T-MAC: TARGET-MOTION ACTOR-CRITIC FOR GENERATIVE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative policies can represent complex and multimodal action distributions, which makes them attractive for continuous control. However, in online reinforcement learning, the target of such a policy is defined by a critic that changes after every update, so the policy keeps chasing a moving target. Existing methods train the policy toward where the target is now, but do not use how the target has moved. We propose Target-Motion Actor-Critic (T-MAC), a flow-based actor-critic method that learns from both signals. In addition to the current-target loss, T-MAC trains the cross-update change of the policy velocity field to follow the target motion. T-MAC estimates the target motion by comparing the Jordan–Kinderlehrer–Otto (JKO) responses of successive critics on the same source actions, and uses a response-magnitude gate to weight each state by how far its target moves. Experimental results on six MuJoCo and four HumanoidBench tasks show that, compared with representative model-free and generative-policy RL baselines, T-MAC achieves state-of-the-art performance with consistent improvements across all ten tasks.

1 INTRODUCTION

Generative policies such as diffusion and flow policies can represent complex and multimodal action distributions, and they have become an expressive alternative to Gaussian policies in continuous-control reinforcement learning (RL) (Wang et al., 2023; Chi et al., 2025; Chen et al., 2023; Ajay et al., 2023; Li et al., 2024; Li, 2023). In online RL, these policies learn from targets defined by a critic. Existing methods either pass critic gradients through the generation process (Wang et al., 2024; Psenka et al., 2024; Lyu et al., 2025; Li & Levine, 2026; McAllister et al., 2026), or build critic-guided target actions and distributions for the policy to match (Yang et al., 2023; Ding et al., 2024; Ma et al., 2025; 2026; Zhu et al., 2026a; Zhan et al., 2026). In both cases, the policy learns only from the current critic.

The critic keeps changing during training, and each critic update moves the target, leaving the policy constantly chasing a moving target. At each update, the policy has been trained toward the previous target, while the missing information for the next update is how the target has moved. Existing objectives only describe the current target, which we refer to as *current-target supervision*, while the change of the target between two critic updates provides an additional signal that we call *target motion*. As shown in Figure 1, different critic histories can end at the same current critic and produce the same current-target supervision, although their target motions are different. A current-target objective cannot distinguish these cases because it only uses the current target.

This raises a natural question: *Can a flow policy learn not only where the target is, but also how it moves?*

We propose **T**arget-**M**otion **A**ctor-**C**ritic (T-MAC), which turns target motion into a supervision signal for flow policy updates. The key observation is that a flow policy is represented by a velocity field over the action space, so each policy update changes this field. Within the same update, the target also moves, and its motion directly guides the change of this field. T-MAC combines the current-target loss with a target-motion loss, so the policy learns both where the target is and how it moves. To measure target motion, T-MAC applies Jordan–Kinderlehrer–Otto (JKO) proximal steps to the same source actions under the previous and updated critics, and uses the difference between the two responses as the target motion. Each step moves the source actions toward higher

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

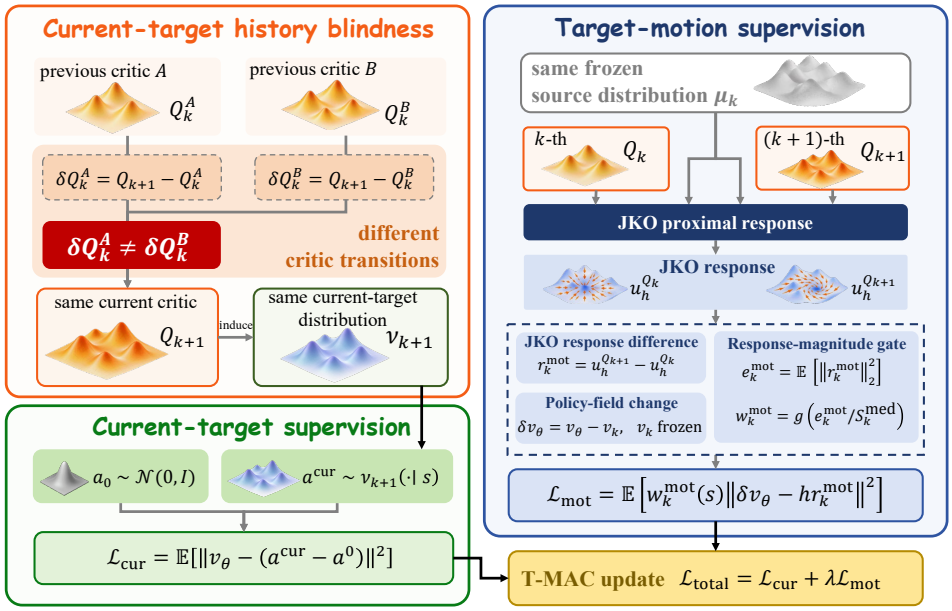


Figure 1: **Overview of T-MAC.** Different critic histories can lead to the same current-target supervision while producing different target motions. T-MAC estimates target motion from JKO responses of successive critics on the same source actions and uses it for policy updates.

value under a transport cost, which gives a local response around the current policy. Because both responses start from the same source actions, this difference isolates the effect of the critic update. A response-magnitude gate then weights each state by the size of its target motion.

Our contributions are summarized as follows:

- We introduce target-motion supervision into flow-based policy learning for online RL. By supervising the cross-update change of the policy velocity field with target motion, T-MAC lets the policy learn how its target changes across critic updates.
- We build a target-motion teacher from the JKO responses of successive critics. By moving the same source actions under the previous and the updated critic, T-MAC obtains a response difference that isolates the effect of the critic update.
- We validate T-MAC on a ten-task benchmark spanning 6 MuJoCo tasks and 4 Humanoid-Bench tasks. T-MAC achieves the best performance on all evaluated tasks and outperforms nine representative baselines, showing that target motion improves generative policies.

2 Preliminaries

2.1 Reinforcement Learning

We consider a Markov decision process (MDP) defined by $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, P is the transition dynamics, r is the reward function, and $\gamma \in [0, 1)$ is the discount factor. A policy $\pi(a \mid s)$ specifies a distribution over actions conditioned on state s and aims to maximize the expected discounted return. The action-value function is

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, s_0 = s, a_0 = a \right]. \tag{1}$$

The corresponding state value and advantage are

$$V^\pi(s) = \mathbb{E}_{a \sim \pi(\cdot|s)}[Q^\pi(s, a)], \qquad A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s). \tag{2}$$

In actor–critic methods, Q^π is approximated by a parametric critic $Q_\phi(s, a)$, which provides the value signal for policy learning.

2.2 Flow Matching

Conditional flow matching considers a base sample $x_0 \sim p_0$ and a target sample $x_1 \sim p_1(\cdot \mid s)$ conditioned on state s (Lipman et al., 2023). For flow time $\tau \in [0, 1]$, the linear probability path and its target velocity are

$$x_\tau = (1-\tau)x_0 + \tau x_1, \qquad u_\tau(x_\tau \mid x_1) = x_1 - x_0. \tag{3}$$

The conditional velocity field v_θ is trained with the conditional flow-matching objective

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{s,x_1,x_0,\tau}\left[\|v_\theta(x_\tau, \tau, s) - u_\tau(x_\tau \mid x_1)\|_2^2\right], \tag{4}$$

where $x_0 \sim p_0$, $x_1 \sim p_1(\cdot \mid s)$, and $\tau \sim \mathcal{U}[0, 1]$.

At inference time, generation starts from $x_0 \sim p_0$ and follows the learned ordinary differential equation (ODE) $dx_\tau/d\tau = v_\theta(x_\tau, \tau, s)$ from $\tau = 0$ to $\tau = 1$. The terminal sample $x_{\tau=1}$ is used as the policy action.

2.3 Wasserstein Gradient Flows and the JKO Scheme

The JKO scheme describes the evolution of probability distributions under the 2-Wasserstein metric (Jordan et al., 1998). For $\mu, \nu \in \mathcal{P}_2(\mathcal{A})$, the squared 2-Wasserstein distance is defined as (Villani et al., 2009)

$$W_2^2(\mu, \nu) = \inf_{\Gamma \in \Pi(\mu,\nu)} \int_{\mathcal{A}\times\mathcal{A}} \|b - b'\|_2^2 \, d\Gamma(b, b'), \tag{5}$$

where b and b' are dummy action variables and $\Pi(\mu, \nu)$ is the set of couplings between μ and ν.

Given an energy functional $\mathcal{F}$ and a source action distribution μ, one JKO step with step size $h > 0$ is

$$\nu_h = \arg\min_{\nu \in \mathcal{P}_2(\mathcal{A})} \left\{ \frac{1}{2h} W_2^2(\nu, \mu) + \mathcal{F}(\nu) \right\}. \tag{6}$$

The Wasserstein term limits the displacement from the source distribution, while $\mathcal{F}$ drives the distribution toward lower energy. When the optimal transport is induced by a map $T_h : \mathcal{A} \to \mathcal{A}$,

$$(T_h)_{\#}\mu = \nu_h,$$

and $T_h(a) - a$ gives the displacement of action a under the proximal step.

3 Methodology

T-MAC separates two signals. The updated critic defines the current target, and the change from one critic update to the next defines target motion. The first signal trains the policy toward the current target; the second supervises how its velocity field should change.

3.1 Current-Target Learning under a Moving Critic

Flow policies evolve along two axes: (a) the flow variable τ indexes the ODE trajectory that generates an action, and (b) the optimization index k counts policy updates made after critic updates. We use k for optimization time so that the MDP interaction time t remains available for trajectories and returns. Standard flow matching models the former and T-MAC focuses on the latter.

(1) The current-target objective is built from the current critic. Online RL does not directly provide target actions for the CFM objective. For a state s, we therefore sample candidate actions from the frozen policy and use the updated critic Q_{k+1} to select the current target action a^{cur}. The selected actions form the current-target distribution $\nu_{k+1}(\cdot \mid s)$ (Appendix D.1). With a base sample $a^0 \sim p_0$ and flow time $\tau \sim \mathcal{U}[0, 1]$, define

$$a_\tau = (1-\tau)a^0 + \tau a^{\mathrm{cur}}, \qquad a^0 \sim p_0, \qquad \tau \sim \mathcal{U}[0, 1]. \tag{7}$$

The current-target loss uses the path in Eq. (7):

$$\mathcal{L}_{\mathrm{cur}}(\theta) = \mathbb{E}_{s,a^{\mathrm{cur}},a^0,\tau}\left[\left\|v_\theta(a_\tau,\tau,s) - \left(a^{\mathrm{cur}} - a^0\right)\right\|_2^2\right], \qquad a^{\mathrm{cur}} \sim \nu_{k+1}(\cdot \mid s). \tag{8}$$

The loss depends on Q_{k+1} and therefore specifies what the policy should match at update $k+1$. It does not specify how this target moved from Q_k.

