HIPC: HUMANOID ITERATIVE PLANNING AND CONTROL VIA GOAL-CONDITIONED MOTION CLOSURE

Anonymous authors
Paper under double-blind review

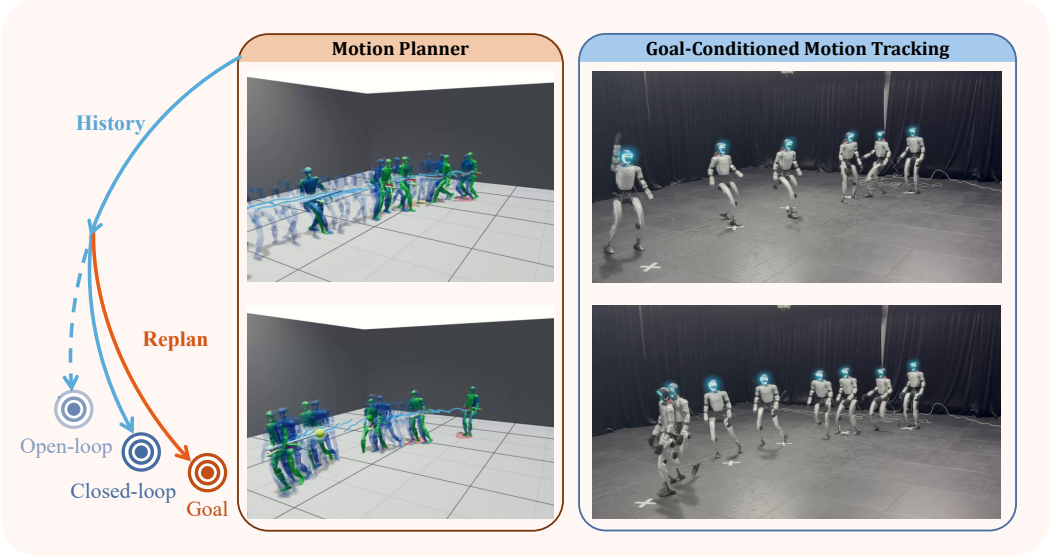

Figure 1: *We propose Humanoid Interactive Planning and Control framework (HIPC), a real-time asynchronous interactive motion planning and control dual-system architecture for goal-reaching tasks. HIPC establishes a unified interface between the robot's current physical state and multimodal commands, enabling the system to interpret and execute heterogeneous control signals, including root positions, joint configurations, end-effector poses, natural-language instructions, and their combinations.*

ABSTRACT

Recent generative models are widely used to provide whole-body motion for humanoid tracking. Current methods combine motion generators with controllers in an open-loop paradigm, which leads to critical issues: tracking errors can accumulate into spatial drift, and the planner cannot adapt to unexpected physical disturbances without feedback from physical execution. However, directly feeding realized states back to the planner in a closed loop can cause discontinuities, resulting in physical jitter and even falls, since such states are out of distribution during training. To bridge this gap, we propose **HIPC**, a real-time **Humanoid Iterative Planning and Control** framework that couples a generative model with motion tracking into a unified planning-and-control paradigm. HIPC introduces a goal-conditioned motion planning trained with a canonical motion representation that avoids gimbal-lock singularities under extreme orientations and supports motion generation on diverse geometric and semantic conditions. Then, HIPC adopts two strategies, deviation-decayed recovery and history re-anchoring, to mitigate the out-of-distribution shift caused by realized-state feedback. Finally, HIPC operates a real-time asynchronous planning-and-control system, with latency compensation to reduce temporal mismatch for inference delay. Extensive experiments confirm the effectiveness of HIPC under switching and composition of diverse conditions. Results show that HIPC achieves state-of-the-art goal-reaching accuracy among existing methods while simultaneously demonstrating robustness to physical perturbations and the ability to recover from falls.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Recent general motion trackers (Luo et al., 2026; Zeng et al., 2026; RoboParty Lab Team, 2026) have made diverse whole-body behaviors executable through a single policy, they provide a reusable low-level control interface, shifting the challenge upstream from task-specific control to motion planning. Yet optimization-based planning is difficult to scale up due to high-dimensional whole-body coordination and complex contact constraints (Gu et al., 2026), motivating generative motion planners that synthesize diverse references from learned motion priors while delegating physical realization to a general tracker (Luo et al., 2026; Liao et al., 2026). However, simply composing a generative planner with a whole-body tracker leaves the planning-to-control interface open-loop (Chen et al., 2026). Physical execution inevitably deviates from kinematic references, creating a *Tracking Gap*. Although the tracker locally corrects such deviations, they remain invisible to the planner and can accumulate into spatial drift or exceed the tracker's recovery capability (Tao et al., 2026; Xu et al., 2026a). Closing this loop requires feeding back the realized physical state from which future motion continues (Chen et al., 2026), rather than only task-level signals such as localization or progress.

However, directly incorporating realized states will cause an *Execution Distribution Gap*, as physical execution histories differ from the kinematic distribution during training. Existing self-rollout strategy mitigate autoregressive distribution shift (Xie et al., 2026; Chen et al., 2026), but do not directly capture structured deviations induced by downstream dynamics. Moreover, replanning from realized states that deviate from the current reference can create discontinuities between successive motion chunks (Chen et al., 2026). Non-negligible *Inference Delay* (Black et al., 2026; 2025) further misaligns planning and execution as the robot continues to evolve during generation. Together, these effects can produce abrupt reference transitions, physical jitter, and even loss of balance. Closed-loop motion-data refinement can incorporate controller feedback during training (Xu et al., 2026b), but couples motion synthesis to a specific downstream controller. The key challenge is therefore to close the planning–control loop while enabling smooth physical execution without sacrificing planner–controller modularity.

In this paper, we introduce **HIPC**, a real-time **Humanoid Iterative Planning and Control** framework that closes the loop between generative motion planning and whole-body tracking. HIPC combines a goal-conditioned motion planner with execution-consistent state feedback and latency-aware asynchronous planning–control. It continuously incorporates realized states into planning while preserving planner-native motion histories, and anticipates the state reached during planner inference for smooth reference handoff. This enables closed-loop whole-body planning without requiring controller-specific rollout data or sacrificing planner–controller modularity. In summary, the primary contributions of this work are three-fold:

Goal-Conditioned Motion Planner with Canonical Representation. We introduce a generative model in a goal reaching paradigm as motion planner, and train this generator with a canonical motion representation conditioned on heterogeneous geometric goals and semantic text.

