ReActor: Reinforcement Learning for Physics-Aware Motion Retargeting

DAVID MÜLLER, Disney Research, Switzerland
AGON SERIFI, Disney Research, Switzerland
SAMMY CHRISTEN, Disney Research, Switzerland
RUBEN GRANDIA, Disney Research, Switzerland
ESPEN KNOOP, Disney Research, Switzerland
MORITZ BÄCHER, Disney Research, Switzerland

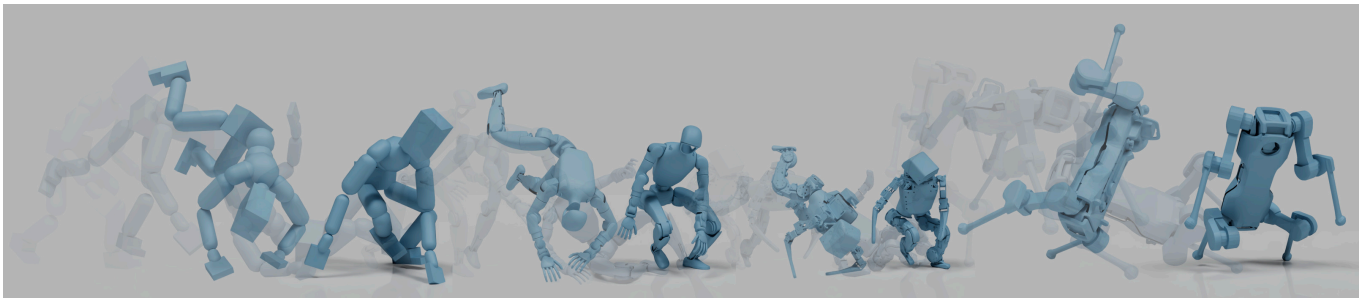

Fig. 1. Physics-aware retargeting of human motion (left) onto two humanoid robots (middle) and a quadruped (right) with varying degrees of freedom and vastly different shapes, sizes, and proportions.

Retargeting human kinematic reference motion onto a robot's morphology remains a formidable challenge. Existing methods often produce physical inconsistencies, such as foot sliding, self-collisions, or dynamically infeasible motions, which hinder downstream imitation learning. We propose a bilevel optimization framework that jointly adapts reference motions to a robot's morphology while training a tracking policy using reinforcement learning. To make the optimization tractable, we derive an approximate gradient for the upper-level loss. Our framework requires only a sparse set of semantic rigid-body correspondences and eliminates the need for manual tuning by identifying optimal values for a parameterization expressive enough to preserve characteristic motion across different embodiments. Moreover, by integrating retargeting directly with physics simulation, we produce physically plausible motions that facilitate robust imitation learning. We validate our method in simulation and on hardware, demonstrating challenging motions for morphologies that differ significantly from a human, including retargeting onto a quadruped.

CCS Concepts: • **Computing methodologies** → **Control methods**; **Reinforcement learning**; **Animation**; • **Mathematics of computing** → *Mathematical optimization.*

1 Introduction

Motion data has become a cornerstone of modern animation and robotics, often serving as reference trajectories for imitation learning with deep reinforcement learning (RL) [Peng et al. 2018]. In practice, such reference motions are typically obtained from human motion capture [Harvey et al. 2020; Mahmood et al. 2019] or reconstructed from video [Goel et al. 2023; Wang et al. 2025]. For character control, however, these motions must be adapted to the target embodiment, which can differ substantially in kinematic structure, body shape, mass distribution, and actuation mechanisms.

To bridge the embodiment gap, reference motions are retargeted to characters or robots via a preprocessing step. Optimization-based approaches minimize pose discrepancies between source and target motions [Araujo et al. 2025; Grandia et al. 2023; Yang et al. 2025a]. However, these methods often require a predefined contact pattern, are prone to local minima, and require substantial manual tuning to scale across diverse motion datasets. Learning-based methods offer an alternative by learning direct mappings from source to target motions [Aberman et al. 2020; Villegas et al. 2018]. However, they often require large datasets of source-target pairs and have primarily been applied to characters with idealized spherical joints, avoiding the complexities of physical characters. Additionally, both approaches can produce physically-implausible motions with artifacts like foot sliding, self-penetration, and abrupt joint movements. These artifacts act as a primary source of performance degradation in downstream tasks such as RL policy training [Araujo et al. 2025].

We instead frame motion retargeting as a reinforcement learning problem within a physics simulation, using a bilevel optimization framework with an RL controller at the lower level, while solving for retargeting parameters in the upper level. The user prescribes coarse correspondences through semantic matching of rigid-body pairs, and the system then solves for optimal offsets between the two embodiments. By jointly optimizing the trajectory and the policy, conflicts between the reference motion and the robot's morphology can be mitigated, thereby minimizing common retargeting artifacts. Our approach inherently respects physical limitations, accounts for discontinuous contact dynamics, and allows the use of non-differentiable objectives.

Authors' Contact Information: David Müller, david.cao@disneyresearch.com, Disney Research, Switzerland; Agon Serifi, agon.serifi@disneyresearch.com, Disney Research, Switzerland; Sammy Christen, sammy.christen@disneyresearch.com, Disney Research, Switzerland; Ruben Grandia, ruben.grandia@disneyresearch.com, Disney Research, Switzerland; Espen Knoop, espen.knoop@disneyresearch.com, Disney Research, Switzerland; Moritz Bächer, moritz.baecher@disneyresearch.com, Disney Research, Switzerland.

It is important to distinguish retargeting from motion imitation. While frameworks such as DeepMimic [Peng et al. 2018] use RL to track a given kinematic reference, our approach addresses the preceding problem of generating a suitable reference to track, bridging the embodiment gap between robot and human source. Unlike motion imitation, we relax strict dynamic requirements by omitting domain randomization and allowing residual force control (RFC) [Yuan and Kitani 2020] to act on the root, which facilitates training a single policy across diverse motions (e.g. AMASS [Mahmood et al. 2019]). Despite these relaxed dynamics, the physics simulation prevents non-physical artifacts like foot sliding, abrupt joint movements, and self-penetration, producing high-quality reference data suitable for downstream tasks.

We demonstrate our retargeting on two humanoid characters, including hardware results on one, and a quadruped (Fig. 1). We validate our method using kinematic metrics against baseline humanoid retargeting methods, demonstrate its effectiveness for the downstream task of learning tracking controllers, and show its applicability to quadrupeds. We further analyze the impact of the bilevel optimization and evaluate generalization to unseen motion data.

Succinctly, we contribute:

- A physics-aware, RL-based retargeting framework producing artifact-free motions without making assumptions on contact patterns.
- A bilevel optimization framework jointly adapting parameterized reference motions and learning tracking policies.
- A retargeting parameterization requiring only sparse, semantic rigid-body correspondences defined by the user in a nominal configuration.

2 Related Work

Motion Retargeting. Motion retargeting has evolved from kinematic optimization minimizing pose discrepancies [Gleicher 1998; Schumacher et al. 2021] to physics-based tracking [Da Silva et al. 2008; Popović and Witkin 1999; Tak and Ko 2005; Zordan and Hodgins 2002]. Other research has addressed varying proportions [Liu et al. 2018; Lyard and Magnenat-Thalmann 2008] and morphologies [Chen et al. 2025b; Hecker et al. 2008] via muscle-based models [Ryu et al. 2021] or interaction-preserving meshes [Ho et al. 2010; Yang et al. 2025a,b]. Data-driven approaches leverage paired supervision [Chen et al. 2025a; Kim et al. 2022; Lee et al. 2023], semantic labels [Gat et al. 2025; Hu et al. 2024], or adversarial objectives [Li et al. 2023; Lim et al. 2019; Villegas et al. 2018; Zhu et al. 2022] to bridge the embodiment gap, often using shared latent spaces [Aberman et al. 2020; Yan et al. 2023] or geometric refinement [Reda et al. 2023; Villegas et al. 2021; Zhang et al. 2023b, 2025, 2023a].

In robotics, retargeting artifacts like foot sliding severely degrade downstream policy training [Araujo et al. 2025]. Consequently, specialized methods for humanoids have been developed [Ayusawa and Yoshida 2017; Darvish et al. 2019; Pollard et al. 2002; Rouxel et al. 2022; Tosun et al. 2015]. Additionally, differentiable simulation can be used to optimize for additional effects such as vibration suppression [Hoshyari et al. 2019]. While recent tools, such as PHC [Luo et al. 2023], ProtoMotions [Tessler et al. 2025], GMR [Araujo et al. 2025], and OmniRetarget [Yang et al. 2025a], streamline sim-to-real transfer, they remain largely kinematic and struggle with temporal coherence. DOC [Grandia et al. 2023] addresses this by optimizing retargeting parameters via differentiable optimal control. Our formulation shares this physics-based focus but differs in three key aspects: we strictly enforce self-collision avoidance, eliminate the need for prescribed contact patterns, and scale to massive datasets using a single retargeting policy.

Physics-based Character Control. Physics-based control has progressed from trajectory optimization [Coros et al. 2010; Hodgins et al. 1995; Hämäläinen et al. 2015; Mordatch et al. 2012; Yin et al. 2007] to imitation-based deep reinforcement learning (RL) [Peng et al. 2018], which is now widely applied in robotics [Fu et al. 2024; Grandia et al. 2024; Liao et al. 2025]. Modern RL methods scale to large datasets [Harvey et al. 2020; Mahmood et al. 2019; Mason et al. 2022] and have moved from residual force control [Luo et al. 2021; Yuan and Kitani 2020; Zhang et al. 2023c] to robust tracking without auxiliary forces [Fussell et al. 2021; Serifi et al. 2024; Wang et al. 2020; Won et al. 2020]. However, most frameworks assume morphological equivalence between the source and target characters, with limited exceptions for body shape variation [Won and Lee 2019] and terrain-optimized design through grammar-based morphologies [Zhao et al. 2020]. Our method plays a complementary role to these physics-based control strategies by generating the morphologically consistent reference motions they require as input.

