OmniXtreme: Breaking the Generality Barrier in High-Dynamic Humanoid Control

Author Names Omitted for Anonymous Review.

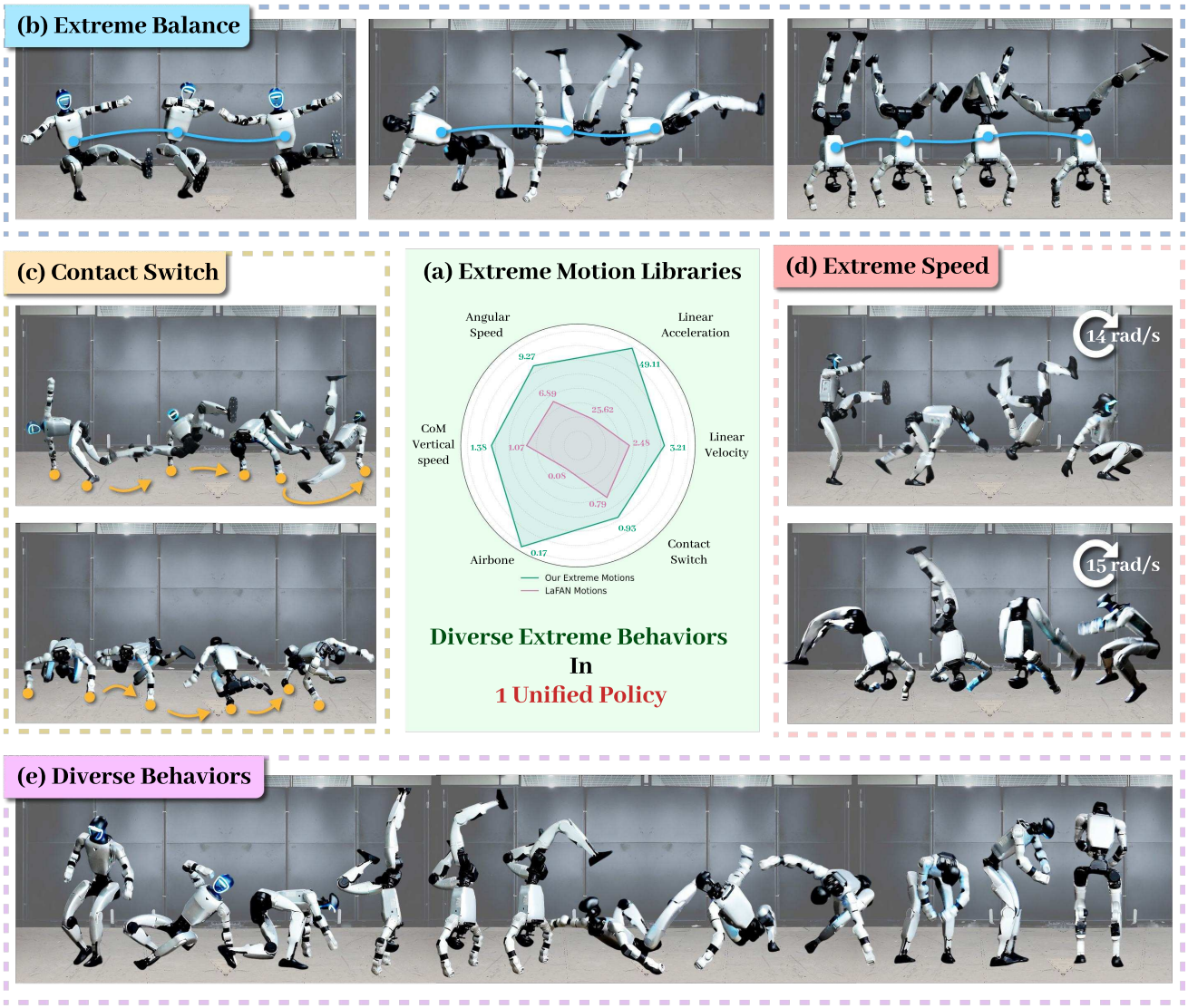

Fig. 1: **Extreme whole-body humanoid control from our unified policy OmniXtreme.** (a) A quantitative comparison shows that our curated extreme motion libraries occupy substantially more challenging regimes than standard multi-motion benchmarks (*e.g.*, Unitree-retargeted LAFAN1). Real-world executions of our unified policy OmniXtreme demonstrate robust and physically executable extreme behaviors drawn from this motion set, including (b) extreme balance behaviors, (c) rapid contact switching with complex support transitions, (d) high-speed motions with large angular velocities, and (e) diverse whole-body behaviors spanning qualitatively distinct motion styles.

***Abstract*—High-fidelity motion tracking serves as the ultimate litmus test for generalizable, human-level motor skills. However, current policies often hit a "generality barrier": as motion libraries scale in diversity, tracking fidelity inevitably collapses—especially for real-world deployment of high-dynamic motions. We identify this failure as the result of two compounding factors: the learning bottleneck in scaling multi-motion optimization and the physical executability constraints that arise in real-**

world actuation. To overcome these, we introduce OMNIXTREME, a scalable framework that decouples general motor skill learning from sim-to-real physical skill refinement. Our approach uses a flow-matching policy with high-capacity architectures to scale representation capacity without the interference-intensive multi-motion RL optimization, followed by an actuation-aware refinement phase that ensures robust performance on physical hardware. Extensive experiments demonstrate that OMNIXTREME maintains high-fidelity tracking across diverse, high-difficulty datasets. On real robots, the unified policy successfully executes multiple extreme motions, effectively breaking the long-standing fidelity–scalability trade-off in high-dynamic humanoid control.

I. INTRODUCTION

We ultimately seek general-purpose humanoids with scalable, human-level whole-body motor skills. A natural and widely used way to study such capability is high-fidelity motion tracking, where a controller must reproduce reference motions accurately while remaining dynamically stable under contacts and disturbances. High-quality tracking is more than an aesthetic goal: it captures whole-body coordination and contact timing that underlie loco-manipulation, expressive interaction, and many downstream core humanoid capabilities [58, 50, 51, 1, 43, 47, 52].

Over the past years, learning-based motion tracking has made striking progress: with carefully designed objectives and reinforcement learning, policies can track individual motions with high precision, including highly dynamic behaviors such as dance, flips, and martial arts [56, 28, 14]. More recent work [5, 30, 9, 23, 11, 6, 12, 57, 54, 26] has taken important steps toward multi-motion controllers that cover broader behavior libraries. Yet a recurring pattern persists: when we scale to larger, more heterogeneous motion libraries spanning diverse styles, contact regimes, and timing modes, motion tracking quality tends to degrade. Controllers become conservative and "average," break on the hardest motions, or prove brittle to the small deviations that inevitably occur in sim-to-real transfer. The degradation is particularly pronounced in *high-dynamic motions*, where even small tracking errors can rapidly cascade into catastrophic failures. This long-standing fidelity–scalability trade-off has effectively capped the level of generality achievable in humanoid motor control, particularly in *high-dynamic regimes*, suggesting a fundamental limitation rather than isolated engineering issues [11, 57, 14, 56].

A central question therefore arises: **why is high-fidelity motion tracking so difficult to scale, especially on real humanoid robots?** We argue that this difficulty stems from two compounding barriers that emerge at different stages of the current sim-to-real training pipeline.

The first barrier is the **learning bottleneck** that arises even in simulation. Several recent works [57, 5, 23, 30, 54, 26, 12, 11] have begun to explore multi-motion humanoid tracking, aiming to improve scalability beyond single-motion imitation. However, existing approaches remain constrained by both representation and optimization. On the representation side, most approaches rely on relatively simple policy parameterizations, such as MLP actors [57, 5, 23, 54, 26, 12, 11, 46]. When required to map observations to highly heterogeneous action targets arising from diverse behaviors and contact patterns, such parameterizations have been observed to exhibit limited scalability as data diversity increases [33]. On the optimization side, jointly training a unified policy across many motions using reinforcement learning exacerbates gradient interference, often leading to conservative averaging and selective failures on *high-dynamic behaviors* [12, 11, 9, 30]. Together, these factors cause tracking fidelity to collapse as motion diversity and difficulty increase.

The second barrier is the **physical executability bottleneck** that emerges at deployment. Even when high-fidelity tracking is achieved in simulation, transferring such behaviors to physical robots remains challenging. In prior humanoid learning pipelines [11, 57, 5, 23, 30, 54, 28, 14, 26, 49], actuation constraints during training are modeled primarily through joint position limits and simple effort bounds. Although these simplifications facilitate learning, they become insufficient in *high-dynamic motions*, where system behavior is dominated by unmodeled actuator nonlinearities [41], such as torque–speed characteristics and velocity-dependent torque losses, as well as power-related effects, including regenerative power phenomena, leading to rapid degradation of execution stability. As a result, fidelity that appears scalable in simulation may still fail to materialize on real robots.

Motivated by this analysis, we propose OMNIXTREME, a scalable training framework designed to explicitly address both barriers, with the goal of enabling a single policy to robustly control diverse and *high-dynamic* humanoid behaviors. To overcome the learning bottleneck, OMNIXTREME adopts a flow matching policy and performs specialist-to-unified generative pretraining via behavior cloning from a collection of motion specialists. This design decouples representation learning from optimization, scaling expressive capacity through a high-capacity generative policy while avoiding interference-heavy multi-motion reinforcement learning.

To overcome the physical executability bottleneck, OMNIXTREME introduces a residual reinforcement learning post-training refinement for execution under realistic actuation constraints, which become particularly critical in *high-dynamic motions*. Rather than relearning motion tracking, this stage refines the pretrained policy to respect real-world actuation constraints through actuation-aware modeling, refined domain randomization, and explicit penalties on power-related effects. This targeted refinement ensures that the scaled tracking policy remains physically executable under realistic hardware dynamics.

We validate OMNIXTREME through extensive simulation and real-robot evaluations on increasingly diverse and high-dynamic motion libraries. Beyond standard multi-motion benchmarks, we curate a set of extreme motions characterized by high speed, frequent contact transitions, and tight timing constraints, and evaluate OMNIXTREME across this full spectrum. As shown in Fig. 1, OMNIXTREME successfully executes a wide range of extreme behaviors on a Unitree G1 humanoid robot, including flips, acrobatics, and breakdancing where even minor deviations can rapidly cascade into failure.

Together, these results constitute a stringent scalability stress test and challenge the prevailing assumption that tracking fidelity must collapse as motion diversity and difficulty increase.

