PRETRAINING FOUNDATION POLICIES FOR PERCEPTIVE HUMANOID LOCOMOTION WITH OFFLINE RL

Anonymous authors
Paper under double-blind review

ABSTRACT

Online reinforcement learning has enabled perceptive humanoids to traverse challenging terrain and to transfer these skills to real hardware. Scaling this paradigm across terrains and robot embodiments remains costly, because each new setting requires substantial environment interaction and careful tuning. Recent approaches reduce this cost in two stages. They first train specialist policies online, then distill their rollouts offline into a more general policy through behavior cloning (BC). BC, however, ignores the recorded returns and imitates every behavior equally, which limits what it can learn from the rollouts of experts of varying quality. We study whether perceptive humanoid locomotion policies that span terrains and embodiments can be learned entirely from such offline data. To this end, we introduce Segment-level Offline Imitation of Diverse Experts (SOFIE), an offline reinforcement learning approach based on advantage-weighted regression that uses the recorded rewards to estimate footstep-level advantages and preferentially imitates the better footsteps. Quantile normalization of the advantages within each robot–terrain group keeps the resulting weights consistent across robots, terrains and reward scales. Across twelve humanoids in two simulators, SOFIE outperforms behavior cloning and offline reinforcement learning baselines with policies shared across robots or terrains, including one perceptive policy for three humanoids on four terrains. When fine-tuned on a small dataset from an unseen humanoid, a policy pretrained with SOFIE also outperforms training from scratch. These results suggest that perceptive foundation policies for humanoid locomotion can be pretrained from recorded rollouts alone, without further environment interaction. Project page: `https://silentgradient.github.io/SOFIE/`

1 INTRODUCTION

Reinforcement learning (RL) in simulation followed by transfer to hardware has become a standard approach to legged locomotion, first for quadrupeds (Tan et al., 2018; Hwangbo et al., 2019; Lee et al., 2020; Kumar et al., 2021; Rudin et al., 2021) and more recently for bipeds and humanoids (Li et al., 2021; Siekmann et al., 2021; Radosavovic et al., 2024; Gu et al., 2024). Challenging terrain such as stairs, gaps and stepping stones also requires policies to perceive the terrain ahead, commonly through height scans or depth images (Miki et al., 2022; Agarwal et al., 2022; Hoeller et al., 2024; Jenelten et al., 2024). Such perceptive policies are often trained with privileged learning, in which an expert with privileged terrain information is optimized online and then distilled into a student that relies on on-board observations (Chen et al., 2019; Lee et al., 2020; Miki et al., 2022). This paradigm has produced increasingly agile locomotion over complex terrain for quadrupeds and humanoids (Zhuang et al., 2023; Cheng et al., 2024; Duan et al., 2024; Zhuang et al., 2024; Long et al., 2025; Wang et al., 2025; Sun et al., 2025; Zhang et al., 2026). Yet these policies are typically trained for one robot and, on challenging terrain, often for one terrain. Multi-task RL can converge to only a subset of the desired behaviors (Rudin et al., 2026; Zhang et al., 2026).

A natural next step is a single perceptive policy that generalizes across terrains and robot embodiments, which we call a *foundation policy*. To generalize across both, the policy must respond to terrain observations while accounting for differences in robot morphology. Across terrains, recent work distills experts into one perceptive policy, sometimes followed by online refinement (Rudin et al., 2026; Zhang et al., 2026). Across embodiments, a growing line of work conditions one policy on the robot morphology, from modular and transformer-based controllers (Huang et al., 2020; Gupta et al.,

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.

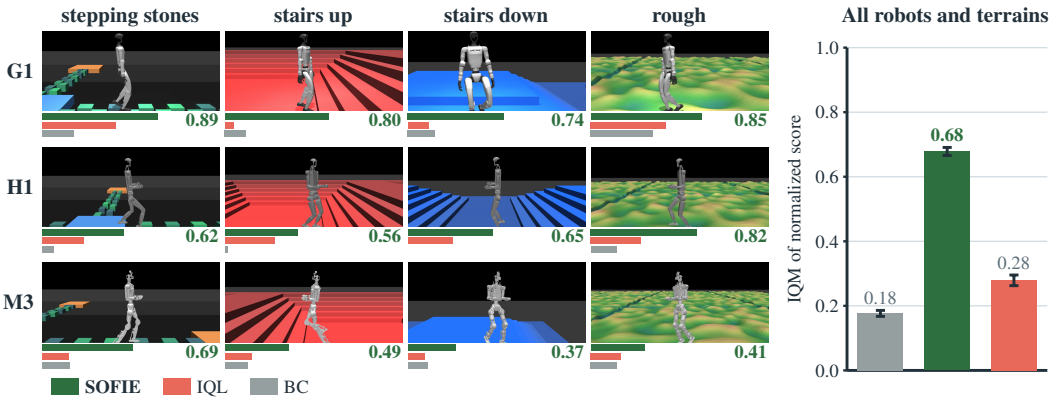


Figure 1: **One SOFIE policy for three humanoids on four terrains.** The policy is trained offline on rollouts from converged and undertrained checkpoints for all twelve robot–terrain pairs. Each image shows the policy on one pair, and the bars below compare its normalized velocity-tracking score with IQL and BC (higher is better). The undertrained and converged checkpoints define scores of 0.0 and 1.0 on each pair. The right panel reports the interquartile mean across pairs with ± 1 bootstrap standard error.

2022) to generalist quadruped and humanoid policies (Bohlinger et al., 2024; Ai et al., 2025). These generalist policies are learned either online, by training on many embodiments jointly, or offline, by distilling rollouts recorded from specialized experts as in GenBot-1K (Ai et al., 2025). Offline RL goes beyond unweighted distillation by using the rewards stored in the dataset to favor better behavior. Kumar et al. (2022) show that offline RL can thereby benefit from imperfect rollouts that cover states and actions relevant to a good policy, whereas behavior cloning (BC) would merely reproduce them. Abe et al. (2026) apply this idea to multi-embodiment pretraining on sixteen robots and report gains over BC on datasets with substantial suboptimal behavior.

We therefore study how to learn one perceptive locomotion policy across robots and terrains entirely offline (Figure 1) from mixed-quality datasets that contain rollouts of converged experts and earlier, undertrained checkpoints. Discarding an undertrained source is too coarse, because its rollouts contain both useful and poor footsteps. BC, at the other extreme, gives every recorded action the same weight and can therefore average conflicting modes when the sources disagree. Identifying useful footsteps on challenging terrain requires looking beyond individual actions. The quality of an action may become clear only at the next touchdown, since the actions of a whole swing jointly place the foot. Reward scales also differ across robots and terrains, so weights based directly on raw advantages can have very different effects across robot–terrain groups.

To address these issues, we introduce Segment-level Offline Imitation of Diverse Experts (SOFIE), an offline RL approach based on advantage-weighted regression (Figure 2). We estimate an advantage for each footstep from recorded rewards and a value function fitted to the dataset. Footstep-level advantage weighting then gives all actions in that footstep one weight, so they are evaluated together. Quantile normalization of the advantages within each robot–terrain group keeps the weights comparable across reward scales. Within each group, footsteps with higher advantages receive more weight, regardless of which checkpoint produced them.

Contributions. **(i)** We propose SOFIE, an offline RL approach that weights whole footsteps by their advantage and normalizes the advantages to quantiles within each robot–terrain group. Its reweighted data distribution stays within the support of the dataset, and its weights are insensitive to the reward scale of each robot and terrain. **(ii)** We show that on mixed-quality datasets SOFIE outperforms BC and offline RL baselines when one policy is shared across humanoids or across terrains. Building on these settings, we learn a single perceptive policy that controls three humanoids on four terrains. **(iii)** Beyond one policy for all robots and terrains, we show that a policy pretrained with SOFIE is a strong starting point for new robots, as fine-tuning it on a small dataset of an unseen humanoid outperforms training from scratch. Together, these results are a step towards pretraining perceptive foundation policies for humanoid locomotion from recorded rollouts alone.

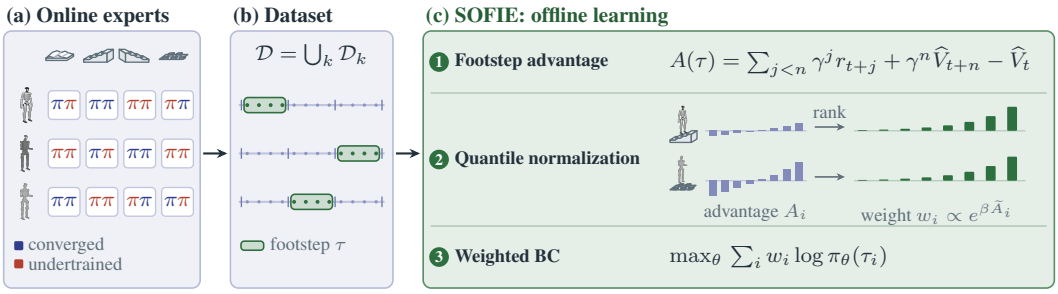


Figure 2: **Overview of SOFIE.** (a) Each robot–terrain pair has an online expert, and rollouts of its converged and undertrained checkpoints are recorded. (b) The rollouts form an offline dataset, grouped by robot–terrain pair and cut into footsteps. (c) SOFIE scores each footstep with an advantage from a value estimate fitted to the dataset, turns the advantages into quantile scores within each group, and gives all transitions of a footstep the same weight in behavior cloning.

2 Related Work

Perceptive locomotion and expert distillation. A common approach trains an expert with privileged terrain information through online RL, then distills it into a student that acts from onboard observations (Chen et al., 2019; Lee et al., 2020; Miki et al., 2022). Using height scans or depth images, such policies have enabled quadrupeds to climb stairs, cross gaps and perform parkour (Agarwal et al., 2022; Hoeller et al., 2024; Jenelten et al., 2024; Zhuang et al., 2023; Cheng et al., 2024). Perceptive humanoids have likewise traversed challenging terrain and sparse footholds (Duan et al., 2024; Zhuang et al., 2024; Long et al., 2025; Wang et al., 2025; Sun et al., 2025). Recent work brings terrain-specific experts together in one perceptive policy. Parkour in the Wild distills them with dataset aggregation (DAgger) and then fine-tunes the shared policy with online RL (Ross et al., 2011; Rudin et al., 2026). RPL distills experts trained with privileged height maps into a transformer policy that uses several depth cameras (Zhang et al., 2026). Both methods continue to use environment interaction when training the shared policy.

Multi-embodiment locomotion. A second line of work shares policies across robot designs. Early approaches use modular policies, transformers over limbs and morphology-conditioned control (Huang et al., 2020; Kurin et al., 2021; Gupta et al., 2022; Xiong et al., 2023). Generalist policies have since been trained on many quadrupeds (Feng et al., 2022; Shafiee et al., 2024). URMA uses online RL to control sixteen legged robots with one policy and later scales to fifty (Bohlinger et al., 2024; Bohlinger & Peters, 2025). Cross-humanoid policies have also shown zero-shot transfer (Xue et al., 2026) and adaptation with limited fine-tuning (Lin et al., 2025). Another approach first trains robot-specific experts and then distills them into a shared policy. GenBot-1K uses BC to distill experts from about a thousand generated embodiments (Ai et al., 2025), while EAGLE alternates between refining humanoid specialists and distilling them into a generalist (Peng et al., 2026). Closest to our offline setting, Abe et al. (2026) learn from proprioceptive walking data of sixteen robots. They find that gradient conflicts between morphologies increase with the number of robots and the share of suboptimal data. They reduce these conflicts by grouping robots with similar morphologies during optimization.

SOFIE brings these two lines together by learning perceptive humanoid policies across terrains and robot embodiments entirely from recorded, mixed-quality rollouts. It weights complete footsteps by their advantages and normalizes those advantages to quantiles within each robot–terrain group, keeping the weights comparable across groups with different reward scales.

3 Background

Markov decision processes. A Markov decision process (MDP) is a tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$ of states $\mathcal{S}$, actions $\mathcal{A}$, transition probabilities $P(s' \mid s, a)$, a reward function $R(s, a)$ and a discount factor $\gamma \in [0, 1)$. We seek a policy π that maximizes the expected discounted re-

turn $J(\pi) := \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)]$. Its value functions are V^π and Q^π, and its advantage is $A^\pi(s,a) := Q^\pi(s,a) - V^\pi(s)$.

