# Direct Dynamic Retargeting for Humanoid Imitation Learning from Videos

Constant Roux[⋆,1], Ludovic De Matteïs[⋆,1], Armand Jordana[1], Valentin Guillet[2], Nicolas Mansard[1,3], Olivier Stasse[1,3], Philippe Souères[1]

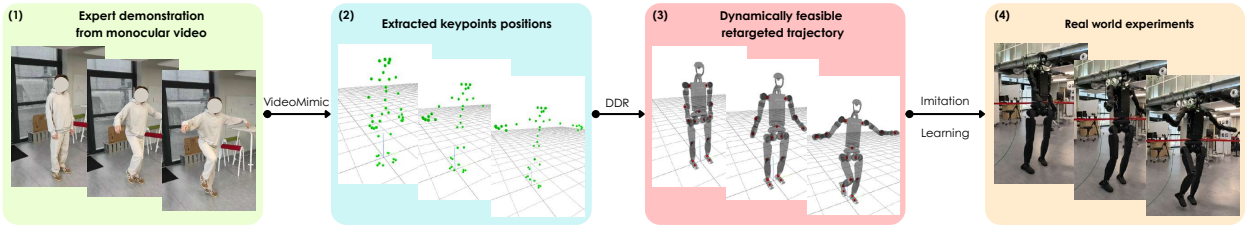



Fig. 1: Our proposed framework uses expert demonstration from monocular video, extracts human keypoints using part of the *VideoMimic* [1] framework and generates retargeted trajectories using our **Dynamically Feasible Retargeting (DDR)** method. Imitation policies obtained from these trajectories can then be ported on a real full-size humanoid robot.

***Abstract*— Imitation Learning from monocular video demonstrations provides a scalable approach for teaching complex skills to humanoid robots. However, translating human motion to humanoids requires overcoming significant morphological mismatches. Standard approaches rely on Geometric Retargeting or Indirect Dynamic Retargeting pipelines. We identify that these intermediate kinematic projections introduce a geometric bias, restricting the search space and yielding suboptimal dynamic behaviors. In this paper, we propose Direct Dynamic Retargeting (DDR), a novel single-stage framework that generates high-fidelity, dynamically feasible trajectories directly from expert videos. By formulating the problem in the task space and leveraging a sampling-based Model Predictive Control solver within a physics simulator, DDR natively optimizes over complex contact sequences while mitigating input drift. Our experiments demonstrate that bypassing the geometric bias allows DDR to outperform state-of-the-art baselines in demonstration tracking accuracy. Furthermore, we establish that providing such physically viable references to RL agents accelerates training convergence and enhances the final execution of agile and balancing behaviors. Source code will be made publicly available.**

## I. Introduction

Humanoid robotics has witnessed major breakthroughs in recent years, demonstrating unprecedented capabilities in locomotion and loco-manipulation [2]. This progress has been largely catalyzed by the rise of Reinforcement Learning (RL) [3]. However, for complex tasks, manual reward engineering becomes a major bottleneck, proving to be both tedious and time-consuming. To bypass this difficulty, Imitation Learning (IL) has emerged as a highly effective alternative. Techniques such as Behavioral Cloning, DeepMimic [4], and Adversarial Motion Priors [5] enable robots to learn complex skills directly from expert demonstrations. Nevertheless, the efficiency of the resulting policies is fundamentally dependent on the quality of the provided references [6], [2]. Current approaches rely either on precise Motion Capture systems (MoCap) or on a posteriori selection of robot trajectories to avoid detrimental demonstrations [7], [8].

Recent works extracting human poses from monocular videos enable the use of vast online datasets [1]. However, the current extraction tools often yield noisy and physically inconsistent trajectories. Furthermore, whether sourced from videos or MoCap, human demonstrations present a morphological mismatch with humanoid robots, leading to both kinematic (size, limb proportions) and dynamic (mass distribution, inertias, actuation limits) discrepancies [9]. Consequently, retargeting human motion to the robot is necessary to provide IL with relevant references.

The current prevailing approach, Geometric Retargeting (GR), aims to find the closest robot posture using Inverse Kinematics (IK) [10], [9], [2], inherently ignoring dynamic constraints. To address this limitation, Indirect Dynamic Retargeting (IDR) methods tracks the GR reference within a physics simulator [11], [12], [13]. We hypothesize that this intermediate IK step restricts the search space and biases the final dynamically feasible solution.

To overcome this, we propose Direct Dynamic Retargeting (DDR), a Model Predictive Control (MPC) framework that directly computes dynamically feasible trajectories from noisy data extracted from videos (See Fig. 1). To solve the optimization, we use the Cross-Entropy Method (CEM) [14]. Unlike gradient-based solvers (e.g., *Crocoddyl* [15]) that require strictly predefined contact sequences, CEM's sampling-based approach intrinsically handles the ambiguities related to the identification of contacts from noisy video data. This allows us to generate high-quality IL retargeted references, completely bypassing the intermediate geometric bias.

We show that DDR outperforms previous methods in terms

⋆ Equal contribution
[1] LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France
[2] IRT Saint-Exupéry, Toulouse, France
[3] Artificial and Natural Intelligence Toulouse Institute (ANITI), Toulouse, France

of physical consistency, demonstration tracking accuracy, and learning efficiency. The main contributions are as follows:

- Introduction of **Direct Dynamic Retargeting**, a novel CEM-based MPC framework that generates dynamically feasible humanoid trajectories directly from demonstrations.
- Quantitative evaluation showing that **eliminating the intermediate geometric bias** significantly improves retargeting accuracy, outperforming GR and IDR baselines on diverse agile motions.
- Experimental evidence establishing that utilizing these **physically consistent references** enhances the learning efficiency and performance of downstream Imitation Learning.
- **Successful real-world deployment** on the *Unitree H1-2* humanoid, achieving zero-shot sim-to-real transfer and validating the viability of the framework.