Bilevel Optimization. Bilevel optimization has been applied to various computational design problems that enforce equilibrium constraints for objectives that involve simulation states (see, e.g., [Coros et al. 2013; Gjoka et al. 2024; Pérez et al. 2015; Tapia et al. 2020]). This paradigm has recently gained traction for RL problems to refine latent dynamics [Zhao et al. 2024] or optimize reward functions [Lu et al. 2026; Xie et al. 2025]. However, we are unaware of the use of stochastic bilevel optimization for retargeting.

The nested nature of these problems presents significant computational challenges and there exists a wide range of algorithmic approaches [Zhang et al. 2024]. A standard technique involves calculating the derivative of the lower-level optima using the implicit function theorem. When RL constitutes the lower-level problem, the primary difficulty lies in differentiating the resulting optimal policy or policy rollout with respect to upper-level decision variables. Unlike previous approaches that rely on the implicit function theorem, our work leverages the specific structure of the retargeting problem to derive a simplified gradient estimate.

3 Bilevel Optimization for Motion Retargeting

Our goal is to retarget a dataset of motions from a source morphology to a target robot by simultaneously finding optimal retargeting parameters $\mathbf{p}$ and learning an optimal retargeting policy $\pi_{\boldsymbol{\phi}}$ parameterized by $\boldsymbol{\phi}$, for a given $\mathbf{p}$. This can be formulated as the following bilevel optimization problem

$$\min_{\mathbf{p}\in\mathcal{P}} \mathcal{L}(\mathbf{p}, \boldsymbol{\phi}^*(\mathbf{p})) \quad \text{subject to} \quad \boldsymbol{\phi}^*(\mathbf{p}) = \arg\max \mathcal{R}(\mathbf{p}, \boldsymbol{\phi}), \quad (1)$$

where $\mathcal{P}$ is a convex set to which the parameters are constrained, $\mathcal{L}(\cdot,\cdot)$ is the upper-level loss function, and $\mathcal{R}(\cdot,\cdot)$ is the lower-level reward function.

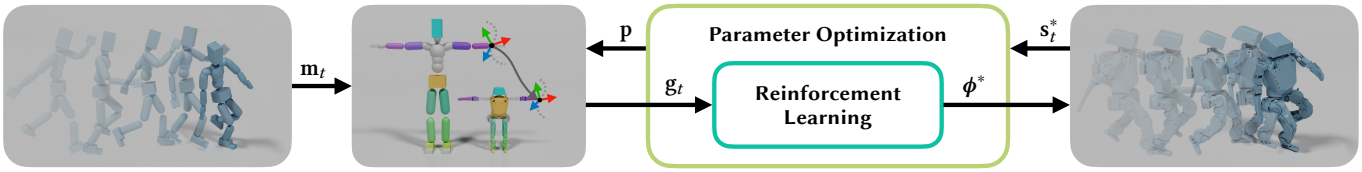

Fig. 2. **Bilevel Optimization for Motion Retargeting.**

As illustrated in Fig. 2, the upper level transforms the source reference motion $\mathbf{m}_t$ (e.g., human motion capture data) into a parameterized reference motion $\mathbf{g}_t$ via a mapping governed by the parameters $\mathbf{p}$. To keep user input to a minimum, we only require semantic correspondences between a sparse set of rigid-body pairs on source and target embodiments. To this end, users select matching rigid bodies on both rigs. The system then automatically solves for the optimal parameters to align the two embodiments.

At the lower level, we employ RL to train an optimal policy tracking the parameterized reference motion. Rolling out the optimal policy results in a state sequence denoted by $\mathbf{s}_t^*$. The upper level compares this simulated state with the reference motion and updates the parameters to minimize the error ℓ, as defined by the upper-level loss function

$$\mathcal{L}(\mathbf{p}, \boldsymbol{\phi}^*(\mathbf{p})) = \mathbb{E}_{\pi_{\boldsymbol{\phi}^*}, \mathbf{s}_0, \mathbf{m}_t}\left[\ell(\mathbf{g}_t - \mathbf{s}_t^*)\right], \tag{2}$$

where the expectation is taken over stochastic rollouts of the optimized policy, given the initial states $\mathbf{s}_0$ sampled as described in Sec. 6, and motions sampled from the dataset. We omit the explicit dependence of $\pi_{\boldsymbol{\phi}^*}$, $\mathbf{g}_t$, $\mathbf{s}_0$, and $\mathbf{s}_t^*$ on $\mathbf{p}$ to simplify the notation.

In the following sections, we first present the optimization algorithm used to solve Eq. (1) (Sec. 4), detail the retargeting parameterization (Sec. 5), and describe the RL setup (Sec. 6).

4 Upper-Level Optimization

A core challenge in our setup is that waiting for the lower-level RL problem to converge before updating the parameters is impractical. We therefore adopt a *single loop* bilevel optimization algorithm [Zhang et al. 2024], which simultaneously updates the lower- and upper-level decision variables. Specifically, we follow the Two-Timescale Approximation (TTSA) [Mingyi Hong et al. 2023], and update the upper-level decision variables at each iteration of the RL algorithm according to

$$\mathbf{p} \leftarrow P_{\mathcal{P}}\left(\mathbf{p} - \eta\, \tilde{\mathrm{d}}_{\mathbf{p}}\mathcal{L}\right), \tag{3}$$

where $P_{\mathcal{P}}(\cdot)$ is the Euclidean projection onto the convex set $\mathcal{P}$, η is the step size, and $\tilde{\mathrm{d}}_{\mathbf{p}}\mathcal{L}$ is a gradient estimate of the upper-level loss, approximating the total derivative $\mathrm{d}_{\mathbf{p}}$ with respect to the parameters. In TTSA, this gradient estimate is constructed in several steps. First, as is standard in bilevel optimization [Zhang et al. 2024], the implicit function theorem is used to derive an expression for $\mathrm{d}_{\mathbf{p}}\mathcal{L}$ based on the optimality conditions of the lower level. Second, since the lower level converges only in the limit, the derived expression is evaluated at the current $\boldsymbol{\phi}$ instead of the optimum. Finally, given the stochastic setting, $\tilde{\mathrm{d}}_{\mathbf{p}}\mathcal{L}$ is computed as an estimate using the sampled data available at the current iteration.

In this work, however, instead of using the implicit function theorem, which requires computing the inverse Hessian of the lower-level problem, we use the structure of the problem to derive a simplified estimate of the upper-level gradient. Consider the total derivative of our error terms

$$\mathrm{d}_{\mathbf{p}}\ell(\mathbf{g}_t - \mathbf{s}_t^*) = \partial_{\mathbf{g}_t}\ell\ \mathrm{d}_{\mathbf{p}}\mathbf{g}_t + \partial_{\mathbf{s}_t^*}\ell\ \mathrm{d}_{\mathbf{p}}\mathbf{s}_t^*, \tag{4}$$

where the sensitivity of the optimal state trajectories $\mathbf{s}_t^*$ with respect to $\mathbf{p}$ is challenging to obtain. We avoid computing this sensitivity by making two assumptions: First, we restrict ourselves to loss functions that depend strictly on the difference between $\mathbf{g}_t$ and $\mathbf{s}_t^*$, implying the property $\partial_{\mathbf{s}_t^*}\ell = -\partial_{\mathbf{g}_t}\ell$ [1]. Second, given that the optimal RL solution depends on $\mathbf{p}$ only through $\mathbf{g}_t$, we can write

$$\mathrm{d}_{\mathbf{p}}\mathbf{s}_t^* = \partial_{\mathbf{g}_t}\mathbf{s}_t^*\ \mathrm{d}_{\mathbf{p}}\mathbf{g}_t, \tag{5}$$

where $\partial_{\mathbf{g}_t}\mathbf{s}_t^*$ represents the change in optimal trajectories given a change in reference motions. We assume this sensitivity takes the form $\alpha\mathbf{I}$, for some $\alpha \in [0, 1]$, intuitively stating that the resulting optimal trajectories adapt (partially) to changes in reference motions.

Substituting these assumptions into Eq. (4) yields a computationally tractable estimate of the upper-level objective gradient

$$\tilde{\mathrm{d}}_{\mathbf{p}}\ell(\mathbf{g}_t - \mathbf{s}_t^*) = (1 - \alpha)\ \partial_{\mathbf{g}_t}\ell\ \mathrm{d}_{\mathbf{p}}\mathbf{g}_t, \tag{6}$$

which eliminates the complex sensitivities of the RL solution and is simply a scaled version of the first term in Eq. (4).

We then proceed similarly to TTSA by evaluating Eq. (6) using the current rather than the optimal trajectories and data sampled at the current iteration. Concretely, given a batch $\mathcal{D}$ of state–reference pairs collected from rollouts of the current policy, we compute

$$\tilde{\mathrm{d}}_{\mathbf{p}}\mathcal{L} = \frac{1}{|\mathcal{D}|}\sum_{(\mathbf{s}_t, \mathbf{g}_t)\in\mathcal{D}} \tilde{\mathrm{d}}_{\mathbf{p}}\ell(\mathbf{g}_t - \mathbf{s}_t). \tag{7}$$

5 Retargeting Parameterization

To define the retargeting objective, the user provides the source and target morphologies in a nominal configuration (e.g., a T-pose) and specifies semantic source-target pairs b of rigid bodies (Fig. 3, Source, Target). We assume that the selected pairs are sparse, meaning that not every body on the source has a corresponding body on the target, and vice versa, as is the case for significantly different morphologies. Furthermore, paired bodies do not need to share the same number of adjacent joints. The user also explicitly selects a *root* pair of bodies, relevant for policy training and simulation (see Sec. 6).

To make our retargeting agnostic to the input, we do not make any assumptions about the location of local coordinate frames on

[1] To see that this identity holds, we introduce the error, $\boldsymbol{\varepsilon} = \mathbf{g} - \mathbf{s}$, omitting super and subscripts, and form the two derivatives, $\partial_{\mathbf{g}}\ell(\boldsymbol{\varepsilon}) = \partial_{\boldsymbol{\varepsilon}}\ell(\boldsymbol{\varepsilon})\ \partial_{\mathbf{g}}\boldsymbol{\varepsilon}$, and, $\partial_{\mathbf{s}}\ell(\boldsymbol{\varepsilon}) = \partial_{\boldsymbol{\varepsilon}}\ell(\boldsymbol{\varepsilon})\ \partial_{\mathbf{s}}\boldsymbol{\varepsilon}$, with $\partial_{\mathbf{g}}\boldsymbol{\varepsilon} = \mathbf{I}$ and $\partial_{\mathbf{s}}\boldsymbol{\varepsilon} = -\mathbf{I}$.

