporal weights, each row and column has fixed total mass, so the structure of the plan is more informative than its absolute scale.

These visualizations provide a direct diagnostic for the local alignment used by FOCUS. A coherent local match produces a concentrated shifted diagonal band when the agent and expert phases differ. When the phases are well-aligned, the plan concentrates near the main diagonal. Diffuse plans indicate ambiguous matches or rollouts that are far from expert support. The row-wise costs used in Eq. 10 are exactly the cost matrix entries weighted by each row of this plan, so the heatmap also shows where the per-step imitation reward originates.

Fig. 3 shows representative plans found during training. Fig. 9 shows the full sequence of saved transport-plan visualizations, ordered by training step.

In practice, we inspect the K candidate plans together with their softmin weights w_k from Eq. 11. A single high-weight plan with a coherent band indicates confident local alignment, while several similar weights indicate multiple plausible expert phases. Persistently diffuse or fragmentary high-weight plans can reveal that the sampled segments do not cover the relevant behavior phase, or that best-of-K is emphasizing local fragments rather than a globally consistent behavior. The plotted plans are the transport plans used to compute the training costs.



(a) 0, $k = 0$ (b) 153.6k, $k = 7$ (c) 307.2k, $k = 6$

(d) 460.8k, $k = 0$ (e) 614.4k, $k = 0$ (f) 768k, $k = 2$

(g) 768k, $k = 3$ (h) 972.8k, $k = 3$ (i) 1.229M, $k = 2$

Figure 9: Full sequence of saved Sinkhorn transport-plan visualizations. Each panel reports the training step and candidate expert-segment index k for the corresponding plan. The two plans at step 768k show different candidate expert segments sampled for the same training stage, including the aligned case highlighted in Fig. 3.

E Robustness and Optimization Analysis

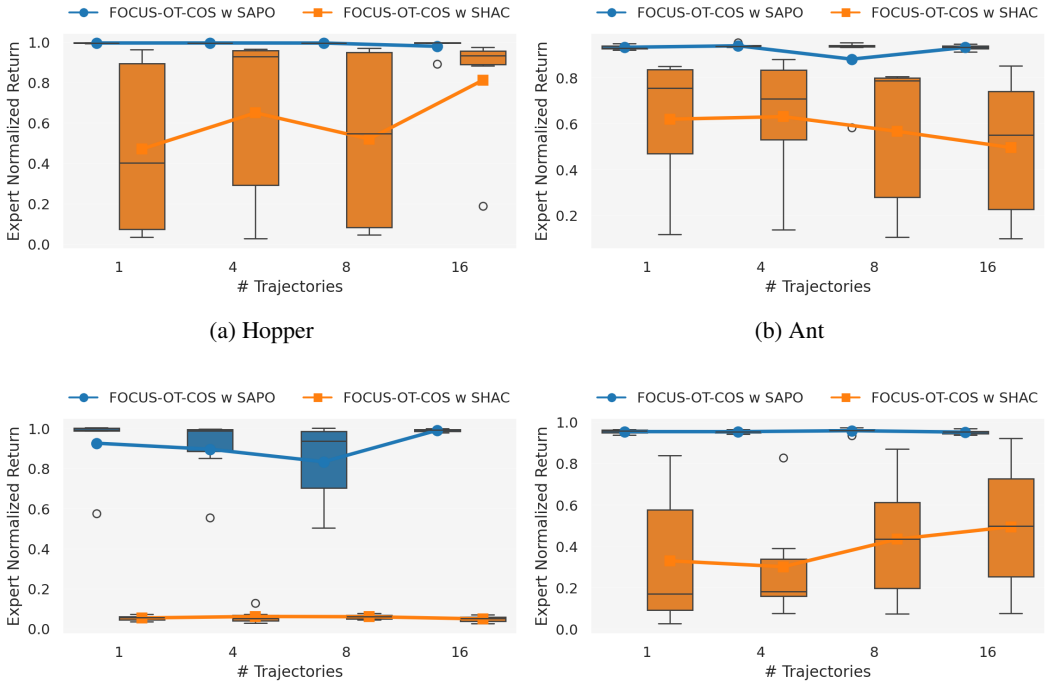
We study sensitivity to expert data, the RL backbone, the optimization route, and candidate-pool size. The best-of-K analysis combines training ablations with an offline diagnostic of phase recovery.