Overall, our contributions are fourfold:

1) We present OmniXtreme, a scalable training framework for high-fidelity humanoid motion tracking that explicitly tackles the fundamental scalability challenge in *high-dynamic* humanoid control.
2) We introduce a specialist-to-unified generative pretraining stage based on flow matching, enabling a unified policy to scale across heterogeneous and high-dynamic motions.
3) We propose an actuation-aware residual reinforcement learning post-training stage that refines the pretrained policy under realistic actuation constraints, ensuring physical executability.
4) We demonstrate through extensive simulation and real-world experiments that OmniXtreme enables a single unified policy to robustly execute diverse and extreme motions, addressing the conventional fidelity–scalability trade-off, especially for high-dynamic motions.

II. Related Work

A. Humanoid Whole-body Control and General Tracking

Recent research in humanoid whole-body control has demonstrated remarkable progress across diverse skills [61, 19, 15, 56, 36], including dance, fall recovery, and parkour. However, achieving both high-fidelity motion tracking and scalability across large and diverse motion libraries remains an open challenge. Frameworks such as ASAP [14] and BeyondMimic [28] demonstrate strong performance in high-quality imitation of individual motion clips, yet extending these approaches to increasingly large motion sets introduces additional optimization complexity. On the other hand, large-scale RL-based trackers including OmniH2O [11], ExBody2 [23], and GMT [5] show promising scalability, though maintaining precise motion fidelity under extensive skill coverage remains challenging. This tension is often reflected as a fidelity–scalability trade-off in practice. To address this issue, OmniXtreme introduces a generative action representation and a specialist-to-unified optimization framework, enabling scalable learning while maintaining strong tracking precision across high-dynamic motion datasets.

B. Diffusion and Flow-based Action Modeling for Robotic Planning and Control

Diffusion and flow-based models [42, 16, 34, 38, 29, 33, 45, 44, 8] have shown strong capability in robot learning, leveraging iterative refinement and stochastic sampling to enhance robustness and diversity in robotic control and planning [33]. While early research focused on high-level trajectory planning or low-frequency visuomotor tasks [22, 18, 8, 48, 4], DiffuseLoco [20] takes a step to apply them to high-frequency quadruped control. To further enhance expressivity and robustness, recent works like Policy Decorator [53] and ResiP [2] introduce residual policy learning on arm-based robots, coupling

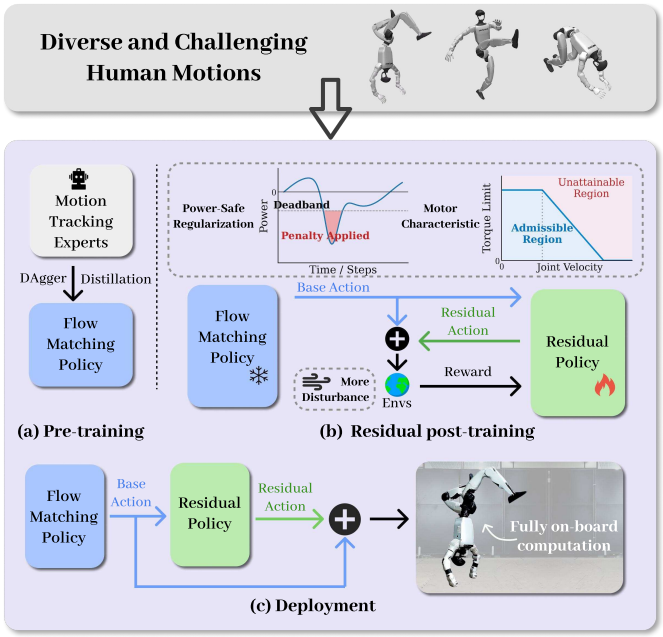

Fig. 2: **Overview of the OmniXtreme.** (a) Pretraining phase: A unified base policy is trained via DAgger-based Flow Matching to aggregate diverse motion priors from different motion tracking experts. (b) Post-training phase: The base policy is frozen while a residual policy is optimized under stringent motor constraints, extensive domain randomization, and power-safety regularization to bridge the sim-to-real gap. (c) Onboard deployment: The whole inference pipeline is real-time and executed entirely onboard, facilitating robust and agile control in physical environments.

frozen base models with refinement layers to handle covariate shift and precision bottlenecks in long-horizon assembly. However, given the vast skill space and inherent instability that distinguish humanoids from quadrupeds and manipulators, current effort such as BeyondMimic [28] focuses on flexible guided control interfaces rather than clearly demonstrating the ability to scale to large and diverse motion libraries. Different from previous work, OmniXtreme introduces a comprehensive training pipeline involving DAgger-based Flow Matching pretraining and residual post-training that pushes the boundaries of low-level scalability and agility, far surpassing the motion diversity and dynamic performance of previous approaches.

C. Actuation-aware Agile Robotic Control

Achieving agility remains a frontier in robotics [25, 27, 13, 24, 21, 32, 55, 17, 7, 60, 61, 49]. ACRL [41] leveraged actuator-constrained RL for high-speed quadrupedal locomotion, while Closing the Reality Gap [59] utilized a current-to-torque calibration and actuator dynamics modeling for dexterous five-finger manipulation. Despite these advancements in other morphologies, learning agile and actuation-aware control policies for humanoids remains an underexplored area. OmniXtreme addresses this gap by integrating physics-informed motor modeling and actuation regularization, pushing the boundaries of agile humanoid performance under realistic hardware constraints.

III. Methodology

In this section, we present OmniXtreme, a two-stage training framework for scalable, high-fidelity humanoid motor skill learning. The **Scalable Flow-based Pretraining** stage focuses on high-fidelity motion imitation and representation capacity acquisition. Specifically, we distill diverse expert behaviors from a collection of motion-specific expert policies into a single unified base policy using flow matching [29]. This generative pretraining stage establishes a shared tracking prior across heterogeneous motions, without relying on interference-prone joint multi-motion reinforcement learning.

To address the gap between simulation and real-world execution, we further introduce an **Actuation-Aware Post-Training** stage based on residual reinforcement learning. Rather than relearning motion tracking, a residual policy is trained to produce corrective actions that complement the pretrained flow matching base policy. This stage aligns the overall system with real-world actuation constraints while introducing substantially more aggressive domain randomization. Through this targeted refinement, the residual policy adapts the pretrained tracking behavior to realistic hardware dynamics, improving physical executability and deployment robustness.

A. Scalable Flow-based Policy Pretraining

1) Problem Formulation: During the pretraining phase, we learn a flow-matching robot policy with Dataset Aggregation (DAgger)-based distillation [39, 45]. Specifically, we consider the observation space of $o = \{p, c, h\}$ covering: (i) robot proprioception p, including joint positions, velocities, base angular velocities, and previous actions; (ii) command c consisting of 6D torso orientation differences along with target joint positions and velocities from the reference motion; and (iii) history information h encompassing past proprioceptive states. Given a reference motion dataset $M = \{m_i\}_{i=1}^{M}$, our goal is to first learn expert policies $\Pi^i_{\text{expert}} = \{\pi^i_{\text{expert}}(a|o)\}_{i=1}^{M}$ for each reference motion, then distilling it into a flow-based general policy $\pi_\theta(a|o)$.

2) Expert Policy Learning: For expert policy training, we draw the reference motion dataset M from a combination of Unitree-retargeted LAFAN1 (LAFAN1) dataset [10], AMASS [31], MimicKit [35], and the Reallusion motion library [37], covering both diverse behavioral patterns and high-dynamic maneuvers. All reference motions are first retargeted to the Unitree G1 humanoid robot using GMR [54, 3]. Subsequently, we train each expert policy π^k_{expert} on the specific motion m_k via Proximal Policy Optimization (PPO) [40].

3) Flow-matching Policy Learning: We learn the flow-matching robot policy with DAgger by first rolling out the current flow-based policy $\pi_\theta(a|o)$ in the simulator and collecting a trajectory of visited states $\{o_1, \cdots, o_N\}$ given reference motion dataset M. For each visited state o, we obtain the expert action a_{expert} by querying the corresponding expert policy. The flow-based model then learn to recover the expert action a_{expert} from noised action by optimizing:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t,\epsilon,a_{\text{expert}}}\left[\|v_\theta(a_t, t, o) - (\epsilon - a_{\text{expert}})\|^2\right], \quad (1)$$

where $a_t = (1-t)a_{\text{expert}} + t\epsilon$ is the noised action interpolated between expert action a_{expert} and random noise $\epsilon \sim \mathcal{N}(0, I)$ depending on flow timestep $t \in [0, 1]$. This objective learns a velocity field $v_\theta(a_t, t, o)$ to predict the target velocity $u = \epsilon - a_{\text{expert}}$, learning the denoising directions at each flow timestep [29]. During the optimization process, the timestep t is sampled from a Beta distribution, $t \sim \text{Beta}(\alpha, \beta)$, to focus the learning process on specific regions of the probability path, thereby enhancing convergence and trajectory refinement. With the velocity field v_θ, we can generate action a_0 from random noise $a_1 \sim \mathcal{N}(0, I)$ by integrating v_θ from $t = 1$ to $t = 0$ via the foward Euler rule:

$$a_{t-\frac{1}{D}} = a_t - \frac{1}{D} v_\theta(a_t, t, o), \quad (2)$$

where D is the number of integration or denoising steps controlling the approximation accuracy. By iteratively rolling out trajectories and supervising them with expert actions using Eq. (1), we learn π_θ as a general policy to map the current observation to appropriate actions. To mitigate distribution shift between expert and learned policies, we align the termination conditions between the expert and student environments, thereby reducing exposure to states that the expert cannot handle. This alignment, combined with DAgger-style corrective supervision, ensures robust performance even when the policy encounters unseen or rare states. The full training procedure is illustrated in Fig. 2(a) and detailed in Alg. 1.

Regarding training cost, the expert policies can be trained fully in parallel, making the approach scalable in practice. Importantly, once the base flow-matching policy acquires sufficient generalization ability, new motions can often be incorporated through post-training(Sec. III-B) without retraining new expert policies, significantly reducing the cost of expanding the motion library.

4) Fidelity-Preserving Randomization and Noise: To maintain a high degree of motion expressivity while ensuring physical stability, we implement a conservative randomization and noise strategy, as detailed in Tab. I, during both the teacher training and pretraining phases. By utilizing moderate noise levels and domain randomization, we prevent the performance collapse often induced by excessive stochasticity. This ensures that the agent accurately captures the underlying physical dynamics, resulting in a flow matching policy that possesses foundational sim-to-real robustness and the predictive certainty necessary for real-world deployment.

