# HUMANOID MOTION FOUNDATION MODEL FOR MULTIMODAL STREAMING CLOSED-LOOP INTERACTION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Recent advances in vision-language-action (VLA) systems have enabled increasingly general-purpose embodied behaviors. However, existing approaches still formulate humanoid interaction as a static conditional generation problem, limiting their ability to support temporally evolving multimodal interaction and execution-aware behavioral adaptation in open-world environments. In this paper, we propose **MSC**, a multimodal streaming behavioral framework for closed-loop humanoid interaction. Instead of conditioning humanoid behaviors on fixed multimodal conditions, **MSC** reformulates humanoid control as a streaming conditional flow process under continuously evolving multimodal and execution-aware states. Built upon a structured conditioning representation and a streaming flow matching mechanism, **MSC** enables continuous interaction and closed-loop behavioral refinement during execution. Experiments show that **MSC** consistently improves motion quality, multimodal alignment, and robustness across text-, vision-, and audio-conditioned humanoid control tasks. We further demonstrate real-world deployment on Unitree G1 humanoids under interleaved multimodal interactions, validating the practicality of streaming closed-loop humanoid interaction in real-world environments.

## 1 INTRODUCTION

Recent advances in vision-language models (VLMs) have enabled powerful multimodal understanding across diverse perceptual tasks Achiam et al. (2023); Bai et al. (2025); Zhu et al. (2025); Team et al. (2025); Comanici et al. (2025), while VLA models further extends these capabilities from passive understanding to embodied action generation Brohan et al. (2022); Zitkovich et al. (2023); Kim et al. (2024); O'Neill et al. (2024); Black et al. (2024). Building upon this progress, an emerging goal in embodied AI is the development of general-purpose humanoid agents capable of interacting with open-world environments through continuous multimodal instructions Driess et al. (2023); Ahn et al. (2022); Zitkovich et al. (2023) and feedback Abramson et al. (2022); Lin et al. (2020); Ramrakhya et al. (2025). Such agents should not only understand multimodal signals jointly, but also adapt their behaviors during execution. For example, while performing an ongoing task, a humanoid agent may receive additional spoken instructions, visual cues, or environmental feedback that alters the intended behavior in real time. This requires action generation to operate as a continuous and feedback-aware process, rather than a static one-shot prediction problem. Enabling such interleaved **multimodal** interaction and **closed-loop** behavioral adaptation therefore remains a fundamental challenge for next-generation humanoid VLA systems.

To move toward this goal, recent research has increasingly adopted a general-purpose paradigm for humanoid control and embodied interaction. Instead of relying on manually designed trajectories Kuindersma et al. (2016); Winkler et al. (2018) or task-specific controllers Peng et al. (2018); Siekmann et al. (2021), these approaches learn motion priors from large-scale human motion data and align generated motion trajectories with high-level instructions Chen et al. (2025); Li et al. (2025d;c); Mees et al. (2022). The resulting trajectories are then tracked and executed as humanoid behaviors, demonstrating strong generalization across diverse tasks and embodiments.

However, current systems still rely on a fixed-conditioning and open-loop generation paradigm that differs from what is required for general-purpose VLA systems. **First, limited multimodal interaction.** Although recent approaches support multimodal input, they typically process only a single



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.

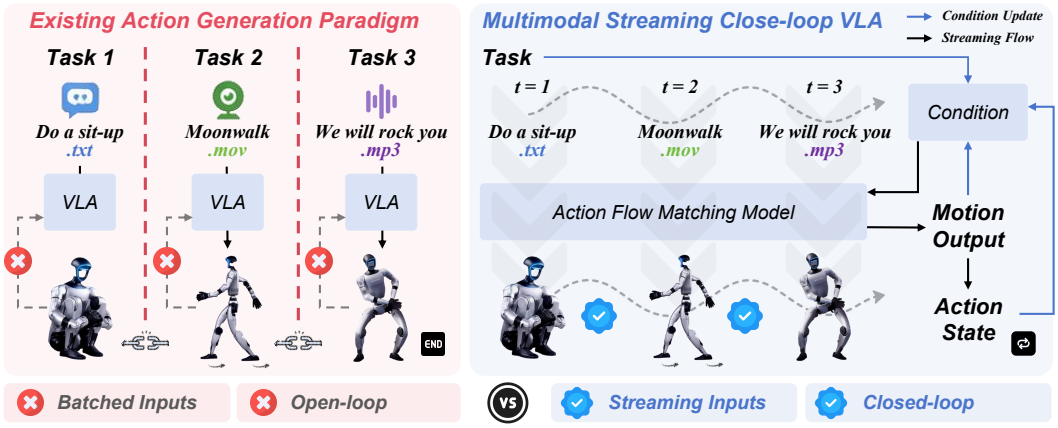



Figure 1: Comparison between existing open-loop humanoid action generation and the proposed multimodal streaming closed-loop VLA framework.