KL-regularized policy improvement. By the performance difference lemma (Kakade & Langford, 2002), a policy improves on a behavior policy μ to the extent that it chooses actions with positive advantage under μ. Relative entropy policy search (Peters et al., 2010) and maximum a posteriori policy optimization (Abdolmaleki et al., 2018) therefore maximize the expected advantage while bounding the Kullback–Leibler (KL) divergence to the behavior policy, so that the states visited by μ stay representative of the new policy,

$$\max_{\pi} \; \mathbb{E}_{s\sim d_\mu,\, a\sim\pi(\cdot|s)}\big[A^\mu(s,a)\big] \qquad \text{s.t.} \qquad \mathbb{E}_{s\sim d_\mu}\big[\mathrm{KL}(\pi(\cdot \mid s) \,\|\, \mu(\cdot \mid s))\big] \le \varepsilon, \tag{1}$$

where d_μ is the state distribution of μ. The solution $\pi^*(a \mid s) \propto \mu(a \mid s)\exp(A^\mu(s,a)/\lambda)$, with Lagrange multiplier $\lambda > 0$, only reweights the actions of μ and therefore stays within its support. Fitting a parametric policy to it gives advantage-weighted regression (Peng et al., 2019), that is, BC in which every transition is weighted by $\exp(A^\mu(s,a)/\lambda)$.

4 Methodology

Problem statement. We learn a locomotion policy $\pi_\theta(a \mid s)$ from a dataset $\mathcal{D} := \bigcup_{k=1}^{K} \mathcal{D}_k$ of recorded rollouts across robots and terrains, without further interaction with the environment. Each subset $\mathcal{D}_k$ holds the rollouts of one robot–terrain pair, recorded from expert policies of varying quality, and μ denotes the resulting mixture of behavior policies. The goal is a policy whose expected discounted return is at least that of μ and that stays within the support of the dataset.

We extend the KL-regularized policy improvement of Section 3 from single actions to segments, and turn the segment advantages into quantile scores so that the weights are comparable across robots and terrains (Section 4.1). Section 4.2 describes how the scores are computed and used to train the policy.

4.1 Segment-level Offline Imitation of Diverse Experts

Segment-level advantage weighting. The objective in Eq. (1) scores each action by its own advantage $A^\mu(s,a)$. In locomotion, however, the effect of an action may become clear only at touchdown, because the actions of a whole swing jointly determine where the foot lands. We therefore score and reweight all transitions between two consecutive events, which in locomotion are footstep boundaries, as one unit.

An event detector $b : \mathcal{S} \to \{0,1\}$ marks each boundary, and the trajectory between two successive events forms a segment $\tau_t := (s_t, a_t, \ldots, s_{t+n-1}, a_{t+n-1})$ of length n. The events only define the unit in which the data are scored and reweighted, and the policy π_θ still acts at every step. The advantage of a segment under the behavior mixture μ, with value function V^μ, is its n-step Bellman residual

$$A^\mu(\tau_t) := \sum_{j=0}^{n-1} \gamma^j r_{t+j} + \gamma^n V^\mu(s_{t+n}) - V^\mu(s_t), \tag{2}$$

with $V^\mu(s_{t+n}) = 0$ at terminal states. It compares the segment's rewards and end-state value with its start value, and equals the discounted sum of the one-step Bellman residuals inside the segment, so the segment's actions are evaluated jointly (Appendix A.1). Regrouping these residuals by segment gives a performance-difference identity over segments (Appendix A.2). Both identities hold for the exact value V^μ, which training replaces with an estimate $\widehat{V}_\phi$. We write A for the resulting estimated segment advantage.

With this advantage, Eq. (1) extends from actions to segments. Let $p_\mu(\tau)$ be the distribution of segments in the dataset. We seek a reweighted segment distribution p that maximizes the expected segment advantage subject to a bound on its KL divergence from p_μ,

$$\max_{p} \; \mathbb{E}_{\tau\sim p}\big[A(\tau)\big] \qquad \text{s.t.} \qquad \mathrm{KL}(p \,\|\, p_\mu) \le \varepsilon. \tag{3}$$

The solution is $p^*(\tau) \propto p_\mu(\tau)\exp(A(\tau)/\lambda)$. Because the KL constraint allows p to be positive only where p_μ is, p^* stays within the support of the dataset and reweights recorded segments. Fitting π_θ to p^* by weighted maximum likelihood weights all transitions of a segment equally (Appendix A.3).

Quantile normalization. The solution of Eq. (3) depends on the scale of the advantage, because rescaling A acts like changing the temperature λ. Robots and terrains differ in reward scale, and the sources recorded in each group shape its advantage distribution, so one temperature can weight different groups very differently.

To remove this scale dependence, we draw on information-geometric optimization, where fitness quantiles make the weights invariant to strictly increasing transformations of the fitness (Ollivier et al., 2017), and apply the same idea to segment advantages. We compute each advantage quantile within its robot–terrain group, write $g(\tau)$ for the group that contains τ, and let $p_{\mu|g}(\tau) := p_\mu(\tau \mid g)$ be the segment distribution of group g. The advantage quantile and its standardized score are

$$q_\mu(\tau) := \Pr_{\tau' \sim p_{\mu|g(\tau)}} \big[A(\tau') \le A(\tau)\big], \qquad \widetilde{A}(\tau) := \sqrt{12}\left(q_\mu(\tau) - \frac{1}{2}\right). \tag{4}$$

If the advantage distribution of each group is continuous, q_μ is uniform on $[0,1]$ in every group (Casella & Berger, 2002). Subtracting $1/2$ centers it and the factor $\sqrt{12}$ gives unit variance, so $\widetilde{A} \sim \mathcal{U}[-\sqrt{3}, \sqrt{3}]$ in every group.

Replacing A with $\widetilde{A}$ in Eq. (3) gives $p^*(\tau) \propto p_\mu(\tau) e^{\beta \widetilde{A}(\tau)}$ with inverse temperature $\beta > 0$, and fitting the shared policy π_θ to this distribution gives the objective

$$\mathcal{L}_{\text{SOFIE}}(\theta) := -\mathbb{E}_{\tau \sim p_\mu}\left[\frac{e^{\beta \widetilde{A}(\tau)}}{Z(\beta)} \sum_{(s,a)\in\tau} \log \pi_\theta(a \mid s)\right], \tag{5}$$

where $Z(\beta) := \mathbb{E}_{\tau \sim p_\mu}[e^{\beta \widetilde{A}(\tau)}]$ normalizes the weights. Because $\widetilde{A}$ has the same distribution in every group, the weights have mean one within each group, so every group keeps its share of the dataset and one policy is trained on all groups.

Proposition 1 (Properties of quantile reweighting)**.** *Suppose the advantage distribution is continuous in each group, let $\beta > 0$, and let $p^*(\tau) \propto p_\mu(\tau) e^{\beta \widetilde{A}(\tau)}$. Then:*

- *(i)* Support preservation. $\operatorname{supp}(p^*) = \operatorname{supp}(p_\mu)$.
- *(ii)* Invariance. *p^* is invariant to any strictly increasing transformation of the advantages applied separately within each group.*

Both properties hold because the weights are positive and a strictly increasing transformation within a group leaves the quantiles unchanged (Appendix A.4). Since the quantile scores have the same bounded distribution in every group, one temperature serves all robots and terrains. The price is that the numerical scale of the advantages is discarded, so tightly clustered advantages receive the same weights as widely separated ones.

4.2 Practical implementation

Computing the objective in Eq. (5) requires a footstep advantage and its quantile score. We describe how both are obtained for perceptive humanoid locomotion. Algorithm 1 summarizes the procedure, and Appendix D lists the hyperparameters.

Segments and reward. Every task has a gait clock, and its gait rewards assign each half cycle of the clock to one foot (Appendix B.1). We define a footstep segment as one such half cycle, ending it sooner if the episode terminates. The advantage uses only the velocity-tracking term of the reward, because the remaining terms mostly reward posture and survival, which a standing policy also collects (Appendix D).

Value function. We estimate V^μ by fitting $\widehat{V}_\phi$ to truncated H-step Monte Carlo returns of each robot–terrain dataset. For a state s_t of a recorded rollout that ends at T, let $H_t := \min(H, T-t)$ and $G_t^{(H)} := \sum_{k=0}^{H_t-1} \gamma^k r_{t+k}$. The value model minimizes

$$\mathcal{L}_{\text{V}}(\phi) := \mathbb{E}_{(s_t, G_t^{(H)})\sim\mathcal{D}}\left[\ell_{\text{V}}\big(\widehat{V}_\phi(s_t), G_t^{(H)}\big)\right]. \tag{6}$$

Compared with temporal-difference targets, Monte Carlo targets avoid bootstrapping error at the cost of higher variance, which the truncation at a fixed horizon H limits at the cost of bias. The footstep

Algorithm 1 Segment-level Offline Imitation of Diverse Experts

Require: dataset $\mathcal{D}$ grouped by robot–terrain pair, policy π_θ, value model $\widehat{V}_\phi$, inverse temperature β
for N_{pre} steps **do** ▷ value pretraining
  Update ϕ with Eq. (6) on a batch of footsteps from $\mathcal{D}$
end for
for N steps **do** ▷ joint training
  Sample a batch of footsteps $\{\tau_i\}_{i=1}^{B}$ from $\mathcal{D}$
  Compute the advantages $A(\tau_i)$ with Eq. (2), using the current $\widehat{V}_\phi$
  Rank the advantages within each group and set $w_i = e^{\beta \widetilde{A}_i}/Z_{g_i}$ (Eq. (8))
  Update θ with Eq. (9) and ϕ with Eq. (6)
end for

length n sets the interval that Eq. (2) scores, whereas H sets how far the value targets reach. Because the return targets are noisy, we use the HL-Gauss loss (Imani & White, 2018; Farebrother et al., 2024). The value model predicts a categorical distribution $\widehat{p}_\phi(s_t)$ over m uniformly spaced bins with centers $c_1 < \cdots < c_m$, and its mean is the value estimate $\widehat{V}_\phi(s_t) := \sum_{j=1}^{m} c_j \widehat{p}_{\phi,j}(s_t)$. A Gaussian centered at $G_t^{(H)}$ assigns the target bin probabilities $y_{t,j}$, and the loss in Eq. (6) is the cross-entropy

$$\ell_{\text{V}}\big(\widehat{V}_\phi(s_t), G_t^{(H)}\big) := -\sum_{j=1}^{m} y_{t,j} \log \widehat{p}_{\phi,j}(s_t). \tag{7}$$

We first train the value model for N_{pre} steps with the policy fixed, so that the first weights already come from a fitted value estimate, and then update both models jointly (Algorithm 1).

Quantile weights. For each of the B footsteps τ_i of a batch, we compute $A(\tau_i)$ with $\widehat{V}_\phi$ in Eq. (2) and estimate $q_\mu(\tau_i)$ by ranking the advantages within their robot–terrain groups. Let $g_i := g(\tau_i)$ be the group of footstep i, and let $B_g := \#\{i : g_i = g\} \geq 2$ be the number of footsteps of group g in the batch. Within its group, footstep i receives a zero-based rank $\rho_i \in \{0, \ldots, B_{g_i} - 1\}$, with ties assigned distinct ranks, and the quantile score

$$\widetilde{A}_i := \sqrt{12}\left(\frac{\rho_i}{B_{g_i} - 1} - \frac{1}{2}\right), \tag{8}$$

which estimates $\widetilde{A}(\tau_i)$ in Eq. (4). With the group normalizer $Z_g := \frac{1}{B_g}\sum_{j:g_j=g} e^{\beta \widetilde{A}_j}$ and the weights $w_i := e^{\beta \widetilde{A}_i}/Z_{g_i}$, which average to one in every group, the shared policy minimizes

$$\mathcal{L}(\theta) := -\frac{1}{B}\sum_{i=1}^{B} w_i \sum_{(s,a)\in\tau_i} \log \pi_\theta(a \mid s). \tag{9}$$

Because ties receive distinct ranks, every group of size B_g has the same scores, so Z_g depends only on B_g and β and not on the advantages.

Policy architecture. A policy shared across robots must accept any number and order of joints. On challenging terrain, the action of one joint also depends on the other joints of its leg and on the other leg. We therefore use the morphology-conditioned policy of Figure 3, which treats the joints as a set of tokens that attend to each other. Each joint j becomes a token $\boldsymbol{x}_j$ built from its angle and velocity history and its descriptor $\bar{\boldsymbol{d}}_j$, a normalized vector of its kinematic and actuation properties. The robot state and the encoded height scan form a context token $\boldsymbol{x}_0$. Self-attention blocks process all tokens, with a bias based on kinematic distance that favors interactions between nearby joints. The encoder maps the context token to a robot feature $\boldsymbol{z}$, which a decoder shared by all joints uses to predict their actions. For joint j, one MLP maps $[\boldsymbol{z}, \bar{\boldsymbol{d}}_j]$ to a feature $\boldsymbol{h}_j$ of width d, a second MLP maps $\bar{\boldsymbol{d}}_j$ alone to the weights $\boldsymbol{w}_j$ and the bias b_j of the last linear layer, and the action is $\hat{a}_j := \boldsymbol{w}_j^\top \boldsymbol{h}_j/\sqrt{d} + b_j$. This decoder generalizes the output layer of a single-robot policy, which learns one such row per joint, to rows computed from the joint descriptors, so no parameter depends on the number or order of the joints (Appendix C).