Execution-Consistent Motion Closure for Real State Feedback. We employ two strategies to re-anchor planned history to the latest robot state. This formulation decouples planning from the executed motion, while preserving consistency between the predicted and actual state.

Latency-Aware Asynchronous Planning and Control. We decouple low-frequency generative planning from high-frequency whole-body tracking through a non-blocking asynchronous architecture. By explicitly accounting for planner inference latency, HIPC maintains continuous real-time control and smooth reference handoff during closed-loop replanning.

2 RELATED WORK

Human Motion Generation. Human motion generation has largely evolved along two complementary paradigms. Diffusion-based models such as MDM (Tevet et al., 2023) model continuous motion distributions through iterative denoising, while autoregressive approaches such as MotionGPT (Jiang et al., 2023) formulate motion generation as sequential prediction in a unified motion–language space. Recent methods increasingly combine the strengths of advanced architectures through masked modeling (Guo et al., 2024; Chen, 2024), hybrid conditions (Zhao et al., 2025; Li et al., 2025), and scaling controllable modeling (Rempe et al., 2026; Zhao et al., 2026). HIPC adopts the autoregressive diffusion paradigm as its motion planner backbone.

Table 1: ***Functional comparison of representative humanoid motion generation and planning–control approaches.*** *"Real-State Feedback" denotes feedback of realized physical states to the planner for online replanning. "Joint" denotes joint-level configuration goals. "Perturbation" and "Get Up" denote recovery from external perturbations and fallen states, respectively.*

Method	Text	Real-Time	Real-State Feedback	Goal Condition				OOD Recovery	
				Root	End Effector	Velocity	Joint	Perturbation	Get Up
Kimodo	✓	✗	✗	✓	✓	✗	✓	✗	✗
TextOp	✓	✓	✗	✗	✗	✗	✗	✗	✗
ARDY	✓	✓	✗	✓	✓	✓	✓	✗	✗
OMG	✓	✓	✗	✗	✗	✗	✗	✗	✗
ReactiveBFM	✓	✓	✓	✓	✗	✗	✗	✓	✓
HIPC (Ours)	✓	✓	✓	✓	✓	✓	✓	✓	✓

Humanoid Whole-Body Control. Physics-based motion imitation established a powerful framework for acquiring dynamic motor skills from reference motion. Early methods such as DeepMimic (Peng et al., 2018) and AMP (Peng et al., 2021) mainly focus on tracking individual motions or limited skill sets, while later systems such as HumanX (Wang et al., 2026b), HAIC (Li et al., 2026a), and VAIC (Li et al., 2026b) demonstrate increasingly agile whole-body behaviors and dynamic interactions. Recent works further scale motion data and model capacity to build generalist or universal motion trackers, such as SONIC (Luo et al., 2026) and ScaleBFM (Zeng et al., 2026), consolidating diverse behaviors into reusable low-level control policies. HIPC builds on this progress by coupling its generative planner with a general-purpose whole-body tracker as the execution interface.

Humanoid Generative Motion Control. Generative motion control first emerged in physics-based character animation, where CLoSD (Tevet et al., 2025) couples an autoregressive diffusion planner with an RL tracker and closes the loop through simulated physical execution. Recent works extend this paradigm to physical humanoids. MotionBricks (Wang et al., 2026a) deploys a motion generator on a humanoid, but leaves the execution-side coupling problem unexplored. Heracles (Tao et al., 2026) identifies large state deviations as a key limitation of rigid reference tracking and introduces a state-conditioned diffusion middleware that preserves tracking near the reference while generating recovery motions under large deviations. TextOp (Xie et al., 2026) demonstrates real-time text-driven motion generation and execution, while ReactiveBFM (Chen et al., 2026) further studies execution feedback, autoregressive distribution shift, and asynchronous replanning. HIPC addresses the resulting execution-history and temporal-alignment issues through history re-anchoring and latency-aware asynchronous planning, while preserving a modular planner–controller interface.

3 Proposed Method

In this section, we present our closed-loop humanoid planning-and-control framework HIPC, as illustrated in Fig. 2. We first introduce canonical motion representation of the goal-conditioned motion planner in Sec. 3.1. Next, Sec. 3.2 addresses the distribution shift caused by directly feeding realized physical states back into the planner through execution-aware data preparation and history reanchoring. Finally, Sec. 3.3 presents the asynchronous planning-and-control system used for real-world deployment, together with latency compensation for continuous real-time replanning.

3.1 Goal-Conditioned Motion Planning

We formulate whole-body planning as conditional autoregressive motion generation. At each replanning step t, the planner observes the most recent $H = 16$ motion frames $\mathbf{x}_{t-H+1:t}$ together with an optional condition set $\mathbf{c}_t$, and predicts a future motion chunk of $F = 64$ frames:

$$\hat{\mathbf{x}}_{t+1:t+F} = G_\theta\left(\mathbf{x}_{t-H+1:t}, \mathbf{c}_t\right). \tag{1}$$

The generated chunk serves as a kinematic reference for the downstream universal whole-body controller. The condition set $\mathbf{c}_t$ supports both geometric goals and semantic text commands, enabling diverse goal-directed whole-body motion generation.