modality at a time during generation Jiang et al. (2025b); Li et al. (2025b); Wang et al. (2025). In practice, humanoid agents should continuously respond to interleaved signals during embodied interaction, such as newly issued language instructions, visual guidance, or environmental feedback that dynamically alter behavior. Existing methods therefore struggle to support asynchronous and temporally evolving multimodal interaction. **Second, open-loop behavioral execution.** Current approaches largely decouple motion generation from execution: trajectories are first generated and then tracked by downstream controllers Li et al. (2025a); Jiang et al. (2025a); Xie et al. (2026), while feedback from ongoing execution rarely influences subsequent behavioral evolution. As a result, these systems cannot adapt behaviors according to execution dynamics and real-time interaction, limiting their ability to support continuous interaction in open environments.

Based on the above analysis, we argue that an ideal general-purpose humanoid VLA system should support both interleaved multimodal interaction and streaming closed-loop action adaptation. Figure 1 illustrates the limitations of existing paradigms and the capabilities required for such systems.

To this end, we propose **MSC, a multimodal streaming behavioral framework for closed-loop humanoid interaction**. Instead of treating humanoid control as an open-loop generation problem, **MSC** reformulates it as a conditioning-driven flow process, where behaviors continuously evolve according to multimodal inputs and execution feedback. Specifically, we introduce a structured conditioning interface that unifies interleaved multimodal signals into a temporally evolving state for multi-source conditioning and feedback fusion. Building on this representation, we develop a streaming closed-loop behavioral generation mechanism based on conditioning-modulated flow matching, enabling continuous trajectory generation and execution-aware adaptation during interaction. Together, these components bridge multimodal understanding and closed-loop humanoid interaction within a unified, real-time feedback-aware framework.

Through extensive experiments, we investigate several key challenges targeted by **MSC** and provide empirical answers to its central research questions. Experimental results show that **MSC** preserves and improves humanoid motion generation performance across multiple modalities, enables streaming multimodal interaction under temporally evolving instructions, and achieves stronger robustness and adaptive behavior generation through closed-loop feedback under execution perturbations. Ablation studies further verify the complementary roles of the structured conditions and reveal that an intermediate policy update frequency provides the best balance between responsiveness and temporal alignment. Finally, we validate **MSC** on a real Unitree G1 humanoid robot, demonstrating the practicality of the proposed framework for real-world streaming multimodal interaction. Overall, our contributions are threefold:

- **Structured multi-rate conditioning.** We propose a structured conditioning state that decomposes interleaved multimodal signals into temporally aligned components, enabling interleaved conditioning.

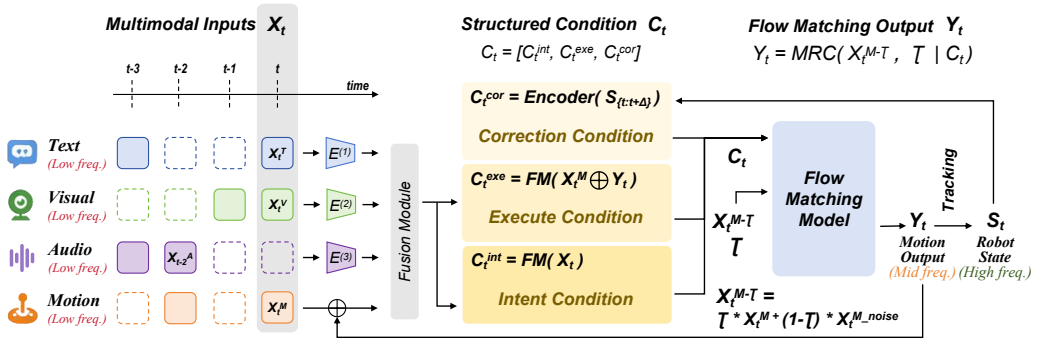



Figure 2: Structured conditioning for multimodal action generation. Interleaved multimodal inputs are fused into a conditioning state $c_t = [c_t^{\text{int}}, c_t^{\text{exe}}, c_t^{\text{cor}}]$ that separates intent, action dynamics, and feedback across different temporal scales. The resulting state modulates the flow matching model to serve streaming, closed-loop generation.