The remainder of this paper is organized as follows. Section II reviews the relevant literature and positions our approach within the current state-of-the-art. Section III details the DDR formulation and describes how our CEM-based MPC framework effectively tracks expert demonstrations. Section IV presents our experimental setup and evaluates the performance of DDR against existing baselines, highlighting improvements in both retargeting quality and downstream learning efficiency. Finally, Section V summarizes our findings and outlines directions for future work.

## II. Related Work

This section provides a review of the literature on motion retargeting for humanoid robots, highlighting the transition from geometric to dynamic approaches. Then, an overview of imitation learning methods is given to contextualize the need for dynamically retargeted reference motions.

### A. Geometric Retargeting

Transferring human motion to humanoid robots requires solving the problem of the morphological mismatch between the two embodiments [16]. Geometric Retargeting typically relies on Inverse Kinematics to map human keypoints or joint angles to the robot's configuration space [17], [18]. While these methods successfully minimize spatial tracking errors and enforce strict joint limits, they inherently ignore the system's dynamics. Consequently, for highly dynamic systems like humanoid robots, purely geometric references frequently result in physically infeasible trajectories that violate balance constraints, torque limits and contact dynamics [19].

### B. Indirect Dynamic Retargeting

To bridge the gap between kinematic similarity and physical validity, IDR approaches introduce a secondary physics-based refinement phase. Methods in this category formulate the problem as a two-stage pipeline: an initial GR step generates a kinematic reference, which is subsequently tracked by a controller or a RL agent within a physics simulator to yield a dynamically feasible trajectory [11], [13], [20], [12]. While these methods ensure dynamic feasibility, they inherently rely on the assumption that the intermediate kinematic reference resides near the optimal dynamic motion. A fundamental limitation of these two-stage pipelines is that the initial kinematic projection intrinsically restricts the optimization space, potentially biasing the final motion away from the true dynamic optimum. To the best of our knowledge, no previous work has successfully bypassed this intermediate step to compute direct, dynamically feasible retargeting for humanoid imitation.

### C. Imitation Learning from Video Demonstrations

Imitation Learning circumvents the need for complex reward shaping by learning humanoid skills directly from expert data [8], [21]. Frameworks ranging from direct reference state tracking [4] to adversarial methods [22], [5] have achieved high-fidelity motion reproduction by mimicking expert data. To scale up these approaches, recent works increasingly focus on extracting human demonstrations directly from large-scale monocular video datasets [1]. Standard imitation pipelines often attempt to train RL agents directly on purely geometric references. However, the learning efficiency and performance of these policies remain sensitive to the quality and dynamical feasiblity of the provided references [6], [2]. Kinematic noise or dynamically infeasible reference trajectory lead to suboptimal behaviors, undesirable artifacts, or learning failures. Since generating these optimized references is an offline process, we distill them into closed-loop RL policies capable of reference tracking under varied initializations and external disturbances, ensuring robust real-time hardware deployment and seamless sim-to-real transfer.

## III. Method

### A. Approach Overview

Consider a humanoid robot with $n_q$ actuated joints - plus a floating base. Let $q \in SE(3) \times \mathbb{R}^{n_q}$ denote the robot configuration. Given a fixed initial configuration $q_0$ with zero initial velocity, a trajectory over $T$ timesteps is defined as $Q = (q_1, \dots, q_T) \in \mathbb{Q}$ where $\mathbb{Q} = (SE(3) \times \mathbb{R}^{n_q})^T$, and the corresponding control sequence is $U = (u_0, \dots, u_{T-1}) \in \mathbb{U} = \mathbb{R}^{n_q \times T}$. To characterize physical feasibility, we define a rollout function $S_{q_0} : \mathbb{U} \to \mathbb{Q}$. Starting from the initial state $q_0$, this function iteratively integrates the robot dynamics under the controls $U$, accounting for contacts with the environment. This rollout function is typically the one of a physics simulator. The Feasibility Set $\mathbb{F}_{q_0}$ in the trajectory space is then defined as the subset of all physically attainable trajectories:

$$\mathbb{F}_{q_0} = \{Q \in \mathbb{Q} \mid \exists U \in \mathbb{U}, S_{q_0}(U) = Q\} \tag{1}$$

Let $x$ be a trajectory in the task space of $m$ 3D keypoints over $T$ timesteps. $x$ is an element of the keypoint space $\mathbb{R}^{3m \times T}$. Consider an extended forward kinematics function $FK : \mathbb{Q} \to \mathbb{R}^{3m \times T}$ that maps a robot trajectory $Q$ to its corresponding keypoints trajectory $x$. We denote by $\widetilde{\mathbb{Q}}$ and $\widetilde{\mathbb{F}}_{q_0}$ the images of $\mathbb{Q}$ and $\mathbb{F}_{q_0}$ under $FK$. These sets represent respectively keypoint trajectories that are kinematically

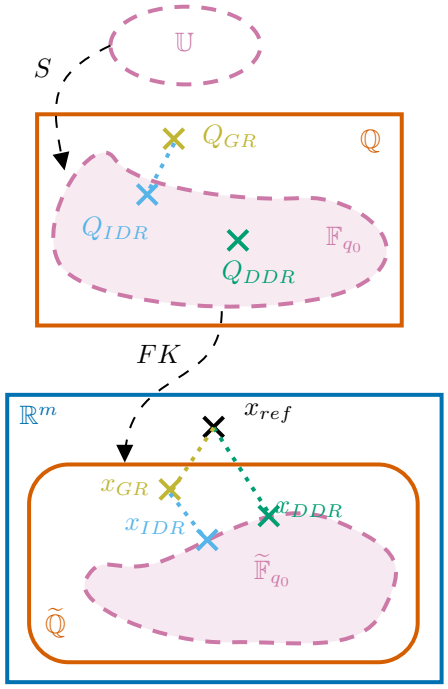



Fig. 2: Schematic comparison of Geometric Retargeting (GR), Indirect Dynamic Retargeting (IDR) and Direct Dynamic Retargeting (DDR). GR generates a pair $(Q_{GR}; x_{GR})$ in the Trajectory Space $\mathbb{Q}$ and its image $\widetilde{\mathbb{Q}}$ while IDR and DDR output pairs $(Q_{IDR}; x_{IDR})$ and $(Q_{DDR}; x_{DDR})$ in the Feasible Space $\mathbb{F}_{q_0}$ - obtained by rollouts from the Control Space $\mathbb{U}$ - and its image $\widetilde{\mathbb{F}}_{q_0}$. IDR fails to minimize the distance to the reference $x_{ref}$ because of the bias induced by the GR reference.