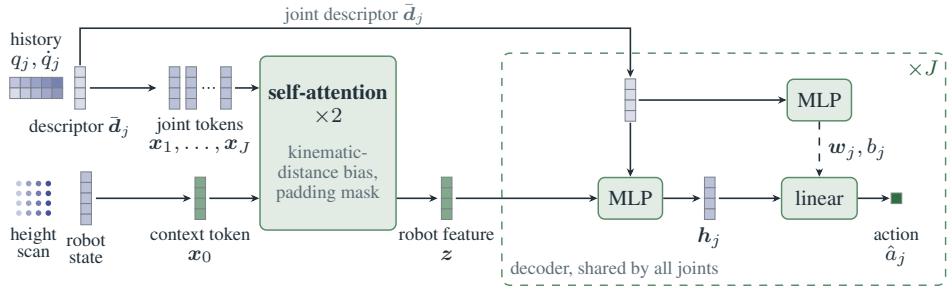


Figure 3: **Morphology-conditioned policy.** Each joint j becomes a joint token $\boldsymbol{x}_j$, built from its angle and velocity history and its descriptor, and the robot state and height scan become a context token $\boldsymbol{x}_0$. The tokens exchange information in self-attention blocks. A decoder shared by all joints maps the robot feature $\boldsymbol{z}$ and the joint descriptor $\bar{\boldsymbol{d}}_j$ to the action $\hat{a}_j$ of joint j, through a last linear layer whose weights and bias a second MLP computes from $\bar{\boldsymbol{d}}_j$ alone.

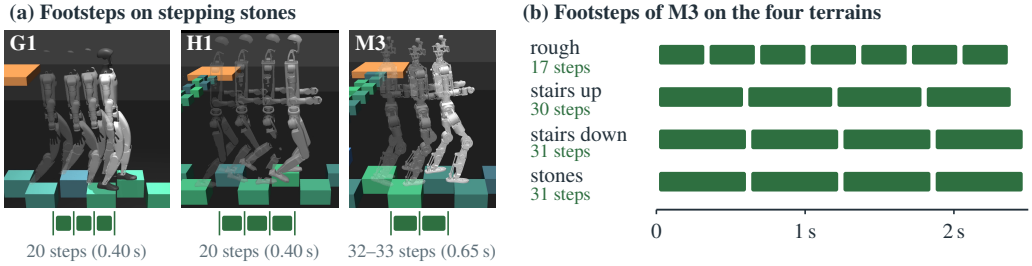


Figure 4: **Footsteps across robots and terrains.** (a) The per-terrain SOFIE policy on stepping stones. Each green bar marks a gait-clock footstep, the segment that receives one training weight. (b) Median footstep duration of the converged M3 checkpoints on each terrain, measured between touchdowns.

5 Experiments

We evaluate SOFIE on tasks in which a humanoid follows a forward velocity command on one terrain (Appendix B). Each robot–terrain pair has a dataset of rollouts, in equal shares from converged and earlier undertrained checkpoints of the same online training run. On **flat ground**, one morphology-conditioned policy controls nine IsaacLab humanoids from proprioception. In mjlab, the Unitree G1 and H1 (Unitree Robotics, 2026) and the M3 (Ly et al., 2026) use height scans to walk on rough ground, up and down stairs, and over stepping stones. We train **one policy per terrain** shared across the three robots, **one policy per robot** shared across four terrains, and **one policy for all robots and terrains**. The first two settings test sharing across robots and terrains separately, while the last tests both together. Because the robots step at different rates, SOFIE defines their footsteps using the gait clock of each robot (Figure 4). On flat ground, where the recorded rollouts contain no falls, it weights single control steps (Appendix E.4).

Baselines. We compare SOFIE with BC, which weights all recorded actions equally as in GenBot-1K (Ai et al., 2025), and with three offline RL methods. IQL (Kostrikov et al., 2022), used for cross-embodiment pretraining by Abe et al. (2026), weights individual actions by advantages from a learned action value. To match the weight scale of SOFIE, we rank those advantages across the whole batch and use the same inverse temperature. GC-IQL also conditions IQL on the command as a goal, while TD3+BC (Fujimoto & Gu, 2021) combines a learned action value with a behavior-cloning term. All methods in a setting use the same dataset, reward and policy architecture (Appendix C). Appendix D gives the hyperparameters and update budgets.

Evaluation. We evaluate each policy on the robots and terrains of its training setting, using 1024 parallel environments at the same terrain level used to record the data (Table 6). Because the total return also rewards standing still, we score the velocity-tracking reward r_{track}, which measures the task objective (Appendix D). We normalize it for each robot–terrain pair so that the undertrained

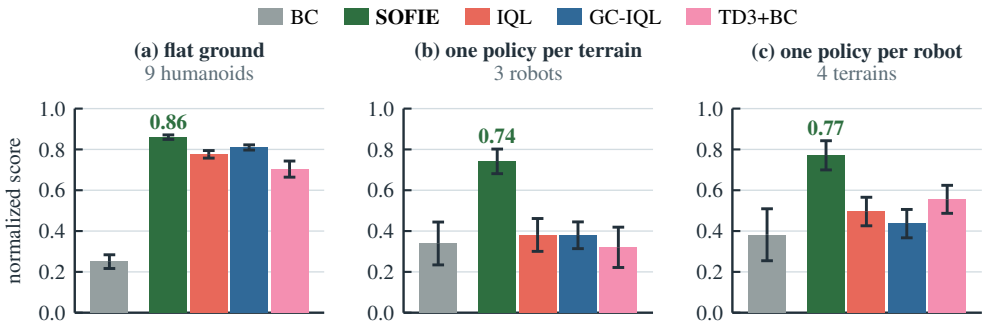
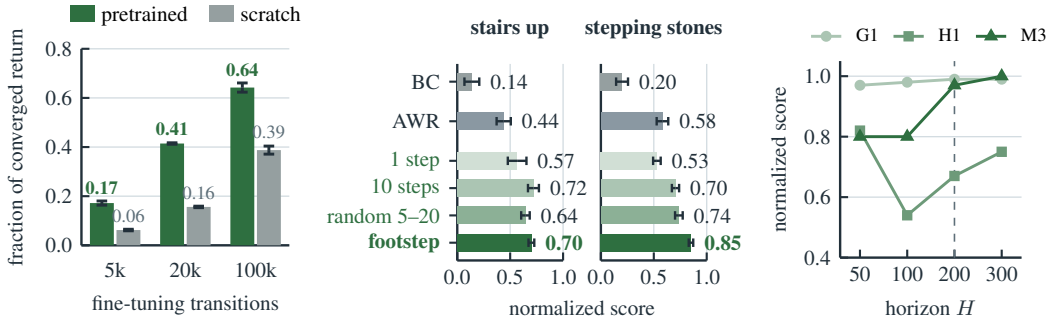


Figure 5: **Aggregate performance on mixed-quality datasets.** Interquartile mean of the normalized velocity-tracking score with ± 1 bootstrap standard error (higher is better). On each robot–terrain pair, the undertrained checkpoint scores 0.0 and the converged checkpoint 1.0. (a) One policy for nine IsaacLab humanoids on flat ground. (b) One policy per terrain, shared by G1, H1 and M3. (c) One policy per robot across four terrains.



(a) Fine-tuning **(b)** Segment ablation **(c)** Monte Carlo horizon

Figure 6: **Pretraining and ablations.** Higher is better throughout. (a) Return on three new humanoids relative to their converged checkpoints after 20k steps of BC fine-tuning, starting from a pretrained SOFIE policy or from scratch. Bars show ± 1 standard error. (b) Normalized score as in Figure 5 for different segments over which an advantage and weight are shared. Green rows vary the segment used by SOFIE, and the bold row uses gait-clock footsteps. AWR weights individual steps by their Monte Carlo advantage. (c) Normalized score for SOFIE trained on each stairs-up dataset alone as the value-target horizon H varies. The dashed line marks $H = 200$, used in the other mjlab experiments.

checkpoint scores 0.0 and the converged checkpoint 1.0. Following Agarwal et al. (2021), we report the interquartile mean (IQM) over pairs and seeds with ± 1 bootstrap standard error. We report falls separately in Appendix E.1 because the tracking score does not capture them.

6 Results

Policies shared across robots or terrains. SOFIE achieves the highest score in all three settings of Figure 5. On flat ground, where one policy controls nine humanoids from proprioception, the two IQL variants approach its score, while TD3+BC trails further behind. BC imitates converged and undertrained rollouts equally and closes only a quarter of the gap between the undertrained and converged checkpoints. Because the undertrained flat-ground rollouts differ mainly in speed throughout, SOFIE can distinguish them from converged rollouts even with single-step advantages (Appendix E.4). When the dataset contains only converged checkpoints, SOFIE scores slightly below BC, suggesting that weighting has little cost when the recorded rollouts are similar in quality (Table 16). On challenging terrain, undertrained checkpoints fall often, and the quality of an action may become clear only when the foot lands. Here the margin over the baselines is larger. With one policy shared across robots on each terrain, SOFIE nearly doubles the score of the strongest

baseline, IQL. With one policy shared across terrains for each robot, it exceeds the strongest baseline, TD3+BC, by more than a third. It also has the fewest falls in every setting (Table 9).

Foundation policy and fine-tuning. We next combine both axes of sharing and train one perceptive policy on all twelve robot–terrain datasets, comprising 49M transitions (Figure 1). SOFIE scores more than twice as high as IQL and almost four times as high as BC. It leads on every robot–terrain pair and falls about half as often as either baseline. In a separate study of transfer to new robots, we pretrain on the stairs-up datasets of G1, H1 and M3, recorded with randomized motor gains so that the policy sees varied actuation. We then fine-tune this policy with BC on converged rollouts from H3-1, T1 and T1VH, a T1 variant with different motors and smaller feet (Figure 6a). With 5k transitions, the pretrained policy needs only 1k fine-tuning steps to reach about twice the return obtained by training from scratch for 20k steps. With 20k transitions, it matches training from scratch on five times as much data (Table 15). On its training robots, the pretrained policy exceeds IQL and BC pretrained on the same data on average. It matches IQL on G1 and H1 and performs substantially better on M3 (Table 14).

Segment ablation. To isolate the role of the segment, we vary only the interval over which one advantage and weight are shared on stairs-up and stepping-stone datasets from the three humanoids (Figure 6b). Sharing a weight across a segment outperforms weighting individual steps on both terrains, whether the single-step advantage comes from a one-step Bellman residual or the Monte Carlo advantage of AWR (Peng et al., 2019). Every weighted variant also improves substantially on BC. Among segment choices, gait-clock footsteps perform on par with fixed ten-step windows on stairs up but better on stepping stones, where each foot must land on a stone. They also yield the fewest falls on both terrains (Table 17). These results support the premise of Section 4.1 that action quality can become apparent only over a longer interval. They also show that aligning that interval with footsteps matters most on stepping stones.

Monte Carlo horizon. We ablate the Monte Carlo value-target horizon H on the stairs-up datasets to examine how far the recorded rewards should extend when fitting the value function (Section 4.2 and Figure 6c). Longer horizons capture more of the consequences of a footstep, but summing rewards over more steps can increase the variance of the target. In the ablation, M3 benefits from a longer horizon, while G1 remains largely unchanged. H1 achieves its highest score at $H = 50$ and scores lower with longer horizons. The balance between a longer return and a less variable target therefore differs across robots. We use $H = 200$ as a shared setting in the mjlab experiments.

7 Conclusion

We introduced SOFIE, an offline RL approach for learning perceptive humanoid locomotion from mixed-quality rollouts. These datasets contain rollouts from converged experts and earlier, under-trained checkpoints, so useful and poor footsteps can occur within the same source. SOFIE estimates an advantage for each complete footstep from recorded rewards and a value function fitted to the dataset, then gives its actions a shared weight. It normalizes the advantages to quantiles within each robot–terrain group so that the weights remain comparable across reward scales. We evaluated policies shared across humanoids and across terrains, from one policy for nine humanoids on flat ground to a single perceptive policy controlling three humanoids on four terrains. In every mixed-quality setting, SOFIE outperformed BC and offline RL baselines. A policy pretrained with SOFIE also adapted to new humanoids faster than training from scratch. Together, these results are a step towards pretraining perceptive foundation policies for humanoid locomotion from recorded rollouts alone, without further environment interaction.

Limitations and future work. Our experiments are limited to perceptive humanoid locomotion, although SOFIE provides a general method for weighting segments defined by events. One promising extension is whole-body control, where motion waypoints or changes in contact could mark segment boundaries so that the actions within each segment are evaluated together. A separate direction is to use a policy pretrained with SOFIE as the starting point for online RL fine-tuning, allowing it to adapt to new robots and tasks when further environment interaction is available.