- **Streaming closed-loop flow matching.** We formulate action generation as a conditioning-modulated flow under a temporally evolving conditioning state, enabling streaming closed-loop refinement.
- **Streaming closed-loop interaction evaluation.** We construct streaming multimodal interaction and closed-loop robustness test for action generation, demonstrating that **MSC** enables continuous multimodal interaction and adaptive behavior generation under dynamically evolving conditions.

## 2 Related Works

### 2.1 Motion Generation

Existing motion generation methods mainly follow two paradigms: two-stage pipelines that first generate human motion trajectories and then retarget them onto robots, and direct robot motion generation methods that produce robot trajectories from high-level instructions. The former benefits from abundant human motion data and mature retargeting techniques, enabling more diverse motions, while the latter avoids retargeting but is often limited by the scarcity of large-scale robot motion data. However, both paradigms largely lack support for continuous multimodal interaction. Even recent multimodal approaches such as UniAct Jiang et al. (2025b) and OmniMotion Li et al. (2025d) typically process only a single modality during generation, making them difficult to extend to interleaved inputs. In contrast, **MSC** supports temporally evolving multimodal conditioning and enables action generation under continuous multimodal interaction.

### 2.2 Streaming Control

Real-time control is a key objective of existing motion generation methods. However, autoregressive models require full-sequence prediction Jiang et al. (2023); Zhang et al. (2023); Lucas et al. (2022), while diffusion-based methods rely on iterative denoising Tevet et al. (2022); Zhang et al. (2024), making them inefficient for latency-sensitive humanoid control. Although recent works improve efficiency through next-sequence prediction or action chunking, they mainly focus on generation speed rather than closed-loop adaptation. In contrast, **MSC** fully leverages streaming motion inputs by incorporating executed actions and historical trajectories as feedback signals, enabling real-time closed-loop motion generation and improved behavioral adaptation.

## 3 Method

Given a multimodal input stream $\mathcal{X} = \{x_t\}_{t=1}^{T}$, where inputs from different modalities arrive asynchronously over time, our goal is to generate humanoid action sequences in a streaming, closed-loop manner. At each step, the model updates its conditioning state from newly arrived inputs and execution feedback, and generates an action segment $Y_{t:t+H}$. Interleaved multimodal inputs are first

encoded into a structured conditioning state $C_t$ with multi-rate updates (Section 3.1); conditioned on $C_t$, a flow matching model generates an action segment $Y_{t:t+H}$ through continuous flow matching. The executed actions further produce environment and proprioceptive feedback, which are used to update $C_t$ for subsequent generation, forming a closed interaction loop (Section 3.2). The overall pipeline is illustrated in Fig. 2 and Fig. 3.

### 3.1 Structured Conditioning for Interleaved Multimodal Control

We introduce an explicit conditioning interface for humanoid control, where low-frequency intent, mid-frequency motion dynamics, and high-frequency state feedback are integrated into a structured state that evolves alongside action generation. At each control step $t$, the model generates a trajectory segment over a finite horizon $H$, denoted as $Y_{t:t+H}$. Unlike encoding the single-modality input on a static embedding, our formulation models conditioning as a dynamic, multi-source process with distinct temporal roles.

**Structured Conditioning Design.** We represent the conditioning signal as a structured state $c_t = [c_t^{\text{int}}, c_t^{\text{exe}}, c_t^{\text{cor}}]$, where each component captures a distinct aspect. The intent component $c_t^{\text{int}}$ encodes high-level semantics from multimodal inputs, the execution component $c_t^{\text{exe}}$ summarizes recent motion tokens and generated trajectories to model motion progression, and the correction component $c_t^{\text{cor}}$ aggregates high-frequency robot states to capture execution feedback and motion deviations. This decomposition explicitly separates semantic intent, motion evolution, and physical feedback within a unified control representation.

**Multi-Rate Condition Updates.** Each component of the conditioning state is updated at a different temporal scale. The intent state $c_t^{\text{int}}$ is updated asynchronously according to the temporal schedule of the multimodal inputs. The execution state $c_t^{\text{exe}}$ evolves with the generated motion segment $Y_{t:t+H}$, capturing mid-frequency trajectory dynamics at control step $t$. The correction state $c_t^{\text{cor}}$ is updated from a short window of high-frequency robot states $S_{t:t+\Delta}$, summarizing execution feedback within each action step. Here, $t$ indexes discrete control steps where trajectory segments are generated, and $H$ denotes the prediction horizon, i.e., the number of future motion steps generated at each control step. In contrast, $\Delta$ defines a shorter interval within a single action step for collecting high-frequency state observations, where typically $\Delta < H$. This distinction enables the model to capture both long-horizon motion structure and fine-grained execution feedback.