reachable in the free space ($\widetilde{\mathbb{Q}}$) and dynamically feasible within the environment under contact constraints ($\widetilde{\mathbb{F}}_{q_0}$).

We denote by $x_{ref}$ the expert demonstration trajectory. Due to anthropometric discrepancies between humans and robots, $x_{ref}$ rarely lie within $\widetilde{\mathbb{Q}}$. Fig. 2 provides a schematic view of the sets and mappings relations and visualizes the resulting pairs of robot and keypoint trajectories for different retargeting approaches.

The retargeting problem aims at finding a trajectory $Q \in \mathbb{Q}$ such that the resulting keypoint trajectory $x = FK(Q)$ minimizes the distance $d$ to a reference demonstration in $\mathbb{R}^{3m\times T}$. We distinguish three major approaches.

The **Geometric Retargeting** approach searches for a pair $(Q_{GR}, x_{GR}) \in \mathbb{Q} \times \widetilde{\mathbb{Q}}$ that minimizes the distance to the expert keypoints $d(x_{GR}, x_{ref})$ while minimizing the distance to the feasible set $\mathbb{F}_{q_0}$.

The **Indirect Dynamic Retargeting** starts from the GR result and refines the trajectory to find a feasible pair $(Q_{IDR}, x_{IDR}) \in \mathbb{F}_{q_0} \times \widetilde{\mathbb{F}}_{q_0}$ minimizing the distance to the GR keypoints $d(x_{IDR}, x_{GR})$ in a physically consistent environment, such as a simulator.

The alternative method proposed in this work, called **Direct Dynamic Retargeting**, directly searches for a pair $(Q_{DDR}, x_{DDR}) \in \mathbb{F}_{q_0} \times \widetilde{\mathbb{F}}_{q_0}$ that minimizes the keypoints distance $d(x_{DDR}, x_{ref})$ without relying on a geometric intermediate.

According to the definition above, the DDR method comes to find a minimum of the reference discrepancy over the dynamically feasible set $\mathbb{F}_{q_0}$:

$$\min_{U\in\mathbb{U}} d(FK(S_{q_0}(U)), x_{ref}) = \min_{Q\in\mathbb{F}_{q_0}} d(FK(Q), x_{\text{ref}}) \quad (2)$$

$$= d(x_{DDR}, x_{ref}) \quad (3)$$

As $Q_{IDR} \in \mathbb{F}_{q_0}$, it follows that:

$$d(x_{DDR}, x_{ref}) \leq d(x_{IDR}, x_{ref}), \quad (4)$$

where $x_{DDR} = FK(Q_{DDR}) \in \widetilde{\mathbb{F}}_{q_0}$ and $x_{IDR} = FK(Q_{IDR}) \in \widetilde{\mathbb{F}}_{q_0}$. This confirms that the DDR approach, by directly optimizing the reference in the task space, establishes a lower bound on the tracking error. Consequently, the IDR method is inherently bottlenecked by a geometric bias. We show in this work that existing state-of-the-art approaches suffer from this bias, converging to suboptimal solutions as the reference $Q_{GR}$ is often significantly far from the feasible manifold $\mathbb{F}_{q_0}$.

### B. SMPL keypoints extraction

The feature extraction pipeline introduced in VideoMimic [1] is used to reconstruct human motion from monocular video inputs via the SMPL model [23] (See Fig. 1.2). To provide sufficient information for high-fidelity imitation, we select specific keypoint trajectories from the resulting SMPL motion. Our reference signal, $x_{ref}$, is composed of trajectories of the torso, shoulders, hands, and feet.

### C. GR and IDR references

The GR reference for a given expert demonstration is generated via the Pyroki integration in VideoMimic, which performs kinematic optimization from the SMPL keypoints [18].

To obtain the IDR reference, we use the same solver as the one used for our proposed DDR approach (detailed in the following section). However, the GR keypoint trajectory $x_{GR}$ is used in place of the human reference $x_{ref}$.

### D. Dynamically Feasible Trajectory

IDR and DDR methods introduced in Section III-A rely on the minimization of the distance within the task space $\mathbb{R}^{3m\times T}$. In order to account for the the human-robot embodiment gap and avoid unnatural motion, we consider the augmented distance introduced in [2], composed of two terms, spatial tracking and relative shape-matching metric:

$$E_p(FK(Q), x_{ref}) = \sum_{i=1}^{T} \|fk(q_i) - \tilde{x}_i\|^2 \quad (5)$$

$$E_l(FK(Q), x_{ref}) = \sum_{i=1}^{T} \frac{1}{m} \|L(fk(q_i) - \tilde{x}_i)\|^2 \quad (6)$$

where $fk(\cdot)$ is the classical forward kinematics, $\tilde{x}$ represents either $x_{ref}$ in DDR or $x_{GR}$ in IDR, $E_p$ is the Euclidean distance, and $E_l$ is a relative metric that uses the Laplacian

matrix $L$ to penalize structural deformations between neighboring keypoints. $E_l$ explicitly preserves the local structural shape of the demonstration, with invariance to global translation and rotation.