E.1 Expert-Trajectory Count and Backbone Ablations

Figures 10, 11, and 12 report the expert-trajectory-count and backbone ablations for each FOCUS variant. The strongest stability results use SAPO; with SHAC, FOCUS variants can collapse on Humanoid, indicating the method is not invariant to the underlying first-order backbone. SHAC is also sensitive on Hopper: for FOCUS-OT-L2, the mean falls sharply from one to four demonstrations and remains below its one-demonstration value at eight and sixteen. The dependence is not monotonic or shared by every variant: FOCUS-L2 improves at eight demonstrations, and FOCUS-OT-Cos has higher means at four and sixteen than at one, although both still show low-return runs. SAPO remains near expert performance across these Hopper settings.

The runs labeled SHAC use the original-style SHAC configuration: ELU activations, Adam, a single critic with a target network, and gradient clipping at 1.0. Both actor and critic use LayerNorm. SAPO combines entropy-augmented actor returns and soft value targets with further changes: state-dependent policy variance, two critics without a target network, mean critic aggregation for the actor and minimum aggregation for value targets, SiLU activations, AdamW, and gradient clipping at 0.5 (Xing et al., 2025, Section 4.2). Thus the comparison changes both the learning objective and the architecture and optimization settings.

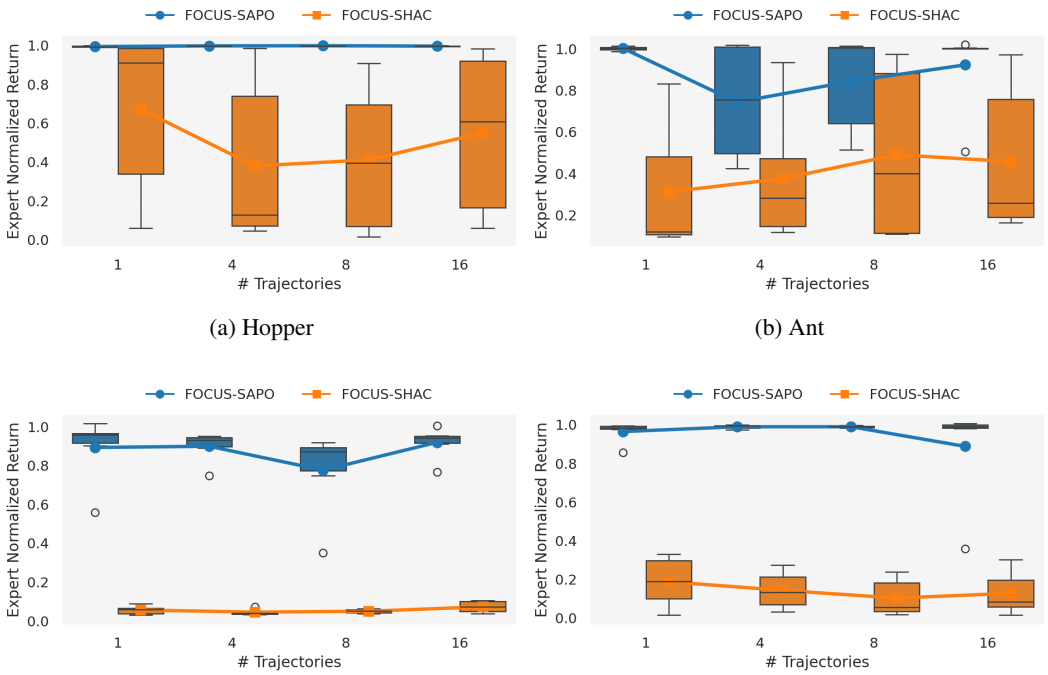
This distinction is also made in SAPO's ablations: removing entropy from both the actor return and value targets retains the other design changes, yielding an enhanced SHAC variant (Xing et al., 2025, Section 6.2 and Appendix F.3). That variant is not the SHAC configuration used in our figures. The SAPO paper reports that these changes alone have limited aggregate impact on its DFlex task-reward benchmarks; this does not establish their effect under our imitation rewards.