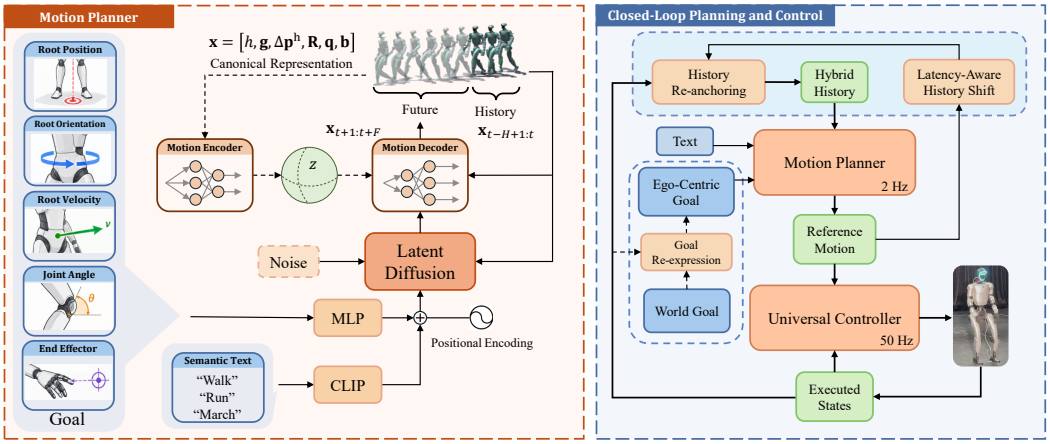

Figure 2: *Interactive text-conditioned Generated Motion Sequence on motion controller.*

Canonical Motion Representation. Existing motion representations commonly decompose root orientation into roll, pitch, and yaw and construct a heading-aligned local frame, which can suffer from gimbal-lock singularities under large body rotations. We instead represent root motion through relative rigid-body transformations between consecutive root frames and treat horizontal and vertical quantities asymmetrically, using incremental horizontal quantities and direct vertical states. This canonical representation is invariant to global horizontal translation and heading and remains continuous across arbitrary root orientations, including upright, tilted, and fallen configurations. We define each motion frame as

$$\mathbf{x}_t = \left[h_t, \mathbf{g}_t, {}^{E_{t-1}}\Delta\mathbf{p}^{\mathrm{h}}_{t-1}, {}^{E_{t-1}}\mathbf{R}_{E_t}, \mathbf{q}_t, \mathbf{b}_t\right], \tag{2}$$

where $\mathbf{R}_t \in \mathrm{SO}(3)$ and ${}^{W}\mathbf{p}_t \in \mathbb{R}^3$ denote the root orientation and position in the world frame, respectively. Here, h_t is the root height above ground, $\mathbf{g}_t = \mathbf{R}_t^\top \mathbf{g} \in \mathbb{R}^3$ is the gravity direction expressed in the current root frame, $\mathbf{q}_t \in \mathbb{R}^{29}$ contains the joint angles of the 29-DoF Unitree G1, and $\mathbf{b}_t \in \{0,1\}^2$ denotes binary foot-contact states. The relative root translation and rotation are defined as ${}^{E_{t-1}}\Delta\mathbf{p}_{t-1} = \mathbf{R}_{t-1}^\top({}^{W}\mathbf{p}_t - {}^{W}\mathbf{p}_{t-1})$ and ${}^{E_{t-1}}\mathbf{R}_{E_t} = \mathbf{R}_{t-1}^\top\mathbf{R}_t$, respectively, with ${}^{E_{t-1}}\Delta\mathbf{p}^{\mathrm{h}}_{t-1}$ obtained by projecting ${}^{E_{t-1}}\Delta\mathbf{p}_{t-1}$ onto the horizontal plane orthogonal to $\mathbf{g}_{t-1}$. All rotations provided to the network are encoded using the continuous 6D representation (Zhou et al., 2019), yielding a 44-dimensional motion representation.

Goal-Conditioned Motion Generation. The planner supports optional geometric and semantic conditions, which can be provided independently, jointly, or omitted entirely. Geometric conditions specify explicit goals at time $t^\star$, including root position ${}^{E_t}\Delta\mathbf{p}^{\mathrm{root}}_{t^\star}$, root orientation ${}^{E_t}\mathbf{R}^{\mathrm{root}}_{t^\star}$, root linear velocity ${}^{E_t}\mathbf{v}^{\mathrm{root}}_{t^\star}$, joint configuration $\mathbf{q}_{t^\star}$, and hand and foot end-effector positions ${}^{E_t}\Delta\mathbf{p}^{\mathrm{ee}}_{t^\star} \in \mathbb{R}^{4\times3}$. An optional time-to-arrival condition resolves temporal ambiguity by specifying when the goal should be reached. At each replanning step, spatial goals are re-expressed in the current root coordinate system, allowing the planner to respond naturally to changing targets despite being trained on fixed motion clips. Text provides complementary semantic guidance for motion styles that are difficult to specify through sparse geometric goals, rather than encoding complex task procedures. We show in Sec. 4 that this formulation supports diverse static and dynamic goal-conditioned applications.

Our planner adopts the VAE–LDM formulation used in DART (Zhao et al., 2025) and TextOp (Xie et al., 2026). We first train a VAE to encode future motion chunks into a compact latent space and reconstruct them conditioned on recent motion history, and then freeze the VAE while training a latent diffusion model over the motion latents. During LDM training, we randomly sample a target frame from each future chunk, use its geometric state to construct the goal condition, and take the corresponding frame offset as the time-to-arrival condition. The sampled goal frame also provides the geometric goal error used for auxiliary supervision. To reduce autoregressive exposure bias, we further adopt self-rollout training, progressively replacing ground-truth history with model-generated motion during sequential chunk prediction.

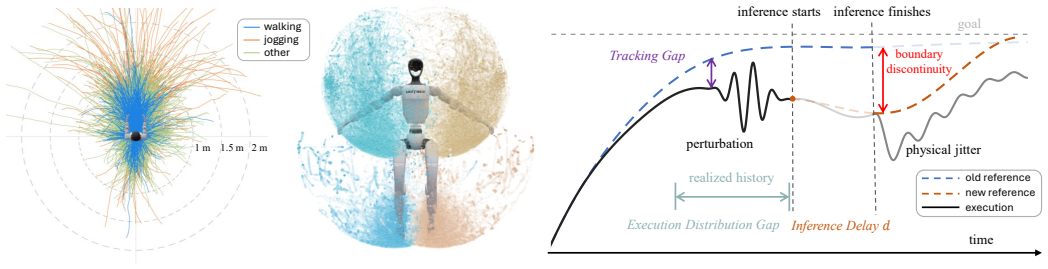

Figure 3: ***Geometry coverage and planning–control gaps.*** *(a) Egocentric root-trajectory distribution in the training data. (b) Data-derived reachable workspace of hand and foot end-effectors. (c) Planning–control gaps caused by tracking deviation, execution-distribution shift, and inference delay, leading to reference discontinuities and physical jitter.*

3.2 Execution-Consistent Motion Closure

Closing the motion loop requires the planner to remain reliable across diverse geometric goals, absorb realized-state feedback without leaving the training distribution, and recover from execution-induced deviations. We address these challenges through goal-aware data construction, state-history fusion, and recovery-oriented training.