B. Actuation-Aware Post-training Phase

1) Residual Policy Modeling: While the pretrained flow matching base policy provides a robust and unified behavioral foundation, it encounters performance gaps when facing real-world physics. To better account for this gap and enable smooth sim-to-real transfer, we propose a post-training refinement stage using a lightweight MLP-based residual-corrective learning. Specifically, we learn the residual correction policy π_ϕ on top of the frozen pretrained policy π_θ by generating

TABLE I: Configurations for noise, domain randomization, and termination thresholds during pre-training and post-training phases. Here $\pm x$ denotes $[-x, x]$.

Parameter Item	Moderate	Aggressive
Noise and Domain Randomization		
Joint Position (rad)	± 0.01	± 0.01
Joint Velocity (rad/s)	± 0.5	± 0.5
Angular Velocity (rad/s)	± 0.2	± 0.2
Torso 6D Rotation (rad)	± 0.05	± 0.05
Base CoM Offset (m)	x: ± 0.025, y, z: ± 0.05	x: ± 0.025, y, z: ± 0.05
Static Friction	$[0.3, 1.6]$	$[0.3, 1.6]$
Dynamic Friction	$[0.3, 1.2]$	$[0.3, 1.2]$
Action Delay (ms)	$[0, 15]$	$[\mathbf{5}, \mathbf{10}]$
Coefficient of Restitution	None	$[\mathbf{0.0}, \mathbf{0.5}]$
Default Calib. (rad)	± 0.01	± 0.01
Init. Pose (rad)	± 0.1	$\mathbf{\pm 0.15}$
Init. Lin. Vel. (m/s)	xy: ± 0.5, z: ± 0.2	$\mathbf{xy} : \mathbf{\pm 0.75}, \mathbf{z} : \mathbf{\pm 0.3}$
Init. Ang. Vel. (rad/s)	RP: ± 0.52, Y: ± 0.78	$\mathbf{RP} : \mathbf{\pm 0.78}, \mathbf{Y} : \mathbf{\pm 1.17}$
Push Frequency (s)	$1.0 - 3.0$	$1.0 - 3.0$
Push Lin. Vel. (m/s)	xy: ± 0.5, z: ± 0.2	xy: ± 0.5, z: ± 0.2
Push Ang. Vel. (rad/s)	RP: ± 0.52, Y: ± 0.78	RP: ± 0.52, Y: ± 0.78
Terrain Surface / Step (m)	None	$[\mathbf{0}, \mathbf{0.01}]/\mathbf{0.01}$
Termination Thresholds		
Torso Pos. Z / Ori. Error	0.25m / 0.8rad	$\mathbf{0.375m/1.2rad}$
End-Effector Z-Error (m)	0.25	$\mathbf{0.375}$

the refined action $a = a_{\text{flow}} + a_{\text{res}}$ and supervising it with cumulative rewards via PPO, detailed in the Appendix.

In particular, the observation space for the residual actor and critic incorporates robot proprioception, motion command, and the current base action a_{flow}. Within the proprioceptive state, the residual policy observes the previous refined action, whereas the flow matching base policy remains conditioned on the previous flow-based action.

2) Actuation-aware Physical Constraint Modeling: To explicitly account for real-world actuation effects, we train the residual policy using environments that incorporate realistic actuation-aware physical constraints and domain randomization, as shown in Fig. 2(b). The actuation-aware physical modeling is detailed as follows:

a) Aggressive Domain Randomization: We substantially increase the range for domain randomization by up to 50% on common domain randomization settings, including initial pose noise, force disturbances magnitude, angular velocity, *etc*., as detailed in Tab. I. We randomize the terrain by adding surface noise and placing vertical steps randomly in the scene. Crucially, we relax the termination thresholds by 1.5× from their base values (*e.g*., orientation error from 0.8 to 1.2 rad). This relaxation allows the residual policy to explore and correct for large-deviation but recoverable states that would otherwise be prematurely terminated.

b) Power-Safe Actuation Regularization: In practice, highly dynamic motions can induce large transient braking loads that are not explicitly regulated in standard training pipelines. To address the issue, we introduce an explicit penalty on excessive negative joint mechanical power to mitigate aggressive motor braking that can trigger overcurrent protection or thermal stress on real robots. Specifically, we use the instantaneous mechanical power $P = \tau \cdot \omega$ calculated from

Algorithm 1 Flow-based Pretraining and Inference

Training: Distill Flow Matching Policy with DAgger
Input: Teacher policy set $\pi_{teacher}$, Flow matching policy π_θ, Motion dataset $\mathcal{M}$, Replay buffer $\mathcal{D}$
repeat
  $\mathcal{D} \leftarrow \emptyset$ ▷ On-policy reset: Clear buffer for the new iteration
  Sample motion $m \sim \mathcal{M}$ and select teacher $\pi^m_{teacher}$
  Rollout π_θ in simulator conditioned on m to obtain states $s_{1:T}$
  for $t = 1$ to T **do**
    $a_{expert,t} \leftarrow \pi^m_{teacher}(s_t)$ ▷ Expert labeling
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, m, a_{expert,t})\}$ ▷ Aggregate data
  end for
  Flow Matching Optimization:
  for each gradient step **do**
    Sample $(s, a_{expert}) \sim \mathcal{D}$
    Sample $t \sim \text{Beta}(\alpha, \beta)$ and $\epsilon \sim \mathcal{N}(0, I)$
    Construct probability path: $x_t = (1-t)a_{expert} + t\epsilon$
    Compute target velocity: $u_t = \epsilon - a_{expert}$
    Update θ using gradient descent on $\|v_\theta(x_t, t, c) - u_t\|^2$
  end for
until convergence

Inference: Action Sampling with Euler Integration
Input: Trained velocity field v_θ, Concatenated condition c, Number of steps N
Set step size $\Delta t = 1/N$
Initialize $x_1 \sim \mathcal{N}(0, I)$ ▷ Start from Gaussian noise
for $k = 0$ to $N - 1$ **do**
  $t = 1 - k \cdot \Delta t$ ▷ Reverse time from 1 to 0
  $v_t \leftarrow v_\theta(x_t, t, c)$
  $x_{t-\Delta t} \leftarrow x_t - v_t \cdot \Delta t$ ▷ Reverse-time Euler step to obtain x_0
end for
return $a = x_0$ ▷ Execute final reconstructed action

the applied joint torque τ and angular velocity ω as a critical policy for actuator safety. We penalize the negative power beyond a predefined deadband to suppress large regenerative braking events for each joint:

$$\mathcal{L}_{\text{neg-power}} = \sum_{j \in \mathcal{J}} \left(\frac{\max(-P_j - P_{\text{db}}, 0)}{K} \right)^2, \tag{3}$$

where P_j, P_{db} denotes power for joint j and the deadband threshold, respectively. K is a normalizing constant. In practice, this term is selectively applied to the knee joints in the context of high-dynamic motions(*e.g*., backflips), as these joints are particularly prone to high braking loads during impacts and recovery phases.

c) Actuation-Aware Torque–Speed Constraints: A major source of sim-to-real discrepancy stems from the oversimplification of actuator modeling, whereas standard torque clipping

techniques neglect velocity-dependent constraints imposed by back-electromotive force and physical power limits. This omission leads to a significant sim-to-real gap when performing high-dynamic motions. To bridge this gap, we integrate a realistic torque-speed operating envelope directly into the simulation, dynamically deriving torque limits based on the instantaneous alignment of torque and angular velocity:

$$\tau_{\max,0} = \begin{cases} \tau_{y1}, & v \cdot \tau_{\text{in}} > 0, \\ \tau_{y2}, & v \cdot \tau_{\text{in}} \leq 0. \end{cases} \tag{4}$$

The admissible torque is then defined as a monotonically decreasing function of joint velocity magnitude:

$$\tau_{\text{clipped}}(v) = \begin{cases} \tau_{\max,0}, & |v| < v_{x1}, \\ \tau_{\max,0}\left(1 - \dfrac{|v| - v_{x1}}{v_{x2} - v_{x1}}\right), & v_{x1} \leq |v| \leq v_{x2}, \\ 0, & |v| > v_{x2}. \end{cases} \tag{5}$$

The commanded torque is finally clipped to this admissible range before being applied to the joint, which ensures that the simulator never samples torque commands that are physically unattainable for the real actuators.

In addition to torque-speed limits, we further model actuator-level internal losses through a nonlinear friction term applied after torque clipping,

$$\tau_{\text{applied}} = \tau_{\text{clipped}} - \left(\mu_s \tanh\left(\frac{v}{v_{\text{act}}}\right) + \mu_d v\right). \tag{6}$$

The smoothed Coulomb component captures the transition from static to kinetic friction, while the viscous term accounts for velocity-dependent dissipation and provides additional damping. The parameters μ_s, v_{act}, and μ_d are constants.

Overall, this structured refinement stage yields controllers that are simultaneously safer, more robust to large disturbances, and more faithfully aligned with real-world actuator dynamics, thereby enabling reliable deployment on robots.

C. Real World Deployment

The integrated real-world deployment pipeline is illustrated in Fig. 2(c). In the deployment phase, we leverage the pelvis IMU as the primary orientation source and compute the torso rotation through Forward Kinematics (FK). To ensure minimal control latency, the entire computational pipeline—including FK-based state estimation, the base flow matching policy, and the residual policy—is optimized and executed via TensorRT. This integrated pipeline achieves an end-to-end inference latency of about 10ms on the Unitree G1's onboard Orin NX. Such optimization enables the robot to execute high-quality motion tracking at a consistent 50Hz frequency in complex physical environments.

IV. Experiments

We present extensive experiments in simulation and on physical robots to evaluate the scalability of our proposed OmniXtreme system as motion libraries grow in diversity and difficulty. Our experiments are organized around the following key questions:

Q1: Scalable high-fidelity tracking. Compared to prior multi-motion baselines, can our approach maintain high-fidelity tracking at scale, both in simulation and on real robots, without collapsing under representation and optimization challenges?

Q2: Fidelity–scalability trade-off (OmniXtreme v.s. from-scratch RL controllers). As motion diversity and difficulty increase, how does tracking performance degrade for from-scratch multi-motion reinforcement learning controllers, and to what extent can our approach extend the scalability frontier?

Q3: Capacity scaling with flow-based (OmniXtreme v.s. MLP-based controllers). Does increasing model capacity improve large-scale multi-motion tracking performance, and does generative pretraining via flow matching enable stronger and more stable scaling behavior than conventional MLP-based motion tracking controllers?

Q4: Real-world executability and robustness. How do aggressive domain randomization, actuation-aware modeling, and power-aware safety mechanisms individually and jointly affect sim-to-real transfer and real-world execution success?

Q5: Qualitative whole-body capability. Beyond scalar tracking metrics, can OmniXtreme demonstrate agile and versatile whole-body behaviors across diverse motion styles and dynamic contact patterns?