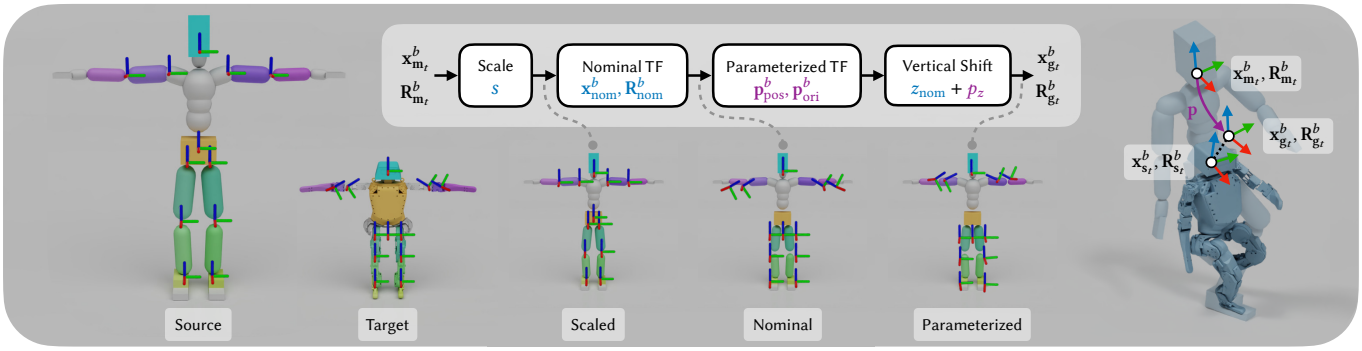

Fig. 3. **Retargeting Parameterization.** A user provides the source and target morphologies in a nominal configuration and defines corresponding rigid-body pairs. A global scale and nominal transformations (TFs) are automatically extracted from the input such that the frames in nominal coordinates align with the corresponding target frames. After this nominal calibration, we introduce parameters in nominal coordinates and a parameterized vertical shift for fine-tuning of source frames during optimization.

the source and corresponding target body. For source rigs, they usually coincide with the joints, but for robots, their location is less standardized. Our goal is now to map the global position $\mathbf{x}^b_{\mathbf{m}_t}$, orientation $\mathbf{R}^b_{\mathbf{m}_t}$, linear velocity $\mathbf{v}^b_{\mathbf{m}_t}$, and angular velocity $\boldsymbol{\omega}^b_{\mathbf{m}_t}$ of the source frame to quantities that we can compare to the corresponding quantities of the moving target body.

To define this mapping, we assume the two nominal configurations to be coarsely aligned and apply a global scaling s to the source configuration, which we derive from the root height ratio $s = h_{\text{target}}/h_{\text{source}}$ (Fig. 3, Scaled). In this aligned nominal configuration, we compute the nominal transformation by expressing the relative offset from the scaled source to the target in the source's local coordinate frame

$$\mathbf{x}^b_{\text{nom}} = (\mathbf{R}^b_{\text{source}})^T(\mathbf{x}^b_{\text{target}} - s\,\mathbf{x}^b_{\text{source}}), \tag{8}$$

$$\mathbf{R}^b_{\text{nom}} = (\mathbf{R}^b_{\text{source}})^T\,\mathbf{R}^b_{\text{target}}, \tag{9}$$

where $(\mathbf{x}^b, \mathbf{R}^b)$ denote the global body frames in the nominal configuration. Note how the frames in nominal coordinates match the ones on the target character after these first two steps (Fig. 3, Nominal, Target).

To enable our outer-level optimization to make adjustments to the location and orientation of these frames, we introduce position and orientation parameters, $\mathbf{p}^b_{\text{pos}}$ and $\mathbf{p}^b_{\text{ori}}$, in local nominal coordinates. Since reference motions from datasets like AMASS often contain floating or penetration artifacts, we precompute a per-motion nominal vertical offset, z_{nom}, following prior work [Luo et al. 2023]. As shown in the supplemental video material, residual floating and penetration remain in the source motions, so we further introduce a learnable per-motion offset p_z to correct artifacts caused by noisy contacts. The full parameterized mapping is therefore

$$\mathbf{x}^b_{\mathbf{g}_t} = \mathbf{R}^b_{\mathbf{m}_t}(\mathbf{R}^b_{\text{nom}}\mathbf{p}^b_{\text{pos}} + \mathbf{x}^b_{\text{nom}}) + s\,\mathbf{x}^b_{\mathbf{m}_t} + (z_{\text{nom}} + p_z)\mathbf{e}_z, \tag{10}$$

$$\mathbf{R}^b_{\mathbf{g}_t} = \mathbf{R}^b_{\mathbf{m}_t}\mathbf{R}^b_{\text{nom}}\text{Exp}(\mathbf{p}^b_{\text{ori}}), \tag{11}$$

$$\mathbf{v}^b_{\mathbf{g}_t} = \boldsymbol{\omega}^b_{\mathbf{m}_t} \times \mathbf{R}^b_{\mathbf{m}_t}(\mathbf{R}^b_{\text{nom}}\mathbf{p}^b_{\text{pos}} + \mathbf{x}^b_{\text{nom}}) + s\,\mathbf{v}^b_{\mathbf{m}_t}, \tag{12}$$

$$\boldsymbol{\omega}^b_{\mathbf{g}_t} = \boldsymbol{\omega}^b_{\mathbf{m}_t}, \tag{13}$$

where we highlight constants that we extract from the nominal configurations and parameters that we optimize. Vector $\mathbf{e}_z$ is the global unit z-axis, and $\text{Exp}(\cdot)$ is the exponential map, mapping the 3D rotation vector $\mathbf{p}^b_{\text{ori}}$ to a rotation matrix [Sola et al. 2018]. While our method is agnostic to the specific parameterization, hence interfaces with user-defined variants, we observe that the above parameterized mapping provides a good balance between simplicity and generalization across diverse morphologies and motions.

The convex set $\mathcal{P}$ is defined by constraining the norm of the optimization parameters

$$\|\mathbf{p}^b_{\text{pos}}\|_2 \le \delta_{\text{pos}}, \quad \|\mathbf{p}^b_{\text{ori}}\|_2 \le \delta_{\text{ori}}, \quad |p_z| \le \delta_z, \tag{14}$$

where δ_{pos}, δ_{ori}, and δ_z are the allowed deviations.

With $\mathbf{g}_t$ fully defined, we can compute differences between the target and simulated state $\mathbf{s}_t$. For position, linear velocity, and angular velocity, we use squared norm loss terms

$$\ell^b_{\mathbf{x}} = \|\mathbf{x}^b_{\mathbf{g}_t} - \mathbf{x}^b_{\mathbf{s}_t}\|^2_2, \quad \ell^b_{\mathbf{v}} = \|\mathbf{v}^b_{\mathbf{g}_t} - \mathbf{v}^b_{\mathbf{s}_t}\|^2_2, \quad \ell^b_{\boldsymbol{\omega}} = \|\boldsymbol{\omega}^b_{\mathbf{g}_t} - \boldsymbol{\omega}^b_{\mathbf{s}_t}\|^2_2. \tag{15}$$

With the rotation term, we penalize the geodesic difference between the two rotations [2]

$$\ell^b_{\mathbf{R}} = \|\text{Log}((\mathbf{R}^b_{\mathbf{s}_t})^T\mathbf{R}^b_{\mathbf{g}_t})\|^2_2, \tag{16}$$

where $\text{Log}(\cdot)$ maps a rotation matrix to a 3D rotation vector [Sola et al. 2018].

Robots frequently have fewer degrees of freedom than digital characters (e.g., 2 DoF quadruped hip joint). In such cases one might want to ignore rotation errors about unactuated axes. We can achieve this by decomposing orientation error into "swing" and "twist" components, where "twist" is the rotation about a user-specified local axis [Dobrowolski 2015]. The orientation loss may then be evaluated on the swing or twist component instead of the full orientation error.

With all tracking losses defined, the upper-level loss function is the sum of all loss terms for all source-target pairs

$$\ell(\mathbf{g}_t - \mathbf{s}_t) = \sum_b (w_{\mathbf{x}}\ell^b_{\mathbf{x}} + w_{\mathbf{R}}\ell^b_{\mathbf{R}} + w_{\mathbf{v}}\ell^b_{\mathbf{v}} + w_{\boldsymbol{\omega}}\ell^b_{\boldsymbol{\omega}}), \tag{17}$$

[2] Even though the loss term is no longer strictly a function of the difference $\mathbf{g} - \mathbf{s}$, Eq. (6) can also be derived on the manifold of rotations [Sola et al. 2018]

where $w_{\mathbf{x}}$, $w_{\mathbf{R}}$, $w_{\mathbf{v}}$, and $w_{\boldsymbol{\omega}}$ are user-specified weights to trade off the relative importance of the error terms. We use the same loss terms in our motion tracking rewards for policy training (see Tab. 1).

6 Lower-Level Reinforcement Learning

As introduced in Sec. 3, the lower level of our bilevel optimization trains a policy $\pi_{\boldsymbol{\phi}}(\mathbf{a}_t \mid \mathbf{o}_t, \mathbf{g}_t)$ to track the parameterized reference motion $\mathbf{g}_t$. In this section, we will define our actions $\mathbf{a}_t$ and observations $\mathbf{o}_t$, and provide a detailed description of the RL problem.

