SONIC: Supersizing Motion Tracking for Natural Humanoid Whole-Body Control

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 "Jim" Fan**[‡], **Yuke Zhu**[‡]

Nvidia

[†] Co-First Authors

[*] Core Contributors

[‡] Project Leads

https://nvlabs.github.io/SONIC/

Abstract

Despite the rise of billion-parameter foundation models trained across thousands of GPUs, similar scaling gains have not been shown for humanoid control. Current neural controllers for humanoids remain modest in size, target a limited behavior set, and are trained on a handful of GPUs over several days. We show that scaling up model capacity, data, and compute yields a generalist humanoid controller capable of creating natural and robust whole-body movements. Specifically, we posit motion tracking as a natural and scalable task for humanoid control, leverageing dense supervision from diverse motion-capture data to acquire human motion priors without manual reward engineering. We build a foundation model for motion tracking by scaling along three axes: network size (from 1.2M to 42M parameters), dataset volume (over 100M frames, 700 hours of high-quality motion data), and compute (9k GPU hours). Beyond demonstrating the benefits of scale, we show the practical utility of our model through two mechanisms: (1) a real-time universal kinematic planner that bridges motion tracking to downstream task execution, enabling natural and interactive control, and (2) a unified token space that supports various motion input interfaces, such as VR teleoperation devices, human videos, and vision-language-action (VLA) models, all using the same policy. Scaling motion tracking exhibits favorable properties: performance improves steadily with increased compute and data diversity, and learned representations generalize to unseen motions, establishing motion tracking at scale as a practical foundation for humanoid control.

1. Introduction

The past decade has witnessed a remarkable journey in artificial intelligence. The GPT (Achiam et al., 2023) family of models trains on 25,000+ GPUs with trillions of tokens. Video and image-generation models (Blattmann et al., 2023; Brooks et al., 2024; Ho et al., 2022; Ramesh et al., 2022; Rombach et al., 2022) leverage thousands of GPUs processing billions of images. These foundation models have shown a consistent pattern: scale unlocks emergent capabilities, generalization, and robustness that smaller models cannot achieve (Bommasani et al., 2021; Hoffmann et al., 2022; Kaplan et al., 2020; Wei et al., 2022). Yet for sim-to-real humanoid control, similar scaling gains have not been achieved. State-of-the-art humanoid control policies are typically small neural networks – often three-layer MLPs with a few million parameters – trained on a single GPU over a few days for a single task. Even more so, due to the manually engineered reward terms for each task, training a policy for too long may even lead to worse performance (Peng et al., 2018, 2021; Sutton, 2019).

Why hasn't humanoid control scaled? The fundamental issue is the task selection. Tasks like locomotion require extensive reward engineering for each scenario – walking naturally forward provides little signal for dancing

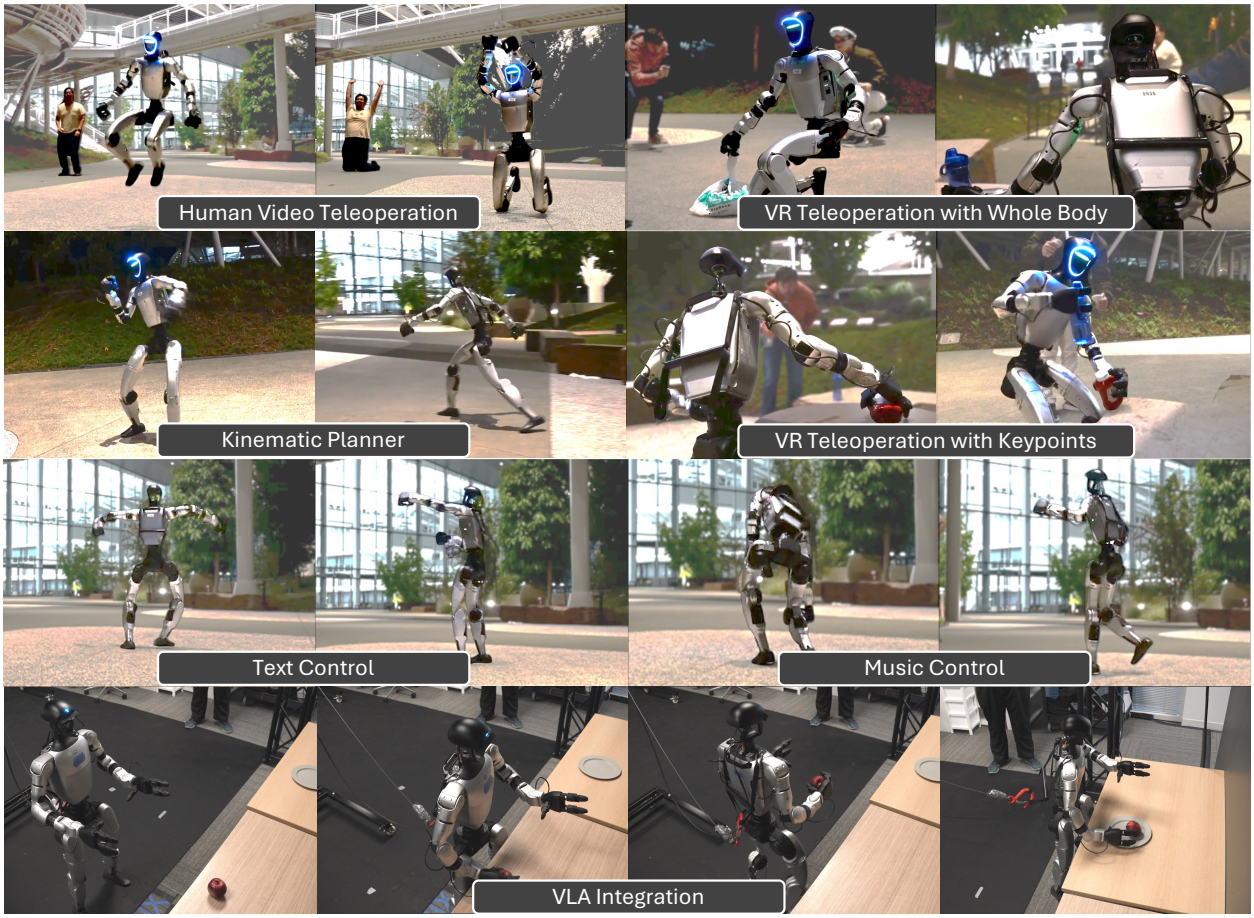

Figure 1: SONIC enables diverse humanoid tasks through a universal control policy that handles diverse input modalities and control interfaces.

(He et al., 2025), getting up from the ground (He et al., 2025; Huang et al., 2025), or teleoperation (Ben et al., 2025; Li et al., 2025; Ze et al., 2025). Each new capability demands redesigned rewards and objectives, making scaling up difficult. Even if we identify a scalable objective that can learn diverse behaviors, a second challenge emerges: how do we support the diverse range of real-world applications? A desired humanoid controller should handle teleoperation, goal-directed tasks, navigation, and even vision-language commands (Ahn et al., 2022; Brohan et al., 2022, 2023; Ma et al., 2024; Open X-Embodiment Collaboration, 2023). Building a system that scales while remaining flexible for different task specifications is non-trivial.

In this work, we address both challenges by identifying motion tracking as the scalable foundation task for humanoid control. Motion tracking leverages human motion capture data, which provides dense, frame-by-frame supervision without reward engineering. Critically, humanoids benefit from decades of motion capture research – datasets covering walking, running, dancing, sports, and object interactions already exist at scale (Li et al., 2021; Mahmood et al., 2019; Punnakkal et al., 2021). While there exists prior art in motion tracking (Chen et al., 2025; He et al., 2024, 2025; Liao et al., 2025; Luo et al., 2023; Wang et al., 2020; Yin et al., 2025; Zeng et al., 2025; Zhang et al., 2025), they are mostly limited to showing whole-body motion tracking results on **training data** and have not demonstrated many downstream tasks beyond motion tracking or navigation. We supersize physics-based motion tracking to unprecedented 100 million frames (at 50 fps) and 128 GPUs training, achieving universal tracking capabilities across diverse human behaviors while maintaining real-time performances. In addition, we show how such a motion tracker can be applied to meaningful downstream tasks, and introduce two key contributions. First, we develop a universal kinematic motion generation system

for interactive control, enabling goal-directed tasks such as interactive locomotion and game-like character control through kinematic planning in motion space. Second, we design a universal token space that supports multimodal control – accepting inputs from teleoperation, human videos, music, text, and vision-language-action models through a unified interface (Kocabas et al., 2020; Li et al., 2021; Tevet et al., 2022; Zhang et al., 2023). This unified framework allows our motion tracker to directly interface with vision-language-action (VLA) models (Bjorck et al., 2025) and diverse control modalities.

We propose *Supersizing mOtion tracking for Natural humanoId Control* (SONIC), a framework that enables natural humanoid control across a wide range of applications. We achieve high-precision teleoperation and interactive control capabilities, including running, jumping, and crawling with natural human-like movement. Leveraging our universal token space, our controller can support cross-embodiment motion tracking where estimated whole-body human motion can be directly mapped to humanoid control signals, bypassing the need for retargeting. We demonstrate seamless integration with multi-modal human motion generation models, supporting video, text, and music control through our universal token space. Furthermore, we show that teleoperation data collected through our system can be used to train vision-language-action foundation models, establishing a complete pipeline from motion tracking at scale to foundation model-based humanoid control. These results validate that massive-scale motion tracking serves as a practical and versatile behavioral foundation model for diverse real-world humanoid applications.

Our contributions are:

- We identify motion tracking as a scalable foundation task for humanoid control, demonstrating that it exhibits favorable scaling properties with both compute and data diversity and scale up humanoid control to 9k GPU hours and 100 million frames of motion sequences, achieving universal tracking capabilities across diverse human behaviors.
- We introduce a kinematic motion generation system for interactive control and a universal token space supporting multimodal inputs, including teleoperation, human videos, motion commands, and VLA foundation models, through single-stage training without distillation.
- We provide a comprehensive evaluation demonstrating empirical humanoid scaling trend, zero-shot transfer to unseen motions, robust sim-to-real deployment on physical humanoid robots, and successful integration with foundation models.

2. Results

We use the Unitree G1 humanoid (Unitree, 2024) to demonstrate our supersizing humanoid motion tracking framework, SONIC. Video results can be seen at our website. We demonstrate SONIC's ability for motion tracking (Sec. 2.1), teleoperation (Sec. 2.4), multi-modal control (Sec. 2.3), and interactive kinematic motion planning (Sec. 2.2), as seen in Fig. 1.

2.1. Motion Tracking

SONIC, trained on 100 million frames of motion over 32,000 GPU hours (128 GPUs over 3 days), exhibits remarkable generalization to **unseen** motions. In this section, we evaluate the generalization capabilities of our tracker on large-scale, previously unseen motion datasets in simulation and the real world.