AI USE STATEMENT

We used generative AI tools to explore and refine research ideas, identify relevant literature, assist with code implementation and analysis and plotting scripts, and draft and edit parts of the manuscript. We also used these tools to discuss possible approaches to mathematical statements and proofs and to cross-check manuscript numbers against saved evaluation outputs. The authors assessed the ideas against prior work, checked the cited literature, and developed and verified the mathematical arguments and proofs themselves. We reviewed and tested AI-assisted code and independently recomputed all reported numbers from the evaluation summaries. The authors take full responsibility for the methods, results, claims, and final manuscript.

ETHICS STATEMENT

This work involves no human subjects or personal data. The datasets consist of simulated rollouts of policies trained by the authors in open simulators. Humanoid locomotion controllers have broad applications. We do not identify additional ethical concerns specific to this work beyond those generally associated with robot learning.

REPRODUCIBILITY STATEMENT

Section 4 presents the learning objective, Section 5 and Appendix D specify the experimental and evaluation procedures, and Appendix B describes the robots, tasks, rewards, and datasets. The tables state their datasets and aggregation procedures in their captions. The code and aggregation scripts will be released upon acceptance.

REFERENCES

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. In *International Conference on Learning Representations (ICLR)*, 2018.

Haruki Abe, Takayuki Osa, Yusuke Mukuta, and Tatsuya Harada. Cross-embodiment offline reinforcement learning for heterogeneous robot datasets. In *International Conference on Learning Representations (ICLR)*, 2026.

Ananye Agarwal, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Legged locomotion in challenging terrains using egocentric vision. In *Conference on Robot Learning (CoRL)*, 2022.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Bo Ai, Liu Dai, Nico Bohlinger, Dichen Li, Tongzhou Mu, Zhanxin Wu, K. Fay, Henrik I. Christensen, Jan Peters, and Hao Su. Towards embodiment scaling laws in robot locomotion. In *Conference on Robot Learning (CoRL)*, 2025.

Nico Bohlinger and Jan Peters. Multi-embodiment locomotion at scale with extreme embodiment randomization. *arXiv preprint arXiv:2509.02815*, 2025.

Nico Bohlinger, Grzegorz Czechmanowski, Maciej Krupka, Piotr Kicki, Krzysztof Walas, Jan Peters, and Davide Tateo. One policy to run them all: an end-to-end learning approach to multi-embodiment locomotion. In *Conference on Robot Learning (CoRL)*, 2024.

George Casella and Roger L. Berger. *Statistical Inference*. Duxbury, 2nd edition, 2002.

Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *Conference on Robot Learning (CoRL)*, 2019.

Xuxin Cheng, Kexin Shi, Ananye Agarwal, and Deepak Pathak. Extreme parkour with legged robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Helei Duan, Bikram Pandit, Mohitvishnu S. Gadde, Bart van Marum, Jeremy Dao, Chanho Kim, and Alan Fern. Learning vision-based bipedal locomotion for challenging terrain. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Ziqi Fan. robot_lab: RL extension library for robots, based on IsaacLab, 2024. `https://github.com/fan-ziqi/robot_lab`, accessed September 2026.

Jesse Farebrother, Jordi Orbay, Quan Vuong, Adrien Ali Taïga, Yevgen Chebotar, Ted Xiao, Alex Irpan, Sergey Levine, Pablo Samuel Castro, Aleksandra Faust, Aviral Kumar, and Rishabh Agarwal. Stop regressing: Training value functions via classification for scalable deep RL. In *International Conference on Machine Learning (ICML)*, 2024.

Gilbert Feng, Hongbo Zhang, Zhongyu Li, Xue Bin Peng, Bhuvan Basireddy, Linzhu Yue, Zhitao Song, Lizhi Yang, Yunhui Liu, Koushil Sreenath, and Sergey Levine. GenLoco: Generalized locomotion controllers for quadrupedal robots. In *Conference on Robot Learning (CoRL)*, 2022.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Xinyang Gu, Yen-Jen Wang, and Jianyu Chen. Humanoid-gym: Reinforcement learning for humanoid robot with zero-shot sim2real transfer. *arXiv preprint arXiv:2404.05695*, 2024.

Agrim Gupta, Linxi Fan, Surya Ganguli, and Li Fei-Fei. Metamorph: Learning universal controllers with transformers. In *International Conference on Learning Representations (ICLR)*, 2022.

David Hoeller, Nikita Rudin, Dhionis Sako, and Marco Hutter. ANYmal parkour: Learning agile navigation for quadrupedal robots. *Science Robotics*, 9(88):eadi7566, 2024.

Wenlong Huang, Igor Mordatch, and Deepak Pathak. One policy to control them all: Shared modular policies for agent-agnostic control. In *International Conference on Machine Learning (ICML)*, 2020.

Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019.

Ehsan Imani and Martha White. Improving regression performance with distributional losses. In *International Conference on Machine Learning (ICML)*, 2018.

Fabian Jenelten, Junzhe He, Farbod Farshidian, and Marco Hutter. DTC: Deep tracking control. *Science Robotics*, 9(86), 2024.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2002.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: Rapid motor adaptation for legged robots. In *Robotics: Science and Systems (RSS)*, 2021.

Aviral Kumar, Joey Hong, Anikait Singh, and Sergey Levine. When should we prefer offline reinforcement learning over behavioral cloning? In *International Conference on Learning Representations (ICLR)*, 2022.

Vitaly Kurin, Maximilian Igl, Tim Rocktäschel, Wendelin Boehmer, and Shimon Whiteson. My body is a cage: the role of morphology in graph-based incompatible control. In *International Conference on Learning Representations (ICLR)*, 2021.

Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47):eabc5986, 2020.

Erich L. Lehmann and Joseph P. Romano. *Testing Statistical Hypotheses*. Springer, 3 edition, 2005.

Zhongyu Li, Xuxin Cheng, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. Reinforcement learning for robust parameterized locomotion control of bipedal robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

Yunfeng Lin, Minghuan Liu, Yufei Xue, Ming Zhou, Yong Yu, Jiangmiao Pang, and Weinan Zhang. H-zero: Cross-humanoid locomotion pretraining enables few-shot novel embodiment transfer. *arXiv preprint arXiv:2512.00971*, 2025.

Junfeng Long, Junli Ren, Moji Shi, Zirui Wang, Tao Huang, Ping Luo, and Jiangmiao Pang. Learning humanoid locomotion with perceptive internal model. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

Thanh Ly, Truong-Duy Dang, Chien Le, Tan-Dzung Do, Tuan Dat Phuong, Cuc T.Trinh, Vien Anh Ngo, and An Thai Le. TACT-ful: Multi-channel terrain affordance and compliance training for payload-robust perceptive humanoid locomotion. In *ECCV Workshop X-Reason: Visual Perception and Reasoning in the Interactable World*, 2026. URL https://openreview.net/forum?id=sHoOIdNzQY.

Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7 (62):eabk2822, 2022.

Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Muñoz, Xinjie Yao, René Zurbrügg, Nikita Rudin, et al. Isaac lab: A GPU-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025.

Yann Ollivier, Ludovic Arnold, Anne Auger, and Nikolaus Hansen. Information-geometric optimization algorithms: A unifying picture via invariance principles. *Journal of Machine Learning Research*, 18(18):1–65, 2017.

Quanquan Peng, Yunfeng Lin, Yufei Xue, Jiangmiao Pang, and Weinan Zhang. Embodiment-aware generalist specialist distillation for unified humanoid whole-body control. *arXiv preprint arXiv:2602.02960*, 2026.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Jan Peters, Katharina Mülling, and Yasemin Altün. Relative entropy policy search. In *AAAI Conference on Artificial Intelligence*, 2010.

Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. Real-world humanoid locomotion with reinforcement learning. *Science Robotics*, 9(89):eadi9579, 2024.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2021.

Nikita Rudin, Junzhe He, Joshua Aurand, and Marco Hutter. Parkour in the wild: Learning a general and extensible agile locomotion policy using multi-expert distillation and RL fine-tuning. *The International Journal of Robotics Research*, 2026. doi: 10.1177/02783649261455067.

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

Clemens Schwarke, Mayank Mittal, Nikita Rudin, David Hoeller, and Marco Hutter. RSL-RL: A learning library for robotics research. *arXiv preprint arXiv:2509.10771*, 2025.

Milad Shafiee, Guillaume Bellegarda, and Auke Ijspeert. ManyQuadrupeds: Learning a single locomotion policy for diverse quadruped robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Jonah Siekmann, Kevin Green, John Warila, Alan Fern, and Jonathan Hurst. Blind bipedal stair traversal via sim-to-real reinforcement learning. In *Robotics: Science and Systems (RSS)*, 2021.

Wandong Sun, Baoshi Cao, Long Chen, Yongbo Su, Yang Liu, Zongwu Xie, and Hong Liu. Learning perceptive humanoid locomotion over challenging terrain. *arXiv preprint arXiv:2503.00692*, 2025.

Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. In *Robotics: Science and Systems (RSS)*, 2018.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

Unitree Robotics. unitree_ros: Robot descriptions of the Unitree robots, 2026. `https://github.com/unitreerobotics/unitree_ros`, accessed September 2026.

Claas A. Voelcker, Axel Brunnbauer, Marcel Hussing, Michal Nauman, Pieter Abbeel, Radu Grosu, Eric Eaton, Amir-massoud Farahmand, and Igor Gilitschenski. Relative entropy pathwise policy optimization. In *International Conference on Learning Representations (ICLR)*, 2026.

Huayi Wang, Zirui Wang, Junli Ren, Qingwei Ben, Tao Huang, Weinan Zhang, and Jiangmiao Pang. BeamDojo: Learning agile humanoid locomotion on sparse footholds. In *Robotics: Science and Systems (RSS)*, 2025.

Zheng Xiong, Jacob Beck, and Shimon Whiteson. Universal morphology control via contextual modulation. In *International Conference on Machine Learning (ICML)*, 2023.

Yufei Xue, Yunfeng Lin, Wentao Dong, Yang Tang, Jingbo Wang, Jiangmiao Pang, Ming Zhou, Minghuan Liu, and Weinan Zhang. Scalable and general whole-body control for cross-humanoid locomotion. *arXiv preprint arXiv:2602.05791*, 2026.

Kevin Zakka, Qiayuan Liao, Brent Yi, Louis Le Lay, Koushil Sreenath, and Pieter Abbeel. mjlab: A lightweight framework for GPU-accelerated robot learning, 2026. `https://github.com/mujocolab/mjlab`.

Yuanhang Zhang, Younggyo Seo, Juyue Chen, Yifu Yuan, Koushil Sreenath, Pieter Abbeel, Carmelo Sferrazza, Karen Liu, Rocky Duan, and Guanya Shi. RPL: Learning robust humanoid perceptive locomotion on challenging terrains. *arXiv preprint arXiv:2602.03002*, 2026.

Ziwen Zhuang, Zipeng Fu, Jianren Wang, Christopher G. Atkeson, Sören Schwertfeger, Chelsea Finn, and Hang Zhao. Robot parkour learning. In *Conference on Robot Learning (CoRL)*, 2023.

Ziwen Zhuang, Shenzhe Yao, and Hang Zhao. Humanoid parkour learning. In *Conference on Robot Learning (CoRL)*, 2024.

A Derivations and proofs

This appendix derives the segment advantage and the performance difference over segments (Appendices A.1 and A.2). It then connects the reweighted segment distribution to the policy objective (Appendix A.3) and proves the properties of quantile reweighting (Appendix A.4).

We assume bounded rewards and $0 < \gamma < 1$. Throughout this appendix, V, Q, and A denote the value, action value, and advantage of the behavior policy μ, with the superscript μ omitted. Each recorded trajectory comes from one expert, so for the Bellman derivations we include that expert's identity in the state and assume that μ is Markov in this state. The learned policy continues to use only its usual observations. In the experiments, we compute advantages from the velocity-tracking reward r_{track} (Section 4.2).