We obtain our retargeted trajectory (see Fig 1.3) by solving this optimization problem using the Cross-Entropy Method (CEM), a derivative-free sampling approach, implemented within the framework introduced in [24]. Although our current pipeline relies on CEM, the underlying framework is modular and could seamlessly accommodate other stochastic optimizers given appropriate tuning. We implement a Model Predictive Control (MPC) scheme with a receding horizon, allowing to solve for any time horizon with minimal additional computational burden. Although MPC can sometimes exhibit myopic behavior, we found it viable for our tasks; however, the framework is compatible with alternative solutions if more foresight is required [11].

### *E. Imitation via Reinforcement Learning*

Because retargeted trajectories are computed offline, we distill them into closed-loop RL policies that track the reference, robustly recovering from initialization variations and external disturbances to ease sim-to-real transfer (Fig. 1.4). Specifically, we train one policy per motion to imitate its retargeted reference on the *Unitree H1-2* humanoid.

*1) Problem Formulation:* The imitation task for a given motion is formulated as a Constrained Markov Decision Process (CMDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \gamma, \mathcal{T}, \{\mathcal{C}^i\}_{i\in I} \rangle$ where $\mathcal{S}$ represents the state space, $\mathcal{A}$ is the action space, $\gamma$ the discount factor, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ the reward function which yields the scalar reward $r(s_t, a_t)$ for taking action $a_t$ in state $s_t$ at timestep $t$, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^+$ the probabilistic transition dynamics, and $\{\mathcal{C}^i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}\}_{i\in I}$ the constraints. The objective is to compute a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes the discounted sum of future rewards:

$$\max_{\pi} \mathbb{E}_{\tau\sim\pi,\mathcal{T}} \left[ \sum_{t=0}^{\infty} \gamma^t r\left(s_t, a_t\right) \right], \tag{7}$$

while ensuring that constraints are satisfied:

$$\mathbb{P}_{(s,a)\sim\rho_\gamma^\pi,\mathcal{T}} \left[ c^i(s,a) > 0 \right] \leq \epsilon_i, \quad \forall i \in I. \tag{8}$$

*2) Framework:* The CMDP is solved via an actor-critic architecture. The policy is trained using the *Constraints-as-Terminations* framework [25], [26], built upon the *skrl* implementation of Proximal Policy Optimization (PPO) [27], [28] in *IsaacLab* [29]. At timestep $t$, the observation vector $o_t$ contains joint positions $q_t$, joint velocities $\dot{q}_t$, previous actions $a_{t-1}$, base linear velocity $v_{b,t}$, base angular velocity $\omega_t$, base height $h_t$, projected gravity $g_t$, and a phase variable $\phi_t \in [0, 1]$ that parameterizes the reference motion. The action vector $a_t \in \mathbf{R}^{21}$ specifies the desired joint positions. These outputs are scaled, added to nominal joint offsets, and tracked by a PD-controller. To encourage imitation of the retargeted trajectory, the reward function formulation follows the deepMimic approach [4] and is summarized in the table I. Hardware safety constraints limit maximum joint positions, velocities, and torques. Training utilizes the reference state initialization and early termination techniques [4].

TABLE I: Reward terms for RL tracking policy.

| Term | Expression |
|---|---|
| | Tracking terms |
| Joint Position | $0.5 \exp\left(-\|q - q_{\text{ref}}\|_2^2 / 2.0^2\right)$ |
| Joint Velocity | $0.1 \exp\left(-\|\dot{q} - \dot{q}_{\text{ref}}\|_2^2 / 10.0^2\right)$ |
| Root Pose | $0.15 \exp\Big( - \big(\|p_b - p_{b,\text{ref}}\|_2^2 + 0.1\Delta\theta_b^2\big) / 0.45^2 \Big)$ |
| Root Velocity | $0.1 \exp\Big( - \big(\|v_b - v_{b,\text{ref}}\|_2^2 + 0.1\|\omega_b - \omega_{b,\text{ref}}\|_2^2\big) / 1.0^2 \Big)$ |
| End-Effector Position | $0.15 \exp\left(-\|p_{\text{ee}} - p_{\text{ee,ref}}\|_2^2 / 0.32^2\right)$ |
| | Penalty terms |
| Joint Acceleration | $-10^{-7}\|\ddot{q}\|_2^2$ |
| Joint Torques | $-10^{-7}\|\tau\|_2^2$ |
| Action Rate | $-0.1\|a_t - a_{t-1}\|_2^2$ |
| Joint Velocity | $-0.005\|\dot{q}\|_2^2$ |
| Foot Slip | $-0.2 \sum \|v_{\text{foot},xy}\|_2 \cdot \mathbb{I}_{F_c > F_{\text{th}}}$ |

(a) Squat (b) Kung fu (c) Long one-foot balance

(d) Pistol Squat (e) Balancing Stick

Fig. 3: Experimental motion dataset for the IL benchmark. This set of agile motions include (a) a stable squat, (b) a static kung fu pose, (c) an extended one-foot balance, (d) a dynamic pistol squat, and (e) a high-amplitude balancing stick pose.

## IV. Evaluation

### *A. Motion collection*

To assess tracking performance and stability, we select motions with varying levels of difficulty (See Fig. 1.1):

- **Stability Baseline**: We start from a stable **squat** to establish basic tracking benchmarks when stability is easily attained.
- **Balancing Tasks**: We then move to a **kung fu** pose and a **long one-foot balance** to evaluate balancing capabilities of the dynamic retargeting and its ability to handle extended horizons.

TABLE II: Percentage of physically infeasible segments in retargeted trajectories.

| Movement | GR | IDR | DDR (ours) |
|---|---|---|---|
| Squat | 21.74% | **0.00%** | **0.00%** |
| Kung fu | 15.81% | **0.00%** | **0.00%** |
| One-foot Balance | 28.38% | **2.86%** | 3.36% |
| Pistol Squat | 18.16% | 1.06% | **0.00%** |
| Balancing Stick | 3.08% | 1.02% | **0.20%** |

- **Dynamic Motions**: Finally, we use a **pistol squat** and a **balancing stick** pose to test the methods under more unstable and dynamic conditions.