Metrics. We employ a comprehensive set of pose-based and physics-based metrics to measure motion imitation performance. The primary measure is the success rate (Succ), where an imitation attempt is deemed unsuccessful if the humanoid deviates too far from the reference motion trajectory. We further report the root-relative mean per-joint position error (MPJPE) E_{mpjpe} (in mm), quantifying the local accuracy of the imitation. To assess physical fidelity, we also calculate differences in acceleration (E_{acc}, mm/frame^2) and velocity (E_{vel}, mm/frame) between the simulated humanoid and the reference human motion.

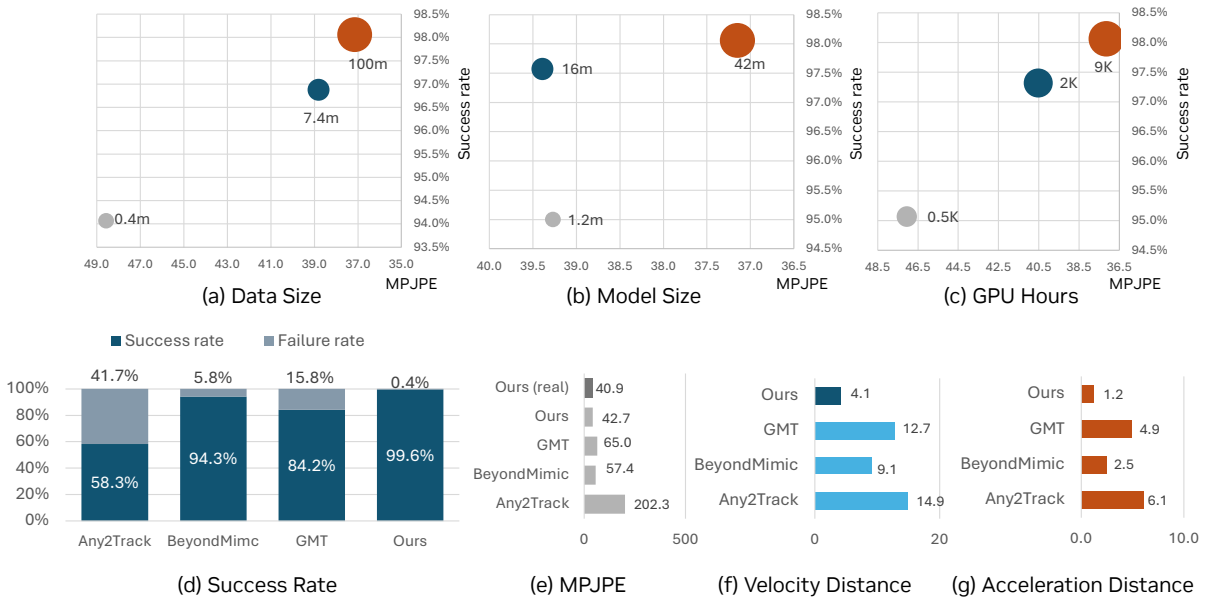

Figure 2: (a-c) Effect of scaling to different sizes of dataset, model, and compute. Mean per joint position error (MPJPE) indicates motion imitation error; lower is better. For (a), we measure the dataset size in millions of frames. (d-g) Comparing our method with baselines on tracking out-of-distribution motion sequences. (d) Success rate of tracking. (e-g) Different tracking accuracy metrics are evaluated on trajectories that are successfully tracked.

Our evaluation is conducted on a uniformly random subset of retargeted AMASS (Mahmood et al., 2019) data (9 hours, 1,602 trajectories), as used in TWIST (Ze et al., 2025). Notably, our test set is orders of magnitude larger than those in prior work (Zeng et al., 2025) and comparable in scale to some existing training sets. Importantly, SONIC is *not* trained on the AMASS dataset, underscoring the robustness of its generalization.

Scaling Up Motion Tracking. In Fig. 2, we analyze the impact of supersizing our humanoid motion tracking system along three key dimensions: GPU hours, model size, and motion dataset size. All evaluations are conducted in Isaac Lab (NVIDIA et al., 2025), with models trained to convergence over 3 to 7 days. For dataset size, we compare with training on LaFAN (0.4M frames), a subset of our in-house dataset (7.4M frames), and our full dataset (100M frames). For GPU hours comparison, we train all models till convergence for training with 8, 32, and 128 GPUs. Across the board, scaling along any of these three axes leads to consistent improvements in motion imitation performance. Notably, increasing the size of the motion dataset yields the most substantial gains, while increased model size and compute (GPU hours) further enhance results. Notice that for GPU hours, parallelizing over more GPUs (e.g., 8 vs 128) is essential, as we notice that training on a smaller number of GPUs yields worse asymptotic performance than a larger number of GPUs. For evaluation, motion imitation is considered unsuccessful if, at any point during tracking, the humanoid's body height deviates by more than 0.25m from the reference motion or if the root orientation differs by more than 1 radian.

Comparison with Baselines. We demonstrate that our approach yields a universal motion tracker capable of accurately tracking a wide range of unseen motions. In Fig. 2, we compare our method to state-of-the-art trackers — Any2Track (Zhang et al., 2025), BeyondMimic (Liao et al., 2025), and GMT (Chen et al., 2025). The baselines are trained on different datasets (Any2Track and BeyondMimic on LaFAN; GMT on AMASS) but are all evaluated on the same unseen dataset for consistency. We use the officially released models when available (GMT) and use the publicly released code to train the respective models (Any2Track and BeyondMimic) when

not. To ensure a fair comparison, all evaluations are performed in MuJoCo (Todorov et al., 2012), which is supported by all baseline implementations. For these experiments, we adopt a more relaxed termination criterion to better reflect real-world deployment scenarios: imitation is only considered unsuccessful if the humanoid falls, defined as its root height deviating by more than 0.25m from the reference. Across all evaluation metrics, our method significantly outperforms the baselines, achieving higher success rates and improved tracking accuracy on unseen motion sequences.

Real-World Evaluation. We assess the real-world performance of SONIC by deploying it on 50 diverse motion trajectories, including dance, jumps, and loco-manipulation tasks. As presented in Fig. 2, our policy achieves motion imitation in the real world that closely matches its simulation results, operating in a true zero-shot manner. Remarkably, it succeeds on all sequences without a single failure (100% success rate), underscoring the robustness and reliability of our tracker in challenging real-world scenarios.

2.2. Interactive Motion Control

In this section, we showcase the scalability and robustness of SONIC in whole-body, real-time interactive control tasks. We present a kinematic runtime generative motion planner that guides the robot's policy through user interaction. Our approach employs an autoregressive framework that continually regenerates future kinematic motions conditioned on the previous robot states and incoming user commands. For each planning step, the model generates motion segments lasting between 0.8s and 2.4s, where the duration is automatically determined by the neural planner to maximize flexibility and robustness. The planner achieves inference times under 5 ms on a standard laptop and 12 ms on a Jetson Orin GPU. Replanning is triggered as frequently as every 100 ms, or immediately when user commands are updated, ensuring highly responsive control.

SONIC supports a variety of applications, including but not limited to: (1) navigation control with arbitrary velocity, direction, and style commands; (2) interactive entertainment tasks such as boxing; (3) locomotion skills such as squatting, crawling, kneeling, etc., which are useful for downstream applications like teleoperation. Because both our kinematic planner and tracking policy are trained on the same large-scale dataset. Utilizing the scalable nature of SONIC, we note that all of the applications above were designed afterwards without retraining the planner or the tracking policy. People with minimal animation or programming background can easily design and specify new desired behaviors.

For navigation control, SONIC supports velocity commands ranging from 0.0m/s to 6.0m/s, as well as arbitrary direction commands spanning 0 to 360 degrees. We employ a critically damped spring model to smooth impractical commands that cannot be achieved by the robot within one motion segment. SONIC achieves scalable, responsive and robust navigation control as shown in the first two rows of Fig. 3. SONIC also supports different styles such as drunken walking, injured walking, happy walking, stealth walking, etc., as shown in the third row of Fig. 3. The capacity of the model to generate robust inbetweening motions showcases the flexibility of SONIC, and the potential of more natural human-robot interaction.

SONIC further extends its versatility to interactive entertainment tasks such as boxing, as illustrated in the last two rows of Fig. 3. Existing academic solutions (Starke et al., 2021; Won et al., 2021) and industrial approaches (Unitree, 2025) often utilize a limited collection of boxing clips, requiring switching between multiple expert models or action labels. This approach leads to discontinuous, unnatural transitions and even pauses. SONIC on the other hand enables much more fluid, responsive, and natural motion generation, while retaining the robot's full freedom of movement throughout the task.

To enable downstream manipulation or navigation in confined environments, skills such as squatting, kneeling, and crawling are essential. As illustrated in Fig. 4, SONIC supports a rich spectrum of postural modes, including squatting and kneeling, allowing the pelvis height to be smoothly controlled from 0.3m to 0.8m. This greatly enhances the agent's reachability and adaptability in scenarios such as teleoperation. For navigation in especially tight spaces, SONIC also enables crawling. The robot can move omnidirectionally using its elbows and knees at