(a) Hopper (b) Ant

(c) Humanoid (d) SNU Humanoid

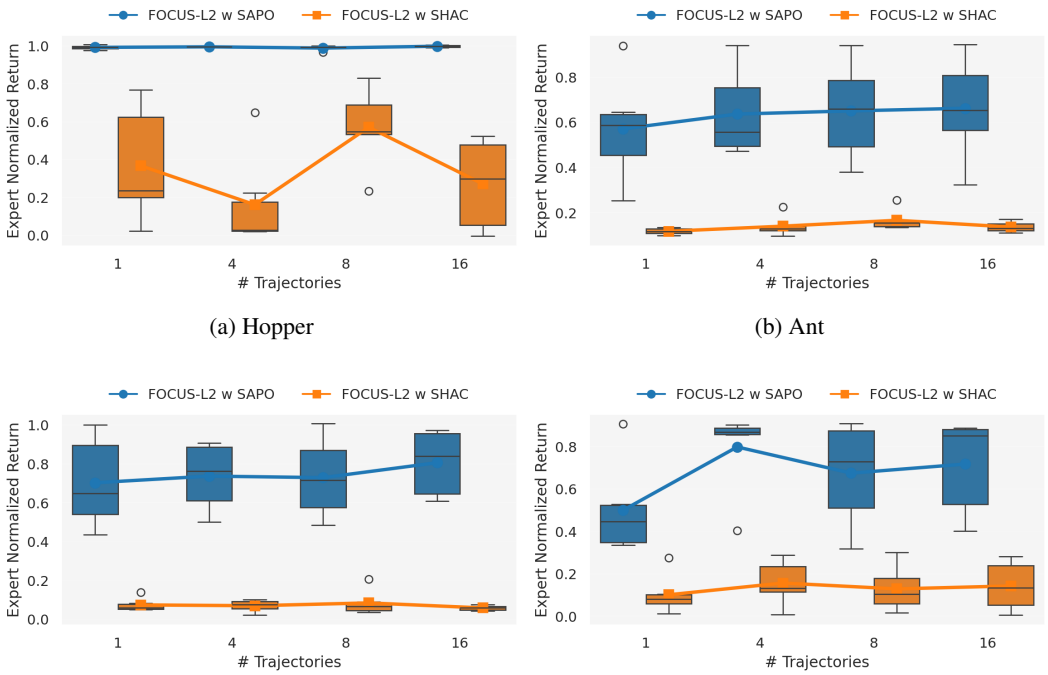
Figure 10: Expert-normalized return of FOCUS-OT-Cos with SHAC and SAPO as the underlying RL algorithm, evaluated across 1, 4, 8, and 16 expert trajectories.



(a) Hopper (b) Ant

(c) Humanoid (d) SNU Humanoid

Figure 11: Expert-normalized return of FOCUS-OT-L2 with SHAC and SAPO as the underlying RL algorithm, evaluated across 1, 4, 8, and 16 expert trajectories.



(a) Hopper (b) Ant

(c) Humanoid (d) SNU Humanoid

Figure 12: Expert-normalized return of FOCUS-L2 with SHAC and SAPO as the underlying RL algorithm, evaluated across 1, 4, 8, and 16 expert trajectories.

E.2 First-Order vs Zeroth-Order Optimization

To assess the contribution of differentiating through the simulator, we compare each FOCUS variant with ZO-FOCUS, which uses the same imitation reward construction with SAC or PPO. Each comparison retains the state-transition representation, local matching rule, best-of-K segment selection, and reward transformation. ZO-FOCUS treats r_t^{FOCUS} from Eq. 12 as a scalar reward, without backpropagating through the dynamics. Table 13 reports expert-normalized returns at the same environment-interaction budgets.