(2) What information does current-target learning discard? The limitation of (8) is informational. If two critic histories produce the same current target, they define the same population policy objective even when their target motions differ. Current-target training therefore specifies what the policy should match at the current update, but does not describe how the target changes across critic updates. Appendix B.2 formalizes this history-blindness property.

Let θ_k be the policy parameters before the current policy update and set $v_k := v_{\theta_k}$. The cross-update velocity change is

$$\delta v_\theta(a,\tau,s) := v_\theta(a,\tau,s) - v_k(a,\tau,s), \tag{9}$$

where v_k is frozen during the update. This is a change between policy updates, not the flow-time change used to generate one action. We next construct the target-motion signal that supervises δv_θ.

3.2 Target-Motion Actor-Critic

T-MAC uses the change between successive critics to add target-motion supervision to current-target policy learning. It first turns the critic change into an action-space response, then matches the policy's cross-update velocity change to that response, and finally relates the tracking error to policy improvement.

(1) How to construct a target-motion teacher. To isolate the critic change from changes in the source distribution, both critics use the same frozen source distribution $\mu_k(\cdot \mid s)$. For a fixed state, the critic defines the free energy

$$\mathcal{E}_Q(\nu \mid s) := -\mathbb{E}_{a\sim\nu(\cdot|s)}[Q(s,a)] + \alpha\,\mathrm{KL}(\nu(\cdot \mid s) \,\|\, \varpi_s), \tag{10}$$

where ϖ_s is a fixed reference measure. The source distribution $\mu_k(\cdot \mid s)$ stays fixed during policy update k. A JKO step of size $h > 0$ starts from this source and gives

$$\nu_h^Q := \arg\min_{\nu\in\mathcal{P}_2(\mathcal{A})}\left\{\frac{1}{2h}W_2^2(\nu,\mu_k(\cdot \mid s)) + \mathcal{E}_Q(\nu \mid s)\right\}. \tag{11}$$

Let T_h^Q transport $\mu_k(\cdot \mid s)$ to ν_h^Q. Its response is

$$u_h^Q(a \mid s) := \frac{T_h^Q(a \mid s) - a}{h}, \qquad (T_h^Q)_{\#}\mu_k(\cdot \mid s) = \nu_h^Q. \tag{12}$$

Define the critic change and target-motion response by

$$\delta Q_k(s,a) := Q_{k+1}(s,a) - Q_k(s,a), \qquad r_k^{\mathrm{mot}}(a \mid s) := u_h^{Q_{k+1}}(a \mid s) - u_h^{Q_k}(a \mid s). \tag{13}$$

The response in Eq. (13) depends on the critic update, not on a change in sampled actions. The following result gives its first-order form.

Theorem 1 (Shared-Source Target-Motion Recovery)**.** *Under the regularity conditions in Appendix B.3, there is $C(s) < \infty$ such that, for sufficiently small h,*

$$\left\|r_k^{\mathrm{mot}}(\cdot \mid s) - \nabla_a \delta Q_k(s,\cdot)\right\|_{L^2(\mu_k(\cdot|s))} \le C(s)h. \tag{14}$$

Proof. See Appendix B.3. □

Thus, the response defined in Eq. (13) estimates the action-space gradient of δQ_k to first order. If different source distributions μ_k and $\widetilde{\mu}_k$ are used, the first-order response contains the extra source-drift term

$$-\alpha\nabla_a \log \frac{d\widetilde{\mu}_k}{d\mu_k}. \tag{15}$$

The shared source removes this term to first order. For each critic, the JKO step gives a minimum-displacement response toward its proximal distribution.

Some critic updates produce little target motion. We use the response energy to define a response-magnitude gate:

$$e_k^{\mathrm{mot}}(s) := \mathbb{E}_{a\sim\mu_k(\cdot|s)}\left[\|r_k^{\mathrm{mot}}(a \mid s)\|_2^2\right], \qquad w_k^{\mathrm{mot}}(s) := g\big(e_k^{\mathrm{mot}}(s)/S_k^{\mathrm{med}}\big), \tag{16}$$

where S_k^{med} is the median response energy over the states in the update and g is nondecreasing with $g(0) = 0$. The gate is scale free, so one setting transfers across tasks whose return scales differ by several orders of magnitude. Appendix D.3 gives the form used in the experiments.

Theorem 1 also gives

$$\left|\sqrt{e_k^{\mathrm{mot}}(s)} - \|\nabla_a \delta Q_k(s,\cdot)\|_{L^2(\mu_k(\cdot|s))}\right| \le C(s)h. \tag{17}$$

The response energy therefore measures the size of the target motion, which follows the action-dependent part of the critic change. The gate is large on states where the target moves more than on the other states in the same update.

Action-independent critic shifts give an exact finite-step null property.

Proposition 1 (Exact Null under Action-Independent Critic Shifts)**.** *If $Q_{k+1}(s,a) = Q_k(s,a)+c(s)$ for every action, then $r_k^{\mathrm{mot}}(\cdot \mid s) = 0$ and $w_k^{\mathrm{mot}}(s) = 0$. The statewise motion loss and its policy gradient are therefore zero.*

Proof. If $Q_{k+1}(s,a) = Q_k(s,a) + c(s)$, then every candidate distribution ν satisfies

$$\mathcal{E}_{Q_{k+1}}(\nu \mid s) = \mathcal{E}_{Q_k}(\nu \mid s) - c(s).$$

The additive term is independent of ν, so the two proximal minimizers and their optimal transport maps are identical. Hence $r_k^{\mathrm{mot}} = 0$, which gives $e_k^{\mathrm{mot}} = 0$ and $w_k^{\mathrm{mot}} = 0$. The motion-loss contribution and its policy gradient are therefore zero. □

(2) How to track target motion. Let χ_k be the training law over (s,a,τ). The response in Eq. (13) does not depend on flow time, so we use the same vector at every τ:

$$\bar{r}_k^{\mathrm{mot}}(a,\tau,s) := r_k^{\mathrm{mot}}(a \mid s).$$

In implementation, values away from the source particles are obtained by the action-space interpolation specified in Appendix D.3. The critics, source distribution, frozen policy snapshot, response, and gate are stop-gradient quantities. Using the cross-update change in Eq. (9), the motion loss is

$$\mathcal{L}_{\mathrm{mot}}(\theta) := \mathbb{E}_{(s,a,\tau)\sim\chi_k}\left[w_k^{\mathrm{mot}}(s)\left\|\delta v_\theta(a,\tau,s) - \mathrm{stopgrad}\big(h\,\bar{r}_k^{\mathrm{mot}}(a,\tau,s)\big)\right\|_2^2\right], \tag{18}$$

and the full policy objective is

$$\mathcal{L}_{\mathrm{total}}(\theta) := \mathcal{L}_{\mathrm{cur}}(\theta) + \lambda\mathcal{L}_{\mathrm{mot}}(\theta), \qquad \lambda \ge 0. \tag{19}$$

The first term fits the current target in Eq. (8). The second term matches the policy change in Eq. (9) to the target-motion response.

(3) From motion tracking to policy improvement. We next connect motion tracking to the actions produced by the policy. Define the motion-guided reference field and its terminal action distribution as

$$v_{k+1}^{\mathrm{J}}(a,\tau,s) := v_k(a,\tau,s) + h\,\bar{r}_k^{\mathrm{mot}}(a,\tau,s), \qquad \pi_{k+1}^{\mathrm{J}}(\cdot \mid s) := \Phi_s(v_{k+1}^{\mathrm{J}}), \tag{20}$$

where J marks the JKO-guided reference, and $\Phi_s(v)$ denotes the terminal action distribution generated by v. We couple the learned policy and this reference policy with the same initial noise and call their expected terminal-action difference $D_A(\theta;s)$. Under the regularity and coverage conditions in Appendix B.5,

$$W_1\big(\pi_\theta(\cdot \mid s), \pi_{k+1}^{\mathrm{J}}(\cdot \mid s)\big) \le D_A(\theta;s) \le C_{\mathrm{flow}}(s)\epsilon_{\mathrm{motion}}(\theta;s) + \epsilon_{\mathrm{solver}}(s), \tag{21}$$

Algorithm 1 Target-Motion Actor-Critic

Require: Flow policy v_θ, critic Q, replay buffer $\mathcal{B}$, proximal step h, KL coefficient α, gate function g, motion weight $\lambda \geq 0$, candidate number N_{cand}, particle number N_{src}

```
Initialize the policy and critic
for each training iteration k do
    Q_k ← stopgrad(Q); sample transitions from B and update the critic to Q_{k+1}
    θ_k ← stopgrad(θ), v_k ← v_{θ_k}                          ▷ frozen policy snapshot
    Draw N_cand candidates from π_{θ_k} and select a^cur by Q_{k+1}
    if λ > 0 then
        Draw N_src particles from μ_k = π_{θ_k}               ▷ shared by both critics
        Solve (11) for Q_k and Q_{k+1} from these particles to get u_h^{Q_k} and u_h^{Q_{k+1}}
        Form r_k^mot, e_k^mot, and w_k^mot using (13) and (16)
    end if
    Update θ by minimizing (19)
end for
return the learned flow policy v_θ
```

where $C_{\text{flow}}(s) = e^{L_x} C_{\text{cov}}^{1/2}$ and ϵ_{motion} is the unweighted RMS velocity residual.

To connect tracking to policy improvement, let $\nu_{k+1}(\cdot \mid s)$ be the current-target distribution and suppose that it provides the margin

$$\mathbb{E}_{a \sim \nu_{k+1}(\cdot|s)} \left[A^{\pi_k}(s, a) \right] \geq \Delta_{\text{base}}(s) - \epsilon_{\text{cert}}(s). \tag{22}$$

The distance between the reference policy and the current-target distribution is

$$\epsilon_J(s) = W_1 \left(\pi_{k+1}^{\text{J}}(\cdot \mid s), \nu_{k+1}(\cdot \mid s) \right). \tag{23}$$

Let θ_λ minimize $\mathcal{L}_{\text{total}}$ and write $\pi_{k+1}^{\lambda} = \pi_{\theta_\lambda}$. This gives the following guarantee.

Theorem 2 (Policy Improvement with Target-Motion Tracking)**.** *Suppose the current-target margin above holds and $A^{\pi_k}(s, \cdot)$ is L_A-Lipschitz. Then the updated policy satisfies*

$$J(\pi_{k+1}^{\lambda}) - J(\pi_k) \geq \frac{1}{1-\gamma} \mathbb{E}_{s \sim d^{\pi_{k+1}^{\lambda}}} \left[\Delta_{\text{base}}(s) - \epsilon_{\text{cert}}(s) - L_A \epsilon_J(s) - L_A D_A(\theta_\lambda; s) \right]. \tag{24}$$

Proof. See Appendix B.6; the execution bound used in the proof is established in Appendix B.5. □

Theorem 2 connects target-motion tracking to policy improvement. The current-target distribution provides the base margin. Reducing the motion residual directly tightens D_A, while aligning the JKO teacher with the target displacement minimizes ϵ_J. Appendix B.6 bounds these terms, and Appendix C.3 evaluates the teacher alignment.