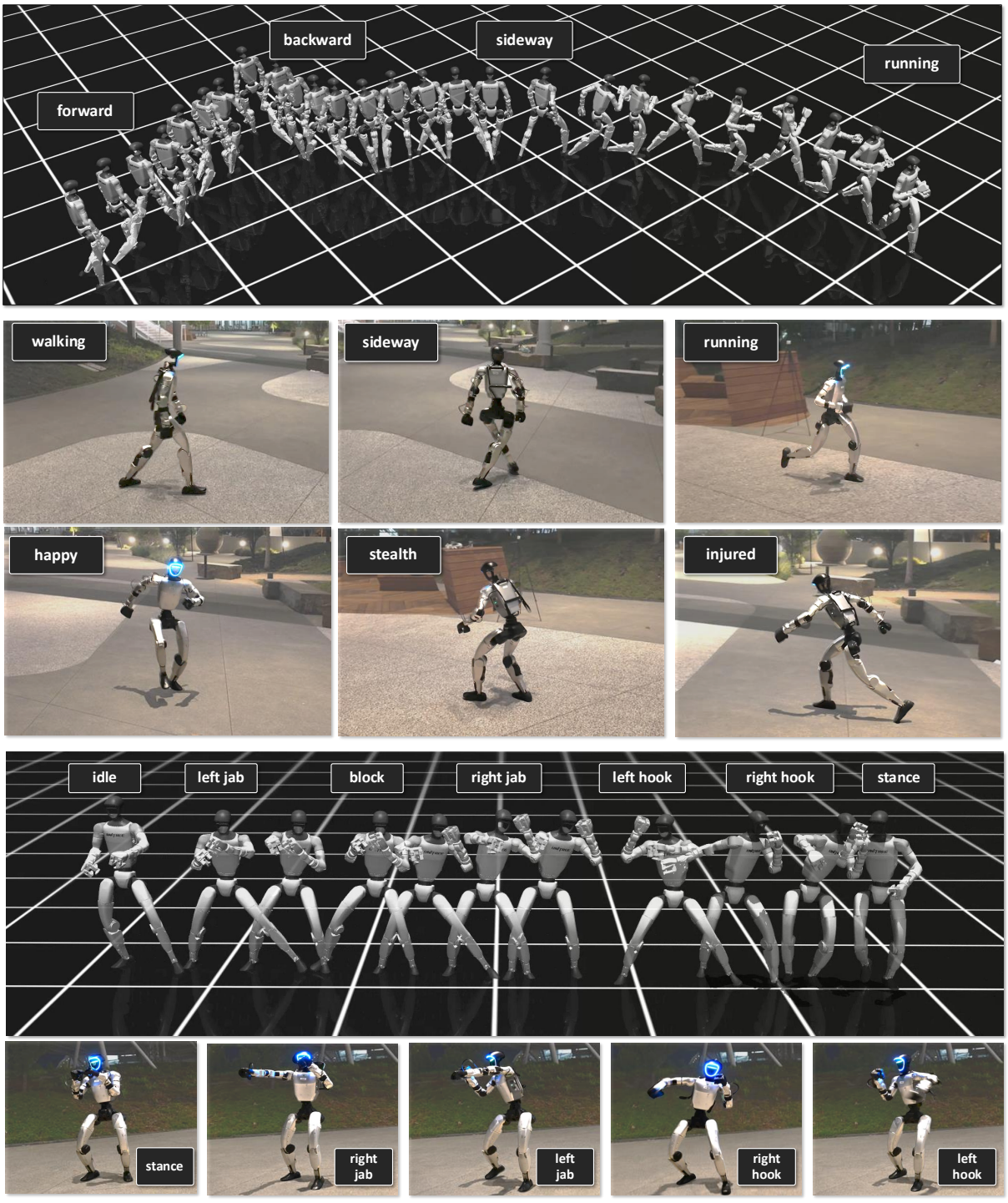

Figure 3: Top three rows: interactive navigation switching between different velocities, directions, and styles. Bottom two rows: SONIC produces high-quality and responsive boxing motions while preserving the robot's complete freedom of movement throughout the task.

velocities from 0.0m/s to 0.5m/s.

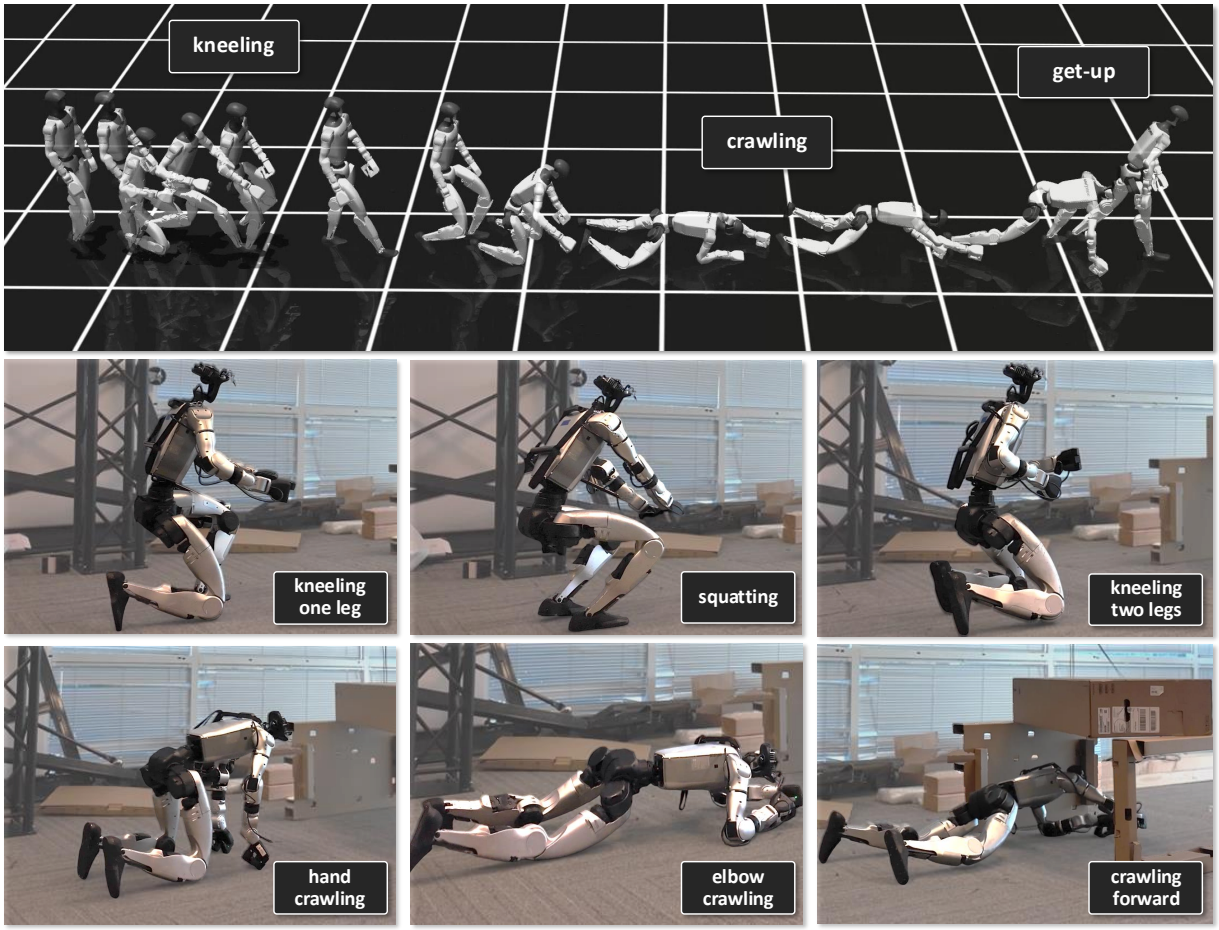

Figure 4: Interactive squatting, kneeling, and crawling. With SONIC, the robot can squat, kneel, and crawl at arbitrary heights, enabling seamless application in real-world downstream scenarios such as teleoperation and navigation in complex environments.

2.3. Video Teleoperation and Multi-Modal Cross-Embodiment Control

As shown in Fig. 5 (top), SONIC also demonstrates a real-time, multimodal, cross-embodiment control framework, enabled by our universal control policy. A unified motion generation system based on GENMO (Li et al., 2025) is designed to generate human motions from three input modalities: human demonstration video, natural-language commands, and music audio. Transitions among modalities are coordinated through the multimodal motion generation system, enabling smooth handoffs without reinitialization.

Video Teleoperation. For video control, the system supports both pre-recorded clips and live monocular webcam streams. Human motion is estimated at ≥ 60 frames per second (fps), enabling interactive teleoperation without specialized motion-capture hardware. Video control provides a higher-fidelity specification of pose and timing, yielding a precise imitation of demonstrated movements.

Music and Text Control. For text control, the system accepts natural-language prompts and synthesizes target motions at ≥ 60 fps. Trained on a large-scale dataset, the policy executes previously unseen instructions in a zero-shot manner. An interactive graphical interface supports free-form prompting at any time with immediate on-robot responses (for example, “walk forward”, “kick left foot”, or “dance like a monkey”).

For music control, the robot generates dance motions conditioned on melodic and rhythmic structure, tracking tempo, and adapting style to musical characteristics.

Figure 5: Video teleoperation, multi-modal control, and VR whole-body teleoperation.

Our system supports seamless transitions between modalities. For example, users can initiate fine-grained control via video, switch to text for general control, and finally hand off to music for performance.

2.4. VR-Based Teleoperation and Connecting to Foundation Models

We present three additional control regimes built on our universal motion tracker and token space: (1) Virtual Reality (VR)-based whole-body teleoperation for full-pose control, (2) VR-based 3-point teleoperation for efficient data collection, and (3) foundation-model-driven mobile manipulation, where a VLA autonomously controls the humanoid. While the teleoperation solutions presented in this section focus on VR-based interfaces, the same universal token-space supports video-based teleoperation; see Sec. 2.3.

2.4.1. VR-Based Whole-Body Teleoperation

We develop a full-body VR teleoperation system using the PICO whole-body motion-tracking interface. This requires wearing the PICO headset, two ankle trackers, and the handheld VR controllers. The PICO interface then provides human motion as full-body human pose estimates in SMPL (Loper et al., 2015) format. The tracked human motion is streamed in real time to our universal control policy, which will be introduced in

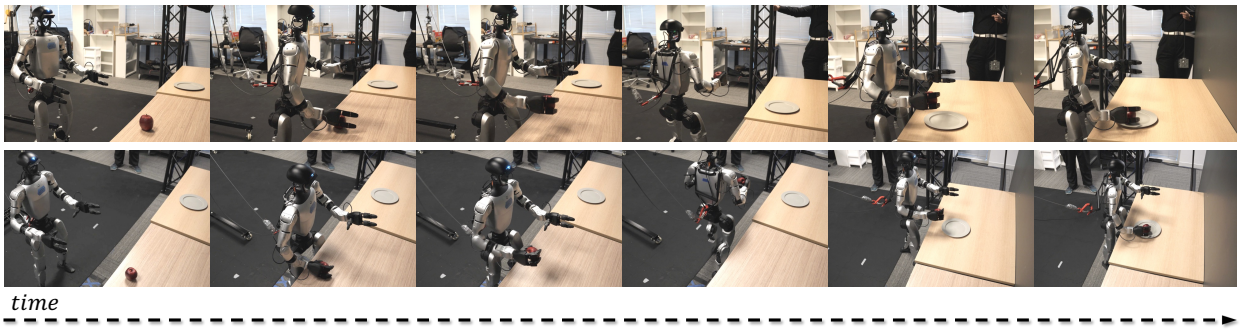

Figure 6: Apple-to-plate mobile bimanual manipulation on the Unitree G1 humanoid robot controlled by a fine-tuned GR00T N1.5 vision-language-action (VLA) model. The VLA emits teleoperation-format commands—poses of head and wrists, base height, and a navigation command—which are executed by SONIC. The model is fine-tuned on 300 VR-teleoperated trajectories and attains 95% success over 20 trials on this proof-of-concept task, demonstrating compatibility between foundation-model planning and our universal control policy.

Sec. 3.2. The human motion will be encoded via its respective encoder into the universal token space, and then decoded by the robot control decoder to yield low-latency, stable, human-like control, as can be seen in Fig. 5 (bottom).

2.4.2. VR-Based 3-Point Teleoperation

To enable scalable and portable data collection for tasks that do not require precise foot placement, we also introduce a lightweight mobile bimanual VR teleoperation interface that operates with the PICO headset and the two handheld controllers (no ankle trackers needed for this mode). The teleoperation interface outputs a compact command consisting of three upper-body SE(3) poses (head and both wrists), finger joint angles, waist height, a locomotion mode (slow walk or fast walk), and a navigation command specifying desired root velocity and heading. These signals are fed into the kinematic motion planner (of Sec. 3.3) and hybrid encoder (introduced in Sec. 3.2), then executed by the universal tracking policy.