This design transforms conditioning from a static embedding into a structured, temporally evolving state aligned with the multi-rate nature of humanoid control. Multimodal perception is updated sparsely when new inputs arrive, motion evolves continuously through generated trajectories, and high-frequency state observations provide real-time feedback. Even without new multimodal inputs, the system continues evolving through ongoing motion execution and state feedback: the execution condition captures recently generated trajectories, while the correction condition reflects real-time deviations from high-frequency observations. As a result, the proposed conditioning interface enables continuous closed-loop adaptation across different temporal scales while maintaining a clear separation between intent, motion evolution, and physical feedback.

### 3.2 Streaming Closed-Loop Flow Matching

**Flow-Based Action Generation.** We adopt flow matching as the underlying action generation mechanism, modeling trajectory generation as a continuous transformation from noise to motion in action space. At each control step $t$, the model operates on an interpolated action variable $x_t^\tau$, where $\tau \in [0, 1]$ denotes the flow time. The model predicts a velocity field

$$Y_t = \text{MSC}(x_t^\tau, \tau \mid c_t),$$

which transports $x_t^\tau$ toward a structured trajectory $a_{t:t+H}$ over horizon $H$. Compared with diffusion-based methods, flow matching enables more efficient trajectory generation without iterative denoising while preserving a continuous formulation consistent with motion dynamics.

**Streaming Flow Matching.** Unlike conventional formulations that assume complete conditioning inputs before generation, we extend flow matching to a streaming setting where conditioning is represented as a temporally evolving state $c_t$ defined in Sec. 3.1. At each control step $t$, the model performs flow-based generation conditioned on the current $c_t$, regardless of whether new multimodal

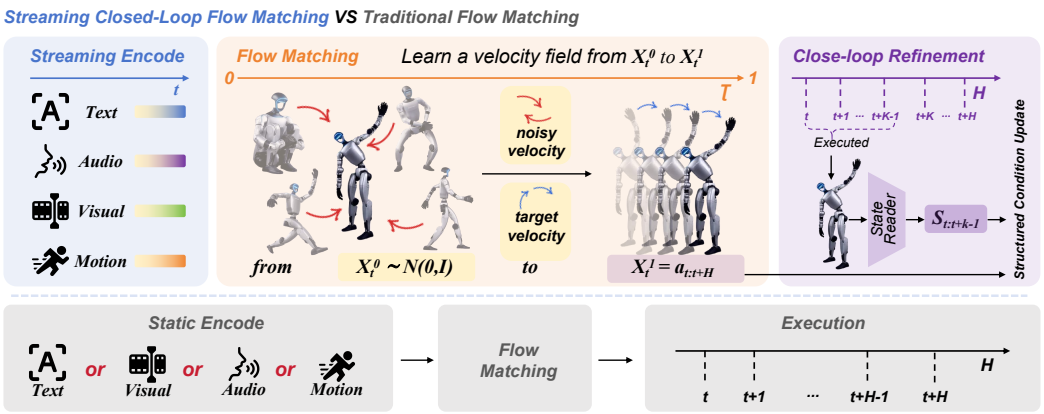



Figure 3: Streaming closed-loop flow matching vs. conventional flow matching. Our approach performs flow-based action generation under temporally evolving conditioning states in a streaming closed-loop setting.

inputs are available. This design decouples input availability from generation: multimodal inputs asynchronously update $c_t^{\text{int}}$, while motion execution and state feedback continuously update $c_t^{\text{exe}}$ and $c_t^{\text{cor}}$. As a result, generation becomes inherently streaming, where conditioning updates are governed by the control loop rather than external inputs. This enables continuous action generation driven by execution and feedback, even without new multimodal observations.

**Closed-Loop Action Refinement.** A key advantage of flow-based action generation is that both the input $x_t^\tau$ and output trajectory lie in the same motion space, enabling seamless integration with closed-loop control. After generating a trajectory segment $a_{t:t+H}$, the executed motion updates the robot state, which is encoded into $c_t^{\text{cor}}$ and fed back into the conditioning state, while the generated trajectory contributes to $c_t^{\text{exe}}$ to capture ongoing motion evolution. At the next control step, a new flow process is initialized using the updated conditioning state, forming a closed loop in action space where generated trajectories, execution feedback, and state observations continuously refine subsequent predictions.