Action Space. The policy outputs joint position setpoints $\mathbf{a}_t^{\text{jts}}$ for Proportional-Derivative (PD) controllers, and auxiliary wrenches $\mathbf{w}_t^{\text{rt}}$, consisting of forces $\mathbf{f}_t^{\text{rt}}$ and torques $\boldsymbol{\tau}_t^{\text{rt}}$, at 50 Hz

$$\mathbf{a}_t \coloneqq (\mathbf{a}_t^{\text{jts}}, \mathbf{w}_t^{\text{rt}}) \qquad \text{with} \qquad \mathbf{w}_t^{\text{rt}} \coloneqq (\mathbf{f}_t^{\text{rt}}, \boldsymbol{\tau}_t^{\text{rt}}). \tag{18}$$

The additional wrench, which acts directly on the character's root [Yuan and Kitani 2020], enables the policy to generalize across large datasets like AMASS [Mahmood et al. 2019], which contain challenging motions such as handstands that are otherwise infeasible due to morphological differences (e.g., characters without hands). To encourage physical realism, we penalize the usage of this external wrench in the reward function. Additionally, we apply a continuous deadband to the wrench action

$$\mathbf{w}_t^{\text{rt}} \coloneqq \operatorname{sgn}(\mathbf{w}_t^{\text{rt}}) \odot \max\left(0, \operatorname{abs}(\mathbf{w}_t^{\text{rt}}) - d\right), \tag{19}$$

with threshold d, to make it easier for the policy to predict exactly zero forces and torques when residuals are unnecessary. The sgn, abs, and max functions return the sign, absolute value, or maximum value of each vector component, and the $\odot$ operator multiplies them component-wise.

Proprioceptive State. The character's proprioceptive state

$$\mathbf{o}_t \coloneqq (h_t^{\text{rt}}, \boldsymbol{\theta}_t^{\text{rt}}, \mathbf{v}_t^{\text{rt}}, \boldsymbol{\omega}_t^{\text{rt}}, \mathbf{q}_t, \dot{\mathbf{q}}_t, \mathbf{a}_{t-1}, \mathbf{a}_{t-2}, \psi_t), \tag{20}$$

contains the height h_t^{rt}, the projected gravity vector $\boldsymbol{\theta}_t^{\text{rt}}$, and the linear and angular velocities $\mathbf{v}_t^{\text{rt}}$ and $\boldsymbol{\omega}_t^{\text{rt}}$, all extracted from the simulation state $\mathbf{s}_t$ of the robot's root body. The observations also include the joint positions $\mathbf{q}_t$ and their velocities $\dot{\mathbf{q}}_t$, and the actions from the previous two time steps, $\mathbf{a}_{t-1}$ and $\mathbf{a}_{t-2}$. Additionally, we introduce a *retargeting phase* variable ψ_t, whose role we will define below.

Initialization. State-of-the-art motion tracking methods typically rely on Reference State Initialization (RSI) [Peng et al. 2018], initializing the robot directly to matching root and joint configurations from the reference trajectory. In our setting, however, the source and target morphologies differ, and the initial joint configuration cannot be extracted directly from the reference. Instead of resolving this mismatch with inverse kinematics, we directly learn the initialization with RL. To this end, we set the root state of the robot to the root state of the source character and sample joint positions from a Gaussian distribution around the nominal robot configuration. To let the policy learn to reach the reference pose from this randomized initial configuration, we use a *retargeting phase* variable $\psi_t \in [0, 1]$ that linearly increases from 0 to 1 at the start of each episode. During this phase, the reference motion is paused and the policy moves the robot toward the start pose before retargeting begins. We also use ψ_t for reward blending and data filtering: Our rigid-body tracking rewards are scaled by ψ_t to avoid large penalties while the character prepares for retargeting, as are the penalties on auxiliary forces and torques to allow the character to use this wrench during initialization. Trajectory segments with $\psi_t < 1$ are excluded from the upper-level data batch $\mathcal{D}$ to ensure that retargeting parameters $\mathbf{p}$ are only optimized once proper tracking is active.

Table 1. **Weighted Reward Terms.** $\boldsymbol{\tau}_t^{\text{jts}}$ and $\ddot{\mathbf{q}}_t$ are joint torques and accelerations.

Name	Reward Term	Weight
Motion Tracking		
Root position xy	$-\ell_{x,y}^{\text{rt}}$	2.0
Root height	$-\ell_{z}^{\text{rt}}$	10.0
Root orientation	$-\ell_{\mathbf{R}}^{\text{rt}}$	2.0
Root lin vel.	$-\ell_{\mathbf{v}}^{\text{rt}}$	0.5
Root ang. vel.	$-\ell_{\boldsymbol{\omega}}^{\text{rt}}$	0.5
Rbs position	$-\ell_{\mathbf{x}}^{b}$	$2.0 \cdot \psi_t$
Rbs orientation	$-\ell_{\mathbf{R}}^{b}$	$2.0 \cdot \psi_t$
Survival	1.0	20
Regularization		
Joint torques	$-\lVert\boldsymbol{\tau}_t^{\text{jts}}\rVert_2^2$	$1.0 \cdot 10^{-4}$
Joint acc.	$-\lVert\ddot{\mathbf{q}}_t\rVert_2^2$	$1.0 \cdot 10^{-6}$
Joint action rate	$-\lVert\mathbf{a}_t^{\text{jts}} - \mathbf{a}_{t-1}^{\text{jts}}\rVert_2^2$	$1.0 \cdot 10^{-2}$
Joint action acc.	$-\lVert\mathbf{a}_t^{\text{jts}} - 2\mathbf{a}_{t-1}^{\text{jts}} + \mathbf{a}_{t-2}^{\text{jts}}\rVert_2^2$	$1.0 \cdot 10^{-2}$
Root Force	$-\lVert\mathbf{f}_t^{\text{rt}}\rVert_1$	$\psi_t \cdot 10^{-2}$
Root Torque	$-\lVert\boldsymbol{\tau}_t^{\text{rt}}\rVert_1$	$\psi_t \cdot 10^{-2}$

Adaptive Motion Sampling. We use an adaptive sampling strategy for the motion clips in the dataset, as they vary in difficulty, to prioritize clips where the policy struggles. For each clip, we maintain a failure count based on early episode terminations, which are triggered when the torso position or orientation reward falls below a threshold. During training, clips are sampled with a probability proportional to their failure rate.

Reward Design. The total discounted reward $\mathcal{R}$ is composed of two terms, a *tracking* reward and a *regularization* reward

$$r_t = r_t^{\text{tracking}} + r_t^{\text{regularization}}. \tag{21}$$

The tracking reward sums up all terms for the rigid-body pairs b, with the root pair treated separately (see Tab. 1). Following common practice in RL, we add regularization rewards to penalize excessive joint torques and encourage smooth joint actions, helping to avoid vibrations and unnecessary effort. We also penalize the use of the auxiliary wrench on the robot's root, encouraging physical plausibility.

7 Results

We evaluate our method on several robotic characters. We compare against two state-of-the-art human motion retargeting methods, GMR [Araujo et al. 2025] and OmniRetarget [Yang et al. 2025a]. We also present ablation studies of our method, demonstrate retargeting of human data onto a quadruped, and showcase real-world use cases: interactive animation of a physics-based character, and physical robot control.

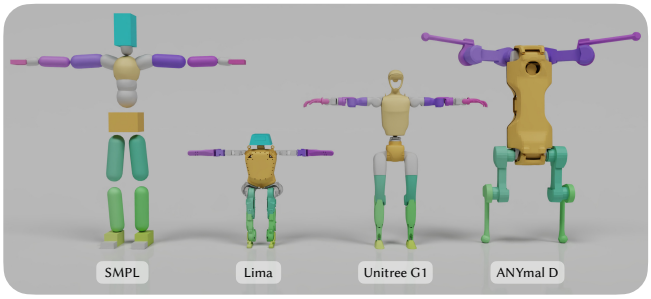

Fig. 4. **Semantic correspondences and nominal configurations.** for the SMPL body model and our target robots: Unitree G1, Lima, and ANYmal D.

Implementation Details. We retarget motions onto two humanoids of different scale: *Unitree G1* (1.27 m, 35 kg, 29 DoF), and *Lima* (0.84 m, 16.2 kg, 20 DoF), a custom small-scale robot. For the Unitree G1, we apply the baseline methods using their provided hyperparameter sets. For Lima, we use the frames after the nominal alignment step as input to the baseline methods. The kinematic correspondences between the source motions and the robot bodies are visualized in Fig. 4. While all our robotic targets have fewer DoF than the human source, the formulation also applies when the target has more DoF, as RL regularization (acceleration, torque, action rate) ensures well-behaved solutions even in under-constrained settings, where additional reward terms could help adjust the results towards a preferred aesthetic goal. Wherever robots have fewer DoF than the source character, we use the twist-swing decomposition. The correspondences were assigned based on structural similarity, without iterative refinement. Unless stated otherwise, we use the AMASS dataset [Mahmood et al. 2019]. Adopting the filtering criteria from PHC [Luo et al. 2023], we remove sequences with human-object interactions or excessive noise. The resulting curated dataset is used directly without additional preprocessing.

We train our policies using PPO [Schulman et al. 2017] with an adaptive learning rate [Rudin et al. 2022]. Both the policy and value function are modeled using multi-layer perceptron (MLP) networks with ELU activations, consisting of three layers with 512 units each. All simulations are performed using Isaac Sim, running 4, 096 environment instances in parallel on a single RTX 5090 GPU. We run our method for 20k iterations (~6 h). Hyperparameters are listed in Tab. 2.

7.1 Baseline Comparison

We compare our method to two state-of-the-art human motion retargeting methods, GMR [Araujo et al. 2025] and OmniRetarget [Yang et al. 2025a], on two humanoid robots at different scales.

Kinematic Evaluation. First, we compare our method against the baselines through several kinematic metrics, as detailed in Tab. 3. The quantitative results are summarized in Tab. 4, with best- and worst-case variability reported in the supplemental material. Additionally, common baseline artifacts are visualized in Fig. 5 and the supplemental video. Our method significantly outperforms both baselines across all metrics on both humanoid platforms. The most severe failures of the optimization-based baselines occur when the solver converges to local minima, typically near kinematic singularities and joint limits. As illustrated in the first column of Fig. 5, the arm over-rotates near a singular configuration, reaches a joint limit, and gets stuck in a local minimum. This leads to severe artifacts and self-penetration, making the retargeted motion unusable.