For each motion, we extract SMPL trajectories from three monocular videos from three different human subjects. We generate retargeted motions using GR, IDR, and our proposed DDR. To account for variance in stochastic sampling, results for IDR and DDR are averaged over five different seeds.

### *B. Retargeted trajectories*

We evaluate the performances of the GR, IDR and DDR method considering five main criterion: the feasibility of the retargeted trajectories, the accuracy of the contacts sequence, the feet slipage during the motion, the success rate of the stochastic methods and the reference tracking.

To compute some of these metrics, we estimate the contact sequence of each retargeted trajectory by considering the distance between a foot and the ground. If at a timestep $t$ this distance is less than 2cm, then the foot is considered to be in contact. These estimated sequences are compared against ground truth data, which was obtained through manual labeling of the source videos.

*1) Feasibility:* We demonstrate that GR fails to guarantee physical feasibility, often producing kinematically valid but dynamically impossible trajectories. To quantify this, we compute the joints velocities and acceleration required to follow a given configuration trajectory. Then, at each time step, we look for the existence of controls and contact forces that achieve the expected accelerations, based on the estimated contact sequence. If no such values exist, this timestep is physically unfeasible. To ensure an unbiased evaluation independent of the simulators used for IDR and DDR, we utilize *Pinocchio* [30] for rigid body dynamics and check for existence of physically feasible controls with *ProxQP* [31]. Table II reports the proportion of the trajectories that are evaluated as unfeasible. We observe that IDR and DDR significantly outperform GR in terms of feasibility. These results indicate that despite incorporating feasibility costs, GR fails to adequately account for contact dynamics, whereas our proposed DDR maintains near-perfect feasibility across most tasks. We can still observe some unfeasible portions in IDR and DDR, suggesting small discrepancies in physical simulation between *Mujoco* [32], used for computing the references, and *Pinocchio* [30].

*2) Contact sequence:* The feasibility gap noted previously for the GR is closely related to the inaccuracy of the contact sequence of the retargeted trajectory, as the reference trajectories cannot be properly tracked without proper contact forces.

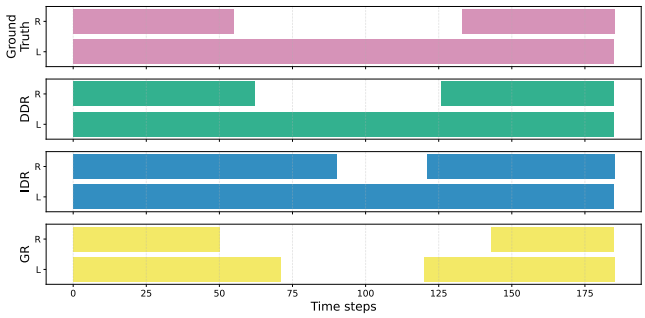



Fig. 4: Qualitative analysis of contact sequence accuracy for the kung fu motion. This figure compares the estimated contact phases of GR, IDR, and DDR against the manually labeled ground truth.

TABLE III: Comparison of contact sequence error rates. table presents the percentage of mismatch between estimated contacts and ground truth.

| Movement | GR | IDR | DDR (ours) |
|---|---|---|---|
| Squat | 32.12% | **0.00%** | **0.00%** |
| Kung fu | 10.53% | 13.03% | **4.23%** |
| One-foot Balance | 21.37% | 24.72% | **13.71%** |
| Pistol Squat | 15.00% | 9.13% | **5.35%** |
| Balancing Stick | 8.19% | 20.73% | **7.86%** |

As illustrated in Fig. 4 for the kung fu motion, GR frequently breaks contact — with feet drifting upward — whereas IDR and DDR maintain significantly more stable grounding. Table III summarizes the contact error rates across all motions. DDR consistently yields the closest match to the reference videos. Notably, IDR performs worse than DDR despite sharing identical cost functions; this suggests that GR-based initialization introduces a kinematic bias that IDR cannot fully overcome. Finally, error rates increase during tasks involving extended horizons (Long Balance) or significant Center of Mass (CoM) displacement (Balancing Stick), highlighting the difficulty of maintaining contact consistency during dynamic motions.

*3) Feet slipage:* The results on feasibility and contact sequence accuracy are largely driven by foot slippage. In GR retargeting, trajectories frequently exhibit sliding artifacts during intended contact phases. This occurs because GR lacks a dynamic contact model and relies purely on a noisy expert demonstration; conversely, these artifacts are mitigated in DDR and IDR, where dynamic consistency constraints enforce stationary contact points.

*4) Success rate:* To further analyze the impact of the kinematic bias introduced by GR on IDR, we evaluate the success rate of the retargeted trajectories. A trial is marked as a failure if the pelvis deviation from the reference exceeds 50 cm, indicating either a fall or a significant tracking failure.

We exclude GR from this comparison as the concept of "falling" is not applicable to a purely geometric method lacking dynamic simulation. As shown in Table IV, DDR

TABLE IV: Success rate of the DDR and IDR methods considering 5 random seeds for each test motion.

| Movement | IDR | DDR (ours) |
|---|---|---|
| Squat | **100.00%** | **100.00%** |
| Kung fu | 66.67% | **100.00%** |
| One-foot Balance | 13.33% | **46.67%** |
| Pistol Squat | 40.00% | **80.00%** |
| Balancing Stick | 0.00% | **66.67%** |

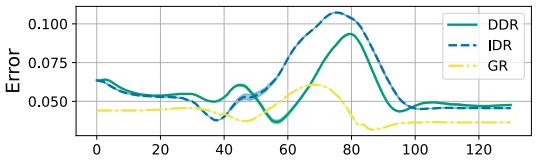



Fig. 5: Temporal evolution of pelvis Laplacian error during a squat motion.

consistently achieves a higher success rate than IDR. These results confirm that IDR is penalized by the geometric bias of its GR initialization; specifically, drifts in the GR trajectory often pull the IDR solver toward unstable regions, leading to optimization failures or simulated falls.