3.3 Training Procedure

Algorithm 1 summarizes one T-MAC update. Both critic responses come from the same frozen source distribution, so the teacher reflects the critic change rather than a change in sampled actions. The JKO teacher and the response-magnitude gate are used only during training, and inference uses the same flow integration as the underlying flow policy.

4 Experiments

We evaluate T-MAC on six MuJoCo continuous-control tasks (Brockman et al., 2016) and four HumanoidBench humanoid tasks (Sferrazza et al., 2024), covering standard locomotion and whole-body humanoid control. We report Total Average Return (TAR) as the primary metric. We compare T-MAC with model-free and generative-policy RL baselines, test whether its updates use target motion, and ablate the source, direction, and construction of the target-motion teacher. Appendix C.4 reports hyperparameter sensitivity.

Table 1: Total Average Return (TAR) on six MuJoCo tasks and four HumanoidBench tasks. Results are reported as mean ± standard deviation over five seeds. **Bold** and underline denote the best and second-best results, respectively. Higher is better.

		Humanoid-v4	Ant-v4	HalfCheetah-v4	Walker2d-v4	Swimmer-v4
Classic Model-Free RL	PPO	1158.3 ± 461.6	3004.8 ± 226.4	3157.1 ± 864.7	2609.9 ± 1211.8	125.2 ± 3.4
	TD3	5400.9 ± 138.9	3794.3 ± 800.7	10640.6 ± 578.7	1792.1 ± 1745.6	86.6 ± 36.3
	SAC	4388.6 ± 1121.1	2574.2 ± 561.2	8786.8 ± 1099.9	3966.0 ± 312.3	69.3 ± 17.3
Generative-Policy RL	QSM	1417.6 ± 282.0	584.7 ± 109.2	10426.1 ± 392.1	899.6 ± 126.9	56.2 ± 6.2
	DIPO	6090.2 ± 276.7	937.4 ± 27.4	7681.1 ± 1087.0	2493.5 ± 918.2	45.5 ± 1.1
	DACER	2076.5 ± 2812.3	4931.3 ± 544.6	10641.8 ± 480.2	3177.5 ± 721.3	73.0 ± 13.0
	FlowRL	5641.0 ± 602.9	2795.1 ± 761.0	4136.8 ± 362.4	1577.5 ± 701.4	37.9 ± 6.5
	QVPO	1462.7 ± 1376.1	1779.4 ± 946.3	6694.7 ± 1282.4	221.6 ± 195.8	52.8 ± 6.3
	DPMD	5339.7 ± 1242.9	5129.4 ± 499.1	10775.3 ± 558.7	3369.7 ± 374.7	56.0 ± 8.1
	T-MAC (ours)	**7183.7 ± 361.1**	**5713.2 ± 83.7**	**11571.8 ± 706.4**	**4449.4 ± 873.2**	**158.4 ± 12.1**
		HumanoidStandup-v4	H1-maze	H1-reach	H1hand-reach	H1hand-crawl
Classic Model-Free RL	PPO	114632.1 ± 14298.8	154.5 ± 25.8	3651.7 ± 162.5	2717.4 ± 219.8	324.0 ± 55.6
	TD3	120747.8 ± 17403.9	213.6 ± 71.4	5385.6 ± 39.9	5134.0 ± 401.8	696.7 ± 97.9
	SAC	116435.5 ± 5462.1	192.0 ± 65.6	5126.3 ± 670.4	5109.4 ± 164.7	649.2 ± 110.4
Generative-Policy RL	QSM	147862.5 ± 14062.9	216.6 ± 25.2	4784.2 ± 1058.9	2662.1 ± 99.7	615.0 ± 18.1
	DIPO	149635.6 ± 8245.9	194.4 ± 38.4	6865.1 ± 378.7	4521.4 ± 192.3	564.8 ± 7.2
	DACER	52415.7 ± 16996.3	254.2 ± 45.1	4772.3 ± 571.0	5244.3 ± 454.8	553.3 ± 70.6
	FlowRL	146440.3 ± 4975.4	282.9 ± 11.3	4492.4 ± 1082.6	3783.3 ± 405.9	695.6 ± 77.4
	QVPO	108635.1 ± 29424.9	125.3 ± 10.3	764.5 ± 468.0	214.8 ± 345.4	401.0 ± 34.9
	DPMD	141867.9 ± 7101.3	236.3 ± 55.2	6757.3 ± 452.0	5418.5 ± 417.4	447.3 ± 33.6
	T-MAC (ours)	**153292.1 ± 4557.8**	**299.0 ± 15.5**	**7417.7 ± 735.5**	**6583.8 ± 1069.7**	**844.5 ± 75.6**

4.1 Experimental Setup

Baselines. We compare T-MAC with six generative-policy methods: DACER (Wang et al., 2024), DPMD (Ma et al., 2025), QVPO (Ding et al., 2024), DIPO (Yang et al., 2023), QSM (Psenka et al., 2024), and FlowRL (Lyu et al., 2025). We also include three model-free online RL baselines: SAC (Haarnoja et al., 2018), PPO (Schulman et al., 2017), and TD3 (Fujimoto et al., 2018). Appendix D.4 and D.5 list the experimental hyperparameters.

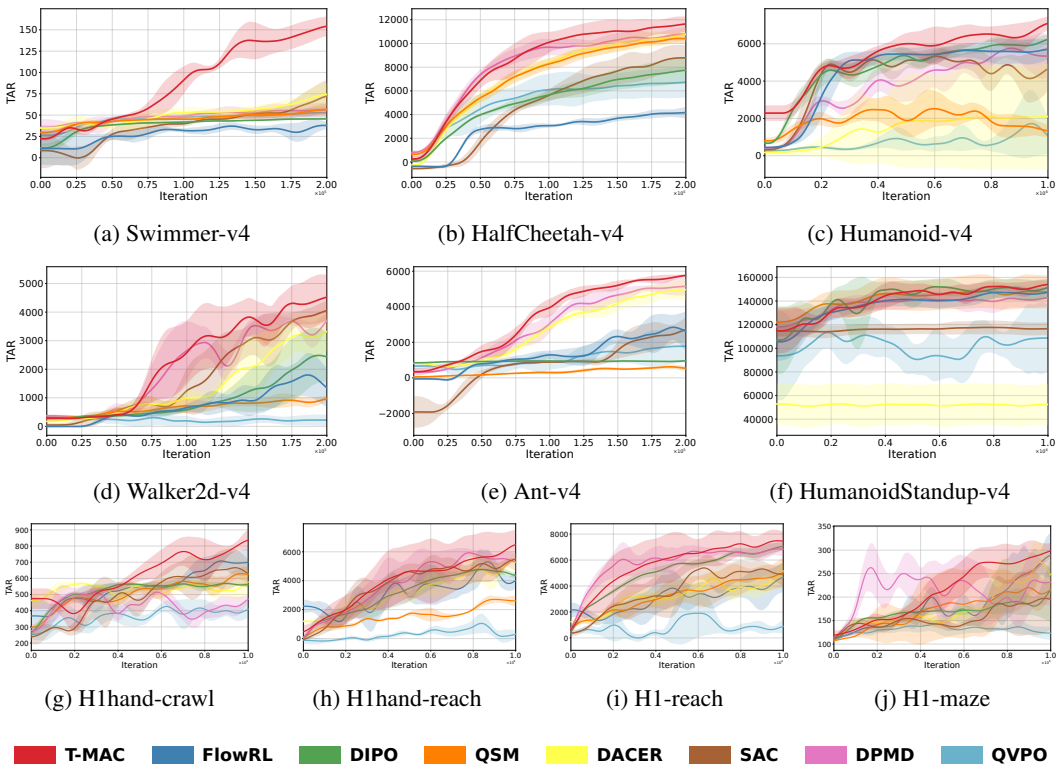
Evaluation Setup. We implement T-MAC in JAX and evaluate six MuJoCo v4 (Brockman et al., 2016) environments: `HalfCheetah-v4`, `Walker2d-v4`, `Ant-v4`, `Swimmer-v4`, `Humanoid-v4`, and `HumanoidStandup-v4`. We also evaluate four HumanoidBench (Sferrazza et al., 2024) tasks: `H1-maze`, `H1-reach`, `H1hand-reach`, and `H1hand-crawl`. All runs use two Intel Xeon Gold 6530 CPUs and eight NVIDIA RTX 5090 GPUs. MuJoCo tasks other than `Humanoid-v4` and `HumanoidStandup-v4` use 1M environment steps, while these two tasks and all HumanoidBench tasks use 5M interactions. Unless stated otherwise, T-MAC uses a one-step flow policy ($N_{\text{flow}} = 1$), $h = 0.05$, and $\lambda = 1$ for all tasks.

4.2 Experimental Results

Figure 2 shows the learning curves, and Table 1 reports the numerical results. T-MAC achieves the highest TAR on all ten environments, covering both MuJoCo locomotion and high-dimensional HumanoidBench humanoid control. Compared with the strongest competing method on each task, T-MAC improves TAR by 17.9% on `Humanoid-v4`, 26.5% on `Swimmer-v4`, 21.5% on `H1hand-reach`, and 21.2% on `H1hand-crawl`. These results show that the advantage of T-MAC extends across both benchmark families. The gains are especially large on whole-body and dexterous-control tasks. T-MAC reaches 7183.7 on `Humanoid-v4` and 6583.8 on `H1hand-reach`, compared with 6090.2 and 5418.5 for the strongest competing baselines, respectively. Together with the highest TAR on the remaining environments, the results show that T-MAC performs strongly from standard locomotion to high-dimensional humanoid control.

4.3 Target-Motion Analysis

Target-motion information. We build two critic histories that end at the same current critic but start from different previous critics, and keep everything else in the update identical (Appendix C.1).