In summary, the proposed formulation unifies flow-based trajectory generation with streaming conditioning and closed-loop feedback. By operating entirely in motion space with a temporally evolving conditioning state, the model supports continuous action generation under asynchronous inputs while maintaining stability through high-frequency feedback. This establishes a unified framework that bridges multimodal perception, trajectory generation, and feedback control across multiple temporal scales.

## 4 Experiments

We organize our experiments around the following questions.

- (1) Can **MSC** preserve or improve motion generation performance?
- (2) How does **MSC** perform on streaming multimodal interaction tasks?
- (3) Does closed-loop design improve robustness under perturbations?
- (4) How do the structured conditions and their update frequencies affect **MSC**?
- (5) Can **MSC** generalize to real-world humanoid platforms?

To answer these questions, we first introduce the experimental setup (Sec. 4.1), followed by evaluations on standard motion benchmarks (Sec. 4.2), streaming multimodal interaction tasks (Sec. 4.3), closed-loop robustness analysis (Sec. 4.4), ablation studies on structured conditioning and update frequencies (Sec. 4.5), and real-world humanoid experiments (Sec. 4.6.

Table 1: Quantitative comparison across different modalities.

| Modality | Method | FID↓ | Diversity↑ | MM-dist↓ | R-precision@1↑ | Success rate↑ |
|---|---|---|---|---|---|---|
| Text | LangWBC | 5.17 | 5.03 | 4.19 | 10.21 | 58.2 |
| | UH-I | 6.39 | 5.41 | 4.45 | 12.73 | 51.3 |
| | OmniH2O+GMR | 3.02 | 5.73 | 3.14 | 18.46 | 65.0 |
| | BeyondMimic+GMR | 2.88 | 5.72 | 2.89 | 24.53 | 65.6 |
| | **MSC** | 1.51 | 5.74 | 2.73 | 46.39 | 81.1 |
| Video | OmniH2O+GMR | 3.27 | 3.86 | 3.57 | 11.02 | 22.5 |
| | BeyondMimic+GMR | 2.94 | 3.97 | 3.22 | 11.68 | 31.7 |
| | **MSC** | 2.26 | 3.15 | 2.56 | 47.34 | 89.3 |
| Audio | OmniH2O+GMR | 2.45 | 4.34 | 3.61 | 15.67 | 44.5 |
| | BeyondMimic+GMR | 1.93 | 4.69 | 3.01 | 17.91 | 58.4 |
| | **MSC** | 1.50 | 4.78 | 2.52 | 51.10 | 83.9 |

Table 2: Results on streaming multimodal interaction tasks.

| Method | MPJPE↓ | S-MPJPE↓ |
|---|---|---|
| TextOp | 127.68 | 294.10 |
| OmniH2O+GMR | 153.41 | 363.87 |
| UniAct | 134.02 | 406.56 |
| **MSC** | 124.83 | 124.83 |

Table 3: Results on closed-loop robustness tasks.

| Method | MPJPE↓ | S-MPJPE↓ |
|---|---|---|
| TextOp | 241.67 | 537.11 |
| OmniH2O+GMR | 310.96 | 629.08 |
| UniAct | 352.25 | 683.72 |
| **MSC** | 188.43 | 188.43 |

### 4.1 Experimental Setup

**Architecture and Backbone.** **MSC** is built upon existing open-source frameworks. We adopt the architecture of $\psi_0$ Wei et al. (2026) as the backbone and adapt it for humanoid action generation. Specifically, we retain its flow-based action expert as the motion generator and extend the conditioning interface for structured multimodal conditioning. We further use Qwen2.5-Omni-3B Xu et al. (2025) to fuse text, vision, and audio inputs into a unified conditioning representation.

**Dataset.** We train **MSC** using datasets from prior works such as TextOp Xie et al. (2026) and UniAct Jiang et al. (2025b), together with additional motion trajectories collected from Internet videos. All trajectories are extracted, retargeted, and refined before training.

**Baselines.** We compare **MSC** with representative humanoid action generation approaches, including both motion-retargeting pipelines OmniH2O He et al. (2024) + GMR Araujo et al. (2025) and end-to-end action generation methods TextOp Xie et al. (2026) and Uniact Jiang et al. (2025b).

**Platform.** All experiments are conducted on the Unitree G1 platform.

Detailed data preparation, training settings, inference configurations, and evaluation protocols are provided in Appendix A, Appendix B, Appendix C, and Appendix D, respectively.