Together, these questions probe the scalability and robustness of OmniXtreme by disentangling the roles of generative pretraining for representation and capacity scaling, and residual post-training for real-world executability.

A. Experimental Setup

1) Motion Libraries: We construct our motion libraries following a two-tier design. First, we use the full LAFAN1 [10], which has been widely adopted in prior multi-motion tracking work and serves as a standard benchmark for evaluating scalability under stylistic and temporal diversity. Second, to evaluate and push the limit of extreme humanoid motions, we curate about 60 highly challenging motions selected from LAFAN1 [10], AMASS [31], MimicKit [35], and Reallusion [37]. These motions exhibit substantially higher dynamic intensity, frequent contact transitions, and tight timing constraints, as shown in Fig. 1(a). We collectively refer to this curated set as the XtremeMotion dataset. Together, LAFAN1 and XtremeMotion form a motion library that spans both standard multi-motion benchmarks and extreme behaviors that probe the limits of fidelity, robustness, and real-world executability.

2) Baselines: We compare against two families of strong baselines designed for multi-motion tracking.

(a) Specialist-to-Unified MLP Distillation. This class of methods [57] first trains per-motion (or per-cluster) specialist policies and then distills them into a single unified MLP tracking policy. Relying on supervised distillation, they benefit from relatively stable and straightforward optimization, but are limited by the representational capacity of the MLP policy.

(b) From-scratch Multi-motion Reinforcement Learning. This class [11, 5, 23, 30, 28] directly trains a single unified tracking policy from scratch using reinforcement learning across all motions, but often suffers from gradient interference and conservative averaging as motion diversity and difficulty increase.

B. Evaluation Metrics

The policy is evaluated through simulated rollouts of motion tracking to extract performance metrics. The primary metric is the **success rate (Succ)**, where an episode is deemed unsuccessful if the humanoid deviates beyond a predefined threshold from the reference motion or becomes unstable. We additionally report the **root-relative mean per-joint position error (MPJPE)** (mm), as well as discrepancies in joint-space **velocity** (Δvel) and **acceleration** (Δacc), to quantify kinematic accuracy and physical fidelity. On physical robots, we evaluate performance using deployment-oriented metrics, including **skill-level success rates** and qualitative assessments of motion fidelity for high-dynamic behaviors.

C. Scalable high-fidelity tracking (Q1)

In this section, we study whether high-fidelity humanoid motion tracking can be preserved by OmniXtreme as motion libraries scale in diversity and difficulty. We compare OmniXtreme with specialist-to-unified MLP distillation (Specialist→Unified MLP) and from-scratch multi-motion reinforcement learning (From-scratch RL) under matched model capacity and identical training data. All methods are trained on the same combined motion library (LAFAN1 + XtremeMotion) and evaluated on three test sets: the full motion library, the high-dynamic XtremeMotion subset, and an unseen motion set (randomly sampled from retargeted AMASS).

Simulation. As summarized in Tab. II, **OmniXtreme consistently outperforms both baselines across all simulation metrics.** The gap becomes substantially larger on XtremeMotion and unseen motions, where baseline methods exhibit reduced success rates and increased tracking errors as motion difficulty increases. This indicates that OmniXtreme preserves tracking fidelity as motion diversity and difficulty scale, rather than degrading under increased complexity.

Real world. We further deploy OmniXtreme on a Unitree G1 humanoid robot using motions drawn from XtremeMotion. For clarity of presentation, motions are grouped into representative skill categories based on shared dynamic structure and contact patterns. A trial is considered successful if the motion is executed without manual intervention or safety-triggered termination. As shown in Tab. III, across 157 real-world trials spanning 24 distinct high-dynamic motions, **OmniXtreme achieves consistently high success rates across diverse skill categories**, including flips, acrobatics, breakdancing, and martial-arts-style motions. These results demonstrate that the scalability gains observed in simulation translate to robust and physically executable behaviors on real hardware.

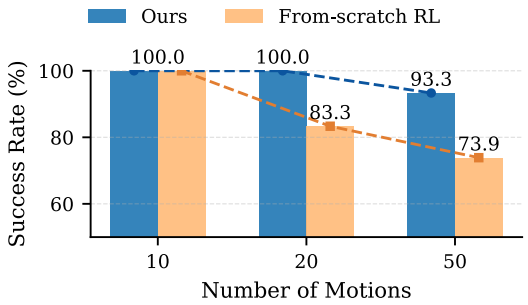

Fig. 3: **Fidelity–scalability trade-off.** Tracking success rate as we progressively scale motion diversity and difficulty, while evaluating all policies on a fixed set of the first 10 motions.

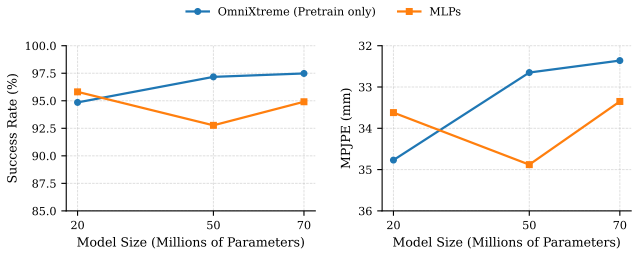

Fig. 4: **Capacity scaling.** Tracking fidelity and robustness as a function of model capacity. OmniXtreme benefit more strongly from scaling, while conventional MLP controllers saturate earlier.

D. Fidelity–scalability trade-off (Q2)

To characterize the fidelity–scalability trade-off in multi-motion tracking, we progressively scale motion diversity by training on an expanding set of motions drawn from XtremeMotion, and analyze how different training paradigms respond under the same evaluation protocol.

from-scratch multi-motion reinforcement learning exhibits earlier and more pronounced performance degradation as scale increases, whereas OmniXtreme maintains higher tracking robustness over a broader scaling range. As shown in Fig. 3, from-scratch multi-motion RL [28] exhibits a characteristic degradation pattern as motion diversity increases: tracking precision deteriorates steadily, followed by a sharp loss of robustness beyond a critical scale. These results indicate that the observed fidelity–scalability trade-off is not inherent, but can be substantially alleviated by a more scalable training paradigm.

E. Capacity scaling (Q3)

We next examine whether increasing model capacity further improves multi-motion tracking performance, and whether our generative policy exhibits stronger scaling behavior than conventional MLP-based controllers [33]. We train a family of models with increasing capacity (e.g., width/depth or Transformer hidden size and layers) under the same data and training recipe. Fig. 4 reports tracking fidelity and robustness as a function of model capacity. We observe that **additional capacity translates more directly into improved tracking quality for our flow matching policies, whereas MLP-based policies show weaker gains.**These results suggest that representational scaling is a practical lever for extending multi-

TABLE II: **Scalable high-fidelity motion tracking under diverse motion sets.** OMNIXTREME consistently achieves lower kinematic errors and higher success rates than baselines, particularly on high-dynamic and unseen motions.

Method	LaFAN1+XtremeMotion				XtremeMotion				Unseen Motions	
	MPJPE ↓	Δvel ↓	Δacc ↓	Succ.(%) ↑	MPJPE ↓	Δvel ↓	Δacc ↓	Succ.(%) ↑	MPJPE ↓	Succ.(%) ↑
From-scratch RL [28]	47.95	10.03	3.27	82.95	54.19	14.04	4.04	79.45	56.87	85.29
Specialist→Unified MLP [57]	33.35	6.70	2.11	94.91	43.43	11.38	2.51	89.22	58.94	85.95
OmniXtreme (Pretrain only)	32.65	6.34	**2.04**	97.17	37.11	10.46	**2.39**	95.16	56.25	89.23
OmniXtreme (Pretrain + Post-train)	**30.93**	**6.19**	2.13	**98.54**	**36.17**	**9.94**	2.58	**95.64**	**56.05**	**89.54**

TABLE III: **Real-world evaluation of OMNIXTREME on Unitree G1.** We evaluate OMNIXTREME on physical hardware using motions drawn from the XtremeMotion motion library.

Skill	#Motions	Attempts	Success (%)↑
Flip	7	55	96.36
Handspring	5	35	88.57
Acrobatics	4	15	80.00
Breakdance	5	22	86.36
Martial arts	3	30	93.33
Total	**24**	**157**	**91.08**

motion tracking fidelity when paired with a scalable training paradigm.

F. Real-world executability and robustness (Q4)

We analyze the contribution of different post-training mechanisms to sim-to-real transfer by incrementally enabling them and evaluating real-world execution at the skill level. Tab. IV summarizes the ablation results.

In summary, **different classes of high-dynamic motions exhibit distinct failure modes, and each execution-oriented mechanism addresses a complementary aspect of real-world executability**. For highly impulsive motions such as flips, enforcing actuator torque-speed constraints is sufficient to enable stable execution, as respecting motor envelopes prevents immediate hardware-level violations. Contact-rich skills such as breakdance and acrobatic motions remain unstable under motor constraints alone, but benefit substantially from aggressive domain randomization, which improves robustness to timing-sensitive contact perturbations. Motions involving high-speed impact buffering, such as acrobatic landings, remain challenging even with aggressive domain randomization, power-safety regularization is critical for these skills, as it mitigates failures caused by excessive transient braking loads and unsafe energy absorption during high-impact contacts. Together, these results show that reliable real-world execution emerges from the combined effects of actuation-aware modeling, robustness-oriented randomization, and energy-aware safety constraints.

G. Qualitative results on extreme motions (Q5)

Finally, we provide qualitative evidence that OMNIXTREME can exhibit agile and versatile whole-body skills across distinct motion styles and contact patterns, beyond what is captured by scalar tracking metrics. We visualize representative rollouts spanning stylistic motions from XtremeMotion. Fig. 5 highlights that OMNIXTREME can track qualitatively different motions with coherent whole-body coordination, complement

TABLE IV: **Ablation of post-training mechanisms.** Real-world executability of different skills under incremental post-training mechanisms. **None**: base pretrained policy only; **MC**: motor constraints; **ADR**: aggressive domain randomization; **PS**: power-safety regularization (overcurrent / regenerative protection). ✓: stable execution; △: unstable or inconsistent execution; ×: consistent failure; ⊖: failures primarily associated with power-safety protection, such as overcurrent or excessive regenerative braking.