Both baselines exhibit ground contact artifacts. GMR relies purely on data preprocessing and does not apply any height correction during retargeting. As a result, it exhibits both ground penetration and floating (Fig. 5, second row). OmniRetarget prevents ground penetration by strictly correcting the motion height at each frame while enforcing ground contact to match heuristically-estimated contact patterns from the reference motion. However, these objectives may conflict, leading to foot floating, as seen in Tab. 4. In contrast, our method avoids both artifacts by design. Note that non-zero values for floating and sliding persist, as our method does not strictly enforce contact matching with the reference. Our method avoids self-penetration by explicitly accounting for contact dynamics within the simulation during retargeting, unlike the baselines.

As seen in Tab. 4, OmniRetarget struggles with the Lima platform, likely due to its non-uniform scaling (approximately half human height but similar width) and non-standard root alignment, which degrades contact estimation. Although per-motion tuning could mitigate these issues, OmniRetarget fails to generalize across the full dataset when using nominal reference parameters. However, GMR behaves worse for G1 but performs better for Lima. In terms of computational cost, parallel training and retargeting on the AMASS

Table 2. **Hyperparameters.** The PPO hyperparameters and bilevel optimization parameters used to train the tracking policy.

Param.	Value	Param.	Value
Num. iterations	20 000	δ_{pos}	0.5
Batch size (envs. × steps)	4096 × 24	δ_{ori}	0.5
Num. mini-batches	4	δ_{z}	0.5
Num. epochs	5	w_{x}	10.0
Clip range	0.2	w_{R}	1.0
Entropy coefficient	0.0025	w_{ω}, w_{v}	0.0
Discount factor	0.97	d	0.1
GAE discount factor	0.95		
Desired KL-divergence	0.009		
Max gradient norm	1.0		

Table 3. **Kinematic Evaluation Metrics.** Based on OmniRetarget, with *self-penetration* and *foot floating* added. Where reference contact state is used, this is estimated following [Shimada et al. 2020].

Metric	Description
Ground Penetration	Fraction of motion frames where penetration exceeds 0.01 m. Reported penetration depth is mean across violating frames. If multiple simultaneous ground contacts, record maximum penetration depth per frame.
Self-Penetration	Time, depth computed as for ground-penetration. Collisions within same kinematic chain ignored, to remove false positives.
Foot Sliding	Mean linear velocity of robot's feet during reference ground contact phases.
Foot Floating	Mean of minimum distances between robot's foot and ground during reference ground contact.

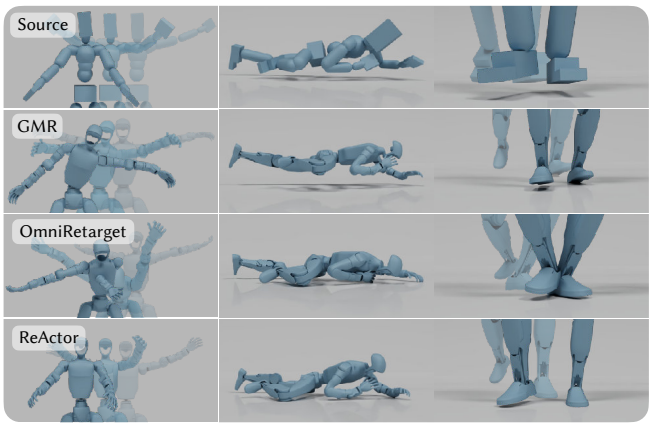

Fig. 5. **Kinematic Artifacts.** Local solver minima, often present near kinematic singularities or joint limits (left). Floating (middle) and self-penetration (right) artifacts.

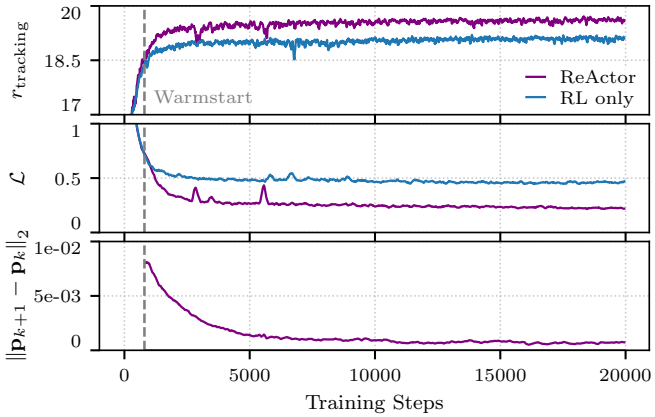

Fig. 6. **Training Curves with and without the Bilevel Optimization.** Tracking reward, upper-level loss, and parameter update rate during training with and without bilevel optimization. The update rate shows outer-loop convergence and is zero without bilevel optimization, as the parameters remain static.

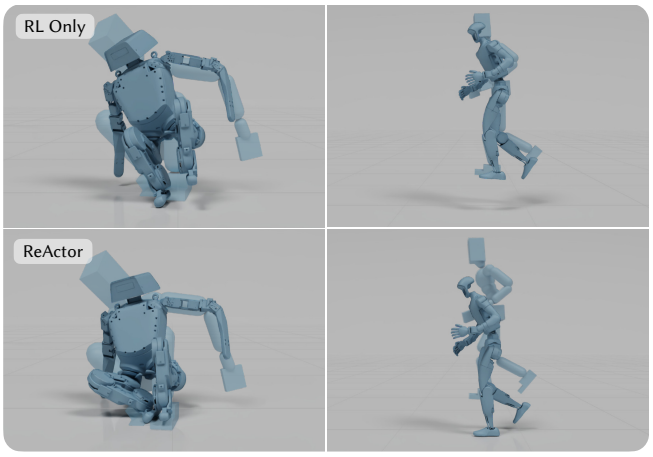

Fig. 7. **Qualitative Comparison With and Without Bilevel Optimization.** With bi-level optimization, results have fewer retargeting artifacts.

dataset requires ~6.5 h on a single GPU, which is comparable to running OmniRetarget (~7 h) and GMR (~5 h) on a CPU.

Downstream RL Performance. A central observation from prior work [Araujo et al. 2025] is that the kinematic quality of retargeted motions strongly influences the success of downstream Reinforcement Learning (RL) training. We train RL tracking policies on the retargeted data from each method and use identical hyperparameters without any method-specific tuning. Unlike [Yang et al. 2025a], which evaluates on a selected subset of 39 sequences from AMASS [Mahmood et al. 2019], we train and evaluate on the entire filtered AMASS dataset [Luo et al. 2023]. We refer to our supplemental material for details about these tracking policies. We report the success rate, measured by the ability of the policy to complete the motion without triggering the termination criteria used during RL training [Yang et al. 2025a], in Tab. 4. Additionally, we report root mean squared errors for the root position, root orientation, and joint position tracking. For each motion, we initialize the episodes with random starting frames and run for 5 seconds, unless the episode ends early due to the termination criteria. Note that the success rate of our approach outperforms the baselines on both G1 and Lima. Similarly, the tracking policies show smaller joint position and root pose errors when trained with data from ReActor.

7.2 Ablation Studies

Parameterization. We study the impact of the proposed parameterization by training policies with and without the upper-level optimization, as shown in Fig. 6, Fig. 7, and the video. The bilevel optimization consistently reduces tracking loss and leads to higher tracking rewards, confirming the effectiveness of the proposed parameterization. Qualitative results further demonstrate that the robot tracks motions in a more natural and physically consistent manner, as the policy and reference are iteratively refined to better align with each other.

In the video, we further explore the impact of source-robot correspondences and target parameterization. We replace the dense set of correspondences with a sparse set of only the root and end effectors. The result deviates further from the source, as would be expected, but the method remains stable. As an additional robustness test against mapping mismatches in the input, we assign the left robot hand to track the motion of the head of the source, and the method remains stable.

We also compare our parameterization against an orientation-only baseline. While both produce stable results, the inclusion of positional targets is beneficial for motions like jumping, where it improves the timing and fidelity of lift-off and touchdown.

Generalization. We evaluate how well a policy, trained on the full AMASS dataset, generalizes to unseen motion data, enabling a

Table 4. **Retargeting Evaluation.** Comparison against baselines on the PHC-filtered AMASS subset. We report mean and standard deviation for ground penetration, self-penetration, foot sliding, and foot floating. For the downstream RL task, we report success rate, and root position, root orientation, and joint root mean squared tracking errors.

	Ground-Pen.		Self-Pen.		Foot Slide	Foot Float	Downstream RL			
Method	Time ↓	Depth [cm] ↓	Time ↓	Depth [cm] ↓	Vel. [cm/s] ↓	Height [cm] ↓	Success [%] ↑	Pos. [cm] ↓	Ori. [deg] ↓	Joints [deg] ↓
Unitree G1										
GMR	0.53 ± 0.30	2.74 ± 0.92	0.07 ± 0.15	5.63 ± 2.80	1.25 ± 3.86	0.34 ± 0.95	89.93	2.99 ± 4.94	4.48 ± 5.09	9.79 ± 12.06
OmniRetarget	**0.00 ± 0.00**	**0.00 ± 0.00**	0.12 ± 0.13	3.27 ± 1.55	2.00 ± 1.23	0.49 ± 0.19	95.51	1.84 ± 3.18	3.32 ± 2.77	6.62 ± 7.17
ReActor	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.17 ± 1.25**	**0.12 ± 0.32**	**97.45**	**1.11 ± 2.39**	**1.87 ± 1.68**	**4.22 ± 2.22**
Lima										
GMR	0.34 ± 0.41	2.42 ± 0.43	0.04 ± 0.13	3.56 ± 1.89	1.97 ± 4.42	1.27 ± 2.59	91.23	3.53 ± 5.25	4.38 ± 5.36	10.45 ± 18.10
OmniRetarget	**0.00 ± 0.00**	**0.00 ± 0.00**	0.09 ± 0.23	3.89 ± 2.17	2.40 ± 2.09	0.31 ± 0.23	79.85	5.86 ± 9.15	6.32 ± 6.44	14.10 ± 16.37
ReActor	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.47 ± 2.38**	**0.02 ± 0.08**	**95.07**	**1.46 ± 1.88**	**3.00 ± 2.07**	**4.38 ± 2.92**

Table 5. **Generalization Evaluation.** Retargeting policy tracking errors on the 100STYLE test set, measured against a pseudo-ground-truth reference (a policy trained on the test set). Rows show a policy trained on the 100STYLE training split and one trained on AMASS.