We use this interface to collect 300 teleoperated demonstrations of a mobile pick-and-place task: the robot walks to a randomly placed apple on a table, grasps it with the right hand, and places it on a randomly positioned plate. This dataset serves as supervision for autonomous control with a vision-language-action (VLA) model in Sec. 2.4.3.

For this mobile manipulation task, teleoperation real-time tracking performance is reported for the right wrist only, since the left wrist remains largely stationary. Over 300 teleoperated trajectories, we measure the latency and tracking error from the right wrist target pose command to the realized wrist pose on the robot, with both quantities expressed in the waist frame. The pipeline exhibits a mean latency of 121.9 ms. The mean right-wrist position error is 6 cm with a 95th percentile of 13.3 cm, and the mean orientation error is 0.145 rad ($8.32°$) with a 95th percentile of 0.267 rad ($15.31°$). Position error is the Euclidean norm and orientation error is the geodesic angle on $\mathrm{SO}(3)$.

2.4.3. Foundation-Model-Driven Mobile Bimanual Manipulation

To demonstrate autonomous control through the same universal token interface, we connect a VLA foundation model to the pipeline used for teleoperation. Using the 300 trajectories collected with the 3-point teleoperation interface, we fine-tune a GR00T N1.5 model (Bjorck et al., 2025,) and evaluate it on the apple-to-plate mobile pick-and-place task described above (Fig. 6). The VLA outputs the same teleoperation-format control signals—three upper-body poses (head and both wrists), base (waist) height, and a navigation command (root linear and angular velocity)—which are then fed into the kinematic planner and hybrid encoder, and then executed via the universal control policy. On this proof-of-concept task, the system attains a 95% success rate over 20 trials,

indicating that the universal motion tracker serves as a robust System 1 controller (fast, reactive whole-body skills) that complements the VLA's System 2 capabilities (slower, deliberative reasoning) (Kahneman, 2011). This experiment is intended only to establish compatibility between foundation-model planning and our motion-tracking controller, not to claim broad generalization; scaling to richer task distributions is left for future work. Looking ahead, interfacing the VLA with the universal token space offers a path toward high-level, whole-body reasoning layered atop the responsiveness, robustness, and human-motion inductive biases of the universal tracker, which is able to operate at substantially higher control rates. Exploring this integration at scale is an important direction for future work.

2.5. Discussion

We cast motion tracking as the core scalable task for learning a single, versatile humanoid controller. By training SONIC on 100 million+ motion frames with up to 128 GPUs, we obtain a single policy that produces natural, robust whole-body behaviors across diverse conditions. Equally important, we build the practical system that makes tracking usable in real deployments: a real-time kinematic motion planner that converts intent into short-horizon reference motions, and a universal token space that unifies heterogeneous interfaces (teleoperation, video, text, and music) within one policy.

We observe consistent improvements as data, model capacity, and compute increase, with generalization to unseen motions in simulation and real-world deployments. These findings support motion tracking as a practical route to acquire broad, transferable whole-body priors without per-task reward engineering.

Relative to prior trackers (e.g., Any2Track (Zhang et al., 2025), BeyondMimic (Liao et al., 2025), GMT (Chen et al., 2025)), our contributions are threefold: (1) scale: orders-of-magnitude larger training and larger unseen evaluation dataset sets; (2) versatility: robot, human, and hybrid commands in a shared latent, enabling cross-embodiment transfer and versatile interface; and (3) practicality: on-board, real-time sim-to-real control on a Unitree G1 with stronger performance under a common evaluation protocol. Together, these advances move motion tracking from narrow imitation toward a general foundation for whole-body humanoid control.

The planner (Sec. 2.2) enables responsive navigation, postural mode control (squatting, crawling), and style modulation without retraining. The universal token space (Sec. 2.3) supports seamless switches among input modalities and cross-embodiment teleoperation. The same interface connects to a vision–language–action model (Sec. 2.4), providing the system 1 capabilities needed for humanoid loco-manipulation.

Limitations include formal treatment of safety, compliance, and energy efficiency for extended deployments, as well as combating noisy input during deployments. Future work will study scaling laws across more diverse dataset, enable VLA instructed whole-body locomanipulation tasks, and explore joint training of planner, tokenizers, and policy to reduce modality gaps.

In summary, scaling motion tracking yields reliable, general whole-body control; pairing it with a planner, a universal token space, and an efficient onboard stack makes it usable as a system. We expect SONIC to serve as a practical foundation upon which higher-level perception and reasoning can be layered to advance general-purpose humanoid autonomy.

3. Materials and Methods

3.1. Humanoid Motion Dataset

Our motion dataset is built from a large-scale motion-capture corpus comprising recordings from 170 human subjects. The participants include a balanced mix of male and female actors, with body heights ranging from 145 cm to 199 cm (mean = 174.3 cm, std = 10.9 cm), approximately following a normal distribution. The dataset spans a broad spectrum of everyday human behaviors, including locomotion, daily activities, gesturing, and a diverse set of combat motions with varied stylistic expressions. Clip durations range from 1 to 180

seconds. In total, the collection covers thousands of unique motion behaviors, with most actions performed by multiple subjects across multiple takes, providing rich intra- and inter-subject variation, as can be seen in Fig. 7. Our in-house dataset contains 700 hours of human motion, resulting in 100+ million frames at 50Hz. The human motion is then retargeted to the humanoid using GMR (Araujo et al., 2025).

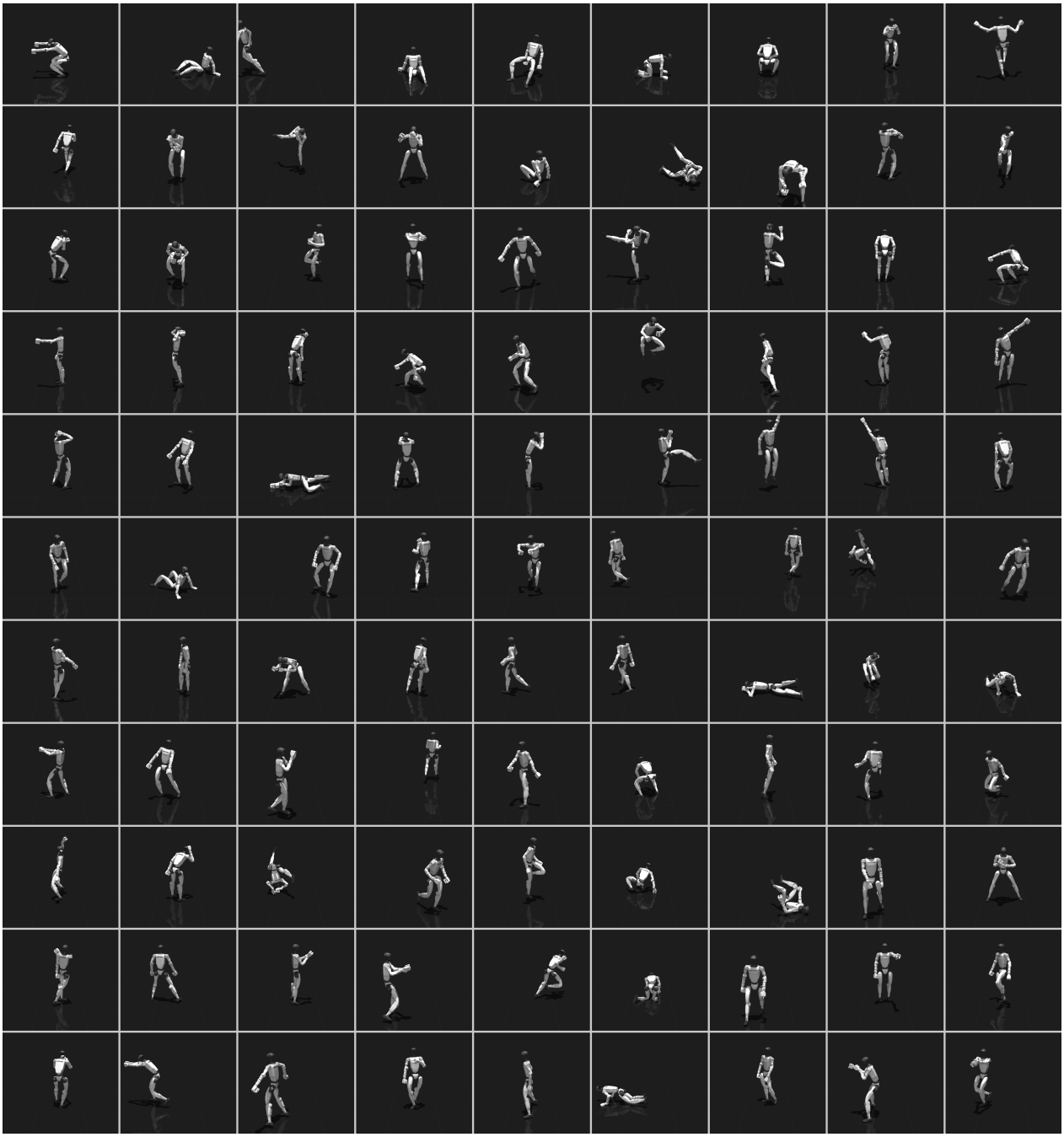

Figure 7: Random samples from our motion dataset.

3.2. Universal Humanoid Motion Tracking

Fig. 8 provides an overview of our approach, SONIC, a universal humanoid motion tracking framework that employs a unified control policy to track diverse motion commands across different embodiments. A key innovation is its ability to seamlessly handle robot motion, human motion, and hybrid motion (combining upper-body keypoints with lower-body robot motions) through a shared latent representation. This cross-

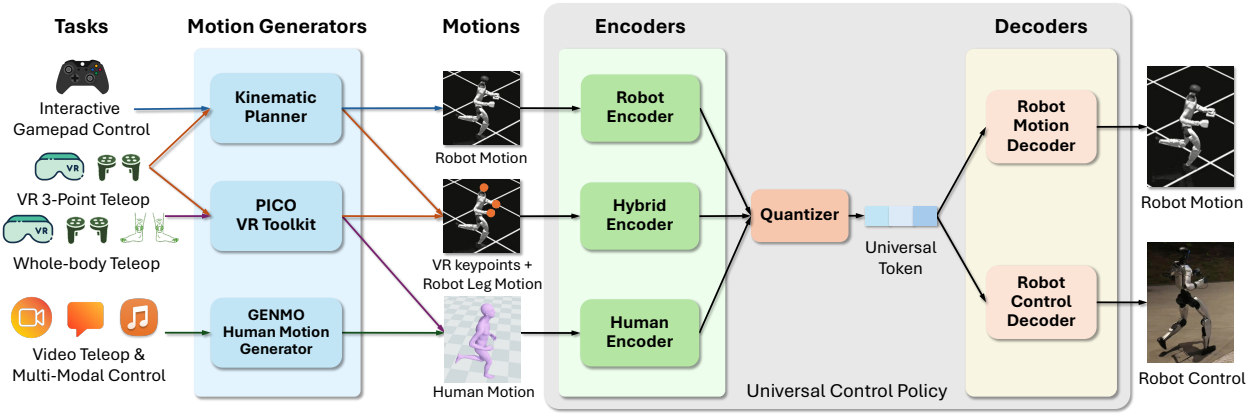

Figure 8: SONIC **enables universal humanoid motion tracking through a universal control policy that handles diverse motion commands and modalities.** Specialized encoders process robot, human, and hybrid motion commands into a universal token that drives robot control and motion decoders. This cross-embodiment design supports diverse applications including gamepad control, VR teleoperation, whole-body teleoperation, video teleoperation, and multi-modal control from text and music.

embodiment capability enables the robot to learn from motion captured and raw video data, bridging the morphological gap between human and robot embodiments. We use various motion generators – kinematic motion planner, VR motion generator, human motion generator (GENMO) – to generate motion commands, which enable diverse applications including interactive gamepad control, VR 3-point teleoperation, whole-body teleoperation, video-based teleoperation, and multi-modal control from text and music.