Skill	None	+MC	+MC+ADR	Full (+MC+ADR+PS)
Flip	△	✓	✓	✓
Breakdance	△	△	✓	✓
Acrobatics	×	△	⊖	✓

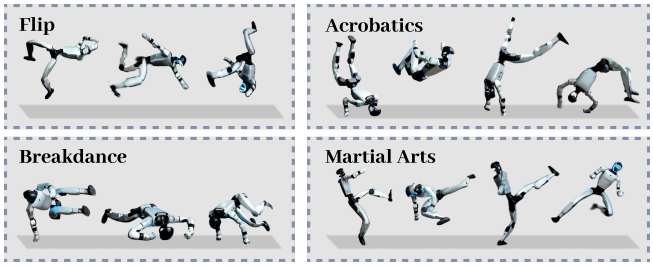

Fig. 5: **Qualitative results.** Representative real-world rollouts produced by OMNIXTREME, executing qualitatively distinct whole-body motions across diverse styles and contact patterns, including flips, acrobatics, breakdance, and martial-arts behaviors. The results illustrate stable and coordinated execution under rapid contact transitions and timing-sensitive phases on physical hardware.

the quantitative metrics in Q1-Q4 and illustrate the breadth of behaviors enabled by scalable generative pretraining and actuation-aware refinement. Please refer to the *Supp.* for additional qualitative results, including video demonstrations.

V. CONCLUSION

We presented OMNIXTREME, a two-stage framework for scalable high-fidelity humanoid motion tracking in high-dynamic regimes. By combining specialist-to-unified flow-based pretraining with actuation-aware residual reinforcement learning, OMNIXTREME mitigates both the learning bottleneck at scale and the physical executability bottleneck at sim-to-real deployment. Extensive simulation results show that OMNIXTREME preserves tracking fidelity substantially deeper into motion diversity than other baselines, and real-robot experiments demonstrate reliable execution of diverse extreme behaviors with a single unified policy, breaking the conventional fidelity–scalability trade-off.

For future research, jointly scaling data diversity and model capacity will be essential for enhancing the generalization of whole-body humanoid motor skills. As learning-based

controllers are pushed toward more dynamic and hardware-constrained regimes, actuation-aware modeling becomes a critical component of the learning pipeline. By incorporating high-fidelity actuation characteristics—such as current, power, torque, and speed-dependent constraints—researchers can further bridge the sim-to-real gap, ensuring that learned behaviors translate seamlessly to physical humanoid robots.

REFERENCES

[1] Arthur Allshire, Hongsuk Choi, Junyi Zhang, David McAllister, Anthony Zhang, Chung Min Kim, Trevor Darrell, Pieter Abbeel, Jitendra Malik, and Angjoo Kanazawa. Visual imitation enables contextual humanoid control. *arXiv preprint arXiv:2505.03729*, 2025.

[2] Lars Ankile, Anthony Simeonov, Idan Shenfeld, Marcel Torne, and Pulkit Agrawal. From imitation to refinement-residual rl for precise assembly. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 01–08. IEEE, 2025.

[3] Joao Pedro Araujo, Yanjie Ze, Pei Xu, Jiajun Wu, and C. Karen Liu. Retargeting matters: General motion retargeting for humanoid motion tracking. *arXiv preprint arXiv:2510.02252*, 2025.

[4] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pre-trained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023.

[5] Zixuan Chen, Mazeyu Ji, Xuxin Cheng, Xuanbin Peng, Xue Bin Peng, and Xiaolong Wang. Gmt: General motion tracking for humanoid whole-body control. *arXiv preprint arXiv:2506.14770*, 2025.

[6] Xuxin Cheng, Yandong Ji, Junming Chen, Ruihan Yang, Ge Yang, and Xiaolong Wang. Expressive whole-body control for humanoid robots. *arXiv preprint arXiv:2402.16796*, 2024.

[7] Xuxin Cheng, Kexin Shi, Ananye Agarwal, and Deepak Pathak. Extreme parkour with legged robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 11443–11450. IEEE, 2024.

[8] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *International Journal of Robotics Research (IJRR)*, page 02783649241273668, 2023.

[9] Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. Humanplus: Humanoid shadowing and imitation from humans. In *Conference on Robot Learning (CoRL)*, 2024.

[10] Felix G Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher J Pal. Robust motion in-betweening. *ACM Transactions on Graphics (TOG)*, 39(4), 2020.

[11] Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *arXiv preprint arXiv:2406.08858*, 2024.

[12] Tairan He, Zhengyi Luo, Wenli Xiao, Chong Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Learning human-to-humanoid real-time whole-body teleoperation. *arXiv preprint arXiv:2403.04436*, 2024.

[13] Tairan He, Chong Zhang, Wenli Xiao, Guanqi He, Changliu Liu, and Guanya Shi. Agile but safe: Learning collision-free high-speed legged locomotion. In *Robotics: Science and Systems (RSS)*, 2024.

[14] Tairan He, Jiawei Gao, Wenli Xiao, Yuanhang Zhang, Zi Wang, Jiashun Wang, Zhengyi Luo, Guanqi He, Nikhil Sobanbab, Chaoyi Pan, et al. Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills. *arXiv preprint arXiv:2502.01143*, 2025.

[15] Xialin He, Runpei Dong, Zixuan Chen, and Saurabh Gupta. Learning getting-up policies for real-world humanoid robots. *arXiv preprint arXiv:2502.12152*, 2025.

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 33: 6840–6851, 2020.

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

[18] Siyuan Huang, Zan Wang, Puhao Li, Baoxiong Jia, Tengyu Liu, Yixin Zhu, Wei Liang, and Song-Chun Zhu. Diffusion-based generation, optimization, and planning in 3d scenes. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[19] Tao Huang, Junli Ren, Huayi Wang, Zirui Wang, Qingwei Ben, Muning Wen, Xiao Chen, Jianan Li, and Jiangmiao Pang. Learning humanoid standing-up control across diverse postures. *arXiv preprint arXiv:2502.08378*, 2025.

[20] Xiaoyu Huang, Yufeng Chi, Ruofeng Wang, Zhongyu Li, Xue Bin Peng, Sophia Shao, Borivoje Nikolic, and Koushil Sreenath. Diffuseloco: Real-time legged locomotion control with diffusion from offline datasets, 2024. URL https://arxiv.org/abs/2404.19264.

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

[22] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

[23] Mazeyu Ji, Xuanbin Peng, Fangchen Liu, Jialong Li, Ge Yang, Xuxin Cheng, and Xiaolong Wang. Ex-body2: Advanced expressive humanoid whole-body control. *arXiv preprint arXiv:2412.13196*, 2024.

[24] Donghyun Kim, Jared Di Carlo, Benjamin Katz, Gerardo Bledt, and Sangbae Kim. Highly dynamic quadruped

locomotion via whole-body impulse control and model predictive control. *arXiv preprint arXiv:1909.06586*, 2019.

[25] Hyeongjun Kim, Hyunsik Oh, Jeongsoo Park, Yunho Kim, Donghoon Youm, Moonkyu Jung, Minho Lee, and Jemin Hwangbo. High-speed control and navigation for quadrupedal robots on complex and discrete terrain. *Science Robotics*, 10(102):eads6192, 2025.

[26] Yixuan Li, Yutang Lin, Jieming Cui, Tengyu Liu, Wei Liang, Yixin Zhu, and Siyuan Huang. Clone: Closed-loop whole-body humanoid teleoperation for long-horizon tasks. In *Conference on Robot Learning (CoRL)*, 2025.

[27] Zhongyu Li, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control. *International Journal of Robotics Research (IJRR)*, 44(5):840–888, 2025.

[28] Qiayuan Liao, Takara E Truong, Xiaoyu Huang, Yuman Gao, Guy Tevet, Koushil Sreenath, and C Karen Liu. Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion. *arXiv preprint arXiv:2508.08241*, 2025.

[29] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2023.

[30] Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Sirui Chen, Fernando Castañeda, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, Xingye Da, Runyu Ding, Cyrus Hogg, Lina Song, Edy Lim, Eugene Jeong, Tairan He, Haoru Xue, Wenli Xiao, Zi Wang, Simon Yuen, Jan Kautz, Yan Chang, Umar Iqbal, Linxi Fan, and Yuke Zhu. Sonic: Supersizing motion tracking for natural humanoid whole-body control. *arXiv preprint arXiv:2511.07820*, 2025.

[31] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019.

[32] Gabriel B Margolis, Ge Yang, Kartik Paigwar, Tao Chen, and Pulkit Agrawal. Rapid locomotion via reinforcement learning. *International Journal of Robotics Research (IJRR)*, 43(4):572–587, 2024.

[33] Chaoyi Pan, Giri Anantharaman, Nai-Chieh Huang, Claire Jin, Daniel Pfrommer, Chenyang Yuan, Frank Permenter, Guannan Qu, Nicholas Boffi, Guanya Shi, et al. Much ado about noising: Dispelling the myths of generative robotic control. *arXiv preprint arXiv:2512.01809*, 2025.

[34] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2023.

[35] Xue Bin Peng. Mimickit: A reinforcement learning framework for motion imitation and control. *arXiv preprint arXiv:2510.13794*, 2025.

[36] Ilija Radosavovic, Sarthak Kamat, Trevor Darrell, and Jitendra Malik. Learning humanoid locomotion over challenging terrain. *arXiv preprint arXiv:2410.03654*, 2024.

[37] Reallusion. 3d animation and 2d cartoons made simple, 2022. http://www.reallusion.com.

[38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[39] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

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

[41] Young-Ha Shin, Tae-Gyu Song, Gwanghyeon Ji, and Hae-Won Park. Actuator-constrained reinforcement learning for high-speed quadrupedal locomotion. *arXiv preprint arXiv:2312.17507*, 2023.

[42] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[43] Zhi Su, Bike Zhang, Nima Rahmanian, Yuman Gao, Qiayuan Liao, Caitlin Regan, Koushil Sreenath, and S Shankar Sastry. Hitter: A humanoid table tennis robot via hierarchical planning and learning. *arXiv preprint arXiv:2508.21043*, 2025.

[44] Chen Tessler, Yunrong Guo, Ofir Nabati, Gal Chechik, and Xue Bin Peng. Maskedmimic: Unified physics-based character control through masked motion inpainting. *ACM Transactions on Graphics (TOG)*, 2024.

[45] Chen Tessler, Yifeng Jiang, Erwin Coumans, Zhengyi Luo, Gal Chechik, and Xue Bin Peng. Maskedmanipulator: Versatile whole-body manipulation. In *ACM SIGGRAPH Asia Conference Proceedings*, 2025.

[46] Yuxuan Wang, Ming Yang, Ziluo Ding, Yu Zhang, Weishuai Zeng, Xinrun Xu, Haobin Jiang, and Zongqing Lu. From experts to a generalist: Toward general whole-body control for humanoid robots. *arXiv preprint arXiv:2506.12779*, 2025.