*5) Reference Tracking:* To conclude our evaluation of retargeted trajectories, we compare tracking accuracy across all methods using the Laplacian distance metric (see Sec. III), which quantifies global shape deformation at each timestep. As shown in Fig. 5 for a squat motion, GR achieves the lowest pelvis tracking error. This is expected, as GR neglects the feasibility constraints enforced by dynamic simulation. Conversely, IDR performs worse than DDR; this suggests that while the GR initialization provides a strong tracking baseline, it introduces a kinematic bias incompatible with dynamic constraints, ultimately degrading the IDR solution. Notably, the tracking gap between GR and DDR narrows when evaluating end-effectors, as illustrated in Fig. 6. Table V summarizes the aggregate tracking error for all keypoints across all motions. While GR generally maintains

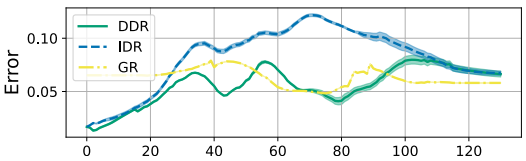
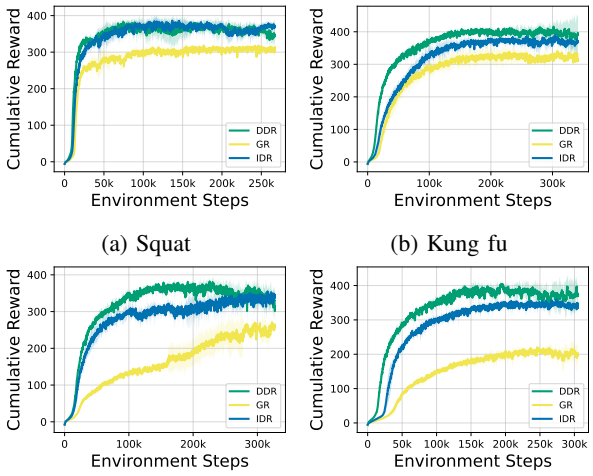



Fig. 6: Temporal evolution of left foot Laplacian error during a squat motion.

TABLE V: Aggregate keypoints tracking performance. This table presents the mean and standard deviation of the Laplacian errors over all keypoints for each motion.

| Movement | GR | IDR | DDR (ours) |
|---|---|---|---|
| Squat | **0.057 (±0.02)** | **0.055 (±0.02)** | **0.055 (±0.01)** |
| Kung fu | **0.062 (±0.02)** | 0.072 (±0.03) | **0.065 (±0.02)** |
| One-foot balance | 0.159 (±0.12) | 0.113 (±0.06) | **0.101 (±0.05)** |
| Pistol Squat | **0.057 (±0.02)** | 0.079 (±0.07) | 0.061 (±0.02) |
| Balancing Stick | **0.048 (±0.02)** | 0.089 (±0.06) | 0.072 (±0.05) |




(a) Squat (b) Kung fu

(c) Long one-foot balance (d) Pistol Squat

Fig. 7: Cumulative Reward evolution in function of iterations learning for (a) squat, (b) kung fu, (c) long one-foot balance, and (d) pistol squat.

a slight advantage in raw tracking, DDR achieves comparable performance while ensuring physical feasibility. IDR consistently underperforms relative to the other methods, further demonstrating that geometric bias from poor initialization hinders the quality of dynamically feasible retargeting.

### C. Imitation Policies

The motivation for generating dynamically feasible references is to improve the performance of downstream imitation learning. To evaluate this, control policies capable of tracking these retargeted trajectories are trained using the RL framework described in Section III-E.

*1) Learning performances:* For each motion, we train policies to track the GR, IDR, and DDR references using the RL framework described in Sec III-E, executing five random seeds per retargeting method to account for training variance and reporting the averaged results.

The evolution of total reward throughout the training process is illustrated in Fig. 7. Quantitative metrics in Table VI confirm that while reference feasibility is necessary, it is not sufficient for optimal learning. Incorporating dynamic feasibility with IDR or DDR provides a significant improvement over the kinematic approach GR. However, while IDR generates physically valid trajectories, it still suffers from a kinematic bias: its optimization remains anchored to the geometric retargeting. Our DDR method bypasses this bias by optimizing dynamics directly. This demonstrates that DDR provides the most coherent learning signal and yielding the highest final reward and fastest convergence across almost all motions.

*2) Reference Tracking:* As a final evaluation, we compare policy rollouts against the original retargeted reference trajectory to assess how effectively each policy reproduces its reference. Table VII summarizes the RMSE of the joint trajectories, the mean absolute cartesian position error, and the mean Laplacian error over the keypoints across each

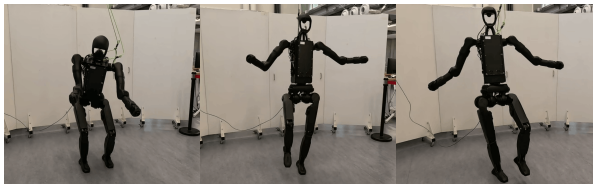
TABLE VI: RL efficiency and convergence metrics. This table compares learning efficiency when training using GR, IDR, and DDR references. **Final** denotes the cumulated reward averaged over the plateau of the training iterations, while **90%** indicates the number of environment steps required to reach 90% of that value.