Motion Tracking Formulation. We formulate humanoid motion tracking as a Markov Decision Process $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$, comprising state space, action space, transition function, reward function, and discount factor γ. We train the policy using proximal policy optimization (PPO) (Schulman et al., 2017) to maximize the expected cumulative discounted return $\mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1} r_t\right]$.

States. The state representation $\boldsymbol{s}_t$ comprises two components: proprioceptive sensing $\boldsymbol{s}_t^{\text{p}}$ and motion command $\boldsymbol{s}_t^{\text{g}}$. Proprioceptive information $\boldsymbol{s}_t^{\text{p}} \triangleq (\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t, \boldsymbol{\omega}_t, \boldsymbol{\psi}_t, \boldsymbol{a}_{t-1})$ includes joint pose $\boldsymbol{q}_t$, joint velocity $\dot{\boldsymbol{q}}_t$, root angular velocity $\boldsymbol{\omega}_t$, gravity vector $\boldsymbol{g}_t$ in the root frame, and previous action $\boldsymbol{a}_{t-1}$. The motion command $\boldsymbol{s}_t^{\text{g}}$ has three types: robot motion $\boldsymbol{g}_r$, human motion $\boldsymbol{g}_h$, or hybrid motion $\boldsymbol{g}_m$ (combining upper-body keypoints with lower-body robot motions), where we drop the subscript t for brevity. All state quantities are expressed in the robot's local heading frame to ensure rotation invariance.

Actions. The policy π outputs target joint positions $\boldsymbol{a}_t$ as actions, which are tracked by proportional-derivative (PD) controllers at each joint.

Rewards. Following Liao et al. (2025), we define the reward as $r_t = \mathcal{R}(\boldsymbol{s}_t^{\text{p}}, \boldsymbol{s}_t^{\text{g}}) + \mathcal{P}(\boldsymbol{s}_t^{\text{p}}, \boldsymbol{a}_t)$, combining tracking reward and penalty terms. The tracking term $\mathcal{R}$ minimizes errors in root position, root orientation, body link positions (relative to root), body link orientations (relative to root), body link linear velocities, and body link angular velocities between the robot state $\boldsymbol{s}_t^{\text{p}}$ and target $\boldsymbol{s}_t^{\text{g}}$. The penalty term $\mathcal{P}$ discourages abrupt action changes, joint limit violations, and undesired contacts. Detailed reward design is presented in Table 1.

Domain Randomization. To enhance robustness and generalization across diverse scenarios, we apply systematic domain randomization during training. We randomize physical parameters which include friction coefficients (μ_s, μ_d), restitution coefficient (e), default joint positions $(\boldsymbol{q}_0)$, and base center-of-mass position. We also periodically apply random perturbations to the robot's root linear and angular velocities to simulate external pushes. Additionally, we apply motion perturbation to the target motion commands $\boldsymbol{s}_t^{\text{g}}$ during training to improve robustness. All domain randomization parameters are detailed in Table 2.

Reward term	Equation	Weight
Tracking rewards $\mathcal{R}(\boldsymbol{s}_t^p, \boldsymbol{s}_t^g)$		
Root orientation	$r_{\text{ori}}^{\text{root}}(t) = \exp\big(-\|\boldsymbol{o}_{t,r}^p - \boldsymbol{o}_{t,r}^g\|_2^2/0.4^2\big)$	0.5
Body link pos (rel.)	$r_{\text{pos}}^{\text{body}}(t) = \exp\Big(-\frac{1}{\lvert\mathcal{B}\rvert}\sum_{b\in\mathcal{B}}\|\boldsymbol{p}_{t,b}^{p,\text{rel}} - \boldsymbol{p}_{t,b}^{g,\text{rel}}\|_2^2/0.3^2\Big)$	1.0
Body link ori (rel.)	$r_{\text{ori}}^{\text{body}}(t) = \exp\Big(-\frac{1}{\lvert\mathcal{B}\rvert}\sum_{b\in\mathcal{B}}\|\boldsymbol{o}_{t,b}^{p,\text{rel}} - \boldsymbol{o}_{t,b}^{g,\text{rel}}\|_2^2/0.4^2\Big)$	1.0
Body link lin. vel	$r_{\text{lin}}^{\text{body}}(t) = \exp\Big(-\frac{1}{\lvert\mathcal{B}\rvert}\sum_{b\in\mathcal{B}}\|\boldsymbol{v}_{t,b}^{p} - \boldsymbol{v}_{t,b}^{g}\|_2^2/1.0^2\Big)$	1.0
Body link ang. vel	$r_{\text{ang}}^{\text{body}}(t) = \exp\Big(-\frac{1}{\lvert\mathcal{B}\rvert}\sum_{b\in\mathcal{B}}\|\boldsymbol{\omega}_{t,b}^{p} - \boldsymbol{\omega}_{t,b}^{g}\|_2^2/3.14^2\Big)$	1.0
Penalty terms $\mathcal{P}(\boldsymbol{s}_t^p, \boldsymbol{a}_t)$		
Action rate	$r_{\text{act}}(t) = \|\boldsymbol{a}_t - \boldsymbol{a}_{t-1}\|_2^2$	-0.1
Joint limit	$r_{\text{jlim}}(t) = \sum_j \mathbb{1}[\boldsymbol{q}_{t,j} \notin [\boldsymbol{q}_{t,j}^{\min}, \boldsymbol{q}_{t,j}^{\max}]]$	-10.0
Undesired contacts	$r_{\text{contact}}(t) = \sum_{c\notin\{\text{ankles, wrists}\}} \mathbb{1}[\|\boldsymbol{F}_c\| > 1.0\text{N}]$	-0.1

Table 1: Reward design. Orientations are represented as 6D rotations (Zhou et al., 2019). Superscript g: goal/target state from $\boldsymbol{s}_t^{\text{g}}$; superscript p: proprioceptive/current state from $\boldsymbol{s}_t^{\text{p}}$; $\mathcal{B}$: set of tracked body links; superscript "rel": quantities expressed in root frame.

Universal Control Policy. A distinguishing characteristic of our tracking framework is its ability to accommodate multiple motion command types from different embodiments through a unified encoder-decoder architecture. We accomplish cross-embodiment learning via specialized encoders that process heterogeneous inputs from both human and robot embodiments into a shared latent representation. This representation undergoes quantization to yield a *universal token*, which subsequently drives a common robot control decoder to generate motor commands. This design enables the policy to leverage motion data from diverse sources — both robot demonstrations and human motion — allowing the robot to imitate human movements despite morphological differences. An auxiliary robot motion decoder is also used to facilitate feature learning and serve as an implicit retargeting module from human to robot embodiment.

Encoders. Three specialized encoders process distinct motion command types: (1) robot motion encoder $\boldsymbol{\mathcal{E}}_r$ encodes robot joint positions and velocities over F_r future frames with a frame interval Δt_r, (2) human motion encoder $\boldsymbol{\mathcal{E}}_h$ encodes 3D human joint positions (Loper et al., 2015) over F_h future frames with a frame interval Δt_h, and (3) hybrid motion encoder $\boldsymbol{\mathcal{E}}_m$ encodes sparse upper-body keypoints (head and hands) of the current frame (for real-time upper-body tracking), combined with lower-body robot motion over F_m future frames with a frame interval Δt_m. Multi-frame inputs enable anticipatory behavior and improve the robustness of the policy. All encoders are implemented as multi-layer perceptrons (MLPs; architecture details in Table 3) that map commands $\boldsymbol{g}_r, \boldsymbol{g}_h, \boldsymbol{g}_m$ into a shared latent space, enabling cross-embodiment representation alignment.

Quantizer. The encoded latent representation is quantized into a *universal token* $\boldsymbol{z}$ using a vector quantizer. Specifically, we use FSQ (Mentzer et al., 2023) as our vector quantizer. The universal token is a D_z-dimensional vector with L_z quantization levels per dimension.

Decoders. The *universal token* $\boldsymbol{z}$ is decoded through two separate decoders. First, a robot control decoder $\boldsymbol{\mathcal{D}}_c$ transforms the *universal token* into motor commands that control the robot's joints. Second, a robot motion decoder $\boldsymbol{\mathcal{D}}_r$ reconstructs the robot motion command, providing auxiliary supervision to improve the latent space and enhance feature learning. Both decoders are implemented as MLPs (Table 3).