(a) Swimmer-v4 (b) HalfCheetah-v4 (c) Humanoid-v4

(d) Walker2d-v4 (e) Ant-v4 (f) HumanoidStandup-v4

(g) H1hand-crawl (h) H1hand-reach (i) H1-reach (j) H1-maze

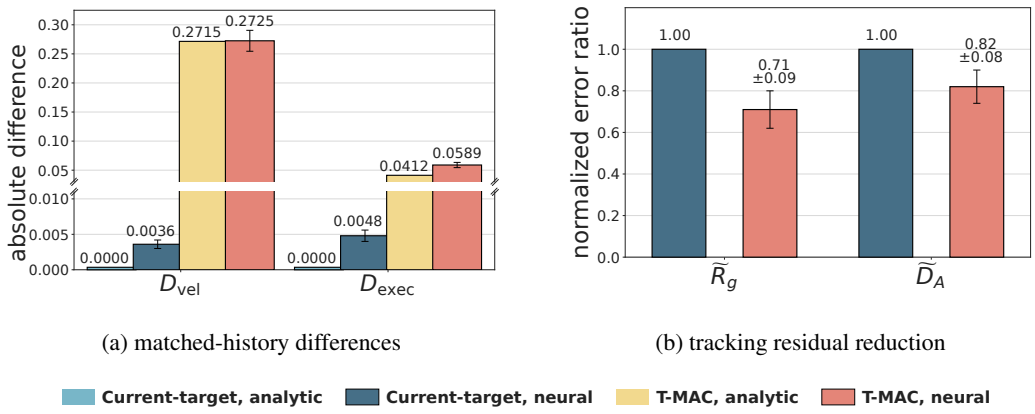
Figure 2: Learning curves on 6 MuJoCo tasks and 4 HumanoidBench tasks with 5 random seeds. Solid lines denote the mean and shaded regions denote ± 1 standard deviation. PPO and TD3 are omitted for visual clarity but included in Table 1.

An update that does not use target motion must give the same result under both histories. We measure the gap between the two updates by the difference in velocity changes, D_{vel}, and the difference in terminal actions, D_{exec}. As shown in Figure 3a, the current-target objective gives almost the same update under both histories, with exactly zero difference in the analytic model and $D_{\text{vel}} = 0.0036$ and $D_{\text{exec}} = 0.0048$ in the neural model. T-MAC gives 0.2725 and 0.0589, showing that it learns from the target motion that the current-target objective cannot see.

Tracking under target-motion supervision. We clone the policy and optimizer state at each checkpoint and apply one current-target update and one T-MAC update with the same critic pair, batch, and random numbers (Appendix C.2). We report the motion residual $\widetilde{R}_g$ and the terminal-action difference $\widetilde{D}_A$, both as ratios over the current-target baseline, so values below one mean better tracking. Figure 3b shows that T-MAC reduces them to 0.71 ± 0.09 and 0.82 ± 0.08.

Teacher direction and source. We compare T-MAC with three controls: 1) **Different sources**, which computes the two responses from the frozen policy and from replay-buffer actions; 2) **Shuffled teacher**, which keeps the teacher magnitude but breaks its alignment with the critic change; 3) **Current target**, which removes the target-motion branch and uses only current-target supervision. As shown in Figure 4a, changing the source distribution, shuffling the teacher, or removing target-motion supervision lowers TAR and increases variation across runs. These results show that both the shared source and the teacher direction are necessary for obtaining the target-motion signal.

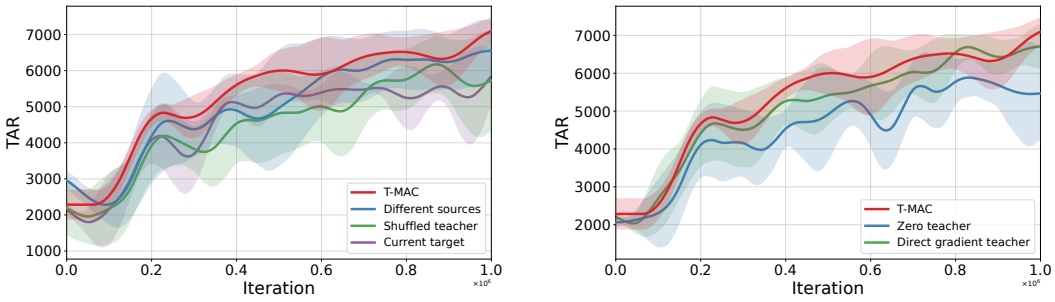
Teacher construction. We then change only the teacher, which separates the effect of its direction from that of the gated penalty. **Zero teacher** sets $r_k^{\text{mot}} = 0$ and keeps the gated penalty, and **Direct gradient teacher** replaces the JKO response difference with its first-order form $\nabla_a Q_{k+1} - \nabla_a Q_k$. As shown in Figure 4b, Zero teacher reaches 5444.2, the same as current-target training, so the gain comes from the teacher direction rather than the penalty. Direct gradient teacher reaches 6672.7 and



(a) matched-history differences

(b) tracking residual reduction

Figure 3: Target-motion diagnostics. (a) Velocity-change and terminal-action differences between two critic histories that share the current critic but differ in the previous critic. The y-axis is broken. (b) Motion residual and terminal-action error of T-MAC, relative to the current-target update.



(a) teacher direction and source

(b) teacher construction

Figure 4: Design ablations on `Humanoid-v4`. Curves show mean TAR over random seeds; shaded regions denote ± 1 standard deviation.

already recovers most of this gain, in line with Theorem 1. The JKO teacher adds another 511.0 with a smaller spread by aligning better with the target displacement, as detailed in Appendix C.3.

5 Conclusion

We present T-MAC, a target-motion-guided method for flow policy optimization in online RL, which augments current-target supervision with a signal for how the policy velocity field should change across updates. To this end, T-MAC measures target motion as the difference between the JKO responses of successive critics on the same source actions, weights it with a response-magnitude gate, and combines the resulting target-motion loss with the current-target loss. Because both responses start from the same source actions, their difference isolates the effect of the critic update. Our analysis explains why this difference captures the action-dependent part of the critic change and connects target-motion tracking to policy improvement. Empirically, T-MAC achieves the highest TAR on all ten tasks across MuJoCo and HumanoidBench, outperforming both model-free and generative-policy baselines from standard locomotion to high-dimensional humanoid control. Matched-history tests and ablations further show that its updates exploit target motion and that the gain needs both the teacher direction and the shared source. Overall, our results show that how the target moves is a learning signal in its own right for online generative policy learning, with particularly strong gains in high-dimensional humanoid control. A natural direction for future work is to extend target-motion supervision to other expressive policy classes such as diffusion policies and to combine it with other current-target constructions.

AI USE STATEMENT

We used generative AI tools to assist with language editing and improve the clarity and organization of the manuscript. AI tools were also used to help discuss theoretical ideas and interpret experimental results during the writing process. The authors reviewed all AI-assisted content and are responsible for the final manuscript.

REFERENCES

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua B. Tenenbaum, Tommi S. Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision making? In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Non-stationary stochastic optimization. *Oper. Res.*, 63(5):1227–1244, 2015.

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

Huayu Chen, Cheng Lu, Chengyang Ying, Hang Su, and Jun Zhu. Offline reinforcement learning via high-fidelity generative behavior modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *Int. J. Robotics Res.*, 44(10-11):1684–1704, 2025.

Shutong Ding, Ke Hu, Zhenhao Zhang, Kan Ren, Weinan Zhang, Jingyi Yu, Jingya Wang, and Ye Shi. Diffusion-based reinforcement learning via q-weighted variational policy optimization. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, Proceedings of Machine Learning Research, pp. 1856–1865. PMLR, 2018.

Eric C. Hall and Rebecca Willett. Dynamical models and tracking regret in online convex programming. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pp. 579–587. JMLR.org, 2013.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Ali Jadbabaie, Alexander Rakhlin, Shahin Shahrampour, and Karthik Sridharan. Online optimization : Competing with dynamic comparators. In Guy Lebanon and S. V. N. Vishwanathan (eds.), *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2015, San Diego, California, USA, May 9-12, 2015*, volume 38 of *JMLR Workshop and Conference Proceedings*. JMLR.org, 2015.

Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker–planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17, 1998.

Qiyang Li and Sergey Levine. Q-learning with adjoint matching. In *The Fourteenth International Conference on Learning Representations*, 2026.

Shengbo Eben Li. *Reinforcement Learning for Sequential Decision and Optimal Control*. Springer Verlag, Singapore, 2023. ISBN 978-981-19-7783-1.

Zechu Li, Rickmer Krohn, Tao Chen, Anurag Ajay, Pulkit Agrawal, and Georgia Chalvatzaki. Learning multimodal behaviors from scratch with diffusion policy gradient. *Advances in Neural Information Processing Systems*, 37:38456–38479, 2024.

Zeyang Li, Sunbochen Tang, and Navid Azizan. Reverse flow matching: A unified framework for online reinforcement learning with diffusion and flow policies. In *Forty-third International Conference on Machine Learning*, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

Lei Lyu, Yunfei Li, Yu Luo, Fuchun Sun, Tao Kong, Jiafeng Xu, and Xiao Ma. Flow-based policy for online reinforcement learning. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025.

Haitong Ma, Tianyi Chen, Kai Wang, Na Li, and Bo Dai. Efficient online reinforcement learning for diffusion policy. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research, 2025.

Haitong Ma, Chenxiao Gao, Tianyi Chen, Na Li, and Bo Dai. Gempo: Generalized measure matching for online diffusion reinforcement learning. *arXiv preprint arXiv:2603.10250*, 2026.

David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow matching policy gradients. In *The Fourteenth International Conference on Learning Representations*, 2026.

David Pfau, Ian Davies, Diana L Borsa, João Guilherme Madeira Araújo, Brendan Daniel Tracey, and Hado van Hasselt. Wasserstein policy optimization. In *Forty-second International Conference on Machine Learning*, 2025.

Michael Psenka, Alejandro Escontrela, Pieter Abbeel, and Yi Ma. Learning a diffusion model policy from rewards via q-score matching. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pp. 41163–41182, 2024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Carmelo Sferrazza, Dun-Ming Huang, Xingyu Lin, Youngwoon Lee, and Pieter Abbeel. Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation. In Dana Kulic, Gentiane Venture, Kostas E. Bekris, and Enrique Coronado (eds.), *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024*, 2024.

Mingyang Sun, Pengxiang Ding, Weinan Zhang, and Donglin Wang. Iterative refinement of flow policies in probability space for online reinforcement learning. *arXiv preprint arXiv:2510.15388*, 2025.

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

Yinuo Wang, Likun Wang, Yuxuan Jiang, Wenjun Zou, Tong Liu, Xujie Song, Wenxuan Wang, Liming Xiao, Jiang Wu, Jingliang Duan, and Shengbo Li. Diffusion actor-critic with entropy regulator. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Zhendong Wang, Jonathan J. Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

Long Yang, Zhixiong Huang, Fenghao Lei, Yucun Zhong, Yiming Yang, Cong Fang, Shiting Wen, Binbin Zhou, and Zhouchen Lin. Policy representation via diffusion probability model for reinforcement learning. *arXiv preprint arXiv:2305.13122*, 2023.