### 4.2 MSC Preserves and Improves Motion Generation Performance

To evaluate the motion generation capability of **MSC** on conventional humanoid action generation tasks, we follow the protocol of UniAct Jiang et al. (2025b) and conduct experiments across text-, vision-, and audio-conditioned motion generation settings. We compare **MSC** against representative humanoid action generation baselines under the same evaluation setup and report standard motion quality and execution metrics.

As shown in Table 1, **MSC** consistently achieves competitive or superior performance across all supported modalities. Despite introducing streaming multimodal interaction and closed-loop action generation, **MSC** not only preserves strong motion generation capability but further improves motion quality, multimodal alignment, and humanoid execution performance compared with existing approaches. These results demonstrate that the proposed framework retains strong generation capability while extending beyond conventional static action generation paradigms.

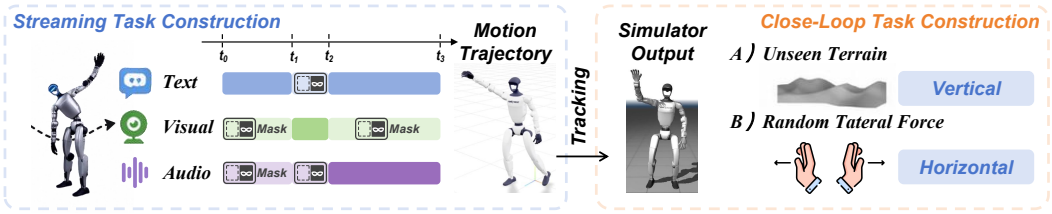



Figure 4: Overview of the proposed tasks' construction. (a) Streaming multimodal interaction task with temporally evolving text, visual, and audio instructions. (b) Closed-loop robustness task with unseen terrain and random lateral perturbations.

### 4.3 MSC Enables Streaming Multimodal Interaction

To evaluate the streaming interaction capability of **MSC**, we construct a new streaming multimodal interaction benchmark to compensate for the lack of existing streaming humanoid action datasets. Specifically, we collect high-quality human action videos from the internet and retarget them onto the Unitree G1 humanoid platform. We further annotate detailed upper-body, lower-body, and full-body motion descriptions, and jointly use the corresponding text descriptions, videos, and audio signals as multimodal interaction sources. To simulate temporally evolving interaction, each sequence is segmented into multiple temporally ordered motion segments, where at least one modality remains active within each time interval. During execution, the active multimodal condition changes according to the temporal schedule of the interaction, requiring continuous motion adaptation and smooth behavior transitions. An example interaction sequence is illustrated in Fig. 4(a). We will release the benchmark together with the closed-loop interaction setting, and provide detailed construction protocols.

Since existing baselines do not support interleaved multimodal streaming, we manually split each sequence according to modality transition timestamps and independently feed each segment into the corresponding single-modality baseline. The generated motion trajectories are then concatenated in temporal order for evaluation. In contrast, **MSC** directly processes the entire interaction sequence within a single continuous inference process. For evaluation, we first report MPJPE to measure motion trajectory similarity against ground-truth motions. In addition, we introduce a new metric termed **S-MPJPE** (**Single-shot Mean Per Joint Position Error**), a new evaluation metric built upon the conventional MPJPE metric. Specifically, the model is allowed to receive the multimodal condition only once at the beginning of the rollout, and the metric is computed as

$$\text{S-MPJPE} = \frac{1}{TJ}\sum_{t=1}^{T}\sum_{j=1}^{J}\left\|\hat{p}_{t,j}^{(1)} - p_{t,j}^{GT}\right\|_2,$$

where $T$ denotes the sequence length, $J$ denotes the number of joints, $p_{t,j}^{GT}$ is the ground-truth joint position, and $\hat{p}_{t,j}^{(1)}$ represents the predicted joint position generated under single-shot conditioning. The purpose of S-MPJPE is to evaluate the long-horizon multimodal consistency of a model without recurrent modality injection, thereby emphasizing the ability to sustain coherent motion generation from a single initial multimodal condition.

As shown in Table 2, **MSC** achieves comparable MPJPE performance to existing baselines despite supporting continuous streaming interaction. In contrast, these baselines process each modality segment independently, requiring separate inference across temporal segments and subsequent trajectory concatenation, making it difficult to maintain consistent multimodal interaction throughout the full sequence. These results demonstrate that **MSC** effectively supports continuous multimodal interaction beyond conventional fixed-conditioning action generation paradigms, although substantial room for future improvement still remains.