A.1 Segment advantage

A segment $\tau_t = (s_t, a_t, \dots, s_{t+n-1}, a_{t+n-1})$ contains n state–action pairs together with their recorded rewards $r_t, \dots, r_{t+n-1}$ and the endpoint s_{t+n}. The notation for τ_t leaves the rewards and endpoint implicit. The one-step Bellman equation gives

$$Q(s_t, a_t) = \mathbb{E}\big[r_t + \gamma V(s_{t+1}) \mid s_t, a_t\big]. \tag{10}$$

Subtracting $V(s_t)$ and using $A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$ yields

$$A(s_t, a_t) = \mathbb{E}\big[r_t + \gamma V(s_{t+1}) - V(s_t) \mid s_t, a_t\big]. \tag{11}$$

We define the one-step Bellman residual

$$\delta_t := r_t + \gamma V(s_{t+1}) - V(s_t), \tag{12}$$

whose conditional expectation equals the action advantage,

$$\mathbb{E}[\delta_t \mid s_t, a_t] = A(s_t, a_t). \tag{13}$$

Because the transition model is the same for every policy, this relation also holds along trajectories generated by another policy. Summing the discounted residuals along a segment then makes the value terms cancel in pairs:

$$\begin{aligned}
\sum_{j=0}^{n-1} \gamma^j \delta_{t+j} &= \sum_{j=0}^{n-1} \gamma^j \big(r_{t+j} + \gamma V(s_{t+j+1}) - V(s_{t+j})\big) \\
&= \sum_{j=0}^{n-1} \gamma^j r_{t+j} + \sum_{j=0}^{n-1} \gamma^{j+1} V(s_{t+j+1}) - \sum_{j=0}^{n-1} \gamma^j V(s_{t+j}) \\
&= \sum_{j=0}^{n-1} \gamma^j r_{t+j} + \gamma^n V(s_{t+n}) - V(s_t).
\end{aligned} \tag{14}$$

The final expression is the segment advantage in Eq. (2). Taking expectations over segments generated by any policy, we can replace each residual by its conditional expectation in Eq. (13) to obtain

$$\mathbb{E}\Big[A(\tau_t)\Big] = \mathbb{E}\left[\sum_{j=0}^{n-1} \gamma^j A(s_{t+j}, a_{t+j})\right]. \tag{15}$$

For a terminal segment, the same identity holds with zero terminal value. In training, we replace V by its estimate $\widehat{V}_\phi$ (Section 4.2). The sum still telescopes, although the resulting score is now an estimate of the behavior advantage.

A.2 Performance difference over segments

Let π and μ start from the same initial-state distribution ρ. Along trajectories of π, the discounted value terms telescope, and their final term vanishes because V is bounded. Hence,

$$J(\pi) - J(\mu) = \mathbb{E}_\pi\left[\sum_{t\geq 0} \gamma^t \delta_t\right] = \mathbb{E}_\pi\left[\sum_{t\geq 0} \gamma^t A(s_t, a_t)\right], \tag{16}$$

which is the performance-difference lemma (Kakade & Langford, 2002). Let $0 = t_0 < t_1 < \cdots$ be successive event times and $n_k := t_{k+1} - t_k$. Applying Eq. (14) to each segment partitions the same sum:

$$J(\pi) - J(\mu) = \mathbb{E}_\pi \left[\sum_{k \geq 0} \gamma^{t_k} A(\tau_{t_k}) \right]. \tag{17}$$

Because Eq. (17) averages over segments generated by π, its right-hand side cannot be evaluated directly from the dataset, whose segments are generated by μ. We instead reweight the recorded segments according to their advantages, with a KL constraint on the change in distribution (Eq. (3)).

A.3 Segment-level policy improvement

We solve Eq. (3) with a generic segment score $f(\tau)$, taking either $f = A$ or $f = \widetilde{A}$. For a positive multiplier λ on the KL constraint and a multiplier η enforcing normalization, the Lagrangian is

$$\mathcal{J}(p, \lambda, \eta) := \mathbb{E}_{\tau \sim p}[f(\tau)] - \lambda\big(\mathrm{KL}(p \,\|\, p_\mu) - \varepsilon\big) - \eta\Big(\int p(\tau)\, \mathrm{d}\tau - 1\Big). \tag{18}$$

Setting its functional derivative with respect to $p(\tau)$ to zero gives $f(\tau) - \lambda(\log[p(\tau)/p_\mu(\tau)] + 1) - \eta = 0$. Normalization therefore yields

$$p^*(\tau) = \frac{p_\mu(\tau) e^{f(\tau)/\lambda}}{\mathbb{E}_{\tau' \sim p_\mu}[e^{f(\tau')/\lambda}]}. \tag{19}$$

Because $\lambda > 0$, the negative KL term makes this stationary point the unique maximum of the Lagrangian. Taking $f = A$ gives exponential advantage weights, while $f = \widetilde{A}$ with $\beta := 1/\lambda$ gives the normalizer $Z(\beta)$ in Eq. (5). In training, we select β directly (Section 4.2).

Projection onto the policy. To turn the reweighted segment distribution into a policy update, we fit π_θ by minimizing $\mathrm{KL}(p^* \,\|\, p_\theta)$. Here p_θ is the segment distribution induced by starting from the dataset's segment initial states and following π_θ. For a recorded segment, the initial-state distribution and environment transitions do not depend on θ, and its event boundaries depend only on the states. Its log probability therefore has the form

$$\log p_\theta(\tau) = C(\tau) + \sum_{(s,a) \in \tau} \log \pi_\theta(a \mid s), \tag{20}$$

where $C(\tau)$ does not depend on θ. Substituting the density ratio p^*/p_μ gives

$$\max_\theta \; \mathbb{E}_{\tau \sim p_\mu} \left[\frac{p^*(\tau)}{p_\mu(\tau)} \sum_{(s,a) \in \tau} \log \pi_\theta(a \mid s) \right]. \tag{21}$$

With $f = \widetilde{A}$, maximizing this expression is equivalent to minimizing Eq. (5). Every transition in a segment receives the same weight.

A.4 Properties of quantile reweighting

For each group g, the cumulative distribution function of the advantage under the conditional segment distribution $p_{\mu|g}$ is

$$F_{\mu|g}(a) := \Pr_{\tau' \sim p_{\mu|g}}[A(\tau') \leq a]. \tag{22}$$

Then $q_\mu(\tau) = F_{\mu|g(\tau)}(A(\tau))$. The reweighted distribution is $p^* = w\, p_\mu$, where $w := e^{\beta \widetilde{A}}/Z(\beta)$. We now prove Proposition 1.

Proof of Proposition 1. (i) Support. Since $-\sqrt{3} \leq \widetilde{A} \leq \sqrt{3}$, the weight is finite and strictly positive. Hence p^* and p_μ have the same support.

(ii) Invariance. Within each group, any strictly increasing transformation preserves the ordering of advantages. This leaves q_μ, the weight, and p^* unchanged even if the transformation differs between groups. □

When the advantages within a group are distinct, their ranks also stay unchanged under strictly increasing transformations. Any statistic with this invariance depends on the advantages only through their ranks within that group (Lehmann & Romano, 2005).

B EXPERIMENTAL SETUP

This appendix describes the mjlab humanoids and the nine IsaacLab humanoids used on flat ground (Appendix B.1), the mjlab tasks (Appendices B.2 and B.3), the reward (Appendix B.4), the expert checkpoints (Appendix B.6) and the construction of the offline datasets (Appendix B.7). The mjlab tasks run in MuJoCo (Todorov et al., 2012) through the GPU-parallel mjlab framework (Zakka et al., 2026).

B.1 ROBOTS

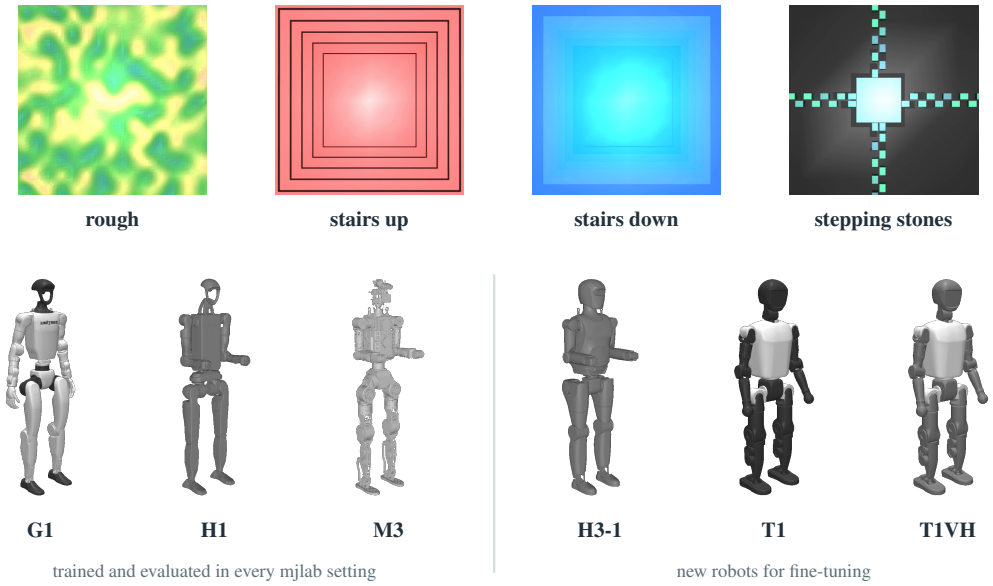


Figure 7: **The mjlab terrains and humanoids.** The top row shows the four terrains at level 6 from above. In the bottom row, G1, H1 and M3 are used in the main mjlab settings, while H3-1, Booster T1 and T1VH are used for fine-tuning. T1VH has the same skeleton as T1 but smaller feet and different motors, with uniform joint stiffness 75 rather than 34 to 63 (Table 1). It is shown with lighter limbs. Robot renders are not to scale.

The main mjlab settings use the Unitree G1 and H1 (Unitree Robotics, 2026) and the M3 (Ly et al., 2026) (Figure 7). For fine-tuning, we use three new robots: H3-1, Booster T1 from mjlab_playground (Zakka et al., 2026), and T1VH, a T1 with different motors and feet. Each robot is controlled through the same twelve leg joints, namely hip pitch, roll, and yaw, knee, and ankle pitch and roll on each leg. The remaining joints are fixed. The three main robots differ in mass by a factor of two and in leg length by a quarter (Table 1), so each faces a different physical challenge on the same terrain.

Actions. The policy outputs one number per joint, $\boldsymbol{a} \in \mathbb{R}^{12}$. The environment turns it into a joint-position target $\boldsymbol{q}^* = \bar{\boldsymbol{q}} + \boldsymbol{s} \odot \boldsymbol{a}$ around the default pose $\bar{\boldsymbol{q}}$, with per-joint scales $\boldsymbol{s}$, and a PD controller with diagonal gain matrices $\boldsymbol{K}_p$ and $\boldsymbol{K}_d$ produces the torques $\boldsymbol{\tau} = \boldsymbol{K}_p(\boldsymbol{q}^* - \boldsymbol{q}) - \boldsymbol{K}_d\dot{\boldsymbol{q}}$, clipped to the effort limits. Gains and limits are those of each robot. On G1 and H1, the gains of a joint with armature I follow the reflected-inertia rule $k_p = I\omega^2$ and $k_d = 2\zeta I\omega$, with $\omega = 2\pi \cdot 10\,\mathrm{Hz}$ and $\zeta = 2$.

Gait clock. Every task carries a gait clock $(\sin 2\pi t/T_\mathrm{g},\ \cos 2\pi t/T_\mathrm{g})$, where t is the time since the reset and the period T_g is $0.8\,\mathrm{s}$ on G1 and H1 and $1.3\,\mathrm{s}$ on M3. The gait rewards of Appendix B.4 tie

the two feet to the two half-cycles of this clock, so a half-cycle is one footstep, which lasts 20 control steps on G1 and H1 and 32 or 33 on M3. This is the segment of SOFIE (Appendix B.7).

Table 1: **Robot specifications** with twelve actuated leg joints. Nominal height is the pelvis height in the default pose. Falling below the crawl height triggers the *crawling* termination. Falling below the pit height ends an episode on stepping stones (Appendix B.2). H3-1, T1, and T1VH are the new robots used for fine-tuning. T1 and T1VH are evaluated only on stairs up, so no pit height applies.

	trained and evaluated			new robots for fine-tuning		
	G1	H1	M3	H3-1	T1	T1VH
mass (kg)	33.3	67.0	51.4	77.8	31.6	31.6
nominal height (m)	0.80	1.03	0.90	0.90	0.67	0.67
leg length, hip to ankle (m)	0.656	0.808	0.765	0.801	0.530	0.535
hip separation (m)	0.233	0.326	0.256	0.267	0.212	0.212
foot length (m)	0.206	0.270	0.264	0.265	0.225	0.200
foot width (m)	0.073	0.080	0.100	0.105	0.100	0.080
stride length (m)	1.28	1.52	1.50	1.20	1.05	1.06
gait period T_g (s)	0.8	0.8	1.3	1.3	0.8	0.8
footstep (control steps)	20	20	32–33	32–33	20	20
crawl height (m)	0.65	0.837	0.731	0.731	0.541	0.545
pit height (m)	0.40	0.515	0.45	0.45	–	–

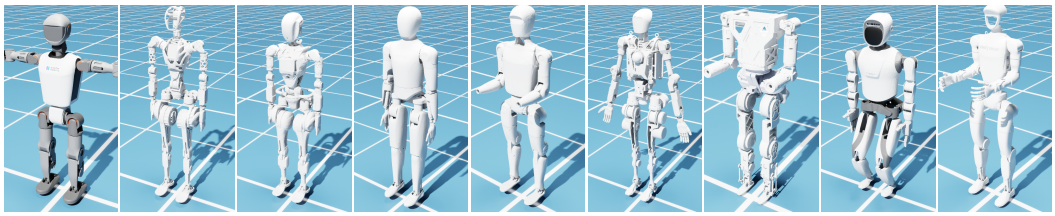


Figure 8: **The nine IsaacLab humanoids used on flat ground**, rendered from the robot_lab models (Fan, 2024) in IsaacLab (Mittal et al., 2025).