Guojian Zhan, Letian Tao, Pengcheng Wang, Yixiao Wang, Yuxin Chen, Yiheng Li, Hongyang Li, Masayoshi Tomizuka, and Shengbo Eben Li. Mean flow policy with instantaneous velocity constraint for one-step action generation. In *The Fourteenth International Conference on Learning Representations*, 2026.

Tianze Zhu, Yinuo Wang, Wenjun Zou, Tianyi Zhang, Likun Wang, Letian Tao, Feihong Zhang, Yao Lyu, and Shengbo Eben Li. Real-time generative policy via langevin-guided flow matching for autonomous driving. *arXiv preprint arXiv:2603.02613*, 2026a.

Zhaoyu Zhu, Shuhan Zhang, Rui Gao, and Shuang Li. Wasserstein proximal policy gradient. *arXiv preprint arXiv:2603.02576*, 2026b.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In Tom Fawcett and Nina Mishra (eds.), *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, pp. 928–936. AAAI Press, 2003.

A Related Work

A.1 Generative Models as Policies for RL

Diffusion models and flow matching provide expressive policy classes for complex and multimodal action distributions (Ho et al., 2020; Lipman et al., 2023). In online RL, one line of work directly uses critic values, gradients, or advantages to guide generative policy learning (Wang et al., 2024; Lyu et al., 2025; Psenka et al., 2024; Li & Levine, 2026; McAllister et al., 2026). Another line constructs critic-induced target distributions through value-weighted, variational, or measure-matching objectives (Ding et al., 2024; Ma et al., 2025; 2026; Li et al., 2026). Related approaches first construct improved action targets and then train a generative policy to match them (Yang et al., 2023; Zhu et al., 2026a; Zhan et al., 2026). These approaches mainly define policy supervision from the critic or critic-induced target at the current update, while T-MAC adds a cross-update signal by using the transition from Q_k to Q_{k+1} to provide target-motion supervision.

A.2 Optimization under Changing Objectives

Several optimization frameworks study learning under changing objectives. Dynamic-regret methods relate tracking error to the variation of a moving comparator (Zinkevich, 2003; Hall & Willett, 2013; Jadbabaie et al., 2015), while non-stationary stochastic optimization studies objectives that change over time (Besbes et al., 2015). In distribution space, Wasserstein gradient flows and the JKO scheme describe distribution updates under an energy and a transport cost (Jordan et al., 1998; Benamou & Brenier, 2000). Recent RL work uses Wasserstein geometry to construct policy updates (Pfau et al., 2025; Zhu et al., 2026b), and applies the JKO scheme to fine-tune a pre-trained flow policy (Sun et al., 2025). T-MAC applies this distribution-space view to successive critics from the same frozen source distribution and uses the difference between their JKO responses as target-motion supervision.

B Theoretical Analysis

B.1 Setup, Notation, and Assumptions

The analysis considers one policy update with Q_k, Q_{k+1}, θ_k, μ_k, and the current-target training distribution held fixed. This separates the policy update from critic optimization and replay-buffer sampling.

Assumption 1 (Current-Target Information Structure)**.** *There is a measurable current-target statistic $\Xi(\mathcal{H})$ such that the current-target data distribution depends on the critic history $\mathcal{H}$ only through $\Xi(\mathcal{H})$:*

$$P_{\text{base}}(dz \mid \mathcal{H}) = P_{\text{base}}(dz \mid \Xi(\mathcal{H})).$$

The corresponding current-target objective is

$$\mathcal{L}_{\text{cur}}(\theta; \mathcal{H}) = \mathbb{E}_{z \sim P_{\text{base}}(\cdot \mid \Xi(\mathcal{H}))}[\ell(\theta; z)].$$

Assumption 2 (Well-Posed Proximal Response)**.** *For every state and critic considered, (11) has a unique minimizer. The source distribution μ_k is absolutely continuous, the optimal transport map is unique μ_k-almost everywhere, and the response fields have finite second moments.*

Assumption 3 (Flow Regularity and Coverage)**.** *The motion-guided reference field v^{J}_{k+1} is L_x-Lipschitz in its action argument, uniformly over flow time. The policy and reference ODEs are well posed. Under their shared-noise coupling, the sum of their expected terminal numerical-integration errors is at most $\epsilon_{\text{solver}}(s)$. The joint action–flow-time distribution along the policy trajectory is dominated by the conditional training distribution $\chi_k(\cdot \mid s)$ with finite coverage constant C_{cov}. When the motion loss is used to control an unweighted residual, we restrict to active states with $w^{\text{mot}}_k(s) \geq w_{\min} > 0$.*

Assumption 4 (Current-Target Policy-Improvement Margin)**.** *Let $\nu_{k+1}(\cdot \mid s)$ be the current-target distribution represented by Ξ_{k+1}. For the current policy π_k, suppose that $\Delta_{\text{base}}(s) \geq 0$ and $\epsilon_{\text{cert}}(s) \geq 0$ satisfy*

$$\mathbb{E}_{a \sim \nu_{k+1}(\cdot \mid s)}[A^{\pi_k}(s, a)] \geq \Delta_{\text{base}}(s) - \epsilon_{\text{cert}}(s). \tag{25}$$

For the best-of-N_{cand} current-target construction used in the experiments, a uniform critic error ϵ_Q gives $\epsilon_{\text{cert}} = 2\epsilon_Q$ under candidate-set comparison. The analysis also applies to other current-target constructions that provide a corresponding policy-improvement margin.

B.2 Current-Target Objectives Are History-Blind

Figure 1 highlights a simple information difference. If policy supervision depends only on the current-target statistic, critic histories that produce the same statistic define the same population training problem.

Lemma 1 (Current-Target Objectives Are History-Blind)**.** *Under Assumption 1, let histories $\mathcal{H}_A$ and $\mathcal{H}_B$ satisfy $\Xi(\mathcal{H}_A) = \Xi(\mathcal{H}_B)$. Then, for every fixed θ,*

$$\mathcal{L}_{\text{cur}}(\theta; \mathcal{H}_A) = \mathcal{L}_{\text{cur}}(\theta; \mathcal{H}_B). \tag{26}$$

If differentiation and expectation can be interchanged, their population gradients are also equal. Their stochastic-gradient distributions are equal in law under the same sampling procedure.

Proof. Under Assumption 1, both histories give the same current-target data distribution because they share the same value of Ξ. Taking the expectation of the same loss, or its gradient, under this distribution gives the result. □

Here, history refers to critic changes across policy updates. Flow time τ remains the standard interpolation variable used by conditional flow matching.

Proposition 2 (Target Motion Separates Current-Target-Equivalent Histories)**.** *Let $\mathcal{H}_A$ and $\mathcal{H}_B$ have the same current-target statistic but different target-motion responses $r_A^{\text{mot}} \neq r_B^{\text{mot}}$ on a set of positive χ_k measure. Then the two histories define the same current-target objective but retain different information in the full objective.*

Let $\partial_\theta(\delta v_\theta)$ denote the Jacobian of the cross-update velocity change with respect to the policy parameters. Suppose the two histories use the same policy field, frozen policy snapshot, gate, and training distribution. If

$$\mathbb{E}_{\chi_k}\left[w_k^{\text{mot}} \partial_\theta(\delta v_\theta)^\top \left(r_B^{\text{mot}} - r_A^{\text{mot}}\right)\right] \neq 0, \tag{27}$$

then their population gradients under the full objective are different.

Proof. Lemma 1 gives the same current-target objective for both histories. The difference between their motion-loss gradients is, up to sign,

$$2\lambda \mathbb{E}_{\chi_k}\left[w_k^{\text{mot}} \partial_\theta(\delta v_\theta)^\top \left(r_B^{\text{mot}} - r_A^{\text{mot}}\right)\right].$$

By (27), this term is nonzero. Therefore, the full objective produces different population gradients for the two histories. □

B.3 Shared-Source Target-Motion Recovery

Fix a state s. Let $\mathcal{A}$ be a compact action domain with periodic or no-flux boundary conditions. Let $\mu_k = \rho_k\, da$ and $\varpi_s = w_s\, da$, where ρ_k and w_s are positive C^3 densities. Assume that $Q_k(s,\cdot)$ and $Q_{k+1}(s,\cdot)$ are C^3, and that the corresponding JKO minimizers and optimal transport maps are smooth for all sufficiently small h. Assume that their densities and first two derivatives are uniformly bounded and that the densities are bounded away from zero.

Proof of Theorem 1. For a density ρ, the first variation of the free energy is

$$\frac{\delta \mathcal{E}_Q}{\delta \rho} = -Q(s,a) + \alpha\left(\log \frac{\rho(a)}{w_s(a)} + 1\right).$$

The corresponding Wasserstein gradient-flow velocity is

$$u_Q[\rho](a) = -\nabla_a \frac{\delta \mathcal{E}_Q}{\delta \rho} = \nabla_a Q(s,a) - \alpha \nabla_a \log \frac{\rho(a)}{w_s(a)}.$$

Let ρ_h^Q be the density of the JKO minimizer and T_h^Q the optimal map from μ_k to ρ_h^Q. The Euler–Lagrange condition gives

$$T_h^Q(a) = a + h\, u_Q[\rho_h^Q]\big(T_h^Q(a)\big),$$

and therefore

$$u_h^Q(a \mid s) = u_Q[\rho_h^Q]\big(T_h^Q(a)\big).$$

Under the stated regularity conditions,

$$\|T_h^Q - \mathrm{id}\|_{L^\infty} = O(h), \qquad \|\rho_h^Q - \rho_k\|_{C^1} = O(h).$$

The smoothness assumptions then give

$$\Big\| u_h^Q(\cdot \mid s) - u_Q[\rho_k] \Big\|_{L^2(\mu_k(\cdot|s))} \le C_Q(s)h.$$

Applying this result to Q_k and Q_{k+1} gives

$$r_k^{\mathrm{mot}} = u_{Q_{k+1}}[\rho_k] - u_{Q_k}[\rho_k] + O_{L^2(\mu_k(\cdot|s))}(h).$$

Because both responses use the same source density ρ_k and reference density w_s,

$$u_{Q_{k+1}}[\rho_k] - u_{Q_k}[\rho_k] = \nabla_a \delta Q_k(s, \cdot).$$

Combining the two relations proves the result. □

Corollary 1 (Source Drift under Different Sources)**.** *Suppose the two critics are evaluated from $\mu_k = \rho_k\, da$ and $\widetilde{\mu}_k = \widetilde{\rho}_k\, da$ under the same regularity conditions. Then*

$$u_h^{Q_{k+1}}[\widetilde{\mu}_k] - u_h^{Q_k}[\mu_k] = \nabla_a \delta Q_k(s, \cdot) - \alpha \nabla_a \log \frac{\widetilde{\rho}_k}{\rho_k} + O_{L^2}(h). \tag{28}$$

Thus, changing the source distribution adds a source-drift term to the first-order response, while the shared-source construction removes this term.