Method	Hopper	Ant	Humanoid	SNU Humanoid
FOCUS-L2	0.951 $[0.938, 0.966]$	0.459 $[0.315, 0.653]$	0.596 $[0.479, 0.714]$	0.582 $[0.419, 0.723]$
FOCUS-OT-L2	$\mathbf{0.988}$ $[\mathbf{0.981}, \mathbf{0.994}]$	0.818 $[0.647, 0.984]$	0.816 $[0.784, 0.847]$	$\mathbf{0.912}$ $[\mathbf{0.902}, \mathbf{0.920}]$
FOCUS-OT-Cos	0.983 $[0.979, 0.986]$	$\mathbf{0.899}$ $[\mathbf{0.884}, \mathbf{0.914}]$	$\mathbf{0.934}$ $[\mathbf{0.895}, \mathbf{0.965}]$	0.894 $[0.876, 0.912]$
ZO-FOCUS-SAC-L2	0.702 $[0.098, 0.872]$	0.431 $[0.291, 0.458]$	0.220 $[0.191, 0.242]$	0.083 $[0.035, 0.102]$
ZO-FOCUS-SAC-OT-L2	0.338 $[-0.017, 0.879]$	0.202 $[0.112, 0.258]$	0.215 $[0.189, 0.238]$	0.118 $[0.109, 0.126]$
ZO-FOCUS-SAC-OT-Cos	0.844 $[0.810, 0.878]$	0.216 $[0.158, 0.236]$	0.243 $[0.217, 0.318]$	0.071 $[0.060, 0.208]$
ZO-FOCUS-PPO-L2	0.657 $[0.521, 0.831]$	0.117 $[0.089, 0.448]$	0.098 $[0.019, 0.138]$	0.023 $[0.019, 0.081]$
ZO-FOCUS-PPO-OT-L2	0.922 $[0.699, 0.936]$	0.113 $[0.086, 0.188]$	0.045 $[0.012, 0.113]$	0.010 $[0.007, 0.053]$
ZO-FOCUS-PPO-OT-Cos	0.588 $[0.401, 0.904]$	0.096 $[0.054, 0.152]$	0.145 $[0.073, 0.156]$	0.054 $[0.052, 0.059]$

Table 13: First-order (FOCUS) vs. matched zeroth-order (ZO-FOCUS) optimization at the final interaction budget. Values are means [bootstrap 95% confidence intervals] across six seeds. ZO-FOCUS results are reported separately for SAC and PPO. Within the FOCUS rows, the highest mean in each environment is bold.

Both OT variants achieve higher mean returns than their SAC and PPO counterparts in every environment, yielding 16 of 16 favorable comparisons with non-overlapping bootstrap 95% confidence intervals. The gaps are particularly large on the humanoid tasks: on Humanoid, the first-order OT variants reach 0.816–0.934 expert-normalized return, compared with 0.045–0.243 for ZO-FOCUS; on SNU Humanoid, the corresponding ranges are 0.894–0.912 and 0.010–0.118. The non-OT L2 variant also attains higher means in all eight matched comparisons. On Ant, however, its advantage over ZO-FOCUS-SAC-L2 is small (0.459 vs. 0.431), and its confidence interval overlaps those of both zeroth-order L2 variants.

These results indicate that the performance of FOCUS depends on how the imitation signal is optimized, beyond the choice of matching objective and reward transformation. In particular, the large gaps on Humanoid and SNU Humanoid support the first-order training procedure used by FOCUS for local transition-matching rewards. The imitation rewards depend on the full matching window, so rewards assigned to earlier transitions can depend on later actions. FOCUS differentiates through these dependencies, whereas the SAC/PPO variants use the assigned scalar rewards in their standard updates. The comparison therefore evaluates complete training procedures with a shared reward construction, including differences in temporal credit assignment and critic learning, rather than isolating the gradient estimator.

E.3 BEST-OF-K TRAINING ABLATION

We evaluate how candidate-pool size affects policy performance, first across all four environments and then at two rollout horizons on Hopper. The phase-recovery diagnostic in Section E.4 separately examines how additional candidates affect local target selection.

Performance across environments. Figure 13 reports expert-normalized final returns for $K \in \{1, 2, 4, 8, 16\}$ with SAPO as the backbone. For FOCUS-OT-L2, the pooled mean increases from $K = 1$ to $K = 8$ before decreasing slightly at $K = 16$. FOCUS-OT-Cos has similar pooled means across $K = 1$–8, with its highest plotted mean at $K = 4$; at $K = 16$, the mean is lower and the confidence interval is wider. These results support moderate candidate pools, including our default $K = 8$, while showing that larger pools do not consistently improve training performance.