Training. We prepare synchronized motion data across all three command types. Each command type $\boldsymbol{g}_r, \boldsymbol{g}_h, \boldsymbol{g}_m$ is encoded via its respective encoder and quantized to produce universal tokens $\boldsymbol{z}_r, \boldsymbol{z}_h, \boldsymbol{z}_m$. For each token, the control decoder $\boldsymbol{\mathcal{D}}_c$ generates motor commands while the motion decoder $\boldsymbol{\mathcal{D}}_r$ reconstructs the robot

Domain Randomization	Sampling Distribution
Physical parameters	
Static friction coefficients	$\mu_s \sim \mathcal{U}[0.3, 1.6]$
Dynamic friction coefficients	$\mu_d \sim \mathcal{U}[0.3, 1.2]$
Restitution coefficient	$e \sim \mathcal{U}[0, 0.5]$
Default joint positions	$\boldsymbol{q}_0 \sim \boldsymbol{q}_0 + \mathcal{U}[-0.01, 0.01]$
Base COM offset (x, y, z)	$\Delta x \sim \mathcal{U}[-0.075, 0.075],\ \Delta y \sim \mathcal{U}[-0.1, 0.1],\ \Delta z \sim \mathcal{U}[-0.1, 0.1]$
Root velocity perturbations (external pushes)	
Root linear vel (x, y, z)	$v_x \sim \mathcal{U}[-0.5, 0.5],\ v_y \sim \mathcal{U}[-0.5, 0.5],\ v_z \sim \mathcal{U}[-0.2, 0.2]$
Push duration	$\Delta t \sim \mathcal{U}[1, 3]$s
Root angular vel	$\omega_{\text{roll}} \sim \mathcal{U}[-0.52, 0.52],\ \omega_{\text{pitch}} \sim \mathcal{U}[-0.52, 0.52],\ \omega_{\text{yaw}} \sim \mathcal{U}[-0.78, 0.78]$
Target motion perturbations ($\boldsymbol{s}_t^g$)	
Target position jitter	$\Delta \boldsymbol{p}^g \sim \mathcal{U}[-0.05, 0.05]^3$ (x,y: ± 0.05, z: ± 0.01)
Target orientation jitter	$\Delta\phi_{\text{roll}}, \Delta\phi_{\text{pitch}} \sim \mathcal{U}[-0.1, 0.1],\ \Delta\phi_{\text{yaw}} \sim \mathcal{U}[-0.2, 0.2]$
Target linear vel jitter	$\Delta \boldsymbol{v}^g \sim \mathcal{U}[-0.5, 0.5]^3$ (x,y: ± 0.5, z: ± 0.2)
Target angular vel jitter	$\Delta\omega_{\text{roll}}, \Delta\omega_{\text{pitch}} \sim \mathcal{U}[-0.52, 0.52],\ \Delta\omega_{\text{yaw}} \sim \mathcal{U}[-0.78, 0.78]$
Target joint jitter	$\Delta \boldsymbol{q}_t^g \sim \mathcal{U}[-0.1, 0.1]$

Table 2: Domain randomization parameters applied during training. $\mathcal{U}[\cdot]$: uniform distribution.

Module	Architecture	Dims
Network configuration		
Quantizer	FSQ	token dimensions $= D_z$; quantization levels $= L_z$
Encoder (g1)	MLP	hidden$= [2048, 1024, 512, 512]$
Encoder (teleop)	MLP	hidden $= [2048, 1024, 512, 512]$
Encoder (smpl)	MLP	hidden $= [2048, 1024, 512, 512]$
Decoder (actions)	MLP	hidden $= [2048, 2048, 1024, 1024, 512, 512]$
Decoder (refs)	MLP	hidden $= [2048, 1024, 512, 512]$
Action dimension	Diagonal Gaussian	29
Critic	MLP	hidden $= [2048, 2048, 1024, 1024, 512, 512]$
Motion command		
Future frames	$F_r = F_h = F_m = 10$ frames	
Frame interval	$\Delta t_r = t_m = 0.1s$, $\Delta t_h = 0.02s$	

Table 3: Universal control policy hyperparameters.

motion command. The total loss comprises:

$$\mathcal{L} = \mathcal{L}_{\text{ppo}} + \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{token}} + \mathcal{L}_{\text{cycle}} \tag{1}$$

$$\mathcal{L}_{\text{recon}} = \|\boldsymbol{\mathcal{D}}_r(\boldsymbol{z}_r) - \boldsymbol{g}_r\|^2 + \|\boldsymbol{\mathcal{D}}_r(\boldsymbol{z}_h) - \boldsymbol{g}_r\|^2 + \|\boldsymbol{\mathcal{D}}_r(\boldsymbol{z}_m) - \boldsymbol{g}_r\|^2 \tag{2}$$

$$\mathcal{L}_{\text{token}} = \|\boldsymbol{z}_r - \boldsymbol{z}_h\|^2 \tag{3}$$

$$\mathcal{L}_{\text{cycle}} = \|\boldsymbol{\mathcal{E}}_r(\boldsymbol{\mathcal{D}}_r(\boldsymbol{z}_h)) - \boldsymbol{z}_r\|^2 \tag{4}$$

where $\mathcal{L}_{\text{ppo}}$ denotes the standard PPO loss. $\mathcal{L}_{\text{recon}}$ represents the reconstruction loss for the robot motion command across different input modalities. Notably, when the input command is human motion $\boldsymbol{g}_h$, the encoder-decoder acts as a retargeting pipeline from human to robot motion, and $\mathcal{L}_{\text{recon}}$ serves as a retargeting loss that enables cross-embodiment transfer. $\mathcal{L}_{\text{token}}$ measures the discrepancy between the robot token $\boldsymbol{z}_r$ and the human motion token $\boldsymbol{z}_h$, explicitly encouraging the encoder networks to produce aligned representations across embodiments. This loss is crucial for achieving cross-embodiment learning, as it ensures that motions from different embodiments (human vs. robot) are mapped to similar representations in the shared latent space. $\mathcal{L}_{\text{cycle}}$ is a cycle consistency loss between the original robot token $\boldsymbol{z}_r$ and the token produced by re-encoding the reconstructed robot motion from the human token, i.e., $\boldsymbol{\mathcal{E}}_r(\boldsymbol{\mathcal{D}}_r(\boldsymbol{z}_h))$. This loss further reinforces alignment of the latent space and supports consistent cross-embodiment representation learning, ensuring that the translation from human to robot motion and back preserves the essential motion characteristics.

We employ bin-based adaptive motion sampling when selecting the initial frame for each episode. Specifically, the entire motion dataset is partitioned into fixed-duration bins. For each bin, we calculate the corresponding

Training hyperparameter	Value
Num parallel envs per GPU	4096
Num steps per env	24
Learning epochs	5
Num mini-batches	4
Discount γ	0.99
GAE λ	0.95
Clip parameter	0.2
Entropy coefficient	0.013
Value loss coefficient	1.0
Actor learning rate	2×10^{-5}
Critic learning rate	1×10^{-3}
Max gradient norm	0.1
Desired KL	0.01
Adaptive LR min/max	$[1\times10^{-5}, 2\times10^{-4}]$
Init noise std	0.05
Actor std clamp min/max	$[0.001, 0.5]$
Adaptive sampling bin size	1s
Adaptive sampling failure rate cap	$\beta = 200$
Adaptive sampling blending hyperparameter	$\alpha = 0.1$

Table 4: Training hyperparameters.

failure rate f_i of the policy. To prevent excessive sampling from bins with exceptionally high failure rates, the failure rate in each bin is capped at $\beta\bar{f}$, where β is a hyperparameter and $\bar{f}$ is the average failure rate across all bins. The normalized, capped failure rates are then used to derive preliminary sampling weights $\hat{p}_i$ for each bin. The final sampling probability is defined as $p_i = \alpha\hat{p}_i + (1-\alpha)\frac{1}{N}$, where α is a blending hyperparameter and N is the total number of bins. This formulation balances targeted sampling of challenging bins with uniform coverage. A bin is selected according to p_i, and the initial frame is then sampled uniformly from within the chosen bin. All models are trained using Isaac Lab (NVIDIA et al., 2025) as the simulation platform. Training hyperparameters are provided in Table 4.

Tasks and Applications. The universal motion tracking framework enables a wide range of applications:

1. *Interactive Gamepad Control*: Users intuitively control the humanoid via gamepad inputs. A generative kinematic motion planner, introduced in Section 3.3, transforms user commands such as velocities, styles into diverse robot motions. The universal control policy tracks these motions through the robot motion encoder $\boldsymbol{\mathcal{E}}_r$.
2. *VR 3-Point Teleoperation*: Users operate the humanoid using a PICO VR headset and handheld controllers, which provide head and wrist SE(3) pose data. We also use the joysticks of the controllers for setting a desired navigation command (linear and angular root velocity) and root height. The kinematic motion planner generates the lower-body motions, resulting in a hybrid command (upper-body keypoints, lower-body robot motion) tracked by the universal policy with the hybrid motion encoder $\boldsymbol{\mathcal{E}}_m$.
3. *VLA Integration*: We post-train a GR00T N1.5 model on data collected with the VR 3-Point Teleoperation stack. The action of the VLA model is fed into the kinematic motion planner to generate the lower-body motions, resulting in a hybrid command (upper-body direct VLA action, lower-body robot motion) tracked by the universal policy with the hybrid motion encoder $\boldsymbol{\mathcal{E}}_m$.
4. *VR Whole-body Teleoperation*: Users control the humanoid using VR headsets (PICO), handheld controllers, and IMU sensors on the ankles, which allows us to use off-the-shelf toolkits to stream full-body human motion. The universal control policy tracks this motion using the human motion encoder $\boldsymbol{\mathcal{E}}_h$.
5. *Video Teleoperation & Multi-Modal Control*: We employ the multi-modal motion generation model GENMO (Li et al., 2025) to estimate and generate human motions from multi-modal inputs (e.g., video, text, audio). The universal control policy subsequently tracks the generated motions using the human motion encoder $\boldsymbol{\mathcal{E}}_h$. Further details are presented in Section 3.4.

3.3. Generative Kinematic Motion Planner

Our generative kinematic motion planner is a large-scale latent generative model, trained on the same natural whole-body motion data as the motion tracking policy. At a high level, the planning process is formulated as an autoregressive motion in-betweening generation task. The context keyframes capture historical robot states, such as joint positions and root positions, while target keyframes are either navigation guidance keyframes generated from the user commands such as velocity, direction, and style, or skill-specific targets for actions such as squatting, crawling, boxing, etc.

Motion Representation. During training, we sample motion segments of length between 0.8s and 2.4s, extracting the keyframes at both endpoints to serve as the context and target keyframes. Our motion representation is mathematically equivalent to the humanoid pose configuration q_t as introduced in Sec. 3.2. Specifically, we represent kinematic motion using the pelvis-relative joint positions and global joint rotations. During training, we randomly rotate the training samples to enable planning in all initial orientations. Incorporating global rotation instead of local, canonicalized rotation is essential for generating motions such as squatting and crawling, where the notion of heading is ill-defined and greatly hurts the motion planning quality. We refer readers to Meng et al. (2025) and Meng et al. (2025) for similar insights and additional discussion.

Generative Neural Backbone in Latent Space. Planning is conducted in the latent space, where continuous motions are first encoded as a sequence of latent tokens as follows:

$$\{z_t\}_{t=1}^{T/4} = \text{enc}\left(\{p_t, r_t\}_{t=1}^{T}\right), \tag{5}$$

where p_t and r_t denote the pose configuration and root position at frame t, respectively. In practice, the encoder operates with a downsampling rate of 4. The latent token sequence is encoded by models such as Transformers or Conv1D networks to capture temporal consistency.