Response magnitude. Theorem 1 also gives the explicit bound

$$\left| \sqrt{e_k^{\mathrm{mot}}(s)} - \|\nabla_a \delta Q_k(s, \cdot)\|_{L^2(\mu_k(\cdot|s))} \right| \le C(s)h. \tag{29}$$

Thus, the response magnitude measures the size of the target motion, which follows the action-dependent part of the critic change to first order.

The JKO response also has two exact finite-step properties: it is the minimum-displacement response to its proximal distribution, and it is unchanged by an action-independent critic shift.

Proposition 3 (Minimum-Displacement Response and Shift Invariance)**.** *Under Assumption 2, the map T_h^Q minimizes*

$$\int \|T(a) - a\|_2^2\, \mu_k(da \mid s)$$

among measurable maps that transport $\mu_k(\cdot \mid s)$ to the proximal distribution ν_h^Q. Hence, (12) gives the minimum-displacement response associated with this proximal distribution.

For any state-only scalar $c(s)$,

$$\nu_h^{Q+c} = \nu_h^Q, \qquad u_h^{Q+c} = u_h^Q$$

μ_k-almost everywhere.

Proof. The first result follows from the definition of the squared Wasserstein distance and the uniqueness of the optimal transport map. For the second, every probability distribution ν satisfies

$$\mathcal{E}_{Q+c}(\nu \mid s) = \mathcal{E}_Q(\nu \mid s) - c(s).$$

Since $c(s)$ does not depend on ν, the proximal minimizer and its optimal transport map remain unchanged. □

B.4 MOTION-RESIDUAL REDUCTION

Let $\mathcal{X}$ be a Hilbert space of admissible cross-update velocity fields $x = \delta v_\theta$, with norm induced by χ_k. Define

$$L_{\mathrm{cur}}(x) := \mathcal{L}_{\mathrm{cur}}(x), \qquad R_{\mathrm{mot}}(x) := \frac{1}{2}\mathbb{E}_{\chi_k}\left[w_k^{\mathrm{mot}}(s)\left\|x - h\,\bar{r}_k^{\mathrm{mot}}\right\|_2^2\right]. \tag{30}$$

Here, L_{cur} is the current-target loss and R_{mot} is the gate-weighted motion residual. A parameterized policy represents a subset of $\mathcal{X}$.

Theorem 3 (Motion-Residual Reduction)**.** *Assume that L_{cur} and R_{mot} in (30) are lower semicontinuous and that the minimizers below exist. Let*

$$L^* = \inf_{x\in\mathcal{X}} L_{\mathrm{cur}}(x), \qquad \mathcal{S} = \arg\min_{x\in\mathcal{X}} L_{\mathrm{cur}}(x),$$

and choose

$$x_{\mathrm{mot}} \in \arg\min_{x\in\mathcal{S}} R_{\mathrm{mot}}(x), \qquad x_\lambda \in \arg\min_{x\in\mathcal{X}}\{L_{\mathrm{cur}}(x) + 2\lambda R_{\mathrm{mot}}(x)\},$$

for $\lambda > 0$. Then, for every current-target-only minimizer $x_0 \in \mathcal{S}$,

$$R_{\mathrm{mot}}(x_\lambda) \le R_{\mathrm{mot}}(x_{\mathrm{mot}}) \le R_{\mathrm{mot}}(x_0), \tag{31}$$

$$0 \le L_{\mathrm{cur}}(x_\lambda) - L^* \le 2\lambda\big(R_{\mathrm{mot}}(x_{\mathrm{mot}}) - R_{\mathrm{mot}}(x_\lambda)\big) \le 2\lambda R_{\mathrm{mot}}(x_{\mathrm{mot}}). \tag{32}$$

If $R_{\mathrm{mot}}(x_{\mathrm{mot}}) = 0$, the full objective reaches both the current-target minimum and zero motion residual.

Moreover, $R_{\mathrm{mot}}(x_\lambda) < R_{\mathrm{mot}}(x_0)$ if either $R_{\mathrm{mot}}(x_{\mathrm{mot}}) < R_{\mathrm{mot}}(x_0)$, or L_{cur} has a unique minimizer x_0, both functions are Fréchet differentiable at x_0, and $\nabla R_{\mathrm{mot}}(x_0) \neq 0$.

Proof. By the optimality of x_λ compared with x_{mot},

$$L_{\mathrm{cur}}(x_\lambda) + 2\lambda R_{\mathrm{mot}}(x_\lambda) \le L^* + 2\lambda R_{\mathrm{mot}}(x_{\mathrm{mot}}).$$

Since $L_{\mathrm{cur}}(x_\lambda) \ge L^*$, this gives (31) and (32). If $R_{\mathrm{mot}}(x_{\mathrm{mot}}) = 0$, both nonnegative gaps are zero.

If $R_{\mathrm{mot}}(x_{\mathrm{mot}}) < R_{\mathrm{mot}}(x_0)$, the strict reduction follows directly. For the unique-minimizer case, $x_\lambda = x_0$ would violate the first-order condition of the full objective because $\nabla L_{\mathrm{cur}}(x_0) = 0$ and $\nabla R_{\mathrm{mot}}(x_0) \neq 0$. Hence $x_\lambda \neq x_0$. Since x_0 is the unique minimizer of L_{cur}, $L_{\mathrm{cur}}(x_\lambda) > L^*$, and (32) gives $R_{\mathrm{mot}}(x_\lambda) < R_{\mathrm{mot}}(x_0)$. □

Proposition 4 (Approximate Optimization)**.** *Suppose $\widehat{x}_0$ is δ_0-suboptimal for L_{cur} and $\widehat{x}_\lambda$ is δ_λ-suboptimal for $L_{\mathrm{cur}} + 2\lambda R_{\mathrm{mot}}$. Then*

$$R_{\mathrm{mot}}(\widehat{x}_\lambda) \le \min\left\{R_{\mathrm{mot}}(\widehat{x}_0) + \frac{\delta_0 + \delta_\lambda}{2\lambda},\; R_{\mathrm{mot}}(x_{\mathrm{mot}}) + \frac{\delta_\lambda}{2\lambda}\right\}. \tag{33}$$

Consequently,

$$R_{\mathrm{mot}}(\widehat{x}_0) - R_{\mathrm{mot}}(x_{\mathrm{mot}}) > \frac{\delta_\lambda}{2\lambda} \quad\Longrightarrow\quad R_{\mathrm{mot}}(\widehat{x}_\lambda) < R_{\mathrm{mot}}(\widehat{x}_0). \tag{34}$$

Proof. Comparing $\widehat{x}_\lambda$ with $\widehat{x}_0$ gives

$$L_{\mathrm{cur}}(\widehat{x}_\lambda) + 2\lambda R_{\mathrm{mot}}(\widehat{x}_\lambda) \le L_{\mathrm{cur}}(\widehat{x}_0) + 2\lambda R_{\mathrm{mot}}(\widehat{x}_0) + \delta_\lambda.$$

Using $L_{\mathrm{cur}}(\widehat{x}_\lambda) \ge L^*$ and $L_{\mathrm{cur}}(\widehat{x}_0) \le L^* + \delta_0$ gives the first bound in (33).

Comparing $\widehat{x}_\lambda$ with x_{mot} gives

$$L_{\mathrm{cur}}(\widehat{x}_\lambda) + 2\lambda R_{\mathrm{mot}}(\widehat{x}_\lambda) \le L^* + 2\lambda R_{\mathrm{mot}}(x_{\mathrm{mot}}) + \delta_\lambda.$$

Using $L_{\mathrm{cur}}(\widehat{x}_\lambda) \ge L^*$ gives the second bound. Condition (34) then gives the strict motion-residual reduction. □

B.5 From Motion Tracking to Executed Actions

Lemma 2 (Action-Space Stability under Flow Integration). *Under Assumption 3, couple the policy flow and the motion-guided reference flow in (20) with the same initial action. Let*

$$\epsilon^2_{\text{motion}}(\theta; s) := \mathbb{E}_{(a,\tau)\sim\chi_k(\cdot|s)} \left\| \delta v_\theta(a, \tau, s) - h\, \bar{r}^{\text{mot}}_k(a, \tau, s) \right\|_2^2 . \tag{35}$$

For a fixed state, the gate-weighted contribution in (30) is

$$R_{\text{mot}}(\theta; s) = \frac{1}{2} w^{\text{mot}}_k(s)\, \epsilon^2_{\text{motion}}(\theta; s). \tag{36}$$

Hence, on active states with $w^{\text{mot}}_k(s) \geq w_{\min} > 0$, $\epsilon_{\text{motion}}(\theta; s) \leq \sqrt{2R_{\text{mot}}(\theta; s)/w_{\min}}$. Let $\widehat{X}_\theta(1)$ and $\widehat{X}_{\text{J}}(1)$ be their numerically integrated terminal actions under the same initial draw. Define the expected terminal-action difference as

$$D_A(\theta; s) = \mathbb{E}\left[\|\widehat{X}_\theta(1) - \widehat{X}_{\text{J}}(1)\|_2 \right]. \tag{37}$$

Then

$$W_1\big(\pi_\theta(\cdot \mid s), \pi^{\text{J}}_{k+1}(\cdot \mid s)\big) \leq D_A(\theta; s) \leq e^{L_x} C^{1/2}_{\text{cov}} \epsilon_{\text{motion}}(\theta; s) + \epsilon_{\text{solver}}(s). \tag{38}$$

Proof. Let X_θ and X_{J} be the coupled exact ODE trajectories and write $v_{\text{J}} = v_k + h\, \bar{r}^{\text{mot}}_k$. Adding and subtracting $v_{\text{J}}(X_\theta, \tau, s)$ gives

$$\frac{d}{d\tau}\|X_\theta - X_{\text{J}}\| \leq \|v_\theta(X_\theta) - v_{\text{J}}(X_\theta)\| + L_x \|X_\theta - X_{\text{J}}\|.$$

Gronwall's inequality and Cauchy–Schwarz over τ bound the expected terminal difference by e^{L_x} times the root-mean-square motion residual along the policy trajectory. The coverage assumption then gives $C^{1/2}_{\text{cov}} \epsilon_{\text{motion}}$. Adding the numerical integration error gives the bound on D_A. Finally, the shared-noise coupling is a valid coupling, so its expected terminal-action difference upper-bounds W_1. □

B.6 Proof of Theorem 2

This section proves Theorem 2. We first bound the reference–target difference $\epsilon_J(s)$ in Eq. (23) by the alignment between the target-motion teacher and the displacement that the current-target distribution asks for. Let T^ν_s be the optimal transport map from $\mu_k(\cdot \mid s)$ to $\nu_{k+1}(\cdot \mid s)$, and define the target displacement field and its flow-time extension as

$$d_k(a \mid s) := \frac{T^\nu_s(a \mid s) - a}{h}, \qquad \bar{d}_k(a, \tau, s) := d_k(a \mid s). \tag{39}$$