Table 2: **The nine IsaacLab humanoids.** Number of actuated joints and total link mass of each robot_lab model (Fan, 2024).

maker	Booster	FFTAI	FFTAI	MagicLab	MagicLab	Openloong	RoboParty	RobotEra	Unitree
robot	T1	GR1T1	GR1T2	Gen1	Z1	Loong	ATOM01	Xbot	G1
joints	23	32	32	14	14	12	23	28	29
mass (kg)	31.6	53.5	54.0	66.4	39.3	77.4	32.0	53.0	33.3

The nine IsaacLab humanoids. The flat-ground suite of Section 6 uses nine humanoids of the robot_lab collection (Fan, 2024) for IsaacLab (Mittal et al., 2025) (Figure 8), namely Booster T1, FFTAI GR1T1 and GR1T2, MagicLab Gen1 and Z1, OpenLoong, RoboParty ATOM01, RobotEra Xbot and Unitree G1, with twelve to thirty-two actuated joints (Table 2). The model descriptions are provided by robot_lab. Every robot walks forward on flat ground under a velocity command and observes proprioception only, so the suite isolates the objective from perception. The expert source of a robot is a converged checkpoint of its own online training run, and its undertrained sources are checkpoints saved earlier in the same run. A recorded row holds the base angular velocity, the projected gravity, the command, the joint positions and velocities, the previous action and a gait clock. The joint terms are zero-padded to thirty-two slots and masked, so that one morphology-conditioned policy reads every robot, and the embodiment descriptor of Appendix B.7 identifies the robot. The datasets of the nine robots with the same share of undertrained rollouts form one training set, on which each method trains one policy.

B.2 ENVIRONMENT

Simulation. The physics step is 5 ms, with the implicit-fast integrator and the Newton solver, and the policy acts every fourth step, at 50 Hz. An episode lasts 20 s, or 1000 control steps, unless a termination ends it earlier.

Observations. One control step of the actor observation holds the base angular velocity and the projected gravity in the body frame, the velocity command, the twelve joint positions relative to $\bar{\boldsymbol{q}}$, the twelve joint velocities, the previous action, the gait clock and a height scan of 17×11 rays on a 1.6×1.0 m grid below the pelvis, 234 numbers in all. The policy receives the last five steps. There is no depth image, so the scan is the only view of the terrain ahead. Uniform noise is added to the proprioceptive terms and the scan at every step, before the history is stacked.

Commands. The command is a forward velocity v_x^c, with zero lateral and yaw components. During expert training, v_x^c is drawn uniformly from a range that widens with the terrain curriculum from $[0.3, 0.5]$ m/s to $[0.3, 0.7]$ m/s ($[0.3, 0.8]$ m/s on stepping stones), redrawn every 3 to 8 s, and 5% of the environments are commanded to stand still. The datasets are recorded and the students evaluated with $v_x^c \sim U(0.3, 0.8)$ m/s, redrawn every 5 to 10 s (Appendix B.7).

Randomization and disturbances. At start-up, the foot friction coefficient is drawn from $U(0.3, 1.2)$, the pelvis center of mass is shifted by up to ± 5 cm per axis (± 6 cm on M3) and every joint encoder receives a bias of up to ± 0.015 rad. At a reset, the base is placed within ± 0.5 m of the tile origin with a uniform yaw, or within ± 0.15 m and ± 0.15 rad on stepping stones, where the platform is small and the command points along the bridge. The joints start within ± 0.1 rad and ± 0.5 rad/s of the default pose. Every 5 to 7 s, the base is pushed with a velocity of up to ± 0.5 m/s along x, and on M3 also up to ± 0.35 m/s along y.

Terminations. An episode ends at the time-out of 20 s, when the base tilts more than 70° from upright (*fell over*), when the pelvis stays below the crawl height of Table 1 for 25 consecutive control steps (*crawling*), and on stepping stones when the base drops below the pit height (*fell into pit*). On M3 stairs and rough ground, the episode also ends as soon as the pelvis is less than 0.5 m above the terrain. Every termination except the time-out counts as a fall and receives the termination penalty of Appendix B.4.

Terrain grid and curriculum. Each task is a grid of 10 rows and 14 columns of tiles (10 columns on stepping stones). The rows are difficulty levels. The tiles of row r are generated at difficulty $(r + \eta)/10$ with $\eta \sim U(0, 1)$, so that the stair rise, for instance, grows from 5 cm in the first row to 20 cm in the last. Each environment is assigned to one tile and is reset there. Following Rudin et al. (2021), an environment on stairs or rough ground moves up one row at the end of an episode if it walked more than 4 m, half a tile, and down one row if it walked less than half the commanded distance $\|v^c\| \cdot 20$ s. On stepping stones, where a bridge ends at a platform, the thresholds are 4.5 m up and 1.5 m down on G1, scaled by leg length on the other robots (5.24 and 1.75 m on M3, 5.54 and 1.85 m on H1). Environments start in a random row between 0 and 5 (row 0 on G1 and H1 stepping stones), and the *terrain level* of an expert is the mean row of its environments.

B.3 TASKS

Figure 9 shows the four tasks. In each of them, the robot starts on a platform or tile at the center of its terrain and is commanded to walk forward at a speed drawn from the ranges of Appendix B.2. The episode ends when the robot falls or after 20 s, and the task measures how well the robot follows the command while staying upright. Stairs and rough ground have the same geometry for every robot, because real stairs do not shrink for a smaller robot. Stepping stones are scaled to the body instead, because they are an abstract foothold task whose difficulty should be the same for a 0.66 m leg and a 0.81 m one. Table 3 lists the generator parameters.

Rough ground. The robot walks across uneven ground with no structure to plan on. The terrain is a height field on a 0.1 m grid, interpolated from heights drawn uniformly from $[0.05, 0.20]$ m in 2 cm increments on a coarser 0.4 m grid. The relief is the same in every row, so rough ground has no difficulty in the curriculum. It is higher than the usual $[0.02, 0.10]$ m because the height scan carries ± 0.1 m of noise, which would obscure smaller terrain variations.

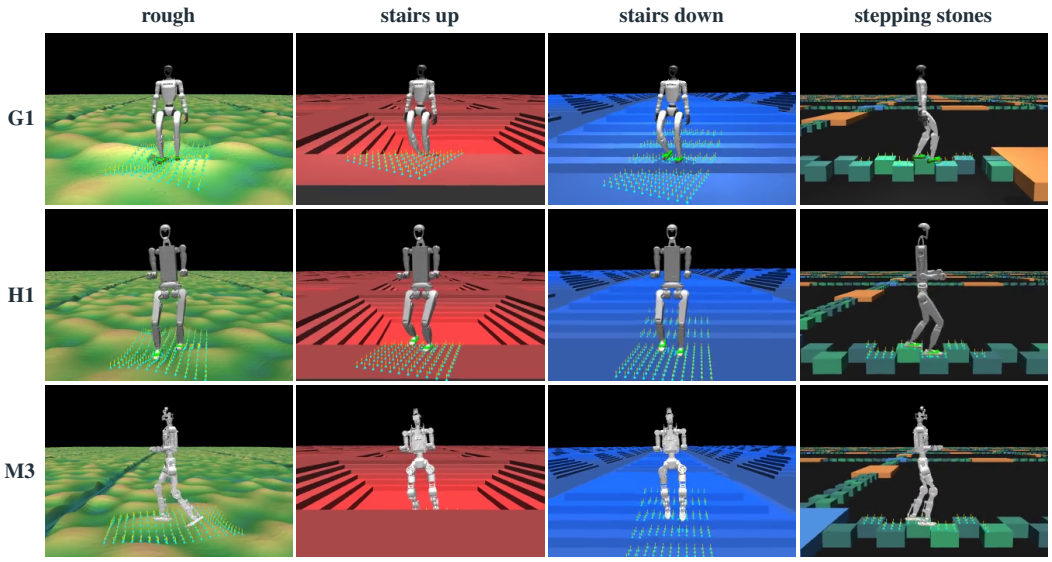


Figure 9: **The four tasks on the three humanoids.** Frames from the experts' rollouts on difficulty row 5, with the height scan of the policy drawn as dots on the ground. The robot walks forward over a random height field (rough), out of a pit of pyramid stairs (stairs up), down a pyramid of stairs (stairs down) and over bridges of stones above a pit (stepping stones). The stairs have the same size for every robot, and the stones are scaled to the body.

Stairs up and stairs down. The robot climbs or descends a flight of stairs whose next step it can only see in the height scan. The terrain is a pyramid of stairs with a run of 0.30 m and a rise that grows linearly with the difficulty from 0.05 to 0.20 m, around a central platform 3 m wide on which the robot spawns. On stairs up the pyramid is inverted into a pit that the robot climbs out of, and on stairs down the robot descends from the top. At level 6, on which every dataset of Appendix B.7 is recorded, the rise is 14.75 cm.

Stepping stones. The robot crosses a pit on small stones that require precise foot placement, so every footstep must be chosen from the scan. Following Zhang et al. (2026), a central platform is joined to four outer platforms by bridges of rectangular stones in a plus layout. Each bridge carries two rows of stones, one per foot, offset by half a pitch so that the feet alternate. The stones are 0.2 m thick and all at the same height, above a pit 1 m deep, and falling into the pit ends the episode. With the difficulty, the stones shrink, and the gaps grow (Table 4). The G1 geometry is the reference. On the other robots, the stone extents scale with the foot, the gaps with the stride, the row spacing with the hip separation, and the tile with the leg length. In expert training, three of the ten columns are flat ground, and the datasets are recorded on stone columns only.

Table 3: **Terrain generators** at the terrain level of every dataset and evaluation. The stepping-stone geometry is per robot (Table 4).

	rough	stairs up, stairs down	stepping stones
tile (m)	8×8	8×8	12 to 14.8 square, per robot
geometry	relief $U(0.05, 0.20)$ m in 2 cm steps on a 0.4 m grid	rise 0.1475 m, run 0.30 m, central platform 3 m	platform 1.5 m, two staggered rows, stones 0.2 m thick, pit 1 m deep, stone size and gap in Table 4

Table 4: **Stepping-stone geometry per robot.** Ranges are (easy, hard). The other robots are scaled from G1, the stone extents by foot size, the gaps by stride, the row spacing by hip separation and the tile by leg length. The last column gives the stone extent and the gap at level 6.

	tile (m)	length (m)	width (m)	gap (m)	spacing (m)	stone, gap at level 6 (m)
G1	12.0	$(0.32, 0.24)$	$(0.32, 0.14)$	$(0.10, 0.40)$	0.25	0.27×0.20, 0.30
H1	14.8	$(0.42, 0.31)$	$(0.35, 0.15)$	$(0.12, 0.49)$	0.34	0.35×0.22, 0.36
M3	14.0	$(0.41, 0.31)$	$(0.44, 0.19)$	$(0.12, 0.47)$	0.28	0.34×0.28, 0.34

B.4 Reward function

The reward at a control step is a weighted sum of terms, $r_t = \sum_k w_k\, r_{k,t}$, using the weights selected for each robot–terrain expert. Between 29 and 36 terms are active per robot and terrain, in six groups. The two *task* terms track the command. The *gait* terms tie the contact of each foot to its half of the gait clock and reward swing time, foot separation, and flat, soft footfalls without slipping. The *posture* terms keep the base and pelvis upright, the joints near the default pose, and the base at a target height above the terrain. The *regularization* terms penalize the action rate, joint velocities and accelerations, torques, power, joint and torque limits, self-collisions and contacts other than the feet. The *terrain* terms penalize a foot that overhangs a step edge, stands on a steep gradient, or misses a target clearance in swing, and on stepping stones they reward landing on the center of a stone, and the distance walked. The *command* terms penalize not moving under a forward command, and every termination other than the time-out receives a large penalty.

The task terms are the velocity-tracking kernels

$$r_{\text{lin}} = \exp\left(-\frac{\|v^c_{xy} - v_{xy}\|^2}{\sigma^2}\right), \qquad r_{\text{yaw}} = \exp\left(-\frac{(\omega^c_z - \omega_z)^2}{\sigma^2}\right), \qquad \sigma = 0.5, \tag{23}$$

where v and ω are the base linear and angular velocities in the body frame, and v^c and ω^c the command. The weighted linear term $w_{\text{lin}}\, r_{\text{lin}}$ is the tracking reward r_{track} of the evaluation. Its weight is 2 on M3, 3.5 on G1 and H1 stairs and rough ground, and 5 on their stepping stones, so r_{track} is bounded by that weight and comparable only within a robot and terrain. This is why the evaluation normalizes the score per robot and terrain (Section 5).