The inbetweening process in the token space is guided by two constraints: the starting and target keyframes, denoted as $\{p_t, r_t\}_{t=1}^{4}$ and $\{p_t, r_t\}_{t=T-4}^{T}$ respectively. Rather than training the network to predict the entire sequence of tokens from these sparse constraints in a single pass, we adopt a masked token prediction approach (Guo et al., 2024; Luo et al., 2024; Pinyoanuntapong et al., 2024; Yu et al., 2023). In this framework, the neural backbone iteratively predicts and finalizes the subset of tokens for which it has the highest confidence, progressively refining the prediction.

$$h = \mathcal{F}\left(\{p_t, r_t\}_{t=1}^{4}, \{p_t, r_t\}_{t=T-4}^{T}, \{z_t\}_{t=1}^{T/4}\right), \tag{6}$$

$$\text{Prob}(z_t) = \sigma(h). \tag{7}$$

This process is iterative, where $\mathcal{F}(\cdot)$ denotes the neural backbone, and h represents the logits for each token position. Token probabilities are computed by applying a softmax function $\sigma(\cdot)$ to the logits. At the first iteration, all latent tokens are unknown, and we initialize the latent embedding with a learnable mask embedding, z_{masked}. During training, the proportion of masked tokens is uniformly sampled from the range $[100\%, 0\%]$. During inference, a cosine schedule determines the proportion of tokens to finalize at each iteration, specifically $1.0 - \cos\left(\frac{\pi}{2} \cdot \frac{L}{L_{\max}}\right)$, where L is the current iteration and $L_{\max}$ is the maximum number of iterations. After finalization of all tokens, the predicted tokens are used to reconstruct the kinematic motions and generate the robot control signals.

Root trajectory spring model. We propose to use an intuitive critical damped spring model to generate the

root position and heading of the keyframes from user commands as follows:

$$x(t) = \left(x_T - x_0 + \left(v_0 + \frac{c}{2}\left(x_T - x_0\right)\right) t\right) e^{-\frac{c}{2}t}, \tag{8}$$

where x_T denotes the target value, x_0 the initial value, v_0 the initial velocity, and c the damping coefficient. We apply this critically damped spring model to three quantities: 1) the pelvis position along the x-axis, 2) the pelvis position along the y-axis, and 3) the projected heading angle of the pelvis. Damping coefficients of $5\ln(2)$ and $20\ln(2)$ are used for position and heading respectively. The target values may be obtained directly from the controllers. Alternatively, if the controller only specifies a desired velocity we can compute the expected target positions after 1.0s using the desired velocity. The target keyframes are then placed at the position and heading with $x(1.0)$, as computed by the spring model in Eq. (8). In practice, we find that our generative kinematic motion planner is highly robust to the choice of damping coefficients. In fact, the spring model could often be omitted entirely, as the planner's ability to generate motions of variable length (ranging from 0.8s to 2.4s) and its strong inbetweening capability makes it adaptable to a wide range of root trajectory commands. Nevertheless, incorporating the spring model improves behavioral predictability and helps safeguard against unrealistic commands, such as abruptly reversing direction from 6.0m/s to -6.0m/s.

Keyframe Module and Application Integration. Traditional motion planning methods often rely on complex target keyframe generation, such as detailed footstep planning. In contrast, our system provides keyframes in a far more intuitive manner, requiring minimal effort or expertise for keyframe specification.

For navigation control, target keyframes are generated by placing a randomly selected segment from the navigation clips of the desired style at the target root trajectory. Despite this simplicity, our model consistently produces natural and smooth motions that align well with the specified style. We attribute this to the model's flexible, variable-length motion generation and its robust inbetweening capabilities. Additionally, the autoregressive replanning ensures that the generated motion is continually refreshed before reaching the end of any given clip, thus minimizing dependence on the specific spatial details of the chosen target keyframes. This approach generalizes to other motion styles such as walking, running, and crawling.

For entertainment tasks such as boxing, target keyframes are determined by selecting the most expressive segment (e.g., the frames with maximal arm extension for a punch) from motion clips that match the desired style. We also support motion layering, where the upper body is specified and the lower body is generated accordingly by the planner, enabling the introduction of predefined behaviors.

For interactive modes needed in manipulation tasks, such as squatting or kneeling, keyframes are retrieved online from the motion clip library according to the desired height. Unlike traditional approaches that require an extensive motion library, our system needs only a single clip to generate the full distribution of transitional motions for a given skill.

3.4. Multi-modal Motion Generation Model

We adopt GENMO (Li et al., 2025) to support multi-modal conditioning within one framework. The core idea is to treat estimation from videos as *constrained generation*: the model synthesizes a complete motion trajectory that must satisfy observed evidence (e.g., video keypoints), while the same network can also generate diverse motions from abstract conditions (e.g., text or audio). This unification lets generative priors regularize estimation under occlusion or noise, and conversely lets abundant in-the-wild video data improve generative diversity.

Conditioning modalities and temporal layout. The model accepts mixed, time-varying conditions, including text prompts, audio features, and visual observations. Conditions may appear over different temporal intervals (e.g., text at the start, audio segments mid-trajectory, intermittent video spans). We represent these inputs as temporally indexed condition streams. Each stream is encoded by a modality-specific encoder into a sequence

of features aligned to a common motion frame rate. Missing intervals are handled natively (no special-case branching) by simply providing empty or masked tokens; downstream fusion treats these as regular positions with null content.

Architecture. Condition streams are fused using a temporal transformer with cross-attention from the motion tokens to the multi-modal condition tokens. A diffusion-based motion prior then operates over the human motion sequence, denoising from Gaussian noise to a kinematically plausible trajectory. The prior is trained to model human motion dynamics and is agnostic to the specific conditioning type; all modality-specific structure lives in the encoders and the fusion layers. We encode the denoised human motion into the universal token space, enabling a single decoder to serve multiple downstream embodiments.

Training objective. Training mixes two complementary objectives. (1) *Generative* learning uses the standard diffusion loss over human motions, conditioned on available modalities (text/audio/video) to learn a broad motion prior. (2) *Estimation-guided* learning adds reconstruction terms when observations are present (e.g., 2D/3D keypoints, trajectory constraints), encouraging the generated motion to match measurements exactly where they exist while relying on the prior elsewhere. Because estimation is realized as generation with hard/soft constraints, the same network parameters serve both tasks. Mixed batches with variable-length sequences and randomly dropped modalities teach the model to seamlessly handle partial and asynchronous evidence without special logic.

Inference modes. At test time, the model supports: (1) *Pure generation* from abstract prompts (text-only or text+audio) to diverse, realistic motions; (2) *Constrained generation* given video frames to produce accurate, temporally coherent reconstructions; and (3) *Hybrid control* where modalities switch over time (e.g., text sets intent, audio refines timing, brief video spans anchor pose). All modes share the same sampler and differ only in which condition streams are non-empty. The model is converted to TensorRT for fast inference.

Integration with our system. Although GENMO supports arbitrary-length motion generation, we use sliding windows with overlap to ensure low-latency motion generation. We modify the diffusion denoising process with inpainting to handle the transitions between windows. More specifically, our diffusion is formulated as the variance-preserving diffusion, where the denoiser predicts the clean human motion at each denosing step. The overlapped part of the denoised human motion is overwritten by the previous window's motions during denoising.

3.5. Deployment

Experiments are conducted on a Unitree G1 platform using the built-in joint-level PD controller. The learned policy outputs desired joint angles are passed directly to the PD interface. All components of the inference and management stack execute *onboard*, leveraging the on-device CPU/GPU to minimize feedback latency and improve timing determinism.

The policy loop runs at 50 Hz. At each cycle, we assemble observations from recent robot state and operator intent, evaluate a learned policy, and synthesize joint-space targets. To ensure crisp, compliant actuation, these targets are streamed by a separate high-rate process at 500 Hz through the Unitree low-level API, continuously publishing the latest user inputs (target motion command) without blocking the policy loop.

Due to the encoder-decoder design, our system can *seamlessly switch between different input interfaces* (e.g., keyboard, gamepad, or networked streams) by changing the input encoder, enabling seamless transitions between various tasks: interactive gamepad control, VR 3-point teleoperation, whole-body teleoperation, video-based teleoperation, and multi-modal control from text and music. The user input is captured independently at 100 Hz.

When needed, the kinematic motion planner is employed at 10 Hz which proposes short-horizon sequences based

on the command from the input interface. These plans are temporally aligned with the 50 Hz policy loop and smoothly blended into the control objectives, providing a motion reference without sacrificing responsiveness. Each periodic process operates on consistent snapshots of the robot state, and we adopt a "latest-data-wins" convention to mitigate jitter, ensuring transient delays do not stall the system.

Both the interactive kinematic motion planner and the policy inference execute *onboard* the Jetson Orin GPU using TensorRT with *CUDA Graph* acceleration. This configuration yields reliable, low-variance execution with 1–2 ms per forward pass for policy and 12 ms per forward pass for motion generation. The resulting architecture emphasizes timing robustness and smoothness. It produces a stable motion, while decoupled loops isolate sensing and planning from control, stabilizing end-to-end latency. Unless otherwise noted, all reported experiments use this onboard configuration (50/10/500/100 Hz) with timing logs enabled for diagnostics and reproducibility.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

[2] Michael Ahn, Anthony Brohan, Noah Brown, and et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. URL https://arxiv.org/abs/2204.01691. 2

[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. 11

[4] Qingwei Ben, Feiyu Jia, Jia Zeng, Junting Dong, Dahua Lin, and Jiangmiao Pang. Homie: Humanoid loco-manipulation with isomorphic exoskeleton cockpit. *arXiv preprint arXiv:2502.13013*, 2025. 2

[5] Johan Bjorck, Valts Blukis, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Xiaowei Jiang, Kaushil Kundalia, Jan Kautz, Zhiqi Li, Kevin Lin, Zongyu Lin, Loic Magne, Yunze Man, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Jing Wang, Qi Wang, Shihao Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Seonghyeon Ye, Zhiding Yu, Yizhou Zhao, Zhe Zhang, Ruijie Zheng, and Yuke Zhu. Gr00t n1.5: An improved open foundation model for generalist humanoid robots. https://research.nvidia.com/labs/gear/gr00t-n1_5/, June 2025. Blog post; accessed 2025-10-30. 9

[6] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. doi: 10.48550/arXiv.2503.14734. URL https://arxiv.org/abs/2503.14734. 3, 9

[7] Andreas Blattmann, Tim Dockhorn, Robin Rombach, and Patrick Esser. Stable video diffusion: Scaling latent video diffusion models. *arXiv preprint arXiv:2311.15127*, 2023. URL https://arxiv.org/abs/2311.15127. 1

[8] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, and et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. URL https://arxiv.org/abs/2108.07258. 1

[9] Anthony Brohan, Noah Brown, Carlos Carbajal, and et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. URL https://arxiv.org/abs/2212.06817. 2

[10] Anthony Brohan, Noah Brown, Ilya Chelombiev, and et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *Nature*, 2023. doi: 10.1038/s41586-023-06475-7. 2

[11] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI*, 2024. 1

[12] 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. 2, 4, 10

[13] Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. Momask: Generative masked modeling of 3d human motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1900–1910, 2024. 16