Let $v^\nu_{k+1} := v_k + h\, \bar{d}_k$ and $\pi^\nu_{k+1}(\cdot \mid s) := \Phi_s(v^\nu_{k+1})$ be the corresponding reference field and terminal action distribution. Away from the source particles, $\bar{d}_k$ uses the same action-space interpolation as $\bar{r}^{\text{mot}}_k$. Define the teacher alignment error and the displacement error as

$$\epsilon_{\text{align}}(s) := h \left\| \bar{r}^{\text{mot}}_k - \bar{d}_k \right\|_{L^2(\chi_k(\cdot|s))}, \qquad \epsilon_{\text{disp}}(s) := W_1\big(\pi^\nu_{k+1}(\cdot \mid s),\, \nu_{k+1}(\cdot \mid s)\big). \tag{40}$$

Lemma 3 (Alignment Bound for $\epsilon_J(s)$). *Under Assumption 3, for every state s,*

$$\epsilon_J(s) \;\leq\; C_{\text{flow}}(s)\, \epsilon_{\text{align}}(s) + \epsilon_{\text{solver}}(s) + \epsilon_{\text{disp}}(s). \tag{41}$$

Proof. By the triangle inequality for W_1,

$$\epsilon_J(s) \leq W_1\big(\pi^{\text{J}}_{k+1}(\cdot \mid s), \pi^\nu_{k+1}(\cdot \mid s)\big) + W_1\big(\pi^\nu_{k+1}(\cdot \mid s), \nu_{k+1}(\cdot \mid s)\big).$$

The second term is $\epsilon_{\text{disp}}(s)$ by definition. For the first term, π^{J}_{k+1} and π^ν_{k+1} are generated by the velocity fields $v_k + h\bar{r}^{\text{mot}}_k$ and $v_k + h\bar{d}_k$ from the same initial noise. Their velocity difference is $h(\bar{r}^{\text{mot}}_k - \bar{d}_k)$, so Lemma 2 applied to this pair gives

$$W_1\big(\pi^{\text{J}}_{k+1}(\cdot \mid s), \pi^\nu_{k+1}(\cdot \mid s)\big) \leq C_{\text{flow}}(s)\, \epsilon_{\text{align}}(s) + \epsilon_{\text{solver}}(s),$$

which proves the claim. □

The term $\epsilon_{\text{disp}}(s)$ represents the error of replacing the optimal transport map T_s^ν with a single constant-velocity flow step. This error vanishes when $d_k(\cdot \mid s)$ is constant over the source, and it is determined by the current-target construction. Substituting (41) into Theorem 2 replaces ϵ_J by the alignment error, which Appendix C.3 measures.

Proof of Theorem 2. By the triangle inequality and Lemma 2,

$$\begin{aligned} W_1\left(\pi^\lambda_{k+1}(\cdot \mid s), \nu_{k+1}(\cdot \mid s)\right) &\le W_1\left(\pi^\lambda_{k+1}(\cdot \mid s), \pi^{\text{J}}_{k+1}(\cdot \mid s)\right) + \epsilon_J(s) \\ &\le D_A(\theta_\lambda; s) + \epsilon_J(s). \end{aligned}$$

Since $A^{\pi_k}(s,\cdot)$ is L_A-Lipschitz,

$$\begin{aligned} \mathbb{E}_{a\sim\pi^\lambda_{k+1}(\cdot|s)}[A^{\pi_k}(s,a)] \ge \mathbb{E}_{a\sim\nu_{k+1}(\cdot|s)}[A^{\pi_k}(s,a)] \\ - L_A W_1\left(\pi^\lambda_{k+1}(\cdot \mid s), \nu_{k+1}(\cdot \mid s)\right). \end{aligned}$$

Applying (25) gives

$$\mathbb{E}_{a\sim\pi^\lambda_{k+1}(\cdot|s)}[A^{\pi_k}(s,a)] \ge \Delta_{\text{base}}(s) - \epsilon_{\text{cert}}(s) - L_A\epsilon_J(s) - L_A D_A(\theta_\lambda; s).$$

Substituting this bound into the performance-difference identity and taking the expectation under $d^{\pi^\lambda_{k+1}}$ proves the policy-improvement bound. The bound on D_A follows from Lemma 2 under Assumption 3. □

The current-target distribution provides the base improvement margin $\Delta_{\text{base}} - \epsilon_{\text{cert}}$, while target-motion tracking controls the action-space tracking term D_A. Since D_A is bounded by the motion residual, better target-motion tracking gives a tighter policy-improvement bound.

C Extended Target-Motion Analysis

C.1 Matched-History Protocol and Full Results

This controlled test asks whether a policy update can distinguish two critic histories with the same current target but different target motion. We construct

$$Q_0 = (0,0,0), \qquad Q_{\text{mid}} = (0,1,0), \qquad Q_{\text{final}} = (2,0,0),$$

and compare the two critic changes

$$\mathcal{H}_A : Q_0 \to Q_{\text{final}}, \qquad \mathcal{H}_B : Q_{\text{mid}} \to Q_{\text{final}}.$$

Both histories use Q_{final} as the current critic but have different previous critics.

To isolate this difference, the two cases use the same policy and optimizer states, source distribution, training samples, random numbers, current-target objective, and optimization budget. Only the previous critic used by the target-motion teacher changes. We compare the current-target objective with $\lambda = 0$ and T-MAC with $\lambda = 1$.

Let $\delta v_\theta^{(A)}$ and $\delta v_\theta^{(B)}$ denote the cross-update velocity changes under the two histories. We measure their difference as

$$D_{\text{vel}} := \left(\mathbb{E}_{(s,a,\tau)\sim\chi_k}\left[\|\delta v_\theta^{(A)}(a,\tau,s) - \delta v_\theta^{(B)}(a,\tau,s)\|_2^2\right]\right)^{1/2}. \tag{42}$$

We also integrate both learned velocity fields from the same initial noise and measure the terminal-action difference

$$D_{\text{exec}} = \mathbb{E}_{z\sim p_0}\left[\|\widehat{a}_A(z) - \widehat{a}_B(z)\|_2\right], \tag{43}$$

where $\widehat{a}_A(z)$ and $\widehat{a}_B(z)$ are the generated actions. This measures whether the difference in the velocity fields remains after flow integration.

We evaluate an analytic three-action simplex model and a neural velocity model. The neural model is a 3-to-32-to-3 MLP trained for the same number of steps in both histories, with results averaged over eight runs. The resulting measurements are reported in Figure 3a.

C.2 Paired Checkpoint Comparison

At each checkpoint, we clone the policy and optimizer state and apply one current-target update and one T-MAC update using the same critic pair, batch, source samples, and random numbers. Let $R_{g,j} = 2R_{\mathrm{mot,j}} := \mathbb{E}[w_k^{\mathrm{mot}} \|\delta v_\theta - h\,\bar{r}_k^{\mathrm{mot}}\|_2^2]$ be the gated motion residual at checkpoint j. The factor of two matches the definition of R_{mot} in Eq. (30). Let $D_{A,j}$ be the terminal-action tracking error under the same initial-noise coupling, as defined in Appendix B.5.

We report

$$\widetilde{R}_{g,j} = \frac{R_{g,j}^{\mathrm{full}}}{R_{g,j}^{\mathrm{base}} + \varepsilon}, \qquad \widetilde{D}_{A,j} = \frac{D_{A,j}^{\mathrm{full}}}{D_{A,j}^{\mathrm{base}} + \varepsilon},$$

and average these ratios over checkpoints.

The aggregate ratios are reported in Figure 3b. The current-target update is normalized to one for each metric.

C.3 Target-Motion Teacher Consistency

This section evaluates the alignment error $\epsilon_{\mathrm{align}}$ from Lemma 3. We measure it for the JKO target-motion response and for two controls. The direct gradient teacher replaces the JKO response difference with $\nabla_a Q_{k+1} - \nabla_a Q_k$. The shuffled control permutes teacher vectors across states.

For each checkpoint pair, we fix the evaluation states and compute the selected actions under Q_k and Q_{k+1}. Their difference is the target displacement $h\,\bar{d}_k$ in Eq. (40), and we compare it with the scaled teacher $h\,\bar{r}_k^{\mathrm{mot}}$. The relative error is $\epsilon_{\mathrm{align}}$ normalized by the target displacement. The direction cosine and the magnitude correlation separate the directional and the scale components of this error. Active states are the states with $w_k^{\mathrm{mot}}(s) \geq w_{\min}$, and we use $w_{\min} = 0.5$. Lemma 2 controls the unweighted residual on this set, so the fraction of active states is the fraction on which that bound applies.

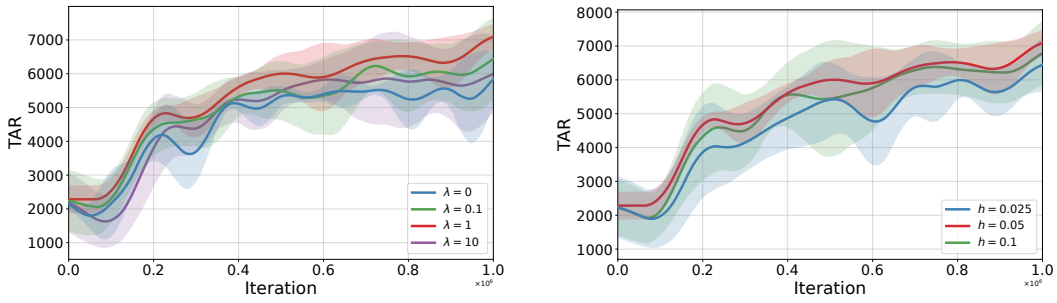
Table 2 reports the results. The JKO teacher gives a relative error of 0.69, so $\epsilon_{\mathrm{align}}$ is about 0.69 of the target displacement. The direct gradient teacher is worse on all three alignment measures. The shuffled control raises the relative error to 1.19, which exceeds the displacement itself and gives no useful bound in Lemma 3. It reports the same fraction of active states as the JKO teacher because permuting the teacher vectors leaves the set of response energies unchanged.

Table 2: Alignment between each target-motion teacher and the measured best-of-N_{cand} action difference. Values are means $\pm$ standard deviations across three checkpoint groups.