Motion Clips with Geometry Coverage. To make training motion cover the geometry goal as much space as possible, we use retargeted AMASS (Mahmood et al., 2019), LAFAN (Harvey et al., 2020) and HiPHI (Ji et al., 2026) through GMR (Araujo et al., 2025), segmented into fixed-length clips, and converted to the canonical representation in Eq. (2). We geometrically filter the retargeted motions by removing clips with severe self-collision or sustained center-of-mass displacement far outside the support region of the feet, and further discard motion categories that do not admit meaningful geometric goal specification. After filtering, we obtain 33,488 training clips and 1,764 validation clips across 189 action categories. To further improve directional coverage of geometric goals, we increase the sampling probability of clips associated with less frequent target directions during training. The resulting egocentric root and end-effector distributions are shown in Fig. 3(a,b), covering a broad range of reachable goals for real-world deployment.

Hybrid State via History Re-anchoring. At planning time t, the realized robot state generally deviates from its planner prediction. We construct each hybrid history state by applying the current tracking deviation consistently to the generated history:

$$\mathbf{s}_\tau^{\mathrm{hyb}} = \boldsymbol{\delta}_t \oplus \mathbf{s}_\tau^{\mathrm{pred}}, \qquad \boldsymbol{\delta}_t = \mathbf{s}_t^{\mathrm{real}} \ominus \mathbf{s}_t^{\mathrm{pred}}, \tag{3}$$

where τ denotes an arbitrary frame in the planner history. The operators $\ominus$ and $\oplus$ denote state-wise deviation and composition, respectively, using rigid-body composition for the root pose and additive offsets for joint configurations. In particular, $\mathbf{s}_t^{\mathrm{hyb}} = \mathbf{s}_t^{\mathrm{real}}$, so the latest realized state is directly incorporated into the planner history, and earlier frames are reconstructed from planner-generated kinematics rather than controller-executed trajectories, avoiding dynamics-dependent execution histories and their high-frequency fluctuations. The reanchored states are then converted to the canonical motion representation $\mathbf{x}$ and used as planner history.

Deviation-Decayed Recovery. To expose the planner to tracking deviations encountered by the hybrid history at deployment, we apply the same uniformly sampled deviation $\tilde{\boldsymbol{\delta}}$ coherently to the entire history using the state transformation in Eq. (3). Over the future trajectory, the deviation progressively decays within a random recovery horizon, guiding the motion back toward the original ground-truth trajectory. This constructs a recoverable neighborhood around the motion manifold, allowing the planner to correct tracking deviations before they accumulate into persistent planning drift. For get-up motions, we apply stronger deviations only to the preceding fallen states while preserving the original get-up trajectory. This constructs additional recovery samples from diverse fallen configurations that converge to plausible pre-get-up states before entering the original get-up motion, increasing both the diversity and proportion of get-up examples in training.

3.3 Asynchronous Planning and Control Mechanism

For real-time deployment, we propose two mechanisms for asynchronous planning and control via a fast-in-slow dual-system architecture and latency-aware history shifting. Asynchronous system try to keep high-frequency control non-blocking and reducing reference discontinuity caused by inference latency. Although asynchronous execution prevents planner inference from blocking the controller, the robot continues following the previous reference during generation and may therefore advance beyond the state from which the new chunk was planned.

Fast-in-Slow Dual-System Architecture. Planning and control run as independent processes and communicate through a publisher–subscriber protocol. The planner replans at 2 Hz from the latest hybrid history with the goal re-expressed in the current egocentric frame, and publishes the resulting reference motion for the controller to track at 50 Hz; the realized robot state is then fed back to both the controller and planner. During planner inference, the controller continues tracking the latest available reference and switches to the newly published trajectory once it arrives. This fast-in-slow design enables non-blocking replanning while maintaining uninterrupted high-frequency control.

Latency Compensation via History Shift. To reduce the discontinuity caused by inference latency, RTC (Black et al., 2026; 2025) notes that blending independently generated chunks can produce invalid intermediate actions and instead accounts for committed actions during inference or training. However, we regard it as a temporal misalignment of the autoregressive history and compensate for it through a predictive history shift. Let d denote the expected number of motion steps before the new reference becomes available, estimated by an exponential moving average of recent inference times. We select the predicted history window $[t+d-H+1, t+d]$ and construct it bidirectionally from the latest realized state $\mathbf{s}_t^{\text{real}}$, using the predicted relative kinematics to reconstruct the past and propagate forward toward the anticipated handoff state. The goal is then re-expressed relative to the anticipated handoff state $\mathbf{s}_{t+d}^{\text{hyb}}$. This simple shift aligns both history and goal conditioning with the expected reference switch, reducing latency-induced handoff inconsistency without output blending, inference-time inpainting, or delay-aware retraining.

4 Experiments

To systematically analyze our proposed method, we adopt a comprehensive experiments covering **Motion Quality and Executability**, **Goal-Reaching Precision**, and **Robustness and Recovery**, and answering the following questions: (**Q1**) Can HIPC generate diverse and executable motions that can be tracked by a general controller? (**Q2**) Can HIPC accurately reach different geometric goal conditions? (**Q3**) Is HIPC robust from out-of-distribution physical states?