B.5 Expert training

mjlab experts. We train one expert from scratch for each robot–terrain pair in parallel mjlab environments. Training uses REPPO (Voelcker et al., 2026), an on-policy actor-critic method with pathwise policy gradients through a learned action value. Each run uses the tasks of Appendix B.3, the terrain curriculum of Appendix B.2 and the reward of Appendix B.4. The actor observes the same inputs as the student policy (Appendix B.2). The critic also receives privileged simulator state, including the true base velocity and foot contacts. The final checkpoint is the expert source, while earlier checkpoints from that run provide undertrained sources (Appendix B.7). We use the same training procedure for the stairs-up experts of H3-1, T1, and T1VH used in fine-tuning.

IsaacLab experts. We train a separate expert for each of the nine flat-ground humanoids using the proximal policy optimization (PPO) (Schulman et al., 2017) implementation of rsl_rl (Schwarke et al., 2025) on the velocity task of its robot_lab model (Appendix B.1). As in mjlab, the converged checkpoint provides the expert source, while earlier checkpoints from the same run provide the undertrained sources.

B.6 Expert checkpoints

Each of the twelve mjlab experts climbs the curriculum to a mean terrain level above 5.5 of 9, and a checkpoint from the end of its training is the expert source of Appendix B.7. Table 5 evaluates these checkpoints with deterministic actions on the hardest row of their terrain, in 256 environments for 1500 control steps with 10 s episodes at 0.6 m/s. On stairs, they fall in at most 4.4% of the episodes; on stepping stones and rough ground in 2.9 to 10.6%, and they cover three to six meters per episode.

Table 5: **The twelve expert checkpoints.** *Level* is the mean terrain level of the environments of the expert at the end of training, with stochastic actions. *Falls* and *distance* are the share of episodes that end in a fall and the mean distance covered, with deterministic actions on the hardest row (difficulty 1.0).

robot	terrain	level	falls	distance (m)
G1	stairs up	6.04	0.5%	4.68
	stairs down	5.97	0.1%	5.31
	stepping stones	5.98	9.6%	5.59
	rough	5.85	10.6%	5.07
H1	stairs up	6.03	4.4%	5.10
	stairs down	5.97	0.1%	5.25
	stepping stones	6.13	2.9%	5.83
	rough	5.86	9.4%	5.30
M3	stairs up	5.93	0.0%	3.04
	stairs down	6.06	0.4%	4.11
	stepping stones	6.15	7.2%	4.78
	rough	5.58	5.3%	4.17

B.7 Recording and dataset construction

Recording. A source is one checkpoint rolled out with its policy mean in 256 environments for 1000 control steps, which gives 256 000 rows, or in 1024 environments (1.024 M rows) for the larger datasets. The command is $v_x^c \sim U(0.3, 0.8)$ m/s, redrawn every 5 to 10 s. The terrain is set to level 6, difficulty 0.65, and every environment is reset onto its own tile, so that a source contributes the same number of rows at the same difficulty however often it falls. Rough ground, which has no levels, is recorded on its standard terrain. Each row stores the policy observation of Appendix B.2, the action, the scalar reward and every reward term of Appendix B.4, the termination flag, the terrain tile and a validation flag drawn with probability 0.1. An episode ends only at a fall, so an expert source holds about 256 trajectories and an undertrained one up to about 3 800.

Sources and contamination. Each robot–terrain pair has seven candidate sources. They are the expert checkpoint of Table 5, rolled out twice, and five undertrained checkpoints, saved earlier in the same training run before the expert had learned the terrain, which track the command poorly and fall often. Two of the five are selected per pair by the footstep advantage that SOFIE weights. A value estimate is fitted for 100k steps on all seven sources, the footstep advantage of Eq. (2) is computed for every footstep, and the two checkpoints whose footsteps are best separated from those of the expert, measured by the area under the curve of $\Pr[A_{\text{expert}} > A_{\text{checkpoint}}]$, are kept. On stairs, both undertrained sources come from one such checkpoint, recorded twice. The dataset of a pair holds the two expert sources and the two undertrained ones, four sources of equal size, so half of its rows come from undertrained checkpoints, and no row records its source. We call the share of undertrained sources in a dataset its contamination level.

Footsteps. Footsteps are cut on the gait clock. A segment is one half-cycle of the clock, 20 control steps on G1 and H1, and 32 or 33 on M3, and a segment also ends at every episode end. This is the segment τ of Section 4.1.

Student inputs. A student receives the policy observation of Appendix B.2 without the previous action, that is, the five-step proprioceptive history, the velocity command and the height scan. A policy shared across robots also receives an embodiment descriptor read from the robot model, with nine numbers for the body (mass, center of mass, pelvis height, principal inertia and joint count) and twenty per joint (placement in the kinematic tree, range, effort and velocity limits, actuator parameters and gains, and action scale). The policy outputs the twelve joint targets, and the imitation loss is the squared error to the recorded action, which is the negative log-likelihood of a Gaussian policy with fixed variance.

Evaluation. A student is evaluated on its training tasks at terrain level 6, in 1024 environments for 1500 control steps, under the command distribution of the recording. A shared policy is evaluated on every robot and terrain it was trained on.

C POLICY ARCHITECTURE

Section 4.2 and Figure 3 describe the morphology-conditioned policy, and this section gives its details. Policies trained on individual robot–terrain pairs use an MLP encoder over the same inputs (Appendix E.2).

Tokens. The joint descriptor of Appendix B.7 is rescaled by fixed constants and extended by Fourier features of the joint anchor, which makes left and right joints distinguishable at the input. The result $\bar{\boldsymbol{d}}_j$ has 44 entries. One shared linear map turns the position and velocity history of a joint and $\bar{\boldsymbol{d}}_j$ into a token of width 128. The context token is formed in the same way from the base terms, the command, the body descriptor and, where the task has one, the height scan, which an MLP encodes first. Previous actions are not an input, because with them the per-joint decoder learned to repeat the last action of its joint. During training, every descriptor is perturbed by 5 % multiplicative noise, so that the policy does not rely on its exact values.

Encoder. Two pre-normalization self-attention blocks with four heads act on the context token and the J joint tokens. The attention logits carry an additive bias looked up from the kinematic distance between the two joints, so the kinematic tree acts as a prior rather than an input, and padded joint slots are masked. After attention, the context token is mapped to the robot feature $\boldsymbol{z}$ of width 384.

Decoder. One decoder serves every joint. A two-layer MLP maps the 428 entries of $[\boldsymbol{z}, \bar{\boldsymbol{d}}_j]$ through 256 units to the joint feature $\boldsymbol{h}_j$ of width $d = 128$. A second MLP, with 128 SiLU units, maps $\bar{\boldsymbol{d}}_j$ alone to 129 outputs, the weights $\boldsymbol{w}_j$ and the bias b_j of the last linear layer, and the action is $\hat{a}_j = \boldsymbol{w}_j^\top \boldsymbol{h}_j / \sqrt{d} + b_j$. This is the fixed output head of a single-robot policy with its free rows replaced by rows computed from the joint descriptors, allowing the same network to control robots with twelve or thirty-two joints. The decoder does not read the joint's own token.

D TRAINING AND EVALUATION PROTOCOL

Table 6 summarizes the datasets, batches and evaluations, and Table 7 gives the update budgets. In mjlab, we evaluate the checkpoint with the lowest loss on a validation split comprising ten percent of the rows from each source. The pretraining study and the flat-ground setting instead evaluate the last checkpoint. We normalize scores using the converged and undertrained checkpoints of each robot–terrain pair. In mjlab, these checkpoints follow the policy evaluation protocol. On flat ground, their scores are the mean tracking rewards of their recorded rollouts. The standstill rate is the share of steps with a command of at least 0.1 m/s and a base speed below 0.15 m/s.

Hyperparameters. All methods use a learning rate of 3×10^{-4}, with batch sizes given in Table 6. SOFIE ranks advantages within robot–terrain groups and weights footsteps with inverse temperature $\beta = 3$. Its value model has a 101-bin HL-Gauss head and receives 100k updates before actor training. The truncated Monte Carlo targets use $\gamma = 0.999$ and $H = 200$ in mjlab (Figure 6c), and $\gamma = 0.95$ and $H = 100$ on flat ground, where segments contain single steps. IQL and GC-IQL use expectile 0.9, Polyak rate 0.005 and the same inverse temperature on advantages ranked across the batch. TD3+BC uses $\alpha = 2.5$ and Polyak rate 0.005.

Choice of metric. Velocity tracking measures the task objective, whereas the total return also includes posture and survival rewards that a standing policy can collect. Episode returns additionally summarize only episodes that end within the evaluation window. For the runs in Figure 5b and c, SOFIE leads under tracking, mean and median episode return, and horizon return (Table 8). It also has the lowest standstill rate, at 8 to 9 percent of commanded steps compared with 10 to 29 percent for the baselines.

Table 6: **Training and evaluation protocol.** Transitions denotes the training dataset size. Evaluation gives environments $\times$ control steps.

setting	transitions	batch	evaluation
flat ground, 9 robots	9.0M	2048	1024×1000
one policy per terrain	12.3M	1024	1024×1500
one policy per robot	12.3M	1024	1024×1500
foundation policy	49.2M	2048	1024×1500
segment ablation	12.3M	1024	1024×1500
pretraining	6.1M	512	1024×1500
fine-tuning	5k–100k	512	512×1000

Table 7: **Update budgets.** The SOFIE column gives value pretraining and actor updates, and each baseline receives the listed number of updates. Actor budgets match in mjlab and IsaacLab. The foundation policy uses a value model from an earlier run on the same dataset, and all fine-tuning uses the BC loss.

setting	SOFIE	baselines
flat ground (IsaacLab)	100k + 200k	200k
one policy per terrain or per robot	100k + 1M	1M
foundation policy	100k + 500k	500k
segment ablation	100k + 500k	500k
pretraining	100k + 500k	500k
fine-tuning on a new robot	20k	20k

Table 8: **Evaluation statistics** for the runs in Figure 5b and c, averaged over robot–terrain pairs and seeds. Tracking is the per-step velocity-tracking reward. Mean and median episode returns include episodes ending within the 1500-step window. Horizon return sums rewards until the first fall in each environment. Standstill is the share of commanded steps with base speed below 0.15 m/s. The best result in each setting is bold.

method	tracking	episode return	median episode	horizon return	standstill
One policy per terrain					
SOFIE	**2.55**	**56.4**	**41.9**	**92.3**	**8 %**
BC	2.00	36.5	25.9	69.1	29 %
IQL	2.11	37.1	29.2	50.1	24 %
GC-IQL	2.09	37.6	28.4	51.0	26 %
TD3+BC	1.94	19.7	14.9	22.8	19 %
One policy per robot					
SOFIE	**2.54**	**61.9**	**46.3**	**100.0**	**9 %**
BC	2.05	35.1	25.1	65.0	29 %
IQL	2.26	35.4	26.9	45.2	19 %
GC-IQL	2.22	38.0	28.5	50.6	21 %
TD3+BC	2.32	40.4	28.6	57.5	10 %

E ADDITIONAL RESULTS

E.1 FALLS

Table 9 gives the falls per 1000 steps of the policies in Figures 1 and 5. SOFIE falls least in every setting. On flat ground, where no recorded rollout falls, BC falls about four times as often as the offline RL baselines.

Table 9: **Falls per 1000 steps** of the policies in Figures 1 and 5 (lower is better), in the environments and steps of Table 6. Every termination before the time-out counts as a fall (Appendix B.2). Lowest per setting in bold.

setting	BC	SOFIE	IQL	GC-IQL	TD3+BC
flat ground, nine humanoids (Figure 5a)	4.8	**0.7**	1.2	1.3	1.1
one policy per terrain (Figure 5b)	4.3	**2.1**	4.2	4.0	5.6
one policy per robot (Figure 5c)	4.2	**2.1**	3.7	3.4	3.3
one policy for all robots and terrains (Figure 1)	4.3	**2.2**	4.0	–	–

E.2 RESULTS PER ROBOT AND TERRAIN

Tables 11 and 12 give the robot–terrain scores behind Figure 5b and c. Table 10 summarizes them alongside policies trained separately on each pair of G1 and H1 (Table 13). Each pair has a dataset of 4.096M transitions, split equally between converged and undertrained checkpoints from one online training run. We evaluate per-step velocity tracking in 1024 environments over 1500 steps at terrain level 6, with commands from 0.3 to 0.8 m/s. Scores are normalized per pair, with 0.0 for the undertrained and 1.0 for the converged checkpoint. Negative values fall below the undertrained checkpoint. The separately trained policies share no parameters and use an MLP encoder (Appendix C). Their difference from the shared policies therefore reflects both policy sharing and architecture.