[47] Haoyang Weng, Yitang Li, Nikhil Sobanbabu, Zihan Wang, Zhengyi Luo, Tairan He, Deva Ramanan, and Guanya Shi. Hdmi: Learning interactive humanoid whole-body control from human videos. *arXiv preprint arXiv:2509.16757*, 2025.

[48] Zhou Xian and Nikolaos Gkanatsios. Chaineddiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation. In *Conference on Robot Learning (CoRL)*. Proceedings of Machine Learning Research, 2023.

[49] Weiji Xie, Jinrui Han, Jiakun Zheng, Huanyu Li, Xinzhe

Liu, Jiyuan Shi, Weinan Zhang, Chenjia Bai, and Xuelong Li. Kungfubot: Physics-based humanoid whole-body control for learning highly-dynamic skills. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=LCPoXt0pzm.

[50] Lujie Yang, Xiaoyu Huang, Zhen Wu, Angjoo Kanazawa, Pieter Abbeel, Carmelo Sferrazza, C Karen Liu, Rocky Duan, and Guanya Shi. Omniretarget: Interaction-preserving data generation for humanoid whole-body loco-manipulation and scene interaction. *arXiv preprint arXiv:2509.26633*, 2025.

[51] Shaofeng Yin, Yanjie Ze, Hong-Xing Yu, C Karen Liu, and Jiajun Wu. Visualmimic: Visual humanoid loco-manipulation via motion tracking and generation. *arXiv preprint arXiv:2509.20322*, 2025.

[52] Runyi Yu, Yinhuai Wang, Qihan Zhao, Hok Wai Tsui, Jingbo Wang, Ping Tan, and Qifeng Chen. Skillmimic-v2: Learning robust and generalizable interaction skills from sparse and noisy demonstrations. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–11, 2025.

[53] Xiu Yuan, Tongzhou Mu, Stone Tao, Yunhao Fang, Mengke Zhang, and Hao Su. Policy decorator: Model-agnostic online refinement for large policy model. *arXiv preprint arXiv:2412.13630*, 2024.

[54] Yanjie Ze, Zixuan Chen, Joao Pedro Araújo, Zi-ang Cao, Xue Bin Peng, Jiajun Wu, and C Karen Liu. Twist: Teleoperated whole-body imitation system. *arXiv preprint arXiv:2505.02833*, 2025.

[55] Chong Zhang, Nikita Rudin, David Hoeller, and Marco Hutter. Learning agile locomotion on risky terrains. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11864–11871. IEEE, 2024.

[56] Tong Zhang, Boyuan Zheng, Ruiqian Nai, Yingdong Hu, Yen-Jen Wang, Geng Chen, Fanqi Lin, Jiongye Li, Chuye Hong, Koushil Sreenath, et al. Hub: Learning extreme humanoid balance. *arXiv preprint arXiv:2505.07294*, 2025.

[57] Zhikai Zhang, Jun Guo, Chao Chen, Jilong Wang, Chenghuai Lin, Yunrui Lian, Han Xue, Zhenrong Wang, Maoqi Liu, Jiangran Lyu, et al. Track any motions under any disturbances. *arXiv preprint arXiv:2509.13833*, 2025.

[58] Siheng Zhao, Yanjie Ze, Yue Wang, C Karen Liu, Pieter Abbeel, Guanya Shi, and Rocky Duan. Resmimic: From general motion tracking to humanoid whole-body loco-manipulation via residual learning. *arXiv preprint arXiv:2510.05070*, 2025.

[59] Zhe Zhao, Haoyu Dong, Zhengmao He, Yang Li, Xinyu Yi, and Zhibin Li. Closing the reality gap: Zero-shot sim-to-real deployment for dexterous force-based grasping and manipulation. *arXiv e-prints*, pages arXiv–2601, 2026.

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

[61] Ziwen Zhuang, Shenzhe Yao, and Hang Zhao. Humanoid parkour learning. In *Conference on Robot Learning (CoRL)*, 2024. URL https://openreview.net/forum?id=fs7ia3FqUM.

APPENDIX

A. XtremeMotion Motion Library

The challenging motions selected from the AMASS dataset, Reallsusion, MimicKit, and LAFAN1 are detailed in Tab. V.

B. Motion Complexity Metrics

Given a motion sequence with frame index $t \in \{0, \dots, T-1\}$, frame time-step Δt, joint positions $q_t \in \mathbb{R}^J$, joint velocities $\dot{q}_t \in \mathbb{R}^J$, root/body positions $x_t \in \mathbb{R}^{B\times 3}$ and velocities $v_t \in \mathbb{R}^{B\times 3}$, we compute a set of scalar metrics that summarize its kinematic and contact complexity.

a) Joint/base kinematics.: From joint or base kinematics we extract:

$$v_{\max} = \max_{t,j} \left\|\dot{q}_t^{(j)}\right\|, \tag{7}$$

$$a_{\max} = \max_{t,j} \left\|\ddot{q}_t^{(j)}\right\|, \tag{8}$$

$$j_{\max} = \max_{t,j} \left\|\dddot{q}_t^{(j)}\right\|, \tag{9}$$

where $\ddot{q}_t$ and $\dddot{q}_t$ are obtained by finite differencing, e.g., $\ddot{q}_t \approx (\dot{q}_{t+1} - \dot{q}_t)/\Delta t$ and $\dddot{q}_t \approx (\ddot{q}_{t+1} - \ddot{q}_t)/\Delta t$. We use base (torso) linear and angular kinematics to define the same three scalars (maximum linear velocity, linear acceleration and angular velocity).

b) Center-of-mass vertical velocity.: Let z_t^{CoM} be the center-of-mass height and $\dot{z}_t^{\mathrm{CoM}}$ its vertical velocity. We measure the peak vertical CoM speed

$$v_{z,\max}^{\mathrm{CoM}} = \max_t \left|\dot{z}_t^{\mathrm{CoM}}\right|. \tag{10}$$

This captures the dynamic "liveliness" of the motion (e.g., jumps).

c) Airborne ratio.: Given a set of foot body indices $\mathcal{F}$, a frame is considered *airborne* if all feet are above a small height threshold h_{air} above the ground. The airborne ratio is the fraction of airborne frames

$$\text{Airborne} = \frac{1}{T}\sum_{t=0}^{T-1} \mathbf{1}\left[\min_{b\in\mathcal{F}} z_t^{(b)} > h_{\text{air}}\right], \tag{11}$$

where $z_t^{(b)}$ denotes the vertical position of body b at time t.

d) Contact switch frequency.: Independently, we obtain a binary contact state $c_t \in \{0,1\}^K$ for K end-effectors and define the contact switch frequency as the number of contact state flips per second:

$$f_{\text{switch}} = \frac{1}{(T-1)\,\Delta t}\sum_{t=0}^{T-2} \mathbf{1}[c_{t+1} \neq c_t] \ \ [\text{Hz}]. \tag{12}$$

In the radar plots, the *contact switch* axis corresponds to a normalized version of this scalar, $s_{\text{sw}} = \min(f_{\text{switch}}/10, 1)$.

e) Difficulty scores.: For comparability across metrics with different physical units we map the main scalar metrics to $[0,1]$ "difficulty" scores using simple clamping and linear scaling, mirroring the implementation used in our analysis code. Let

$$s_{ang} = \min\left(\frac{ang_{\max}}{20}, 1\right), \tag{13}$$

$$s_v = \min\left(\frac{v_{\max}}{20}, 1\right), \tag{14}$$

$$s_a = \min\left(\frac{a_{\max}}{200}, 1\right), \tag{15}$$

$$s_{\text{air}} = \text{clip}\big(\text{Airborne}, 0, 1\big), \tag{16}$$

$$s_{\text{com}} = \min\left(\frac{v_{z,\max}^{\mathrm{CoM}}}{2}, 1\right), \tag{17}$$

$$s_{\text{sw}} = \min\left(\frac{f_{\text{switch}}}{10}, 1\right), \tag{18}$$

where $\text{clip}(x,0,1) = \max(0, \min(x,1))$. The resulting 6-D score vector

$$\mathbf{s} = \left[s_{ang}, s_v, s_a, s_{\text{com}}, s_{\text{air}}, s_{\text{sw}}\right] \tag{19}$$

is what we use in the radar plots to summarize and compare motion complexity across datasets.

C. Teacher Training

The teacher policy is trained using BeyondMimic [28]——a RL motion tracking algorithm within the IsaacLab simulator, with the specific reward constituents detailed in Tab. VII. Regarding the robot configuration, a full-mesh representation is employed for the collision geometry to ensure high-fidelity physical interactions. System parameters—including armature, k_p, k_d, and action scale—are kept consistent with the BeyondMimic framework, which is specifically tailored to the motor specifications of each joint.

The Teacher Policy employs an MLP architecture with hidden dimensions of [512, 256, 128] for both actor and critic. The actor observation is defined by

$$\mathbf{o} = [\boldsymbol{\psi}, \mathbf{e}_{\text{torso}}, \mathcal{V}_{\text{imu}}, \boldsymbol{q} - \boldsymbol{q}^0, \dot{\boldsymbol{q}}, \mathbf{a}_{\text{last}}],$$

including reference motion joint positions and velocities $\boldsymbol{\psi} = [\boldsymbol{q}^{\text{ref}}, \dot{\boldsymbol{q}}^{\text{ref}}]$, $\mathbf{e}_{\text{torso}}$ which includes torso position difference and torso orientation difference from reference represented by the 6D orientation from $R_{\text{torso}}^{\text{ref}} R_{\text{torso}}^{\top}$, base linear and angular velocities $\mathcal{V}_{\text{imu}} \in \mathbb{R}^6$, relative joint positions $\boldsymbol{q} - \boldsymbol{q}^0$ computed by joint position $\boldsymbol{q}$ subtract default joint position $\boldsymbol{q}^0$ and joint velocities $\dot{\boldsymbol{q}}$, and the previous action $\mathbf{a}_{\text{last}}$, all of which are subject to noise described in Tab. I to enhance robustness. In contrast, the critic observation includes these same elements in their ground-truth form, supplemented by privileged information such as the robot's precise body position and orientation, providing a comprehensive and noise-free state representation to facilitate stable value function estimation during training.

The detailed training parameters are shown in Tab. VI. To speed up training, early termination is introduced when tracking errors is larger than a threshold. Tracking errors are defined as $\mathbf{e}_{p,b} = \mathbf{p}_b^{\text{ref}} - \mathbf{p}_b$ for position and $\mathbf{e}_{R,b} = \log(R_b^{\text{ref}} R_b^{\top})$ for

TABLE V: Motion information summary.