4.1 Experimental Setup and Real-World Deployment

All experiments use the 29-DoF Unitree G1, with HIPC as the high-level planner and SONIC (Luo et al., 2026) as the downstream tracker. HIPC replans asynchronously at 2 Hz, while SONIC tracks the latest reference at 50 Hz. For comparison, each baseline uses the same SONIC tracker and

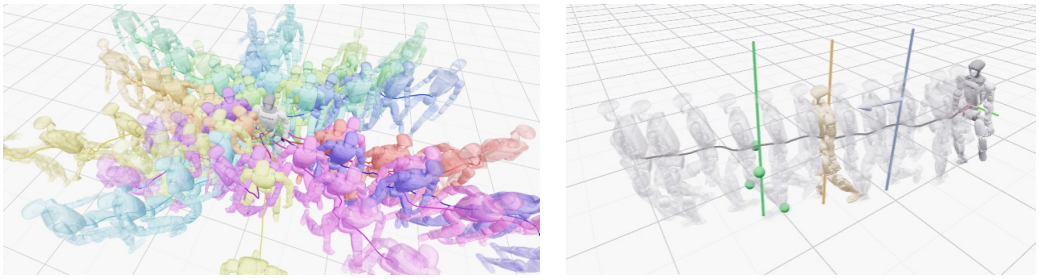

Figure 4: ***Static and switching goal reaching.*** *Trajectories from diverse initial states converge to fixed goals and smoothly adapt to sequentially switched goals.*

Table 2: ***Comparison of planning-motion quality and tracker-conditioned execution.*** *"→" denotes closer agreement with the dataset reference.* ***Bold*** *values denote the best results excluding the dataset. Mean ± standard deviation over three rollout seeds.*

(a) Replanning smoothness and geometric quality of planning motion

Method	Replanning Smoothness		Ground Penetration		Self-Penetration		Foot–Ground Contact	
	PJ →	AUJ ↓	Time (%) ↓	Depth (cm) ↓	Time (%) ↓	Depth (cm) ↓	Skate (cm/s) ↓	Float (cm) ↓
Dataset	$0.166_{\pm 0.717}$	$0.915_{\pm 1.420}$	$93.000_{\pm 14.551}$	$2.963_{\pm 0.784}$	$29.440_{\pm 37.016}$	$3.119_{\pm 1.688}$	$0.886_{\pm 1.631}$	$0.139_{\pm 1.058}$
Kimodo	$0.982_{\pm 0.778}$	$4.991_{\pm 3.505}$	$44.427_{\pm 19.989}$	$2.462_{\pm 1.697}$	$9.710_{\pm 14.081}$	$2.981_{\pm 1.604}$	$\mathbf{6.885}_{\pm \mathbf{76.076}}$	$0.129_{\pm 1.368}$
ARDY	$0.822_{\pm 0.482}$	$4.170_{\pm 2.883}$	$\mathbf{28.057}_{\pm \mathbf{11.985}}$	$2.291_{\pm 1.217}$	$15.250_{\pm 17.508}$	$\mathbf{2.964}_{\pm \mathbf{1.535}}$	$29.663_{\pm 117.040}$	$0.445_{\pm 2.240}$
TextOp	$1.003_{\pm 0.959}$	$6.946_{\pm 6.606}$	$70.987_{\pm 21.212}$	$5.163_{\pm 5.111}$	$34.243_{\pm 27.974}$	$3.831_{\pm 2.146}$	$103.696_{\pm 288.884}$	$1.146_{\pm 4.905}$
OMG	$0.915_{\pm 0.461}$	$6.388_{\pm 2.847}$	$52.363_{\pm 19.597}$	$2.909_{\pm 3.089}$	$30.160_{\pm 27.405}$	$3.685_{\pm 1.884}$	$85.147_{\pm 163.188}$	$0.940_{\pm 3.133}$
HIPC (Ours)	$\mathbf{0.492}_{\pm \mathbf{0.264}}$	$\mathbf{2.894}_{\pm \mathbf{1.166}}$	$90.920_{\pm 13.026}$	$\mathbf{2.080}_{\pm \mathbf{0.730}}$	$\mathbf{5.350}_{\pm \mathbf{17.259}}$	$3.773_{\pm 2.104}$	$17.378_{\pm 62.129}$	$\mathbf{0.042}_{\pm \mathbf{0.380}}$

(b) Goal-reaching, distributional, and semantic quality of tracking result

Method	Execution		Motion Distribution		Text–Motion Alignment			
	Fall (%) ↓	E_{pos} (m) ↓	FID ↓	DIV →	R@1 ↑	R@2 ↑	R@3 ↑	MM-Dist ↓
Dataset	–	–	$0.000_{\pm 0.000}$	$1.188_{\pm 0.008}$	$0.178_{\pm 0.022}$	$0.270_{\pm 0.021}$	$0.339_{\pm 0.017}$	$1.301_{\pm 0.080}$
Kimodo	$7.000_{\pm 25.515}$	$0.457_{\pm 0.408}$	$0.621_{\pm 0.001}$	$\mathbf{1.190}_{\pm \mathbf{0.007}}$	$\mathbf{0.132}_{\pm \mathbf{0.024}}$	$\mathbf{0.210}_{\pm \mathbf{0.024}}$	$\mathbf{0.286}_{\pm \mathbf{0.023}}$	$1.329_{\pm 0.002}$
ARDY	$1.667_{\pm 12.802}$	$0.242_{\pm 0.238}$	$0.605_{\pm 0.006}$	$1.228_{\pm 0.005}$	$0.100_{\pm 0.012}$	$0.173_{\pm 0.024}$	$0.241_{\pm 0.024}$	$1.338_{\pm 0.002}$
TextOp	$24.667_{\pm 43.107}$	–	$0.743_{\pm 0.009}$	$1.238_{\pm 0.002}$	$0.079_{\pm 0.005}$	$0.157_{\pm 0.000}$	$0.231_{\pm 0.006}$	$1.333_{\pm 0.000}$
OMG	$7.000_{\pm 25.515}$	–	$0.747_{\pm 0.032}$	$1.194_{\pm 0.012}$	$0.075_{\pm 0.011}$	$0.139_{\pm 0.016}$	$0.203_{\pm 0.022}$	$\mathbf{1.322}_{\pm \mathbf{0.012}}$
HIPC (Ours)	$\mathbf{0.000}_{\pm \mathbf{0.000}}$	$\mathbf{0.101}_{\pm \mathbf{0.067}}$	$\mathbf{0.574}_{\pm \mathbf{0.000}}$	$1.194_{\pm 0.000}$	$0.106_{\pm 0.000}$	$0.175_{\pm 0.000}$	$0.236_{\pm 0.000}$	$1.324_{\pm 0.000}$

2 Hz closed-loop protocol; evaluations are offline unless otherwise stated to avoid inference-latency effects.

For real-world deployment, HIPC and SONIC run off-board on an NVIDIA RTX 4070 GPU, with motion capture providing robot poses for egocentric goal construction. Across five goal configurations with three trials each, HIPC responds online to goal switches and corrects spatial deviations, including target overshoot. If the robot falls during locomotion, it can recover to standing and continue toward the goal without manual intervention. Additional details are provided in the Appendix.