Table 10: **Mean normalized tracking score** across the robot–terrain pairs of each setting, with $\pm$ standard deviation across pairs. The per-pair setting covers G1 and H1 only. Its policies share no parameters and use an MLP encoder, so differences from the shared settings reflect both policy sharing and architecture.

setting	BC	SOFIE	IQL	GC-IQL	TD3+BC
per terrain	0.39 ± 0.31	$\mathbf{0.72 \pm 0.15}$	0.39 ± 0.21	0.40 ± 0.20	0.31 ± 0.28
per robot	0.40 ± 0.31	$\mathbf{0.72 \pm 0.21}$	0.47 ± 0.19	0.45 ± 0.19	0.54 ± 0.19
per pair	0.36 ± 0.24	$\mathbf{0.88 \pm 0.15}$	0.69 ± 0.18	0.65 ± 0.19	0.55 ± 0.17

Table 11: **One policy per terrain**, shared across the three robots. Normalized tracking score for each robot–terrain pair, reported as mean ± standard deviation over seeds. Scores are not clipped.

terrain	robot	BC	SOFIE	IQL	GC-IQL	TD3+BC
stepping stones	G1	0.29 ± 0.05	**0.90 ± 0.03**	0.68 ± 0.04	0.76 ± 0.07	0.33 ± 0.03
	H1	0.04 ± 0.02	**0.61 ± 0.10**	0.15 ± 0.12	0.30 ± 0.11	0.22 ± 0.09
	M3	0.42 ± 0.09	**0.67 ± 0.10**	0.35 ± 0.12	0.21 ± 0.05	0.66 ± 0.09
stairs up	G1	0.40 ± 0.10	**0.90 ± 0.02**	0.41 ± 0.13	0.39 ± 0.02	0.41 ± 0.06
	H1	0.07 ± 0.02	0.56 ± 0.06	**0.56 ± 0.07**	0.53 ± 0.06	0.01 ± 0.20
	M3	**0.95 ± 0.06**	0.85 ± 0.19	0.28 ± 0.28	0.69 ± 0.26	0.77 ± 0.12
stairs down	G1	0.41 ± 0.32	**0.81 ± 0.02**	0.74 ± 0.02	0.48 ± 0.09	0.47 ± 0.09
	H1	0.02 ± 0.07	0.44 ± 0.09	**0.52 ± 0.20**	0.51 ± 0.16	0.09 ± 0.08
	M3	**0.95 ± 0.01**	0.86 ± 0.04	0.39 ± 0.20	0.33 ± 0.11	0.59 ± 0.08
rough	G1	0.53 ± 0.12	**0.80 ± 0.05**	0.33 ± 0.08	0.32 ± 0.16	0.32 ± 0.17
	H1	0.21 ± 0.04	**0.72 ± 0.13**	0.06 ± 0.04	0.09 ± 0.07	-0.07 ± 0.06
	M3	0.42 ± 0.12	**0.56 ± 0.07**	0.22 ± 0.04	0.23 ± 0.09	-0.06 ± 0.12

Table 12: **One policy per robot**, shared across its four terrains. Normalized tracking score for each robot–terrain pair, reported as mean ± standard deviation over seeds. Scores are not clipped.

robot	terrain	BC	SOFIE	IQL	GC-IQL	TD3+BC
G1	stepping stones	0.54 ± 0.08	**0.94 ± 0.01**	0.79 ± 0.08	0.82 ± 0.04	0.77 ± 0.04
	stairs up	0.37 ± 0.37	**0.89 ± 0.02**	0.66 ± 0.08	0.48 ± 0.28	0.83 ± 0.02
	stairs down	0.51 ± 0.35	**0.85 ± 0.03**	0.66 ± 0.07	0.57 ± 0.14	0.67 ± 0.07
	rough	0.58 ± 0.11	**0.88 ± 0.00**	0.56 ± 0.04	0.54 ± 0.12	0.40 ± 0.17
H1	stepping stones	0.04 ± 0.03	0.44 ± 0.21	0.34 ± 0.08	0.25 ± 0.18	**0.53 ± 0.05**
	stairs up	-0.01 ± 0.06	0.37 ± 0.14	0.65 ± 0.05	**0.65 ± 0.11**	0.40 ± 0.09
	stairs down	-0.01 ± 0.04	0.37 ± 0.17	0.51 ± 0.02	**0.55 ± 0.09**	0.44 ± 0.17
	rough	0.29 ± 0.04	**0.67 ± 0.09**	0.28 ± 0.04	0.26 ± 0.09	0.42 ± 0.15
M3	stepping stones	0.14 ± 0.13	**0.88 ± 0.13**	0.19 ± 0.02	0.36 ± 0.25	0.73 ± 0.13
	stairs up	0.79 ± 0.18	**0.79 ± 0.15**	0.38 ± 0.31	0.38 ± 0.31	0.65 ± 0.06
	stairs down	**0.88 ± 0.04**	0.86 ± 0.16	0.30 ± 0.25	0.21 ± 0.16	0.45 ± 0.16
	rough	0.66 ± 0.05	**0.66 ± 0.09**	0.34 ± 0.21	0.30 ± 0.11	0.17 ± 0.07

Table 13: **One MLP policy per robot–terrain pair** for G1 and H1. Normalized tracking score for each pair, reported as mean ± standard deviation over seeds. Scores are not clipped.

robot	terrain	BC	SOFIE	IQL	GC-IQL	TD3+BC
G1	stepping stones	0.48 ± 0.09	**0.99 ± 0.01**	0.94 ± 0.01	0.93 ± 0.04	0.51 ± 0.30
	stairs up	0.50 ± 0.17	**0.99 ± 0.00**	0.70 ± 0.05	0.57 ± 0.03	0.57 ± 0.16
	stairs down	0.65 ± 0.06	**1.00 ± 0.01**	0.84 ± 0.01	0.74 ± 0.13	0.82 ± 0.01
	rough	0.56 ± 0.04	**0.96 ± 0.01**	0.75 ± 0.09	0.71 ± 0.13	0.62 ± 0.12
H1	stepping stones	0.33 ± 0.08	**0.83 ± 0.02**	0.35 ± 0.04	0.32 ± 0.06	0.68 ± 0.01
	stairs up	0.03 ± 0.01	0.65 ± 0.06	0.74 ± 0.00	**0.78 ± 0.01**	0.29 ± 0.07
	stairs down	0.01 ± 0.03	0.65 ± 0.06	**0.66 ± 0.08**	0.64 ± 0.16	0.35 ± 0.15
	rough	0.33 ± 0.14	**0.94 ± 0.01**	0.54 ± 0.16	0.47 ± 0.18	0.54 ± 0.05

E.3 PRETRAINING AND FINE-TUNING

Table 14 evaluates the policies on the three robots used for pretraining. SOFIE has the highest mean score. IQL matches it closely on G1 and H1 but falls more often, while SOFIE leads clearly on M3. BC scores near the undertrained checkpoints on G1 and H1. On G1, it seldom falls but does not track better than the undertrained checkpoint. Table 15 reports all three fine-tuning dataset sizes. Starting from the pretrained SOFIE policy yields higher final returns than training from scratch at every size.

Table 14: **Pretrained policies on their training robots.** Stairs-up rollouts of G1, H1 and M3 were recorded with randomized motor gains, half from converged and half from undertrained checkpoints. Entries give normalized tracking score / horizon return / falls per 1000 steps, averaged over seeds in 1024 environments for 1500 steps. The undertrained and converged checkpoints define scores of 0.0 and 1.0. SOFIE receives 100k value and 500k actor updates, and IQL and BC receive 500k updates.

start	G1	H1	M3	mean score
SOFIE	0.92 / 158 / 0.5	0.75 / 80 / 1.6	0.74 / 75 / 1.1	0.80
IQL	0.91 / 95 / 1.4	0.74 / 42 / 3.0	0.35 / 39 / 1.7	0.66
BC	0.04 / 116 / 0.4	-0.01 / 16 / 5.3	0.42 / 77 / 0.8	0.15

Table 15: **Fine-tuning on new robots.** Return on H3-1, T1 and T1VH relative to their converged checkpoints after 1k or 20k steps of BC fine-tuning. Each robot contributes 5k, 20k or 100k transitions from its converged checkpoint. Training starts from the pretrained SOFIE policy of Table 14 or from scratch. Entries are means over robots and seeds with ± 1 standard error, evaluated in 512 environments for 1000 steps.

	5k transitions		20k transitions		100k transitions	
start	1k steps	20k steps	1k steps	20k steps	1k steps	20k steps
SOFIE (pretrained)	$0.13 \pm .010$	$0.17 \pm .008$	$0.21 \pm .007$	$0.41 \pm .003$	$0.22 \pm .007$	$0.64 \pm .019$
scratch	$0.00 \pm .004$	$0.06 \pm .003$	$0.01 \pm .006$	$0.16 \pm .003$	$0.00 \pm .006$	$0.39 \pm .016$

E.4 FLAT GROUND

Table 16 adds falls and the fraction of environments still upright to Figure 5a, and repeats the comparison using converged-checkpoint data alone. On the mixed-quality dataset, SOFIE scores highest at 0.86 and has the fewest falls, 0.7 per 1000 steps. Since the recorded flat-ground rollouts contain no falls, the undertrained checkpoints differ mainly in speed throughout the rollout, and SOFIE weights individual steps. With only converged checkpoints, every method except TD3+BC scores within 0.03 of BC.

Table 16: **Flat ground, nine humanoids.** Normalized score as in Figure 5a, reported as the IQM over robots and seeds with ± 1 bootstrap standard error. Falls are counted per 1000 steps, and upright is the fraction of environments still standing after 1000 steps. Each method receives 200k updates in total, including value pretraining for SOFIE.

	converged checkpoints only			50 % undertrained		
method	score	falls	upright	score	falls	upright
BC	0.96 ± 0.00	0.1	94 %	0.25 ± 0.03	4.8	50 %
TD3+BC	0.86 ± 0.02	1.3	73 %	0.70 ± 0.04	1.1	66 %
IQL	0.95 ± 0.00	0.1	94 %	0.78 ± 0.02	1.2	81 %
GC-IQL	0.95 ± 0.00	0.1	93 %	0.81 ± 0.01	1.3	81 %
SOFIE	0.94 ± 0.00	0.2	89 %	0.86 ± 0.01	0.7	81 %

E.5 SEGMENT ABLATION

Table 17 adds the falls of each variant to the scores in Figure 6b. The ablation trains on stairs-up and stepping-stone data from G1, H1 and M3. Each run receives 100k value updates and 500k actor

updates. Segments ending at recorded foot-contact events score 0.62 on stairs up and 0.81 on stepping stones, below the gait-clock footsteps used by SOFIE.

Table 17: **Segment ablation.** Normalized tracking score (IQM over robots and seeds, ± 1 bootstrap standard error) and falls per 1000 steps on the stairs-up and stepping-stone datasets of G1, H1 and M3 (Figure 6b). Among the SOFIE rows, only the segment over which one advantage and one weight are shared changes.

	stairs up		stepping stones	
segment	score	falls	score	falls
BC (no weighting)	0.14 $\pm$0.07	2.3	0.20 $\pm$0.06	6.2
AWR, one step, Monte Carlo advantage	0.44 $\pm$0.07	3.1	0.58 $\pm$0.05	2.4
SOFIE, one step	0.57 $\pm$0.09	2.9	0.53 $\pm$0.04	2.5
SOFIE, ten steps	**0.72** $\pm$0.05	2.5	0.70 $\pm$0.03	1.9
SOFIE, random length of 5 to 20 steps	0.64 $\pm$0.04	2.7	0.74 $\pm$0.04	2.1
SOFIE, gait-clock footstep	0.70 $\pm$0.03	**2.2**	**0.85** $\pm$0.02	**1.7**

E.6 Network size

To test whether a larger policy can narrow the gap between BC and SOFIE, we compare two network sizes on the stepping-stone task of the per-terrain setting (Figure 5b). The dataset contains 12.3M transitions, half from undertrained checkpoints. Each policy trains with batch size 1024 and 1M actor updates. SOFIE also receives 100k value updates before actor training. The evaluation covers stepping stones for G1, H1 and M3 under the protocol of Section 5.

BC uses the default network with 1.45M parameters and a wider and deeper network with 6.19M. The latter doubles the width of every layer and uses four attention blocks instead of two. SOFIE uses the default network. Increasing the size raises the BC score from 0.25 to 0.41 and reduces its falls from 5.8 to 3.8 per 1000 steps. Even with this gain, BC remains below SOFIE, which scores 0.73 and falls 2.4 times per 1000 steps with the default network (Table 18). Thus increasing network capacity does not replace footstep-level advantage weighting.

Table 18: **Network size.** One policy for stepping stones in the per-terrain setting (Figure 5b), using the default network or a wider and deeper one. The latter doubles the width of every layer and has four rather than two attention blocks. With the default network, BC and SOFIE differ only in their training objectives. The normalized score is the IQM over G1, H1, M3 and seeds, with ± 1 bootstrap standard error. Falls are per 1000 steps.

method	network	parameters	score	falls
BC	default	1.45M	0.25 $\pm$0.02	5.8
BC	wider and deeper	6.19M	0.41 $\pm$0.04	3.8
SOFIE	default	1.45M	0.73 $\pm$0.03	2.4