Training	Pos. [cm] ↓	Ori. [deg] ↓	Joints [deg] ↓
100STYLE	0.12 ± 0.07	5.57 ± 2.46	5.79 ± 2.04
AMASS	0.19 ± 0.15	6.18 ± 2.01	6.93 ± 1.53

single retargeting policy to be reused across datasets and within real-time user applications. As there is no absolute retargeting ground truth, we train a retargeting policy directly on the test data and use its output as a pseudo-ground-truth reference. To this end, we randomly partition the 100STYLE dataset by selecting 50% of the motions for training and reserving the remaining half for testing. We train separate retargeting policies for Lima on the 100STYLE training subset, the full AMASS dataset, and the 100STYLE test subset (pseudo-ground-truth). We compare the retargeting errors between these policies on the pseudo-ground-truth reference in Tab. 5. Note that these errors are smaller than in Tab. 4, as they evaluate the retargeting policy itself, which uses residual forces. In contrast, Tab. 4 evaluates a separate downstream tracking policy trained without residual forces.

External Force. The external force penalty weight is the most sensitive hyperparameter. Other parameters, such as regularization weights, primarily suppress high-frequency jitter without significantly affecting motion quality. Thus, we analyze the force penalty weight trade-off on retargeting performance in Fig. 8. Increasing the penalty encourages greater physical realism but can lead to failures on more challenging motions, whereas weaker penalties improve retargeting success at the cost of physical plausibility. We report the mean over motions of the maximum applied forces and torques, together with the mean upper-level loss $\mathcal{L}$ and the total failure count. A motion is considered a failure if, after training, the retargeting policy triggers the termination condition on the root pose (orientation error > 45° ∨ position error > 1 m).

7.3 Use Cases

Retarget Human Data onto a Quadruped. We demonstrate the versatility of our approach by applying our method to a quadruped,

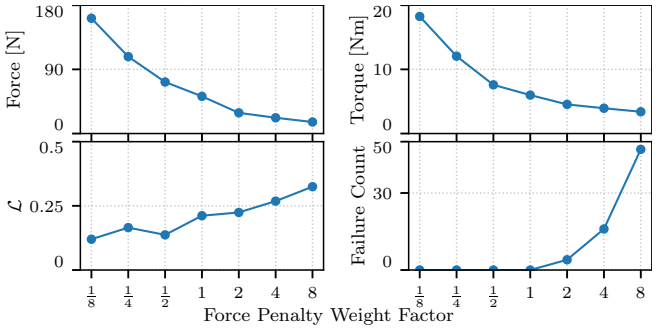

Fig. 8. **Effect of the External Force Penalty.** Mean over motions of the maximum applied forces and torques, the mean upper-level loss $\mathcal{L}$, and the total number of failures as a function of the force penalty weight.

ANYmal D (50 kg, 12 DoF). See Fig. 4 for semantic correspondences, and the video for retargeted motions. Even with widely different embodiments, the motion's visual appearance is preserved, while also providing valuable insights into the method's limitations. As the embodiment gap becomes larger, more nuanced reward tuning becomes necessary. In the video, we show an example where ANYmal gives up tracking in favor of reducing external force usage.

Interactive Animation of Physics-based Character. To demonstrate practical applicability, we deploy the retargeting policy in an interactive user setting, where an artist modifies a motion sequence on the fly while the policy retargets the motion in real time to the robot morphology (see video). The system runs at 88.3 Hz, exceeding real-time requirements and enabling seamless, responsive motion editing. We also see applications in the real-time retargeting of a performance onto a robot during a capture session.

Physical Robot Control. As shown in Tab. 4, our method significantly improves downstream reinforcement learning performance. We further validate this by deploying goal-conditioned tracking policies, trained with a DeepMimic-style reward formulation [Peng et al. 2018] on data retargeted by **ReActor**, directly on the physical Lima robot. We evaluate a range of challenging motions within the

robot's hardware limits, showcased in the video. The successful sim-to-real transfer demonstrates that our retargeting pipeline produces motion references suitable for real-world robotic deployment.

8 Conclusion

This paper introduces a novel bilevel optimization framework that effectively bridges the embodiment gap between human motion and diverse robotic morphologies. By framing retargeting as a joint problem where retargeting parameters and an RL tracking policy are optimized simultaneously, the system significantly reduces common artifacts. This integrated approach, supported by a simplified gradient estimate for computational efficiency, allows the framework to produce physically plausible motions that serve as high-quality reference data for downstream imitation learning tasks. Moreover, we also see applications in training generative motion models and real-time retargeting, e.g. during live mocap sessions.

The current external force penalty weight serves as a tuning parameter that enables a user to prioritize either physical realism or successful retargeting of the most extreme motions in the dataset. Ultimately, the retargeting of physically impossible motions remains an ill-posed problem, where it is unclear if the robot should walk up the virtual staircase or if the retargeting method should project the motion to the ground. Regardless, providing more user-control over the result is desirable.

While ReActor establishes a strong foundation for physics-aware retargeting, several avenues for future research remain. Currently, the optimized parameterization is assumed to be constant over time. Exploring time-varying parameterizations could further increase the solution space, though it may introduce new challenges. Moreover, automating the semantic correspondence selection could further reduce user input. More broadly, the bilevel formulation presented herein holds significant promise for more complex scenarios where reference tracking must be balanced with auxiliary objectives, such as obstacle avoidance, manipulation, or even automated robot design. We believe that such tasks should be approached in an integrated, bilevel manner, rather than as sequences of decoupled steps.

References

Kfir Aberman, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. 2020. Skeleton-aware networks for deep motion retargeting. *ACM Trans. Graph.* 39, 4 (2020). doi:10.1145/3386569.3392462

Joao Pedro Araujo, Yanjie Ze, Pei Xu, Jiajun Wu, and C. Karen Liu. 2025. Retargeting Matters: General Motion Retargeting for Humanoid Motion Tracking. doi:10.48550/arXiv.2510.02252

Ko Ayusawa and Eiichi Yoshida. 2017. Motion Retargeting for Humanoid Robots Based on Simultaneous Morphing Parameter Identification and Motion Optimization. *IEEE Trans. Robot.* 33, 6 (2017). doi:10.1109/TRO.2017.2752711

Ling-Hao Chen, Yuhong Zhang, Zixin Yin, Zhiyang Dou, Xin Chen, Jingbo Wang, Taku Komura, and Lei Zhang. 2025b. Motion2Motion: Cross-topology Motion Transfer with Sparse Correspondence. In *ACM SIGGRAPH Asia.* doi:10.1145/3757377.3763811

Xingyu Chen, Hanyu Wu, Sikai Wu, Mingliang Zhou, Diyun Xiang, and Haodong Zhang. 2025a. Implicit Kinodynamic Motion Retargeting for Human-to-humanoid Imitation Learning. doi:10.48550/arXiv.2509.15443

Stelian Coros, Philippe Beaudoin, and Michiel van de Panne. 2010. Generalized biped walking control. *ACM Trans. Graph.* 29, 4 (2010). doi:10.1145/1778765.1781156

Stelian Coros, Bernhard Thomaszewski, Gioacchino Noris, Shinjiro Sueda, Moira Forberg, Robert W. Sumner, Wojciech Matusik, and Bernd Bickel. 2013. Computational design of mechanical characters. *ACM Trans. Graph.* 32, 4 (2013). doi:10.1145/2461912.2461953

M. Da Silva, Y. Abe, and J. Popović. 2008. Simulation of Human Motion Data using Short-Horizon Model-Predictive Control. *Comput. Graph. Forum.* 27, 2 (2008). doi:10.1111/j.1467-8659.2008.01134.x

Kourosh Darvish, Yeshasvi Tirupachuri, Giulio Romualdi, Lorenzo Rapetti, Diego Ferigo, Francisco Javier Andrade Chavez, and Daniele Pucci. 2019. Whole-Body Geometric Retargeting for Humanoid Robots. In *Int. Conf. Humanoid Robots.* doi:10.1109/Humanoids43949.2019.9035059

Przemysław Dobrowolski. 2015. Swing-twist decomposition in clifford algebra. *arXiv preprint arXiv:1506.05481* (2015).

Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. 2024. HumanPlus: Humanoid Shadowing and Imitation from Humans. In *Conf. Robot Learn.*

Levi Fussell, Kevin Bergamin, and Daniel Holden. 2021. SuperTrack: motion tracking for physically simulated characters using supervised learning. *ACM Trans. Graph.* 40, 6 (2021). doi:10.1145/3478513.3480527

Inbar Gat, Sigal Raab, Guy Tevet, Yuval Reshef, Amit Haim Bermano, and Daniel Cohen-Or. 2025. AnyTop: Character Animation Diffusion with Any Topology. In *ACM SIGGRAPH.* doi:10.1145/3721238.3730621

Arvi Gjoka, Espen Knoop, Moritz Bächer, Denis Zorin, and Daniele Panozzo. 2024. Soft Pneumatic Actuator Design using Differentiable Simulation. In *ACM SIGGRAPH.* doi:10.1145/3641519.3657467

Michael Gleicher. 1998. Retargetting motion to new characters. In *ACM SIGGRAPH.* doi:10.1145/280814.280820

Shubham Goel, Georgios Pavlakos, Jathushan Rajasegaran, Angjoo Kanazawa, and Jitendra Malik. 2023. Humans in 4D: Reconstructing and Tracking Humans with Transformers. In *Int. Conf. Comput. Vis.* doi:10.1109/ICCV51070.2023.01358

Ruben Grandia, Farbod Farshidian, Espen Knoop, Christian Schumacher, Marco Hutter, and Moritz Bächer. 2023. DOC: Differentiable Optimal Control for Retargeting Motions onto Legged Robots. *ACM Trans. Graph.* 42, 4 (2023). doi:10.1145/3592454