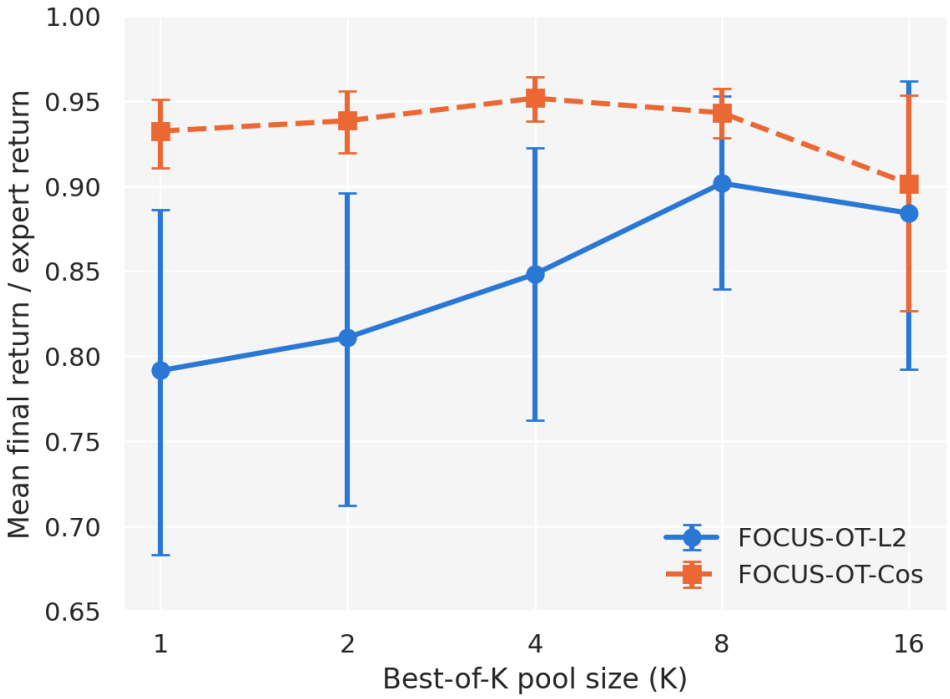


Figure 13: Effect of candidate-pool size on training performance across Hopper, Ant, Humanoid, and SNU Humanoid. Each point averages 24 final returns, with six seeds per environment and each return normalized by its environment's expert return. Error bars show 95% bootstrap confidence intervals. Both OT variants use SAPO; a normalized return of 1 corresponds to expert performance.

Interaction with rollout horizon. Figure 14 compares $H \in \{16, 32\}$ on Hopper. At $H = 32$, both OT variants remain close to expert return across the tested pool sizes. At $H = 16$, increasing K improves mean return most clearly for FOCUS-OT-Cos; the L2 variant varies non-monotonically, although both approach expert return at $K = 16$. The wider intervals at small K indicate greater uncertainty in these means. Changing H also changes the differentiation horizon, the bootstrap point, and the per-step reward scale: with uniform OT row mass $1/H$, a constant ground cost c gives per-step cost c/H. We apply no compensation for this scaling, so changing H also changes the shaped rewards.