| Movement | GR | | IDR | | DDR (Ours) | |
|---|---|---|---|---|---|---|
| | Final | 90% | Final | 90% | Final | 90% |
| Squat | 307.0 | 41.3k | **374.8** | 38.2k | 363.2 | **21.1k** |
| Kung fu | 322.7 | 91.1k | 366.8 | 95.6k | **398.4** | **71.6k** |
| One-foot Bal. | 253.7 | 219.9k | 301.6 | **57.3k** | **360.0** | 76.3k |
| Pistol Squat | 200.7 | 128.4k | 340.8 | 93.5k | **383.7** | **79.6k** |
| Bal. Stick | 252.6 | 103.2k | - | - | **364.5** | **78.9k** |

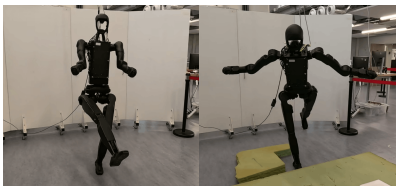
motion. These metrics further validate the limitations of kinematic retargeting and explicitly highlight the negative impact of kinematic bias. While IDR improves upon GR by providing a physically valid reference, it still generally exhibits higher positional and laplacian errors compared to DDR. This performance gap stems from a fundamental difference in how the references handle the inherent dynamics discrepancies between human and humanoid morphologies. GR provides a target that is often physically impossible for the robot to execute, forcing the RL agent into a constant conflict between tracking the reference and adhering to physical laws, which yields high positional errors. Conversely, while IDR corrects these immediate physical violations, its optimization remains anchored to the initial geometric retargeting. This embeds a kinematic bias that pushes the trajectory into marginal areas near dynamic unfeasibility, resulting in motions that the agent still struggles to track. By optimizing the dynamics directly without patching an intermediate kinematic retargeting, DDR avoids this bias entirely. Consequently, the RL agent can track the DDR target motion with the highest fidelity, without having to fight a marginally feasible or physically conflicting reference.

### D. Real World experiments

To validate the practical utility of DDR, we deploy the trained control policies on physical hardware. The experiments are conducted on the human-size *Unitree H1-2* humanoid, utilizing onboard proprioception and a Mocap system for control. We evaluate the zero-shot transferability of our approach by deploying the squat, pistol squat, kung fu, one-foot balance, and balancing stick policies. As shown in Fig. 8 and the supplementary video, the robot successfully executes the target motions without any task-specific manual tuning. Notably, during the pistol squat, the policy exhibits a one-foot recovery jump, as shown in the supplementary material, highlighting the robustness and practical utility of the RL framework.



(a) Squat (b) Kung-fu (c) Long one-foot balance



(d) Pistol Squat (e) Balancing Stick

Fig. 8: The *Unitree* robot *H1-2* performing (a) a stable squat, (b) a static kung fu pose, (c) an extended one-foot balance, (d) a dynamic pistol squat, and (e) a balancing stick pose.

## V. Conclusion

In this work, we introduce Direct Dynamic Retargeting (DDR), a novel framework for generating high-quality, dynamically feasible reference trajectories for imitation learning. By formulating the reference generation problem directly in the task space using a Laplacian graph distance, our approach effectively mitigates the impact of noisy and drifting human data extracted from videos. Furthermore, by employing a sampling-based solver directly within a physics simulator, DDR inherently optimizes over complex contact sequences while guaranteeing the physical viability of the resulting motions.

Our evaluations demonstrate that DDR consistently outperforms state-of-the-art retargeting baselines, yielding superior tracking accuracy by eliminating the geometric bias inherent to previous methods. These higher-fidelity references translate directly into downstream gains: Reinforcement Learning agents trained on DDR trajectories converge faster and exhibit more robust behavior across diverse balancing tasks.

However, DDR has limitations. The objective is defined purely in task space, which can lead to ill-conditioned solutions or slower convergence. Finally, performance remains sensitive to task-space weighting and initialization, which currently require manual tuning.

Future work will scale this framework to handle large, diverse sets of motion variables from in-the-wild web videos, enabling the creation of a comprehensive dataset of physically viable humanoid reference motions for the community.


## Acknowledgment

This project was provided with computing HPC and storage resources by GENCI at IDRIS thanks to the grant 2026-AD011016104R1 on the supercomputer Jean Zay's V100 partition.


## References


[1] A. Allshire, H. Choi, J. Zhang, D. McAllister, A. Zhang, C. M. Kim, T. Darrell, P. Abbeel, J. Malik, and A. Kanazawa, "Visual imitation enables contextual humanoid control," in *Proceedings of the Conf. on Robot Learning*, 2025.

TABLE VII: Reference tracking metrics of imitation policies: Root Mean Square Error (RMSE) on the joint configurations, mean cartesian position error, and mean Laplacian error between the reference and the imitated trajectory.

| **Movement** | **Joints RMSE [rad]** | | | **Mean Pos. Error [m]** | | | **Mean Laplacian Error [m]** | | |
|---|---|---|---|---|---|---|---|---|---|
| | GR | IDR | DDR (ours) | GR | IDR | DDR (ours) | GR | IDR | DDR (ours) |
| Squat | 0.916 (±0.04) | 0.806 (±0.00) | **0.750** (±0.00) | 0.410 (±0.29) | 0.311 (±0.23) | **0.265** (±0.22) | 0.280 (±0.16) | 0.229 (±0.13) | **0.203** (±0.13) |
| Kung fu | 0.735 (±0.00) | 0.672 (±0.00) | **0.627** (±0.00) | 0.331 (±0.21) | 0.289 (±0.20) | **0.248** (±0.19) | 0.255 (±0.17) | 0.240 (±0.18) | **0.194** (±0.17) |
| One-foot balance | 0.848 (±0.09) | **0.657** (±0.00) | 0.713 (±0.00) | 0.416 (±0.18) | 0.420 (±0.29) | **0.255** (±0.18) | 0.324 (±0.19) | 0.263 (±0.19) | **0.229** (±0.17) |
| Pistol Squat | 0.976 (±0.06) | 0.807 (±0.02) | **0.729** (±0.00) | 0.541 (±0.37) | 0.486 (±0.31) | **0.316** (±0.26) | 0.329 (±0.19) | 0.284 (±0.17) | **0.244** (±0.16) |
| Balancing Stick | 1.021 (±0.12) | - | **0.667** (±0.00) | 0.414 (±0.23) | - | **0.295** (±0.19) | 0.339 (±0.23) | - | **0.255** (±0.18) |


[2] L. Yang, X. Huang, Z. Wu, A. Kanazawa, P. Abbeel, C. Sferrazza, C. K. Liu, R. Duan, and G. Shi, "Omniretarget: Interaction-preserving data generation for humanoid whole-body loco-manipulation and scene interaction," 2025.