Ruben Grandia, Espen Knoop, Michael Hopkins, Georg Wiedebach, Jared Bishop, Steven Pickles, David Müller, and Moritz Bächer. 2024. Design and Control of a Bipedal Robotic Character. In *Robotics: Science and Systems XX.* doi:10.15607/RSS.2024.XX.103

Félix G. Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. 2020. Robust motion in-betweening. *ACM Trans. Graph.* 39, 4 (2020). doi:10.1145/3386569.3392480

Chris Hecker, Bernd Raabe, Ryan W. Enslow, John DeWeese, Jordan Maynard, and Kees van Prooijen. 2008. Real-time motion retargeting to highly varied user-created morphologies. In *ACM SIGGRAPH.* doi:10.1145/1399504.1360626

Edmond S. L. Ho, Taku Komura, and Chiew-Lan Tai. 2010. Spatial relationship preserving character motion adaptation. *ACM Trans. Graph.* 29, 4 (2010). doi:10.1145/1778765.1778770

Jessica K. Hodgins, Wayne L. Wooten, David C. Brogan, and James F. O'Brien. 1995. Animating human athletics. In *ACM SIGGRAPH.* doi:10.1145/218380.218414

Shayan Hoshyari, Hongyi Xu, Espen Knoop, Stelian Coros, and Moritz Bächer. 2019. Vibration-minimizing motion retargeting for robotic characters. *ACM Trans. Graph.* 38, 4 (2019). doi:10.1145/3306346.3323034

Lei Hu, Zihao Zhang, Chongyang Zhong, Boyuan Jiang, and Shihong Xia. 2024. Pose-Aware Attention Network for Flexible Motion Retargeting by Body Part. *IEEE Trans. Vis. Comput. Graph.* 30, 8 (2024). doi:10.1109/TVCG.2023.3277918

Perttu Hämäläinen, Joose Rajamäki, and C. Karen Liu. 2015. Online control of simulated humanoids using particle belief propagation. *ACM Trans. Graph.* 34, 4 (2015). doi:10.1145/2767002

Sunwoo Kim, Maks Sorokin, Jehee Lee, and Sehoon Ha. 2022. HumanConQuad: Human Motion Control of Quadrupedal Robots using Deep Reinforcement Learning. In *ACM SIGGRAPH Asia Emerg. Technol.* doi:10.1145/3550471.3564762

Sunmin Lee, Taeho Kang, Jungnam Park, Jehee Lee, and Jungdam Won. 2023. SAME: Skeleton-Agnostic Motion Embedding for Character Animation. In *ACM SIGGRAPH Asia.* doi:10.1145/3610548.3618206

Tianyu Li, Jungdam Won, Alexander Clegg, Jeonghwan Kim, Akshara Rai, and Sehoon Ha. 2023. ACE: Adversarial Correspondence Embedding for Cross Morphology Motion Retargeting from Human to Nonhuman Characters. In *ACM SIGGRAPH Asia.* doi:10.1145/3610548.3618255

Qiayuan Liao, Takara E. Truong, Xiaoyu Huang, Yuman Gao, Guy Tevet, Koushil Sreenath, and C. Karen Liu. 2025. BeyondMimic: From Motion Tracking to Versatile Humanoid Control via Guided Diffusion. doi:10.48550/arXiv.2508.08241

Jongin Lim, H. Chang, and J. Choi. 2019. PMnet: Learning of Disentangled Pose and Movement for Unsupervised Motion Retargeting. In *Brit. Mach. Vis. Conf.*

Zhiguang Liu, Antonio Mucherino, Ludovic Hoyet, and Franck Multon. 2018. Surface based motion retargeting by preserving spatial relationship. In *ACM SIGGRAPH.* doi:10.1145/3274247.3274507

Renzhi Lu, Jie Wang, Zonghe Shao, Ruijuan Chen, Lijun Zhu, Yuzhi Jiang, Yunyi Pang, Dongfang Liang, Yang Shi, and Han Ding. 2026. Deep Reinforcement Learning for Real-World Humanoid Robot Locomotion Control with Automatic Reward Learning. *Research* 0, ja (2026). doi:10.34133/research.1123

Zhengyi Luo, Jinkun Cao, Alexander Winkler, Kris Kitani, and Weipeng Xu. 2023. Perpetual Humanoid Control for Real-time Simulated Avatars. In *Int. Conf. Comput. Vis.* doi:10.1109/ICCV51070.2023.01000

Zhengyi Luo, Ryo Hachiuma, Ye Yuan, and Kris M. Kitani. 2021. Dynamics-regulated kinematic policy for egocentric pose estimation. In *Advances in Neural Information Processing Systems.*

Etienne Lyard and Nadia Magnenat-Thalmann. 2008. Motion adaptation based on character shape. *Comput. Animat. Virtual Worlds* 19, 3-4 (2008). doi:10.1002/cav.233
Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. 2019. AMASS: Archive of Motion Capture as Surface Shapes. doi:10.48550/arXiv.1904.03278
Ian Mason, Sebastian Starke, and Taku Komura. 2022. Real-Time Style Modelling of Human Locomotion via Feature-Wise Transformations and Local Motion Phases. *Proc. ACM Comput. Graph. Interact. Tech.* 5, 1 (2022). doi:10.1145/3522618
Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. 2023. A Two-Timescale Stochastic Algorithm Framework for Bilevel Optimization: Complexity Analysis and Application to Actor-Critic. *SIAM J. Optim.* (2023). doi:10.1137/20M1387341
Igor Mordatch, Emanuel Todorov, and Zoran Popović. 2012. Discovery of complex behaviors through contact-invariant optimization. *ACM Trans. Graph.* 31, 4 (2012). doi:10.1145/2185520.2185539
Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. 2018. DeepMimic: example-guided deep reinforcement learning of physics-based character skills. *ACM Trans. Graph.* 37, 4 (2018). doi:10.1145/3197517.3201311
Jesús Pérez, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, José A. Canabal, Robert Sumner, and Miguel A. Otaduy. 2015. Design and fabrication of flexible rod meshes. *ACM Trans. Graph.* 34, 4 (2015). doi:10.1145/2766998
N.S. Pollard, J.K. Hodgins, M.J. Riley, and C.G. Atkeson. 2002. Adapting human motion for the control of a humanoid robot. In *IEEE Int. Conf. Robot. Autom.* doi:10.1109/ROBOT.2002.1014737
Zoran Popović and Andrew Witkin. 1999. Physically based motion transformation. In *ACM SIGGRAPH*. doi:10.1145/311535.311536
Daniele Reda, Jungdam Won, Yuting Ye, Michiel van de Panne, and Alexander Winkler. 2023. Physics-based Motion Retargeting from Sparse Inputs. *Proc. ACM Comput. Graph. Interact. Tech.* 6, 3 (2023). doi:10.1145/3606928
Quentin Rouxel, Kai Yuan, Ruoshi Wen, and Zhibin Li. 2022. Multicontact Motion Retargeting Using Whole-Body Optimization of Full Kinematics and Sequential Force Equilibrium. *IEEE/ASME Trans. Mechatron.* 27, 5 (2022). doi:10.1109/TMECH.2022.3152844
Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. 2022. Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning. In *Conf. Robot Learn.*
Hoseok Ryu, Minseok Kim, Seungwhan Lee, Moon Seok Park, Kyoungmin Lee, and Jehee Lee. 2021. Functionality-Driven Musculature Retargeting. *Comput. Graph. Forum.* 40, 1 (2021). doi:10.1111/cgf.14191
John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. doi:10.48550/arXiv.1707.06347
Christian Schumacher, Espen Knoop, and Moritz Bächer. 2021. A Versatile Inverse Kinematics Formulation for Retargeting Motions Onto Robots With Kinematic Loops. *IEEE Robot. Autom. Lett.* 6, 2 (2021). doi:10.1109/LRA.2021.3056030
Agon Serifi, Ruben Grandia, Espen Knoop, Markus Gross, and Moritz Bächer. 2024. VMP: Versatile Motion Priors for Robustly Tracking Motion on Physical Characters. In *Symp. Comput. Anim.* doi:10.1111/cgf.15175
Soshi Shimada, Vladislav Golyanik, Weipeng Xu, and Christian Theobalt. 2020. PhysCap: physically plausible monocular 3D motion capture in real time. *ACM Trans. Graph.* 39, 6 (2020). doi:10.1145/3414685.3417877
Joan Sola, Jeremie Deray, and Dinesh Atchuthan. 2018. A micro lie theory for state estimation in robotics. *arXiv preprint arXiv:1812.01537* (2018).
Seyoon Tak and Hyeong-Seok Ko. 2005. A physically-based motion retargeting filter. *ACM Trans. Graph.* 24, 1 (2005). doi:10.1145/1037957.1037963
Javier Tapia, Espen Knoop, Mojmir Mutný, Miguel A. Otaduy, and Moritz Bächer. 2020. MakeSense: Automated Sensor Design for Proprioceptive Soft Robots. *Soft Robotics* 7, 3 (2020). doi:10.1089/soro.2018.0162
Chen Tessler, Yifeng Jiang, Xue Bin Peng, Erwin Coumans, Yi Shi, Haotian Zhang, Davis Rempe, Gal Chechik, and Sanja Fidler. 2025. ProtoMotions3: An Open-source Framework for Humanoid Simulation and Control. https://github.com/NVLabs/ProtoMotions.
Tarik Tosun, Ross Mead, and Robert Stengel. 2015. A General Method for Kinematic Retargeting: Adapting Poses Between Humans and Robots. In *ASME Int. Mech. Eng. Congr. Expo.* doi:10.1115/IMECE2014-37700
Ruben Villegas, Duygu Ceylan, Aaron Hertzmann, Jimei Yang, and Jun Saito. 2021. Contact-Aware Retargeting of Skinned Motion. In *Int. Conf. Comput. Vis.* doi:10.1109/ICCV48922.2021.00958
Ruben Villegas, Jimei Yang, Duygu Ceylan, and Honglak Lee. 2018. Neural Kinematic Networks for Unsupervised Motion Retargetting. In *IEEE Conf. Comput. Vis. Pattern Recog.* doi:10.1109/CVPR.2018.00901
Tingwu Wang, Yunrong Guo, Maria Shugrina, and Sanja Fidler. 2020. UniCon: Universal Neural Controller For Physics-based Character Motion.
Yufu Wang, Ziyun Wang, Lingjie Liu, and Kostas Daniilidis. 2025. TRAM: Global Trajectory and Motion of 3D Humans from in-the-Wild Videos. In *Eur. Conf. Comput. Vis.*, Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (Eds.). Springer Nature Switzerland.
Jungdam Won, Deepak Gopinath, and Jessica Hodgins. 2020. A scalable approach to control diverse behaviors for physically simulated characters. *ACM Trans. Graph.* 39, 4 (2020). doi:10.1145/3386569.3392381
Jungdam Won and Jehee Lee. 2019. Learning body shape variation in physics-based characters. *ACM Trans. Graph.* 38, 6 (2019). doi:10.1145/3355089.3356499
Weiji Xie, Jinrui Han, Jiakun Zheng, Huanyu Li, Xinzhe Liu, Jiyuan Shi, Weinan Zhang, Chenjia Bai, and Xuelong Li. 2025. KungfuBot: Physics-Based Humanoid Whole-Body Control for Learning Highly-Dynamic Skills. In *Advances in Neural Information Processing Systems.*
Yashuai Yan, Esteve Valls Mascaro, and Dongheui Lee. 2023. ImitationNet: Unsupervised Human-to-Robot Motion Retargeting via Shared Latent Space. In *Int. Conf. Humanoid Robots.* doi:10.1109/Humanoids57100.2023.10375150
Lujie Yang, Xiaoyu Huang, Zhen Wu, Angjoo Kanazawa, Pieter Abbeel, Carmelo Sferrazza, C. Karen Liu, Rocky Duan, and Guanya Shi. 2025a. OmniRetarget: Interaction-Preserving Data Generation for Humanoid Whole-Body Loco-Manipulation and Scene Interaction. doi:10.48550/arXiv.2509.26633
Lujie Yang, H. j Terry Suh, Tong Zhao, Bernhard Paus Graesdal, Tarik Kelestemur, Jiuguang Wang, Tao Pang, and Russ Tedrake. 2025b. Physics-Driven Data Generation for Contact-Rich Manipulation via Trajectory Optimization. In *Robotics: Science and Systems XXI.*
KangKang Yin, Kevin Loken, and Michiel van de Panne. 2007. SIMBICON: simple biped locomotion control. *ACM Trans. Graph.* 26, 3 (2007). doi:10.1145/1276377.1276509
Ye Yuan and Kris Kitani. 2020. Residual Force Control for Agile Human Behavior Imitation and Extended Motion Synthesis. In *Advances in Neural Information Processing Systems.*
Haotian Zhang, Ye Yuan, Viktor Makoviychuk, Yunrong Guo, Sanja Fidler, Xue Bin Peng, and Kayvon Fatahalian. 2023c. Learning Physically Simulated Tennis Skills from Broadcast Videos. *ACM Trans. Graph.* 42, 4 (2023). doi:10.1145/3592408
Jiaxu Zhang, Junwu Weng, Di Kang, Fang Zhao, Shaoli Huang, Xuefei Zhe, Linchao Bao, Ying Shan, Jue Wang, and Zhigang Tu. 2023b. Skinned Motion Retargeting with Residual Perception of Motion Semantics & Geometry. In *IEEE Conf. Comput. Vis. Pattern Recog.* doi:10.1109/CVPR52729.2023.01332
Jia-Qi Zhang, Miao Wang, Fu-Cheng Zhang, and Fang-Lue Zhang. 2025. Skinned Motion Retargeting With Preservation of Body Part Relationships. *IEEE Trans. Vis. Comput. Graph.* 31, 9 (2025). doi:10.1109/TVCG.2024.3423426
Yunbo Zhang, Deepak Gopinath, Yuting Ye, Jessica Hodgins, Greg Turk, and Jungdam Won. 2023a. Simulation and Retargeting of Complex Multi-Character Interactions. In *ACM SIGGRAPH.* doi:10.1145/3588432.3591491
Yihua Zhang, Prashant Khanduri, Ioannis Tsaknakis, Yuguang Yao, Mingyi Hong, and Sijia Liu. 2024. An Introduction to Bilevel Optimization: Foundations and applications in signal processing and machine learning. *IEEE Signal Process. Mag.* 41, 1 (2024). doi:10.1109/MSP.2024.3358284
Allan Zhao, Jie Xu, Mina Konaković-Luković, Josephine Hughes, Andrew Spielberg, Daniela Rus, and Wojciech Matusik. 2020. RoboGrammar: graph grammar for terrain-optimized robot design. *ACM Trans. Graph.* 39, 6 (2020). doi:10.1145/3414685.3417831
Wenshuai Zhao, Yi Zhao, Joni Pajarinen, and Michael Muehlebach. 2024. Bi-Level Motion Imitation for Humanoid Robots. In *Conf. Robot Learn.*
Wentao Zhu, Zhuoqian Yang, Ziang Di, Wayne Wu, Yizhou Wang, and Chen Change Loy. 2022. MoCaNet: Motion Retargeting In-the-Wild via Canonicalization Networks. *Proc. AAAI Conf. Artif. Intell.* 36, 3 (2022). doi:10.1609/aaai.v36i3.20274
Victor Brian Zordan and Jessica K. Hodgins. 2002. Motion capture-driven simulations that hit and react. In *Symp. Comput. Anim.* doi:10.1145/545261.545276