### 4.4 Closed-Loop Feedback Improves Robustness

To evaluate the robustness brought by the proposed closed-loop design, we further extend the streaming interaction benchmark under more challenging execution conditions in simulation. Specifically, we modify the terrain in Isaac-Sim to introduce uneven ground surfaces that disrupt the robot's vertical balance. In addition, we randomly apply external lateral forces during execution to perturb the

Table 4: Ablation study on the three structured conditions of **MSC**. We progressively remove $C^{int}$, $C^{exe}$, and $C^{cor}$ to evaluate their individual and complementary contributions.

| Method | $\mathbf{C^{int}}$ | $\mathbf{C^{exe}}$ | $\mathbf{C^{cor}}$ | S-MPJPE↓ |
|---|---|---|---|---|
| **MSC (Full)** | ✓ | ✓ | ✓ | **124.83** |
| w/o $C^{int}$ | | ✓ | ✓ | 133.27 |
| w/o $C^{exe}$ | ✓ | | ✓ | 130.44 |
| w/o $C^{cor}$ | ✓ | ✓ | | 126.85 |
| $C^{int}$ only | ✓ | | | 135.06 |
| $C^{exe}$ only | | ✓ | | 137.79 |
| $C^{cor}$ only | | | ✓ | 144.32 |
| **Backbone** | | | | 143.61 |

Table 5: Ablation study on different update frequencies. Low, Mid, and High denote 1, 3, and 10 Hz, respectively.

| Setting | Update Rate | S-MPJPE↓ | RTF↓ |
|---|---|---|---|
| Low | 1 Hz | 153.41 | 0.312 |
| Mid(MSC) | 3 Hz | 124.83 | 0.94 |
| High | 10 Hz | 197.16 | 3.12 |

Table 6: Runtime latency breakdown of **MSC** across different stages of the inference pipeline.

| | MM Fusion | Condition | Flow Matching | Decoding |
|---|---|---|---|---|
| MSC | 204ms | 14ms | 85ms | 9ms |
| **Total** | | **312 ms** | | |

robot's horizontal stability. These modifications significantly alter the execution states compared with standard flat-ground settings, requiring the model to continuously adapt future motion generation according to dynamically changing robot states. We report MPJPE and S-MPJPE under the same protocol.

As shown in Table 3, all methods experience noticeable performance degradation under these additional perturbations. Although the robot execution may fail to fully complete some challenging interaction sequences under these perturbations, MSC still consistently outperforms existing baselines by a large margin. We attribute this limitation to the fact that **MSC** is currently trained without explicit scene interaction or perturbation-aware supervision. Nevertheless, the strong S-MPJPE performance demonstrates the potential of the proposed closed-loop generation framework under dynamically changing execution conditions, which we believe can be further improved through future large-scale interaction training and environment-aware scaling.

### 4.5 Ablation

**Structured Conditioning.** Table 4 demonstrates that the three structured conditions are all essential to maintaining effective closed-loop behavior generation. Removing any one of $C^{int}$, $C^{exe}$, or $C^{cor}$ consistently degrades performance, with the removal of $C^{int}$ causing the largest degradation among the three. This is consistent with the distinct role of $C^{int}$ as the primary carrier of the evolving action intent: without it, the model no longer receives explicit information about what behavior should be performed next, and can only rely on flow matching to continue the previously generated motion, which progressively increases deviation from the intended behavior. Meanwhile, the degradation caused by removing $C^{exe}$ or $C^{cor}$ indicates that motion evolution and execution feedback are also indispensable for maintaining the closed-loop interaction process. In other words, removing any condition breaks one part of the feedback loop, preventing the model from jointly maintaining behavioral intent, motion continuity, and execution-aware correction. When all three conditions are removed, the performance approaches that of the worst single-condition setting, further highlighting that the proposed structured conditioning is not redundant but requires the three components to work together.

**Runtime Analysis.** We further analyze the inference latency of MSC to assess its practicality for streaming interaction. As shown in Table 6, most of the inference time is spent on multimodal fusion (204 ms), mainly due to the relatively large 3B multimodal backbone used to process multimodal inputs, while condition construction and decoding introduce only minor overhead. The flow matching stage takes 85 ms, demonstrating the efficiency of the proposed action generation process. Overall, one policy inference takes approximately 312 ms, which closely matches the 333 ms interval of a 3 Hz policy update and therefore places the practical policy update frequency of MSC around 3 Hz. Importantly, the policy update frequency is distinct from the robot motion execution frequency. Each policy call predicts a 30-step future motion chunk, while the generated trajectory is continuously tracked by the SONIC whole-body controller at 50 Hz. We adopt a receding-horizon strategy in which the first 20 steps are committed, while subsequent policy predictions are computed asynchronously and overlap with ongoing motion execution. Thus, the relatively low-frequency