4.2 Analysis for Motion Quality and Executability

To address **Q1**, we evaluate HIPC from two perspectives: the quality of the generated motions and their executed motions. The former focuses on replanning smoothness and geometric plausibility, while the latter evaluates whether the generated references remain stable, goal-directed, and semantically meaningful after physical execution. All metric definitions are provided in the Appendix.

Comparison on Motion Quality. We evaluate planning-motion quality on 100STYLE (Mason et al., 2022), using sampled motion histories and future root positions as planar goals. As shown in Table 2, HIPC substantially improves replanning smoothness and geometric quality, with lower boundary jerk, limited penetration, and reduced foot artifacts, while preserving motion diversity and text–motion alignment comparable to existing generative baselines. These improvements also translate to physical execution: after SONIC tracking, HIPC achieves zero falls and the lowest endpoint error among goal-conditioned baselines (0.101 m), together with the best FID. Overall, the results indicate that HIPC produces references that are not only smoother across replanning boundaries, but also more physically executable without sacrificing the distributional and semantic properties of generated motion.

Ablation on Motion Closure Mechanism. Table 3 isolates the effects of execution feedback, history re-anchoring, and latency compensation. Predicted history avoids execution-induced distribution shift but leaves tracking deviations unobserved, causing spatial drift, while directly using realized histories introduces dynamics-dependent mismatch and less smooth handoff. History re-anchoring resolves this trade-off by anchoring planner-native history to the realized state while preserving its kinematic structure, substantially improving goal accuracy. The additional reconstruction increases average end-to-end planning latency from 92 ms to 120 ms, still within the 2 Hz replanning budget. Latency compensation then accounts for state evolution during this interval, recovering

smooth and stable handoff. Together, these results validate execution-consistent history re-anchoring and latency-aware replanning for closing the planning–control loop.

Table 3: ***Ablation of motion-closure mechanisms.*** *We compare planner-history constructions and handoff strategies that isolate execution feedback, history re-anchoring, and latency compensation. Metrics evaluate goal accuracy, stability, and seam smoothness; results are mean±std over three seeds.*

Iterative Design		Goal Reaching		Smoothness
Planner History	**Chunk Handoff**	E_{pos} **(m)** ↓	**Fall (%)** ↓	**Seam PJ** ↓
Predicted	Direct Switching	0.371 ±0.037	0.000 ±0.000	0.005 ±0.000
Realized	Direct Switching	0.547 ±0.093	4.444 ±3.143	0.012 ±0.002
Hybrid	Direct Switching	0.147 ±0.028	2.222 ±3.143	0.009 ±0.002
Hybrid	**Latency Compensation**	**0.129** ±0.017	**0.000** ±0.000	**0.005** ±0.000

4.3 Analysis for Goal Reaching Ability

To answer **Q2**, we evaluate HIPC from two complementary aspects: unified realization of diverse static goal conditions and online tracking of dynamically updated goals from upstream planners.

Static Goal Achievement. We evaluate HIPC on 330 trials covering 11 goal configurations, with 30 trials per setting and states and targets sampled from 100STYLE (Mason et al., 2022). Table 4 shows that a single planner supports root position, orientation, velocity, joint, and end-effector goals, together with representative compositions. While individual goals are matched accurately, composed conditions exhibit cross-goal interference: increasing the number of constraints does not consistently improve each component and can reduce its accuracy as multiple objectives compete within the same motion. This trade-off is most apparent for full-state goals, and adding text introduces a further modest degradation, showing that semantic and geometric conditioning can be combined but are not completely decoupled.

Dynamic Goal Tracking. As shown in Fig. 5, we evaluate HIPC in six Replica indoor scenes (Straub et al., 2019) imported into Isaac Sim, using two upstream navigation modules: an occupancy-map-based A* planner with obstacle inflation and the vision-and-language navigation model OmniNav (Xue et al., 2025). Their waypoints are transformed into HIPC's egocentric goal conditions for online navigation. Across 50 navigation trials each, A* and OmniNav achieve 90% and 70% success in reaching the final navigation goal, respectively. The gap reflects the greater reliability of map-based geometric planning, while the VLN result shows that HIPC can reliably track dynamically updated navigation goals from higher-level upstream planners, supporting its use as a planner-agnostic whole-body execution interface.

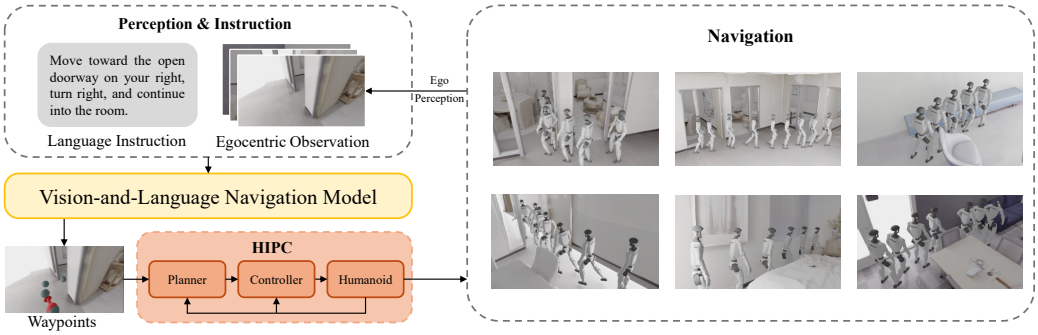

Figure 5: ***Dynamic goal tracking for navigation.*** *HIPC executes dynamically updated waypoints from vision-and-language navigation planners as egocentric whole-body goals in Replica scenes.*