[3] T. Haarnoja, S. Ha, *et al.*, "Learning to walk via deep reinforcement learning," in *Robotics: Science and Systems XV*, 2019.

[4] X. B. Peng *et al.*, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Trans. Graph.*, 2018.

[5] M. Stępień, R. Kourdis, C. Roux, and O. Stasse, "Latent conditioned loco-manipulation using motion priors," in *IEEE-RAS 24th Int. Conf. on Humanoid Robots*, 2025.

[6] Q. Liao, T. E. Truong, X. Huang, Y. Gao, G. Tevet, K. Sreenath, and C. K. Liu, "Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion," 2025.

[7] X.-S. Lei, J. Pan, and J.-B. Su, "Humanoid robot locomotion," in *Int. Conf. on Machine Learning and Cybernetics*, 2005.

[8] T. Zhang, B. Zheng, R. Nai, Y. Hu, Y.-J. Wang, G. Chen, F. Lin, J. Li, C. Hong, K. Sreenath, and Y. Gao, "Hub: Learning extreme humanoid balance," *arXiv preprint arXiv:.07294*, 2025.

[9] X. Bin Peng *et al.*, "Learning agile robotic locomotion skills by imitating animals," in *Robotics: Science and Systems XVI*, 2020.

[10] G. Yang, S. Yang, J. Z. Zhang, Z. Manchester, and D. Ramanan, "Ppr: Physically plausible reconstruction from monocular videos," in *IEEE/CVF Int. Conf. on Computer Vision*, 2023.

[11] V. Dhedin, I. Taouil, S. Omar, D. Yu, K. Tao, A. Dai, and M. Khadiv, "DynaRetarget: Dynamically-Feasible retargeting using Sampling-Based trajectory optimization," 2026.

[12] T. Yoon, D. Kang, S. Kim, J. Cheng, M. S. Ahn, S. Coros, and S. Choi, "Spatio-temporal motion retargeting for quadruped robots," *IEEE Trans. on Robotics*, 2025.

[13] C. Pan, C. Wang, H. Qi, Z. Liu, H. Bharadhwaj, A. Sharma, T. Wu, G. Shi, J. Malik, and F. Hogan, "Spider: Scalable physics-informed dexterous retargeting," 2025.

[14] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of operations research*, 2005.

[15] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard, "Crocoddyl: An Efficient and Versatile Framework for Multi-Contact Optimal Control," in *IEEE Int. Conf. on Robotics and Automation*, 2020.

[16] K. Ayusawa and E. Yoshida, "Motion retargeting for humanoid robots based on simultaneous morphing parameter identification and motion optimization," *IEEE Trans. on Robotics*, 2017.

[17] S. Caron *et al.*, "Pink: Python inverse kinematics based on Pinocchio," 2026. [Online]. Available: https://github.com/stephane-caron/pink

[18] C. M. Kim*, B. Yi*, H. Choi, Y. Ma, K. Goldberg, and A. Kanazawa, "Pyroki: A modular toolkit for robot kinematic optimization," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2025.

[19] T. Moulard, E. Yoshida, and S. Nakaoka, "Optimization-based motion retargeting integrating spatial and dynamic constraints for humanoid," in *IEEE ISR*, 2013.

[20] X. Zhang *et al.*, "Kinodynamic motion retargeting for humanoid locomotion via multi-contact whole-body trajectory optimization," 2026.

[21] Y. Pan, R. Qiao, L. Chen, K. Chitta, L. Pan, H. Mai, Q. Bu, C. Zheng, H. Zhao, P. Luo, and H. Li, "Agility meets stability: Versatile humanoid control with heterogeneous data," *arXiv preprint arXiv:.17373*, 2025.

[22] X. B. Peng *et al.*, "Amp: Adversarial motion priors for stylized physics-based character control," *ACM Trans. on Graphics*, 2021.

[23] M. Loper *et al.*, "Smpl: A skinned multi-person linear model," in *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, 2023.

[24] V. Kurtz, "Hydrax: Sampling-based model predictive control on gpu with jax and mujoco mjx," 2024, https://github.com/vincekurtz/hydrax.

[25] E. Chane-Sane, P.-A. Leziart, *et al.*, "Cat: Constraints as terminations for legged locomotion reinforcement learning," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2024.

[26] C. Roux, E. Chane-Sane, L. de Matteïs, T. Flayols, J. Manhes, O. Stasse, P. Souères, and N. Mansard, "Constrained reinforcement learning for unstable point-feet bipedal locomotion applied to the bolt robot," in *IEEE-RAS 24rd Int. Conf. on Humanoid Robots*, 2025.

[27] J. Schulman *et al.*, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[28] A. Serrano-Muñoz *et al.*, "skrl: Modular and flexible library for reinforcement learning," *Journal of Machine Learning Research*, 2023.

[29] M. Mittal *et al.*, "Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning," *arXiv preprint arXiv:2511.04831*, 2025.

[30] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, "The pinocchio c++ library – a fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *IEEE Int. Symposium on System Integrations*, 2019.

[31] A. Bambade *et al.*, "Proxqp: an efficient and versatile quadratic programming solver for real-time robotics applications and beyond," *IEEE Transactions on Robotics*, 2025.

[32] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *IEEE/RSJ int. conf. on intelligent robots and systems*, 2012.