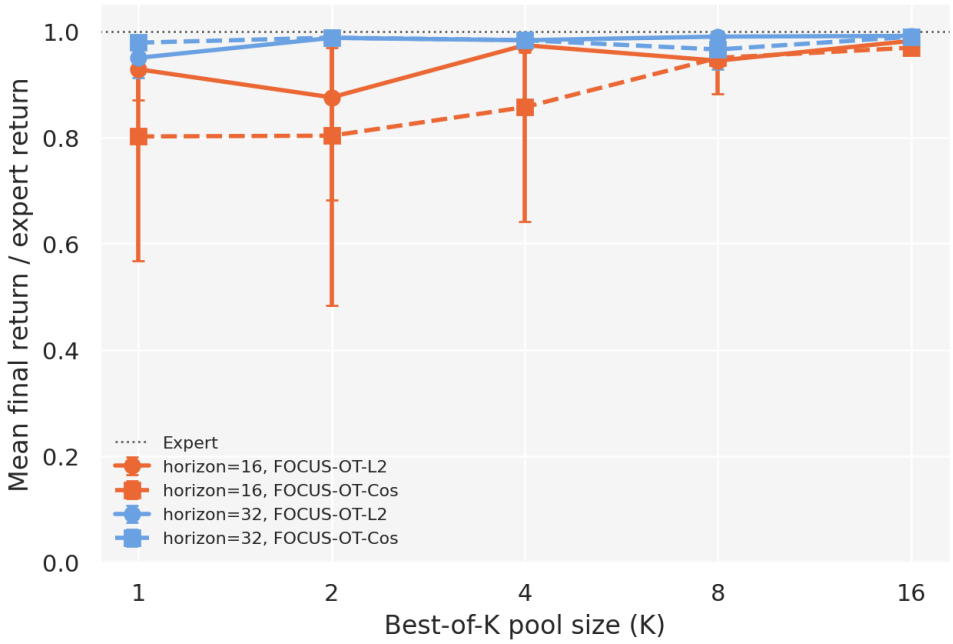


Figure 14: Interaction between candidate-pool size and rollout horizon on Hopper. Points show final expert-normalized returns averaged across six seeds at $H \in \{16, 32\}$; error bars show 95% bootstrap confidence intervals. The dotted line marks expert-level return.

E.4 Best-of-K Phase Recovery

We test whether adding candidate expert segments enables minimum-cost OT selection to recover from a phase mismatch. In each environment, 512 held-out query windows and four candidate pools per query give 2048 paired comparisons. Each window contains 32 state transitions. The $K = 1$ baseline is deliberately mismatched by approximately half a gait cycle. Its paired $K = 8$ pool adds seven uniformly sampled alternatives; the nested $K = 4$ pool keeps the first four candidates, and the $K = 16$ pool appends eight further alternatives. All pools retain the same mismatched baseline. Candidates are ranked by cosine OT cost; this diagnostic evaluates minimum-cost selection on expert windows, separately from the soft-weighted training objective.

Metrics. Reference windows receive discrete phase labels from a shared gait-cycle template. The target phase of a query is determined by a separate Euclidean comparison with the reference bank, independently of its sampled pool and OT scores. For candidate phase $p(c)$, query target phase $p^\star(q)$, and estimated period P, the circular phase error in gait cycles is

$$e(q, c) = \frac{\min\{|p(c) - p^\star(q)|,\ P - |p(c) - p^\star(q)|\}}{P} \in [0, 0.5].$$

A candidate is phase-compatible when $e(q, c) \leq 0.10$. We report *coverage*, the fraction of pools containing a compatible candidate; *rescue*, the fraction of pairs whose mismatched baseline is replaced by a compatible minimum-cost selection; and mean selected phase error. Coverage and rescue are expressed as percentages of all 2048 query–pool pairs. Tables report point estimates with 95% trajectory-cluster bootstrap confidence intervals.

Environment	K	Coverage (%)	Rescue (%)	Mean phase error
Hopper	4	47.3 [44.9, 49.7]	32.6 [30.2, 35.0]	0.200 [0.190, 0.211]
	8	78.2 [76.5, 79.9]	46.3 [43.6, 48.9]	0.147 [0.139, 0.156]
	16	95.8 [94.9, 96.6]	56.6 [54.1, 58.8]	0.118 [0.112, 0.124]
Ant	4	51.8 [49.8, 53.9]	40.1 [38.0, 42.3]	0.168 [0.162, 0.174]
	8	82.0 [80.2, 83.8]	63.4 [60.9, 65.9]	0.102 [0.098, 0.107]
	16	97.7 [97.2, 98.3]	81.4 [79.6, 83.3]	0.073 [0.069, 0.078]
Humanoid	4	56.1 [54.4, 57.9]	43.0 [41.3, 44.8]	0.190 [0.184, 0.196]
	8	85.2 [83.8, 86.7]	64.0 [62.1, 65.9]	0.121 [0.116, 0.125]
	16	98.0 [97.4, 98.7]	77.7 [76.0, 79.3]	0.089 [0.085, 0.093]
SNU Humanoid	4	55.8 [53.7, 58.0]	48.0 [45.9, 50.4]	0.168 [0.163, 0.174]
	8	85.1 [83.8, 86.5]	72.0 [70.1, 73.9]	0.095 [0.092, 0.099]
	16	97.9 [97.4, 98.5]	87.2 [85.6, 88.8]	0.058 [0.055, 0.062]

Table 14: Nested-pool phase-rescue comparison at a target half-cycle mismatch. Entries are point estimates [95% trajectory-cluster bootstrap CI], using 512 held-out query windows and four pools per window. All conditions share the same mismatched baseline; percentages use all 2048 pairs, and mean phase errors are measured in gait cycles.