Table 4: ***Goal reaching under individual and representative composed conditions.*** *For individual goals, "Error" denotes the corresponding goal-specific metric. Representative compositions are evaluated jointly over all specified components; additional combinations are reported in the Appendix.*

(a) Individual goals

Goal Condition	Error
Root Position	0.079 ± 0.049
Root Orientation	0.092 ± 0.026
Joint Configuration	0.129 ± 0.085
Root Velocity	0.361 ± 0.303
End-Effector Position	0.156 ± 0.051

(b) Composed goals

Goal Condition	E_{pos} (m) ↓	E_{ori} (rad) ↓	$E^{\text{goal}}_{\text{MPJPE}}$ (m) ↓	E_{vel} (m/s) ↓	E_{ee} (m) ↓
Position + Orientation	0.110 ± 0.069	$\mathbf{0.082} \pm 0.044$	–	–	–
Position + Joint	0.112 ± 0.060	–	0.116 ± 0.081	–	–
Position + Velocity	0.095 ± 0.061	–	–	$\mathbf{0.576} \pm 0.469$	–
Position + End-Effector	$\mathbf{0.085} \pm 0.038$	–	–	–	$\mathbf{0.146} \pm 0.035$
Full-state Goal	0.130 ± 0.063	0.133 ± 0.053	$\mathbf{0.056} \pm 0.016$	0.577 ± 0.470	0.182 ± 0.076
Full-state Goal + Text	0.142 ± 0.137	0.166 ± 0.121	0.069 ± 0.029	0.579 ± 0.455	0.206 ± 0.142

Table 5: ***Robustness to physical perturbations and fall recovery.*** *(a) evaluates goal reaching after randomized external disturbances. (b) induces falls with a 500 N horizontal force for 0.1 s in a uniformly random direction, then commands a same-position standing goal and reports recovery rate and get-up time over successful trials.*

(a) Robustness under perturbations

Method	Fall (%) ↓	E_{pos} (m) ↓	Response PJ ↓
TextOp	29.167 ± 0.851	1.615 ± 0.065	0.070 ± 0.007
Kimodo	7.986 ± 2.140	0.443 ± 0.014	0.059 ± 0.005
ARDY	2.431 ± 1.299	0.251 ± 0.022	0.045 ± 0.005
OMG	8.681 ± 4.019	2.642 ± 0.109	0.068 ± 0.005
HIPC (Ours)	$\mathbf{0.000} \pm 0.000$	$\mathbf{0.105} \pm 0.005$	$\mathbf{0.034} \pm 0.001$

(b) Recovery after induced falls

Variant	Recovery (%) ↑	T_{getup} (s) ↓
Heading-based Representation	0.000 ± 0.000	–
w/o History Re-anchoring	0.000 ± 0.000	–
w/o Deviation-Decayed Recovery	0.000 ± 0.000	–
HIPC	$\mathbf{53.333} \pm 9.428$	$\mathbf{7.227} \pm 0.318$

4.4 Robustness and Recovery Analysis

To answer **Q3**, we evaluate whether HIPC can close the planning–control loop under large execution deviations caused by transient perturbations and dynamically induced falls. We assess its ability to preserve goal-directed motion after disturbances and recover from fallen states through endpoint error, fall rate, response smoothness, recovery success, and get-up time.

Robustness to Physical Perturbations. Following the same goal-reaching protocol as Sec. 4.2, we apply randomized external forces and evaluate post-perturbation stability, goal preservation, and response smoothness. As shown in Table 5(a), HIPC achieves zero falls together with the lowest endpoint error and response PJ, demonstrating that execution-consistent replanning can correct disturbance-induced deviations while preserving both task accuracy and smooth physical execution.

Fall Recovery. We induce diverse falls using a 500 N horizontal force for 0.1 s with random direction, then command an upright standing goal at the same position. HIPC recovers in 53.33% of trials with a mean get-up time of 7.23 s, whereas the heading-based representation and variants without history re-anchoring or deviation-decayed recovery achieve no successful recoveries (Table 5(b)). This confirms the importance of yaw-free representation, execution-consistent history construction, and recovery-oriented training.

5 Conclusion and Limitation

Conclusion. We presented HIPC, a closed-loop humanoid planning-and-control framework that integrates goal-conditioned generative motion planning with execution-consistent feedback and latency-aware asynchronous control. By closing the loop between motion generation and physical execution while preserving planner–controller modularity, HIPC supports accurate goal reaching, robust perturbation recovery, and recovery from unseen fallen configurations across diverse whole-body tasks.

Limitations. Despite these strengths, HIPC currently closes the loop primarily between motion planning and physical execution, while relying on externally provided goal states rather than direct environment perception. In real-world deployment, accurate spatial goals are therefore obtained from external localization systems such as motion capture, which increases sensing and communication complexity.

REFERENCES

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.

Kevin Black, Allen Z Ren, Michael Equi, and Sergey Levine. Training-time action conditioning for efficient real-time chunking. *arXiv preprint arXiv:2512.05964*, 2025.

Kevin Black, Manuel Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. *Advances in Neural Information Processing Systems*, 38:33383–33407, 2026.

Xiao Chen, Weishuai Zeng, Xiaojie Niu, Zirui Wang, Jianan Li, Huayi Wang, Furui Xu, Jiahe Chen, Weixiang Zhong, Lihe Ding, et al. Reactivebfm: Reactive closed-loop motion planning towards universal humanoid whole-body control. *arXiv preprint arXiv:2606.30362*, 2026.

Xingyu Chen. Text-driven human motion generation with motion masked diffusion model. *arXiv preprint arXiv:2409.19686*, 2024.

Yan Gu, Guanya Shi, Fan Shi, I-Chia Chang, Yen-Jen Wang, Qilong Cheng, Zachary Olkin, Ivan Lopez-Sanchez, Yunchu Feng, Jian Zhang, et al. Evolution of humanoid locomotion control. *Science Robotics*, 11(117):eaed3973, 2026.

Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. Momask: Generative masked modeling of 3d human motions. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1900–1910. IEEE, 2024.

Félix G. Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion in-betweening. 39(4), 2020.