Motion ID	Motion Source	Motion Description
1	CMU-85-05	Handstand walk.
2	CMU-85-10	Handstand spin.
3	CMU-88-09	Back Handspring with a full twist.
4	CMU-88-08	Back Handspring with a half twist.
5	CMU-90-08	Aerial Cartwheel
6	CMU-85-14	B-boying with a rapid-fire string of back handsprings.
7	CMU-85-13	Skip back, pivot through a hand-to-headstand, drop, flip, and bounce back.
8	CMU-90-06	Fly kick.
9	CMU-90-34	Forward roll.
10	CMU-90-01	Backward roll.
11	CMU-90-28	Backspin.
12	CMU-85-08	Thomas Flare.
13	CMU-85-12	Long breaking dance.
14	CMU-85-04	Another long breaking dance.
15	CMU-85-01	Bicycle kick flip.
16	CMU-85-02	Another bicycle kick flip.
17	CMU-88-06	Butterfly kick.
18	CMU-85-06	Webster flip.
19	CMU-49-08	Two consecutive cartwheels.
20	CMU-90-30	Alternating pistol squats.
21	CMU-90-29	Acrobatic gymnastics with cartwheel and back handsprings.
22	CMU-90-19	Crawl forward and backflip.
23	GeneralA11-MilitaryCrawlForward	Stay low and crawl forward.
24	IconicHeroMotion-SwordJudgment	Spinning slash.
25	IconicHeroMotion-SwordHeroic	Another style of spinning slash.
26	HandtoHandCombat-B3AttackReverseTurningKick	Reverse turning kick.
27	HandtoHandCombat-D2AttackPunchSweepKick	Punch and sweep kick.
28	HandtoHandCombat-D4AttackReverseFrontSnapKick	Reverse and front snap kick.
29	HandtoHandCombat-D4DodgeRollBack	Two consecutive rolls in different styles.
30	HandtoHandCombat-G1GetupKipUp	Kip up.
31	HandtoHandCombat-G2GetupHandstandKipUp	Handstand kip.
32	HandtoHandCombat-KO2FalltoGroundAxelDown	Execute a downward Axel into a ground fall.
33	LadyAgent-AgentElbowStrikeSweepKick	Elbow strike and sweep kick.
34	LadyAgent-AgentHandspring	Front handspring.
35	LadyAgent-AgentRollForward	Shoulder roll.
36	LadyAgent-AgentShootSidewardRoll	Quick side roll.
37	LadyAgent-AgentSnapKick	Snapkick.
38	Mimickit-g1spinkick	Spinkick.
39	LAFAN1-dance1subject2 [82.8,106.9]s	Constantly spin in full circles.
40	LAFAN1-dance1subject2 [145.3,161.3]s	Play guitar while hopping on one leg.
41	LAFAN1-dance2subject3 [160.2,224.3]s	Flutter arms and hands.
42	LAFAN1-fightandsports1subject1 [167.0,176.9]s	Continuous long jumps.
43	LAFAN1-fightandsports1subject1 [0.0,17.0]s	Balance on one leg.
44	LAFAN1-dance1subject1 [104.6,119.1]s	Cartwheel twice.
45	LAFAN1-jump1subject1 [70.3,87.6]s	Play hopscotch.
46	LAFAN1-jump1subject1 [89.1,138.4]s	Hop on one foot.
47	LAFAN1-jump1subject1 [0.0,72.0]s	Successive leaps.
48	LAFAN1-fightandsports1subject4 [154.4,219.8]s	Vigorously swing the baseball bat.
49	LAFAN1-fightandsports1subject4 [61.1,85.9]s	Lash the golf club.
50	LAFAN1-fightandsports1subject4 [28.2,61.9]s	Diverse kicking movements.
51	LAFAN1-run1subject2 [20.4,101.0]s	Shuttle run.
52	LAFAN1-run1subject2 [92.0,130.6]s	Run rapidly.
53	LAFAN1-fallandgetup2subject3 [33.8,56.1]s	Kip up twice.
54	LAFAN1-fightandsports1subject1 [17.3,27.2]s	Roundhouse kick.
55	LAFAN1-jump1subject2 [187.9,196.2]s	Push up.
56	LAFAN1-jump1subject2 [196.0,205.8]s	Lateral roll.
57	LAFAN1-jump1subject2 [205.6,244.4]s	Lateral roll and kip up.

orientation. Termination is triggered if the vertical position error $|\mathbf{e}_{p,z,b}|$ of the torso or any end-effector ($\mathcal{B}_{\text{ee}}$) exceeds 0.25m, or if the local gravity vector discrepancy caused by torso's orientation error $\|\mathbf{e}_{R,\text{torso}}\|$ exceeds 0.8.

TABLE VI: **Motion tracking hyperparameters for teacher training** [28].

Hyperparameter	Value
Architecture	
Actor MLP hidden dimensions	[512, 256, 128]
Critic MLP hidden dimensions	[512, 256, 128]
Activation function	ELU
Training	
Steps per environment	24
Clip parameter	0.2
Entropy coefficient	0.005
Value loss coefficient	1.0
Discount factor (γ)	0.99
GAE λ	0.95
Desired KL	0.01
Learning epochs	5
Mini-batches	4

D. Flow matching policy training

The inputs robot proprioception p and motion command c are each mapped to a single token, while the 15-step history h is mapped to 15 individual tokens, resulting in a concatenated sequence of 17 tokens. Each mapping is performed by a dedicated MLP preprocessing unit consisting of two 256-unit layers with ReLU activations. This sequence is then processed by a Transformer encoder, followed by global average pooling and a linear projection to produce a 1024-dimensional embedding.

Simultaneously, the action a_t is projected through a linear layer to a 1024-dimensional space, while the time step t is transformed into a 1024-dimensional embedding using sinusoidal encoding. These three components—the state embedding, action embedding, and time embedding—are concatenated and fed into a deep MLP with hidden dimensions of $[2048, 2048, 2048]$ to predict the velocity v_t at time t.

E. Residual policy training

The Residual Policy is implemented as an MLP with hidden layer sizes of [1024, 512, 256, 128, 64], with the corresponding reward functions detailed in Tab. VIII.

The residual learning framework employs an asymmetric actor-critic architecture, as well. In this setup, the residual actor's observation space is restricted to proprioceptive data, relative motion torso orientation, motion command joint pos and joint velocity, the previous step's total action, and the base action out by the flow matching policy. To enhance sim-to-real robustness, uniform noise is injected into these observations, and the noise is aligned with the flow matching base policy noise.

Conversely, the residual critic utilizes privileged information available only during simulation. The critic's observation space encompasses the full ground-truth state, including noise-free versions of the actor's inputs, as well as critical state variables hidden from the actor, such as the robot's global body position and orientation, base linear velocity, and the precise position of the motion torso.

TABLE VII: Reward function terms and expressions for Teacher Policy training [28].

Reward Term	**Weight**	**Expression**
Global Torso Position	0.5	$\exp(-\|p_{err}\|^2/0.3^2)$
Global Torso Orientation	0.5	$\exp(-\|\theta_{err}\|^2/0.4^2)$
Relative Body Position	1.0	$\exp(-\|p_{rel_err}\|^2/0.3^2)$
Relative Body Orientation	1.0	$\exp(-\|\theta_{rel_err}\|^2/0.4^2)$
Body Linear Velocity	1.0	$\exp(-\|v_{err}\|^2/1.0^2)$
Body Angular Velocity	1.0	$\exp(-\|\omega_{err}\|^2/3.14^2)$
Action Rate	-0.1	$\|a_t - a_{t-1}\|^2$
Joint Limit	-10.0	$\sum \max(0, q - q_{limit})$
Undesired Contacts	-0.1	$\sum 1(F_{contact} > 1.0)$

TABLE VIII: Reward function terms and expressions for Residual Policy training. Refer to Sec. V-K for further details regarding power safety regularization.

Reward Term	**Weight**	**Expression**
Global Torso Position	0.5	$\exp(-\|p_{err}\|^2/0.3^2)$
Global Torso Orientation	0.5	$\exp(-\|(\theta_{err}\|^2/0.4^2)$
Relative Body Position	1.0	$\exp(-\|p_{rel_err}\|^2/0.3^2)$
Relative Body Orientation	1.0	$\exp(-\|\theta_{rel_err}\|^2/0.4^2)$
Body Linear Velocity	1.0	$\exp(-\|v_{err}\|^2/1.0^2)$
Body Angular Velocity	1.0	$\exp(-\|\omega_{err}\|^2/3.14^2)$
Action Rate	-0.1	$\|a_t - a_{t-1}\|^2$
Joint Limit	-10.0	$\sum \max(0, q - q_{limit})$
Undesired Contacts	-0.1	$\sum 1(F_{contact} > 1.0)$
Power Safety Regularization	-10.0	$\sum \left(\frac{\max(0,\ -\tau\dot{q}-150)}{500}\right)^2$

F. Actuator Modeling

The physical parameters of the actuators, including armature inertia I, torque-speed characteristics, and friction coefficients, are summarized in Tab. IX.

TABLE IX: **Actuator modeling parameters.**

Actuator	τ_{y1}	τ_{y2}	v_{x1}	v_{x2}	μ_s	v_{act}	μ_d	I
5020-16	24.8	31.9	30.86	40.13	0.6	0.01	0.06	3.610e-03
7520-14.3	71.0	83.3	22.63	35.52	1.6	0.01	0.16	1.018e-02
7520-22.5	111.0	131.0	14.5	22.7	2.4	0.01	0.24	2.510e-02
4010-25	4.8	8.6	15.3	24.76	0.6	0.01	0.06	4.250e-03

G. PD control

Following the methodology of BeyondMimic, we calculate the PD gains, k_p and k_d, based on the natural frequency (ω), damping ratio (ζ), and the armature inertia (I) of the motors:

$$k_{\text{p},j} = I_j\omega^2, \quad k_{\text{d},j} = 2I_j\zeta\omega$$

For the configuration addressed here, the parameters consist of a 10 Hz natural frequency and a damping ratio of 2. To determine the action scale, we still utilize the maximum torque

TABLE X: Unitree G1 Joint-to-Motor Mapping.