Increasing K improves coverage and rescue rates and reduces mean selected phase error in all four environments (Table 14). At $K = 8$, rescue ranges from 46.3% to 72.0%, and mean phase error is reduced by a factor of approximately 3–5 relative to the mismatched baseline (Table 15). At $K = 16$, coverage exceeds 95% in every environment, but rescue remains lower, ranging from 56.6% on Hopper to 87.2% on SNU Humanoid. Thus, including a phase-compatible candidate does not ensure that OT assigns it the lowest cost. The absolute reduction in mean phase error is also smaller from $K = 8$ to $K = 16$ than from $K = 4$ to $K = 8$ in every environment.

Environment	Period P (steps)	Phase error $K = 1$	Phase error $K = 8$	Reduction factor
Hopper	37	0.486 [0.486, 0.486]	0.147 [0.139, 0.156]	$3.31\times$
Ant	23	0.478 [0.478, 0.478]	0.102 [0.098, 0.107]	$4.69\times$
Humanoid	20	0.500 [0.500, 0.500]	0.121 [0.116, 0.125]	$4.13\times$
SNU Humanoid	30	0.477 [0.477, 0.478]	0.095 [0.092, 0.099]	$5.02\times$

Table 15: Mean circular phase errors in gait cycles [95% trajectory-cluster bootstrap CI]. Phase labels use a single shared reference-cycle template per environment. The reduction factor is the ratio of the reported mean errors ($K = 1$ divided by $K = 8$), computed from the rounded point estimates.

E.4.1 Diagnostic Protocol

Reference data and sampling. The first eight expert demonstrations form the reference bank; the next 32 (zero-based indices 8–39) supply held-out queries and are used only for this diagnostic. Query windows are sampled without replacement from all valid stride-one windows. The shared baseline is sampled uniformly from reference windows with phase error in $[0.46, 0.50]$, corresponding to a tolerance of 0.04 cycles around the target half-cycle mismatch. Additional candidates are sampled uniformly with replacement. The sampling seed is 0; the eight alternatives appended for $K = 16$ use seed 67890. Query windows, phase labels, and the original $K = 8$ pools remain fixed across the nested-pool comparison.

OT settings. Matching uses concatenated successive observations without standardization, cosine ground cost, Sinkhorn regularization 0.1, and 60 iterations. These are the settings of the offline diagnostic; the training settings are reported in Section B.5.

Gait-period estimation. Phase estimation uses a separate copy of the reference observations standardized by the means and population standard deviations over all valid timesteps of the eight reference demonstrations. Dimensions with standard deviation at most 10^{-4} are excluded. After discarding the first 100 timesteps, each integer lag ℓ is scored by the squared difference between

standardized observations at t and $t + \ell$, averaged over valid timesteps and dimensions within each demonstration and then equally over demonstrations. The inclusive lag ranges are $[10, 90]$ for Hopper, $[8, 80]$ for Ant, and $[8, 120]$ for both humanoids. The estimated period P is the strict interior local minimum with the lowest score; if none exists, we use the lowest-scoring lag over the full range.

Phase labels. The shared template comprises observations at indices $100, \ldots, 100 + P - 1$ of the first reference demonstration. Each reference window receives the phase offset that minimizes squared Euclidean distance between its first eight standardized observations and an eight-observation probe of the template, wrapping modulo P. For each query, we search the full reference bank for the nearest window under time-aligned squared Euclidean distance over the first eight unstandardized transition vectors, using all coordinates of both states in each transition. That reference window's phase defines $p^\star(q)$. Ties are resolved by the first minimum in index order.

Confidence intervals. We use 5000 trajectory-cluster bootstrap draws with seed 12345 and report pointwise 95% percentile intervals. Each draw samples 32 query-trajectory IDs with replacement, retains all sampled windows and pool repetitions from each selected trajectory, and recomputes the statistics. All 32 held-out query trajectories contribute windows. The reference bank and phase template remain fixed, so intervals quantify query-trajectory sampling uncertainty conditional on these references. For Hopper, Ant, and Humanoid, the discrete labels and admissible mismatch interval permit only one baseline phase error: $18/37$, $11/23$, and $10/20$, respectively. Their $K = 1$ intervals therefore have zero width up to floating-point precision. SNU Humanoid permits both $14/30$ and $15/30$, so its baseline error can vary.