Jiahao Ji, Ji Ma, Runhan Zhang, Runyi Yu, Wenjia Wang, Weiheng Chi, Qianqian Peng, Weichao Yan, Yongfei Gu, Ye Tian, et al. Hiphi: A large-scale benchmark for high-precision human motion and object-interaction. *arXiv preprint arXiv:2608.16222*, 2026.

Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems*, 36:20067–20079, 2023.

Dongting Li, Xingyu Chen, Qianyang Wu, Bo Chen, Sikai Wu, Hanyu Wu, Guoyao Zhang, Liang Li, Mingliang Zhou, Diyun Xiang, et al. Haic: Humanoid agile object interaction control via dynamics-aware world model. *arXiv preprint arXiv:2602.11758*, 2026a.

Dongting Li, Qianyang Wu, Xingyu Chen, Liang Li, Yuhang Lin, Sikai Wu, Guoyao Zhang, Mingliang Zhou, Diyun Xiang, Qiang Zhang, et al. Vaic: Vision-guided humanoid agile object interaction control via decoupled commands. *arXiv preprint arXiv:2606.09286*, 2026b.

Jiefeng Li, Jinkun Cao, Haotian Zhang, Davis Rempe, Jan Kautz, Umar Iqbal, and Ye Yuan. Genmo: A generalist model for human motion. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11766–11776. IEEE, 2025.

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. *Science Robotics*, 11(117):eadx8924, 2026.

Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Fernando Castañeda, Sirui Chen, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, et al. Sonic: Supersizing motion tracking for natural humanoid whole-body control. *Science Robotics*, 11(117):eaed4592, 2026.

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*, pp. 5442–5451, 2019.

Ian Mason, Sebastian Starke, and Taku Komura. Real-time style modelling of human locomotion via feature-wise transformations and local motion phases. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 5(1), may 2022. doi: 10.1145/3522618.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4):1–14, 2018.

Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (ToG)*, 40(4):1–20, 2021.

Davis Rempe, Mathis Petrovich, Ye Yuan, Haotian Zhang, Xue Bin Peng, Yifeng Jiang, Tingwu Wang, Umar Iqbal, David Minor, Michael de Ruyter, et al. Kimodo: Scaling controllable human motion generation. *arXiv preprint arXiv:2603.15546*, 2026.

RoboParty Lab Team. Mimiclite: Efficient and effective general humanoid motion tracking. `https://github.com/EGalahad/mimic-lite`, 2026. Technical report: `https://github.com/Roboparty/MimicLite/blob/main/mimic-lite.pdf`.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

Zelin Tao, Zeran Su, Peiran Liu, Jingkai Sun, Wenqiang Que, Jiahao Ma, Jialin Yu, Jiahang Cao, Pihai Sun, Hao Liang, et al. Heracles: Bridging precise tracking and generative synthesis for general humanoid control. *arXiv preprint arXiv:2603.27756*, 2026.

Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=SJ1kSyO2jwu`.

Guy Tevet, Sigal Raab, Setareh Cohan, Daniele Reda, Zhengyi Luo, Xue Bin Peng, Amit Bermano, and Michiel Van de Panne. Closd: Closing the loop between simulation and diffusion for multi-task character control. In *International Conference on Learning Representations*, volume 2025, pp. 46506–46520, 2025.

Tingwu Wang, Olivier Dionne, Michael De Ruyter, David Minor, Davis Rempe, Kaifeng Zhao, Mathis Petrovich, Ye Yuan, Chenran Li, Zhengyi Luo, Brian Robison, Xavier Blackwell, Bernardo Antoniazzi, Xue Bin Peng, Yuke Zhu, and Simon Yuen. Motionbricks: Scalable real-time motions with modular latent generative model and smart primitives, 2026a. URL `https://arxiv.org/abs/2604.24833`.

Yinhuai Wang, Qihan Zhao, Yuen Fui Lau, Runyi Yu, Hok Wai Tsui, Qifeng Chen, Jingbo Wang, Jiangmiao Pang, and Ping Tan. Humanx: Toward agile and generalizable humanoid interaction skills from human videos. *arXiv preprint arXiv:2602.02473*, 2026b.

Weiji Xie, Jiakun Zheng, Jinrui Han, Jiyuan Shi, Weinan Zhang, Chenjia Bai, and Xuelong Li. Textop: Real-time interactive text-driven humanoid robot motion generation and control. *arXiv preprint arXiv:2602.07439*, 2026.

Qingyao Xu, Sheng Yin, Zibo Zhou, Ya Zhang, Siheng Chen, and Yue Hu. Glori: Closed-loop whole-body tracking with global-local reference interaction for humanoid loco-manipulation. *arXiv preprint arXiv:2609.05994*, 2026a.

Weisheng Xu, Qiwei Wu, Jiaxi Zhang, Jing Tan, Yangfan Li, Yuetong Fang, Jiaqi Xiong, Kai Wu, Rong Ou, and Renjing Xu. Iterative closed-loop motion synthesis for scaling the capabilities of humanoid control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16398–16407, 2026b.

Xinda Xue, Junjun Hu, Minghua Luo, Xie Shichao, Jintao Chen, Zixun Xie, Quan Kuichen, Guo Wei, Mu Xu, and Zedong Chu. Omninav: A unified framework for prospective exploration and visual-language navigation. *arXiv preprint arXiv:2509.25687*, 2025.

Weishuai Zeng, Kangning Yin, Xiaojie Niu, Shunlin Lu, Weixiang Zhong, Jiahe Chen, Feiyu Jia, Xiao Chen, Zirui Wang, Furui Xu, et al. Scaling behavior foundation model for humanoid robots. *arXiv preprint arXiv:2607.15163*, 2026.

Kaifeng Zhao, Gen Li, and Siyu Tang. Dartcontrol: A diffusion-based autoregressive motion model for real-time text-driven motion control. In *International Conference on Learning Representations*, volume 2025, pp. 23569–23592, 2025.

Kaifeng Zhao, Mathis Petrovich, Haotian Zhang, Tingwu Wang, Siyu Tang, and Davis Rempe. Autoregressive diffusion with hybrid representation for interactive human motion generation. *ACM Transactions on Graphics (TOG)*, 45(4):1–14, 2026.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5738–5746. IEEE, 2019.