Teacher	Active states (%) ↑	Direction cosine ↑	Relative error ↓	Magnitude correlation ↑
JKO teacher	71.8 ± 3.6	0.81 ± 0.07	0.69 ± 0.10	0.57 ± 0.09
Direct gradient	68.5 ± 4.2	0.68 ± 0.09	0.84 ± 0.12	0.43 ± 0.11
Shuffled teacher	71.8 ± 3.6	0.04 ± 0.06	1.19 ± 0.14	0.02 ± 0.08

C.4 Hyperparameter sensitivity

Figure 5 shows sensitivity to the motion weight λ, the proximal step h, the flow-integration budget N_{flow}, and the gate constant κ on `Humanoid-v4`. Setting $\lambda = 0$ removes the motion branch and gives the current-target objective of Section 4. Setting $\kappa = 0$ gives $w_k^{\mathrm{mot}}(s) = 1$ on every state and recovers the ungated motion loss. Among the tested settings, the default $\lambda = 1$, $h = 0.05$, $N_{\mathrm{flow}} = 1$, and $\kappa = 10^{-2}$ gives the best overall result. This setting already uses one-step flow integration, so target-motion supervision requires no extra inference-time flow steps. We use the same default setting throughout the main benchmark.



(a) sensitivity to λ

(b) sensitivity to h

(c) sensitivity to N_{flow}

(d) sensitivity to κ

Figure 5: Hyperparameter sensitivity on `Humanoid-v4`. Curves show mean TAR over random seeds, and shaded regions show ± 1 standard deviation across runs. The default setting uses $\lambda = 1$, $h = 0.05$, $N_{\text{flow}} = 1$, and $\kappa = 10^{-2}$.

D Implementation Details

D.1 Current-Target Construction

For each state s, we sample N_{cand} candidate actions $a^i \sim \pi_{\theta_k}(\cdot \mid s)$ and select

$$a^{\text{cur}}(s) = a^{i^\star}, \qquad i^\star = \arg\max_{1 \le i \le N_{\text{cand}}} Q_{k+1}(s, a^i).$$

The empirical law of the selected actions is $\nu_{k+1}(\cdot \mid s)$. It supplies a^{cur} in the CFM loss (8).

D.2 T-MAC Training Protocol

At policy update k, Q_k is the critic snapshot before the critic update, and Q_{k+1} is the updated critic. The policy snapshot and source distribution μ_k are frozen during the policy update. The current target is constructed from Q_{k+1}, while the motion branch computes the two critic responses using the same μ_k. The teacher, gate, and critic quantities are treated as stop-gradient values. When $\lambda = 0$, the motion loss and JKO teacher computation are skipped. This gives the best-of-N_{cand} current-target defined by (8).

D.3 JKO Target-Motion Teacher and Solver

At policy update k, we freeze the source distribution $\mu_k(\cdot \mid s) = \pi_{\theta_k}(\cdot \mid s)$ at the beginning of the policy update. For each state, we draw $N_{\text{src}} = 64$ source particles and reuse the same particles for the proximal problems associated with Q_k and Q_{k+1}. For clarity, the implementation uses a regularized particle approximation to the Euler–Lagrange condition of Eq. (11). It does not evaluate the KL divergence between an atomic empirical measure and a continuous reference measure directly. Let a_i be a source particle and x_i its transported location, and define the Gaussian

KDE

$$\widehat{\rho}_{x,\sigma}(a) := \frac{1}{N_{\mathrm{src}}} \sum_{j=1}^{N_{\mathrm{src}}} \phi_\sigma(a - x_j), \qquad \phi_\sigma(z) := (2\pi\sigma^2)^{-d_a/2} \exp\left(-\frac{\|z\|_2^2}{2\sigma^2}\right). \tag{44}$$

The corresponding particle score is $\nabla_a \log \widehat{\rho}_{x,\sigma}(a)$. Since $\varpi_s = \mathcal{U}([-1,1]^{d_a})$ has constant density in the interior of the normalized action box, $\nabla_a \log \varpi_s(a) = 0$ there. Thus, for each critic Q, the particle locations are updated by the damped fixed-point scheme

$$x_i^{(\ell+1)} = x_i^{(\ell)} + \beta \left[\mathrm{clip}_{[-1,1]^{d_a}} \left\{ a_i + h \left(\nabla_a Q(s, x_i^{(\ell)}) - \alpha \nabla_a \log \widehat{\rho}_{x^{(\ell)},\sigma}(x_i^{(\ell)}) \right) \right\} - x_i^{(\ell)} \right], \quad x_i^{(0)} = a_i. \tag{45}$$

The index-wise source–particle coupling is retained throughout the solve, so the resulting method is a fixed-coupling particle approximation rather than an exact empirical W_2 optimization. The KDE bandwidth is fixed for a solve using the source-particle scale, and the box projection enforces the normalized action bounds. The projection is a numerical constraint; the continuous regularity results are interpreted for the interior response under their stated boundary assumptions. The solver runs for at most $S_0 = 50$ iterations and stops when the relative fixed-point residual is below 10^{-3}.

Let $\widehat{T}_h^Q(a_i) := x_i^{(S)}$ denote the resulting particle transport map. We compute the response on each source particle as

$$\widehat{u}_h^Q(a \mid s) := \frac{\widehat{T}_h^Q(a \mid s) - a}{h},$$

and form the target-motion teacher as

$$\widehat{r}_k^{\mathrm{mot}}(a \mid s) := \widehat{u}_h^{Q_{k+1}}(a \mid s) - \widehat{u}_h^{Q_k}(a \mid s).$$

The two responses use the same source particles. For a policy flow state a_τ, the response is extended from the source particles by the normalized Gaussian interpolation

$$\widehat{r}_k^{\mathrm{mot}}(a_\tau \mid s) := \sum_{i=1}^{N_{\mathrm{src}}} \omega_i(a_\tau) \widehat{r}_k^{\mathrm{mot}}(a_i \mid s), \qquad \omega_i(a_\tau) := \frac{\exp\left(-\|a_\tau - a_i\|_2^2/(2\sigma_I^2)\right)}{\sum_j \exp(-\|a_\tau - a_j\|_2^2/(2\sigma_I^2))}. \tag{46}$$

This interpolation is in action space and is independent of flow time except through the query point a_τ; it is not an interpolation over JKO solver iterations. In the motion loss, both $v_\theta(a_\tau, \tau, s)$ and $v_k(a_\tau, \tau, s)$ are evaluated at this same input, as required by Eq. (9). The target-motion response and the gate are treated as stop-gradient quantities. We use $g(x) = x/(x + \kappa)$ with $\kappa = 10^{-2}$, applied to the response energy divided by its median over the states in the update, and we use $w_{\min} = 0.5$ in Assumption 3. The median makes the argument of g dimensionless, and κ is the fraction of the median energy at which supervision is halved. When $\lambda = 0$, both JKO solves and the motion loss are skipped. The JKO solver is used only during training and is not used at inference time.

D.4 Baseline Configurations

We run each baseline with the configuration recommended by its paper or public implementation, except for parameters that determine the environment-interaction budget. We match the number of transitions collected per optimization iteration across methods to keep the total interaction budget consistent. Method-specific settings, such as learning rates, follow each method's recommendation.

For DPMD, the critic learning rate is 3×10^{-4} and the policy learning rate is linearly annealed from 3×10^{-4} to 3×10^{-5} (Ma et al., 2025). For FlowRL, both the policy and critic learning rates are 3×10^{-4} (Lyu et al., 2025).

D.5 T-MAC Hyperparameters

Table 4 reports the default T-MAC configuration, including training settings, flow-policy parameters, target-motion parameters, and network architectures. Unless otherwise specified, the same configuration is used across all MuJoCo and HumanoidBench tasks. Learning rates follow the FlowRL protocol above, and the motion branch uses $\lambda = 1$ by default. The response-magnitude gate follows (16). The source distribution is $\mu_k(\cdot \mid s) = \pi_{\theta_k}(\cdot \mid s)$, frozen at the beginning of policy update k.

Table 3: Key hyperparameters of generative-policy baselines. Shared settings are described in the text, and remaining method-specific settings follow official or released default configurations.

Parameter	DACER	DPMD	QVPO	DIPO	QSM	FlowRL
Policy architecture	3×256	3×256	3×256	3×256	3×256	3×256
Critic architecture	3×256	3×256	3×256	3×256	3×256	3×256
Policy learning rate	1×10^{-4}	linear: $3\times10^{-4}\rightarrow3\times10^{-5}$	1×10^{-4}	1×10^{-4}	1×10^{-4}	3×10^{-4}
Critic learning rate	1×10^{-4}	3×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}	3×10^{-4}
Diffusion / flow sampling steps	20	20	20	20	20	1
Action gradient steps	–	–	–	20	–	–

Table 4: Hyperparameter settings of T-MAC. Best-of-N_{cand} is the number of action candidates used by the current-target branch; N_{src} is the number of shared-source particles used for the two critic responses.

Hyperparameter	Value	Hyperparameter	Value
Training			
Critic learning rate	3×10^{-4}	Policy learning rate	3×10^{-4}
Optimizer	Adam	Discount factor γ	0.99
Target smoothing coefficient	5×10^{-3}	Batch size	256
Replay buffer capacity	1×10^{6}	Buffer warm-up size	3×10^{4} steps
Reward scale	0.2	Policy update interval	2 steps
Policy			
Flow integration steps N_{flow}	1	Motion weight λ	1
Proximal step h	0.05	Action bound	$[-1,1]$
Best-of-N_{cand} candidate number	$N_{\mathrm{cand}}=32$	Source particles per state	$N_{\mathrm{src}}=64$
KL coefficient α	0.1	JKO stopping tolerance	10^{-3} (rel. residual)
Source distribution: $\mu_k(\cdot\mid s)=\pi_{\theta_k}(\cdot\mid s)$, frozen at the start of policy update k			
Reference measure: $\varpi_s=\mathcal{U}([-1,1]^{d_a})$			
JKO initialization: $\nu^{(0)}=\mu_k$ (identity transport map); source particles reused for both critic responses			
Architecture			
Policy hidden layers	3	Critic hidden layers	3
Policy hidden width	256	Critic hidden width	256
Policy/Critic activation	Mish	Time embedding (flow)	sinusoidal, $\dim=16$

We draw $N_{\mathrm{src}}=64$ source particles per state and reuse the same particles for the responses of Q_k and Q_{k+1}. The JKO solver is initialized with the source particles and runs for at most 50 iterations with a relative residual tolerance of 10^{-3}. The reference measure is uniform on the normalized action space. We use $\alpha=0.1$ and $\kappa=10^{-2}$.

Reproducibility Statement We provide the core source code in the supplementary ZIP attachment and include detailed implementation settings and hyperparameters in the supplementary material. All experiments are conducted with five random seeds. The implementation will be publicly available upon publication.

E LIMITATIONS

T-MAC is developed for flow-based generative policies, where policy changes can be represented through velocity-field updates. This formulation allows target-motion supervision to directly guide the evolution of the policy velocity field. Extending target-motion supervision to other generative policy architectures, such as diffusion-based policies, requires designing suitable representations for policy changes and is an interesting direction for future work.