[14] 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. 2

[15] 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. 2

[16] Tairan He, Wenli Xiao, Toru Lin, Zhengyi Luo, Zhenjia Xu, Zhenyu Jiang, Jan Kautz, Changliu Liu, Guanya Shi, Xiaolong Wang, et al. Hover: Versatile neural whole-body controller for humanoid robots. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9989–9996. IEEE, 2025. 2

[17] Xialin He, Runpei Dong, Zixuan Chen, and Saurabh Gupta. Learning getting-up policies for real-world humanoid robots. *arXiv preprint arXiv:2502.12152*, 2025. 2

[18] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 1

[19] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, and et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. URL https://arxiv.org/abs/2203.15556. 1

[20] 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. 2

[21] Daniel Kahneman. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, 2011. ISBN 9780374275631. 10

[22] Jared Kaplan, Sam McCandlish, Tom Henighan, and et al. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. URL https://arxiv.org/abs/2001.08361. 1

[23] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. Vibe: Video inference for human body pose and shape estimation. In *CVPR*, 2020. doi: 10.1109/CVPR42600.2020.01265. 3

[24] Jialong Li, Xuxin Cheng, Tianshu Huang, Shiqi Yang, Ri-Zhao Qiu, and Xiaolong Wang. Amo: Adaptive motion optimization for hyper-dexterous humanoid whole-body control. *arXiv preprint arXiv:2505.03738*, 2025. 2

[25] Jiefeng Li, Jinkun Cao, Haotian Zhang, Davis Rempe, Jan Kautz, Umar Iqbal, and Ye Yuan. Genmo: A generalist model for human motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. 7, 15, 17

[26] Ruilong Li, Shan Li, Angjoo Huang, and et al. Bailando: 3d dance generation via actor-critic gpt with choreographic memory. In *SIGGRAPH Asia 2021*, 2021. doi: 10.1145/3478513.3480495. 3

[27] Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. AI Choreographer: Music conditioned 3d dance generation with aist++. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021. 2

[28] Qiayuan Liao, Takara E Truong, Xiaoyu Huang, 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. 2, 4, 10, 12

[29] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 34(6): 248:1–248:16, 2015. doi: 10.1145/2816795.2818013. URL http://smpl.is.tue.mpg.de. 8, 13

[30] Zhengyi Luo, Jinkun Cao, Alexander W. Winkler, Kris Kitani, and Weipeng Xu. Perpetual humanoid control for real-time simulated avatars. In *International Conference on Computer Vision (ICCV)*, 2023. 2

[31] Zhuoyan Luo, Fengyuan Shi, Yixiao Ge, Yujiu Yang, Limin Wang, and Ying Shan. Open-magvit2: An open-source project toward democratizing auto-regressive visual generation. *arXiv preprint arXiv:2409.04410*, 2024. 16

[32] Yi Ma, Zahid Hazara, Brian Ichter, and et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024. URL https://arxiv.org/abs/2403.12945. 2

[33] 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 the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2, 4

[34] Zichong Meng, Zeyu Han, Xiaogang Peng, Yiming Xie, and Huaizu Jiang. Absolute coordinates make motion generation easy. *arXiv preprint arXiv:2505.19377*, 2025. 16

[35] Zichong Meng, Yiming Xie, Xiaogang Peng, Zeyu Han, and Huaizu Jiang. Rethinking diffusion for text-driven human motion generation: Redundant representations, evaluation, and masked autoregression. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 27859–27871, 2025. 16

[36] Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505*, 2023. 13

[37] NVIDIA, :, Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Muñoz, Xinjie Yao, René Zurbrügg, Nikita Rudin, Lukasz Wawrzyniak, Milad Rakhsha, Alain Denzler, Eric Heiden, Ales Borovicka, Ossama Ahmed, Iretiayo Akinola, Abrar Anwar, Mark T. Carlson, Ji Yuan Feng, Animesh Garg, Renato Gasoto, Lionel Gulich, Yijie Guo, M. Gussert, Alex Hansen, Mihir Kulkarni, Chenran Li, Wei Liu, Viktor Makoviychuk, Grzegorz Malczyk, Hammad Mazhar, Masoud Moghani, Adithyavairavan Murali, Michael Noseworthy, Alexander Poddubny, Nathan Ratliff, Welf Rehberg, Clemens Schwarke, Ritvik Singh, James Latham Smith, Bingjie Tang, Ruchik Thaker, Matthew Trepte, Karl Van Wyk, Fangzhou Yu, Alex Millane, Vikram Ramasamy, Remo Steiner, Sangeeta Subramanian, Clemens Volk, CY Chen, Neel Jawale, Ashwin Varghese Kuruttukulam, Michael A. Lin, Ajay Mandlekar, Karsten Patzwaldt, John Welsh, Huihua Zhao, Fatima Anes, Jean-Francois Lafleche, Nicolas Moënne-Loccoz, Soowan Park, Rob Stepinski, Dirk Van Gelder, Chris Amevor, Jan Carius, Jumyung Chang, Anka He Chen, Pablo de Heras Ciechomski, Gilles Daviet, Mohammad Mohajerani, Julia von Muralt, Viktor Reutskyy, Michael Sauter, Simon Schirm, Eric L. Shi, Pierre Terdiman, Kenny Vilella, Tobias Widmer, Gordon Yeoman, Tiffany Chen, Sergey Grizan, Cathy Li, Lotus Li, Connor Smith, Rafael Wiltz, Kostas Alexis, Yan Chang, David Chu, Linxi "Jim" Fan, Farbod Farshidian, Ankur Handa, Spencer Huang, Marco Hutter, Yashraj Narang, Soha Pouya, Shiwei Sheng, Yuke Zhu, Miles Macklin, Adam Moravanszky, Philipp Reist, Yunrong Guo, David Hoeller, and Gavriel State. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning, 2025. URL https://arxiv.org/abs/2511.04831. 4, 15

[38] Open X-Embodiment Collaboration. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. URL https://arxiv.org/abs/2310.08864. 2

[39] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. In *ACM SIGGRAPH 2018 Papers*, 2018. doi: 10.1145/3197517.3201311. 1

[40] Xue Bin Peng, Zhaoyu Zhou, Stephen Luo, and Michiel van de Panne. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics*, 40(4), 2021. doi: 10.1145/3450626.3459670. 1

[41] Ekkasit Pinyoanuntapong, Pu Wang, Minwoo Lee, and Chen Chen. Mmm: Generative masked motion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1546–1555, 2024. 16

[42] Abhinanda Punnakkal, Michael J. Black, Tushar Kapadi, and Gerard Pons-Moll. Babel: Bodies, action and behavior with english labels. In *CVPR*, 2021. doi: 10.1109/CVPR46437.2021.00756. 2

[43] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 1

[44] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1

[45] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 12

[46] Sebastian Starke, Yiwei Zhao, Fabio Zinno, and Taku Komura. Neural animation layering for synthesizing martial arts movements. *ACM Transactions on Graphics (TOG)*, 40(4):1–16, 2021. 5

[47] Richard S. Sutton. The bitter lesson. http://www.incompleteideas.net/IncIdeas/BitterLesson.html, 2019. 1

[48] Guy Tevet, Sigal Raab, Yuval Shafir, and et al. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022. URL https://arxiv.org/abs/2209.14916. 3

[49] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109. 5

[50] Unitree. Humanoid robot G1_Humanoid Robot Functions_Humanoid Robot Price | Unitree Robotics — unitree.com. https://www.unitree.com/g1, 2024. [Accessed 31-10-2025]. 3

[51] Unitree. Unitree boxing. https://www.unitree.com/boxing, 2025. Accessed: 2024-06-30. 5

[52] Tingwu Wang, Yunrong Guo, Maria Shugrina, and Sanja Fidler. Unicon: Universal neural controller for physics-based character motion. *arXiv preprint arXiv:2011.15119*, 2020. 2

[53] Jason Wei, Yi Tay, Rishi Bommasani, and et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022. URL https://arxiv.org/abs/2206.07682. 1

[54] Jungdam Won, Deepak Gopinath, and Jessica Hodgins. Control strategies for physically simulated characters performing two-player competitive sports. *ACM Transactions on Graphics (TOG)*, 40(4):1–11, 2021. 5

[55] Kangning Yin, Weishuai Zeng, Ke Fan, Minyue Dai, Zirui Wang, Qiang Zhang, Zheng Tian, Jingbo Wang, Jiangmiao Pang, and Weinan Zhang. Unitracker: Learning universal whole-body motion tracker for humanoid robots, 2025. URL https://arxiv.org/abs/2507.07356. 2

[56] Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. Magvit: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10459–10469, 2023. 16

[57] Yanjie Ze, Zixuan Chen, JoÃĢo Pedro AraÃšjo, Zi-ang Cao, Xue Bin Peng, Jiajun Wu, and C Karen Liu. Twist: Teleoperated whole-body imitation system. *arXiv preprint arXiv:2505.02833*, 2025. 2, 4

[58] Weishuai Zeng, Shunlin Lu, Kangning Yin, Xiaojie Niu, Minyue Dai, Jingbo Wang, and Jiangmiao Pang. Behavior foundation model for humanoid robots. *arXiv preprint arXiv:2509.13780*, 2025. 2, 4

[59] Ye Zhang, Tong He, Qingxuan Zhang, and et al. T2m-gpt: Generating human motion from textual descriptions with discrete representations. In *CVPR*, 2023. doi: 10.1109/CVPR52729.2023.00877. 3

[60] Zhikai Zhang, Jun Guo, Chao Chen, Jilong Wang, Chenghuai Lin, Yunrui Lian, Han Xue, Zhenrong Wang, Maoqi Liu, Huaping Liu, et al. Track any motions under any disturbances. *arXiv preprint arXiv:2509.13833*, 2025. 2, 4, 10

[61] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5745–5753, 2019. 13

A. Acknowledgments

We thank Scott Reed, Wei Liu, You Liang Tan, Avnish Narayan, Fengyuan Hu, Qi Wang, Yuqi Xie, Letian (Max) Fu, Mengda Xu, Davis Rempe, Xue Bin (Jason) Peng, Haotian Zhang, Yifeng Jiang, Anna Minx, John Malaska, Chen Tessler for their thoughtful discussions. We thank Leilee Naderi and Amanpreet Singh for supporting with data collection and providing feedback on teleoperation user experience. We thank Jeremy Chimienti and Tri Cao for their work ensuring robot readiness and conducting the necessary repairs. We also thank Tri Cao, Jazmin Sanchez, Demetria Quijada, and Jesse Yang for their help in filming and editing.

B. Contributors

ZL, YY, TW, CL contributed equally as co-first authors. They trained the polices, developed planners, and designed the deployment piplines.

SC, FC, ZC, JL, DM, QB, XD are core contributors towards the whole systems including system ID, VLA training, model speed up, teleoperation, etc.

SY, CH, LS, EL, EJ, TH focused on retargeting and filtering our large-scale humanoid motion data.

RD, TH, HX, WX, ZW helped with brainstorming and codebase setup.

SY, JK, YC, UI, LF, YZ are our leadership team providing guidance.