A Downstream RL Policy Details

Following prior work, we evaluate the retargeting methods on a downstream tracking task by training RL policies on the retargeted motions, with tracking performance serving as a proxy for motion quality [Liao et al. 2025; Yang et al. 2025a]. We train the RL policies following the DeepMimic framework [Peng et al. 2018], where the policy is conditioned on a motion reference and optimized using explicit tracking rewards. The reward terms used for training are detailed in Tab. 6.

We measure the success rate based on the training termination criteria, as proposed in [Yang et al. 2025a]. A trial is considered a failure if the robot's root deviates by more than 1 m from the target root position or if the geodesic distance between the current and target root orientation exceeds 45°.

Table 6. **Reward Terms for Downstream RL Training.** The root position is given by $\mathbf{x}^{\text{rt}}$ and the root height is z^{rt}. The root orientation matrix is $\mathbf{R}^{\text{rt}}$, the root's linear and angular velocities are $\mathbf{v}^{\text{rt}}$ and $\boldsymbol{\omega}^{\text{rt}}$, respectively. We denote rigid body positions as $\mathbf{x}^b$ and rigid body orientations as $\mathbf{R}^b$. The terms $\boldsymbol{\tau}_t^{\text{jts}}$ and $\ddot{\mathbf{q}}_t$ are joint torques and accelerations. The policy actions are given by $\mathbf{a}_t^{\text{jts}}$. Note that in this case $\mathbf{g}_t$ the retargeted trajectory and $\mathbf{s}_t$ denotes the simulation state of the downstream RL policy.

Name	Reward Term	Weight G1	Weight Lima
Motion Tracking			
Root position xy	$-\lVert \mathbf{x}^{\text{rt}}_{\mathbf{g}_t} - \mathbf{x}^{\text{rt}}_{\mathbf{s}_t} \rVert_2^2$	5.0	5.0
Root height	$-(z^{\text{rt}}_{\mathbf{g}_t} - z^{\text{rt}}_{\mathbf{s}_t})^2$	5.0	5.0
Root orientation	$-\lVert \text{Log}((\mathbf{R}^{\text{rt}}_{\mathbf{s}_t})^T \mathbf{R}^{\text{rt}}_{\mathbf{g}_t}) \rVert_2^2$	3.0	3.0
Root lin vel.	$-\lVert \mathbf{v}^{\text{rt}}_{\mathbf{g}_t} - \mathbf{v}^{\text{rt}}_{\mathbf{s}_t} \rVert_2^2$	0.5	0.5
Root ang. vel.	$-\lVert \boldsymbol{\omega}^{\text{rt}}_{\mathbf{g}_t} - \boldsymbol{\omega}^{\text{rt}}_{\mathbf{s}_t} \rVert_2^2$	0.5	0.5
Rbs position	$-\lVert \mathbf{x}^{b}_{\mathbf{g}_t} - \mathbf{x}^{b}_{\mathbf{s}_t} \rVert_2^2$	5.0	5.0
Rbs orientation	$-\lVert \text{Log}((\mathbf{R}^{b}_{\mathbf{s}_t})^T \mathbf{R}^{b}_{\mathbf{g}_t}) \rVert_2^2$	2.5	2.5
Survival	1.0	10.0	1.0
Regularization			
Joint torques	$-\lVert \boldsymbol{\tau}_t^{\text{jts}} \rVert_2^2$	$1.0 \cdot 10^{-4}$	$1.0 \cdot 10^{-3}$
Joint acc.	$-\lVert \ddot{\mathbf{q}}_t \rVert_2^2$	$2.5 \cdot 10^{-8}$	$2.5 \cdot 10^{-6}$
Joint action rate	$-\lVert \mathbf{a}_t^{\text{jts}} - \mathbf{a}_{t-1}^{\text{jts}} \rVert_2^2$	0.15	3.0
Joint action acc.	$-\lVert \mathbf{a}_t^{\text{jts}} - 2\mathbf{a}_{t-1}^{\text{jts}} + \mathbf{a}_{t-2}^{\text{jts}} \rVert_2^2$	$1.0 \cdot 10^{-2}$	1.0

Table 7. **Performance Variability (Lima).** Best and worst case (min/max of the per-motion mean) for the kinematic metrics reported in the main paper.

Metric	ReActor	OmniRetarget	GMR
Ground Pen. [cm]	**0.0** / **0.0**	0.0 / 0.0	0.0 / 9.3
Self Pen. [cm]	**0.0** / **0.0**	0.0 / 17.2	0.0 / 15.3
Foot Slide [cm/s]	**0.0** / **52.3**	0.0 / 98.6	0.0 / 95.5
Foot Float [cm]	**0.0** / **9.4**	0.0 / 21.4	0.0 / 32.4

B Performance Variability

Tab. 7 reports best- and worst-case results (min/max of the per-motion mean) for Lima, complementing the mean and standard deviation reported in the main paper.