Joint Name	Motor Model
Hip Pitch Joint	7520-22.5
Hip Roll Joint	7520-22.5
Knee Joint	7520-22.5
Hip Yaw Joint	7520-14.3
Ankle Pitch Joint	5020
Ankle Roll Joint	5020
Waist Roll Joint	5020
Waist Pitch Joint	5020
Waist Yaw Joint	7520-14.3
Shoulder Pitch Joint	5020
Shoulder Roll Joint	5020
Shoulder Yaw Joint	5020
Elbow Joint	5020
Wrist Roll Joint	5020
Wrist Pitch Joint	4010
Wrist Yaw Joint	4010

values ($\boldsymbol{\tau}_{\text{max}}$) originally defined in the URDF, scaled by a factor of 0.25:

$$\boldsymbol{\alpha} = 0.25\frac{\boldsymbol{\tau}_{\text{max}}}{k_{\text{p},j}}$$

Notably, due to differences in the G1 robot variants, we utilize 7520-22.5 motors for the hip pitch actuators(Tab. X), rather than the 7520-14.3 models used in the original BeyondMimic.

The PD setpoints (target positions) are computed using the following relationship:

$$\boldsymbol{q}^{\text{tar}} = \boldsymbol{q}^0 + \boldsymbol{\alpha} \odot \mathbf{a}$$

where $\boldsymbol{q}^0$ represents the default joint positions, and $\mathbf{a}$ is the action output by the policy. The pre-clipped control torque is then calculated using the standard PD control law:

$$\boldsymbol{\tau} = k_p(\boldsymbol{q}^{\text{tar}} - \boldsymbol{q}) - k_d\dot{\boldsymbol{q}}$$

In this expression, $\mathbf{q}$ and $\dot{\mathbf{q}}$ denote the current joint positions and velocities, respectively.

H. Evaluation Protocol and Metrics

This subsection details the evaluation protocol and metrics used throughout the paper. Our goal is to assess tracking fidelity, robustness, and generalization under controlled conditions that are consistent with training and reflective of sim-to-real performance.

a) Motion segmentation and episode length.: To avoid bias introduced by varying motion durations, all reference motions are segmented into fixed-length clips of **10 seconds** (corresponding to **500 control steps**), which matches the episode length used during training. Motions shorter than 10 seconds are evaluated without further segmentation. All reported metrics in simulation and real world are computed at the clip level.

b) Simulation evaluation settings.: During simulation evaluation, we retain the same sensor noise and base domain randomization used during training, including center-of-mass offsets and default joint position offsets. This ensures that evaluation reflects basic sim-to-real robustness rather than idealized noise-free tracking. Each motion clip is evaluated over **10 independent rollouts**, and metrics are averaged across these trials.

c) Unseen motion set.: In addition to test motions drawn from the training distributions, we evaluate generalization on an unseen motion set. This set is constructed by uniformly sampling **1000** motion clips (10 seconds each) from retargeted AMASS CMU and KIT mocap sequences, explicitly excluding all motions used in training or selected as extreme motions. The unseen set primarily consists of locomotion, turning, and simple dance-like behaviors, and is used to assess generalization beyond the curated extreme motion library.

d) Evaluation metrics.: We employ a set of pose-based and physics-aware metrics commonly used in humanoid motion tracking. All metrics are computed per episode and then averaged across episodes.

Let $t = 1, \ldots, T$ index control steps in an episode with step time Δt (seconds), and let $i = 1, \ldots, N$ index tracked bodies. All per-body errors are computed over the same tracked-body subset.

e) Root-relative body positions.: Reference motions are aligned to the robot using the torso pose. We denote the aligned reference body position as $\mathbf{p}_{t,i}^{\text{ref}}$ and the robot body position as $\mathbf{p}_{t,i}^{\text{rob}}$.

f) MPJPE (mm).:

$$\text{MPJPE} = 1000 \cdot \frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{p}_{t,i}^{\text{ref}} - \mathbf{p}_{t,i}^{\text{rob}}\right\|_2\right). \quad (20)$$

g) Δvel (mm/frame).: Let $\mathbf{v}_{t,i}^{\text{ref}}$ and $\mathbf{v}_{t,i}^{\text{rob}}$ denote linear velocities (in m/s).

$$\Delta v = 1000 \cdot \Delta t \cdot \frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{v}_{t,i}^{\text{ref}} - \mathbf{v}_{t,i}^{\text{rob}}\right\|_2\right). \quad (21)$$

h) Δacc (mm/frame^2).: Acceleration is computed by finite differences on velocity:

$$\mathbf{a}_{t,i} = \frac{\mathbf{v}_{t,i} - \mathbf{v}_{t-1,i}}{\Delta t}.$$

The per-step acceleration error is:

$$e_t^{\text{acc}} = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{a}_{t,i}^{\text{ref}} - \mathbf{a}_{t,i}^{\text{rob}}\right\|_2. \quad (22)$$

We exclude the step immediately following an environment reset. The reported acceleration error is:

$$\Delta a = 1000 \cdot \Delta t^2 \cdot \frac{1}{T}\sum_{t=1}^{T} e_t^{\text{acc}}. \quad (23)$$

i) Success rate.: An episode is considered successful if it terminates due to time-out (i.e., runs for its full allotted duration) rather than early termination. **The termination criteria used during evaluation are identical to those used during training**, including violation of safety thresholds. Success rate is defined as the fraction of successful episodes.

TABLE XI: **Skill-level grouping of real-world evaluation motions.** Each skill category groups multiple motion instances with similar semantic meaning and dynamic structure. Motion IDs correspond to retargeted motions used during hardware evaluation.

Skill category	Motion IDs
Flip	5, 9, 10, 15, 16, 18, 22
Handspring	2, 3, 4, 19, 23
Acrobatics	1, 6, 7, 20, 21
Breakdance	11, 12, 13, 14
Martial arts	8, 17, 38

j) Aggregation.: MPJPE, Δv, and Δa are first averaged over control steps within each episode. Final reported metrics are obtained by averaging over all evaluated episodes (and over motion clips when evaluating multiple motions).

I. Skill-level Grouping for Real-world Evaluation

For real-world evaluation, we organize individual motions into a set of *extreme skill categories*. Each skill groups multiple motion instances that share similar semantic intent and high-level dynamic structure. This grouping is used *only for reporting and analysis*, and does not affect training or policy execution.

The classification is based on human semantic understanding of the motions, taking into account characteristic body coordination patterns, contact configurations, and dominant dynamic features (e.g., aerial phases, rapid contact switching, or impulsive landing). We emphasize that this grouping is not obtained through automated clustering or learned representations, but serves as a transparent and interpretable abstraction for summarizing real-world performance.

Tab. XI lists the correspondence between skill categories and the underlying motion instances used in hardware evaluation.

J. Motion Subsets Used in Fidelity–Scalability Analysis

In the fidelity–scalability analysis Sec. IV-D, we evaluate how tracking performance degrades as the size of the motion library increases. To ensure a controlled and interpretable comparison, we explicitly specify the motion subsets used at each scale.

We begin with a curated set of extreme motions that exhibit high angular velocities, rapid contact switching, and challenging balance conditions. These motions represent a diverse yet highly demanding subset of the full motion library. All quantitative evaluations in Q2 are conducted on the same **first 10 motions** of this set, which include fast flips, contact-rich acrobatics, and extreme balance behaviors. By fixing the evaluation set, we isolate the effect of training-scale growth from changes in evaluation difficulty.

To study scalability, we progressively expand the training motion library to include 10, 20, and all extreme motions. The additional motions are drawn from the same extreme motion pool and increase diversity in contact patterns and dynamic regimes, while preserving a consistent difficulty level. This design yields a controlled yet representative benchmark for analyzing the fidelity–scalability trade-off under increasingly diverse extreme motions. Tab. XII summarizes the motion IDs used at each scale.

TABLE XII: **Motion subsets used in Q2 fidelity–scalability analysis.** All evaluations are performed on the same first 10 extreme motions. Larger training sets (20 and all motions) extend this core set with additional diverse extreme motions.

Training motions	Motion IDs
10	3–10, 13, 14
20	2–10, 13-22
50	All

K. Knee negative power penalty.

For each knee joint j with torque τ_j and joint velocity $\dot{q}_j$, we define the instantaneous mechanical power $P_j = \tau_j \dot{q}_j$ and penalize excessive *negative* power (large braking torques) beyond a deadband threshold. Concretely, for knee joints selected via the regex `".*_knee_joint"` we compute

$$\tilde{P}_j = \max\big(0,\ -P_j - 150\big), \tag{24}$$

where the deadband is 150 (in the same units as power, e.g., W). The normalized per-step penalty is

$$c_{\text{knee}} = \sum_{j \in \mathcal{J}_{\text{knee}}} \left(\frac{\tilde{P}_j}{500} \right)^2, \tag{25}$$

where 500 is the normalization constant `power_norm`. In the reward function this term is added with weight $w = -10$, i.e., $r_{\text{knee}} = w\, c_{\text{knee}}$, so that large negative joint power at the knees is strongly discouraged.

L. TensorRT-Accelerated Deployment

For deployment on the real G1 robot we run the model as an ONNX graph, executed by ONNX Runtime with TensorRT acceleration. At run time the deploy node continuously receives joint states and IMU signals from the robot, builds a compact observation vector, and feeds it to the ONNX base policy. The base policy outputs a 29-dimensional whole-body action, and, the residual policy produces a small corrective action in the same space; the two are added to obtain the final command. This command is interpreted as desired joint positions, which are tracked by a PD controller with safety envelopes on torques and velocities, and converted into low-level motor commands at a control period of $20\,\text{ms}$.

When a GPU with TensorRT is available, ONNX Runtime delegates the forward pass of these ONNX models to TensorRT, giving a lower-latency implementation of the same controller. In our implementation, we use **5 ODE integration steps** (often informally referred to as denoising steps), which strike a favorable balance between action quality and computational cost. All inference is executed on the onboard **NVIDIA Jetson Orin NX** of the Unitree G1 humanoid robot. With TensorRT acceleration, the end-to-end policy inference latency, including the full ODE integration procedure, is approximately

10 ms per control step. This latency comfortably satisfies real-time control requirements and enables closed-loop execution of high-dynamic whole-body motions entirely on the onboard compute, without reliance on offboard processing.

M. Failure Cases and Discussion

While OmniXtreme achieves strong sim-to-real performance overall, we observe a small number of failure cases during real-world deployment. These failures predominantly occur during highly impulsive landing phases of certain extreme motions, where large transient braking loads trigger hardware protection mechanisms, including motor overcurrent, power limits, or battery undervoltage events.

Notably, these motions can be executed reliably in simulation and sim-to-sim transfer, indicating that failures are not caused by degraded tracking accuracy or loss of balance. Instead, they expose residual discrepancies between simulated actuation models and the true hardware capability envelope under extreme dynamic conditions.

These observations suggest that further improving robustness for extreme motions will require more comprehensive modeling of real actuator and power-system limits, including the coupled effects of torque, speed, current, power flow, and battery voltage dynamics, which remain challenging to capture accurately in simulation. ·