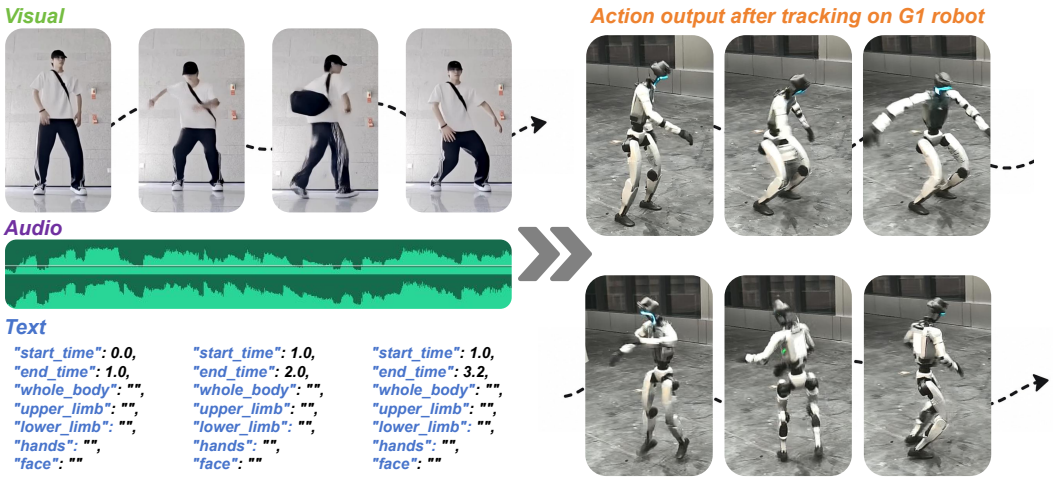



Figure 5: Real-world action output after tracking on G1 robot. More detailed case studies and task construction procedures are provided in Appendix E.

multimodal policy updates do not imply low-frequency robot motion execution, as the underlying motion controller continuously tracks the generated trajectory at high frequency.

**Policy Update Frequency.** We further investigate how the policy update frequency affects streaming behavior generation. Importantly, the policy update frequency is distinct from the robot motion execution frequency: MSC generates new motion plans at the policy level, while the generated trajectories are continuously tracked by the SONIC whole-body controller at 50 Hz. As shown in Table 5, the performance exhibits a clear trade-off between update responsiveness and computational cost. At 1 Hz, the policy updates too infrequently to sufficiently incorporate the evolving motion and execution feedback, resulting in higher S-MPJPE. Increasing the update frequency to 3 Hz substantially improves performance and achieves the best S-MPJPE. This setting is also closely aligned with the computational budget of MSC: the measured inference latency of 312 ms is slightly below the 333 ms interval of a 3 Hz policy update, allowing MSC to operate near real time. In contrast, increasing the update frequency to 10 Hz requires a new policy prediction every 100 ms, which is substantially shorter than the 312 ms inference latency. As reflected by the RTF of 3.12, the policy therefore cannot sustain real-time execution at this frequency. These results suggest that the effective streaming frequency of multimodal behavioral policies is currently constrained primarily by policy generation latency, rather than by the underlying robot motion execution rate.

### 4.6 Real-World Evaluation

We further deploy **MSC** on the Unitree G1 humanoid robot for real-world evaluation. For motion tracking and execution, we adopt the NVIDIA SONIC Luo et al. (2025) framework to convert generated motion trajectories into executable humanoid actions. We demonstrate a representative streaming multimodal interaction sequence on the real robot, as illustrated in Fig. 5. The results show that **MSC** can continuously respond to temporally evolving multimodal inputs and execute corresponding behavior transitions on a real humanoid platform, further validating the practicality and real-world feasibility of the proposed streaming closed-loop action generation framework.

## 5 Conclusion

We propose **MSC**, a multimodal streaming behavioral framework for closed-loop humanoid interaction. By combining structured multimodal conditioning with streaming flow matching, **MSC** enables continuous action generation and execution-aware behavioral adaptation under dynamically evolving inputs. Experimental demonstrate the effectiveness and robustness of the proposed framework for next-generation humanoid VLA systems.

## AI DISCLOSURE

In this work, we used generative AI tools to edit and polish the manuscript, including improving its grammar, clarity, and readability. We also used generative AI tools to assist with code development. All AI-assisted textual revisions were reviewed by the authors to ensure that they preserved the intended meaning and accurately reflected the research methods and results. AI-assisted code was reviewed and verified for correctness by multiple authors. We take full responsibility for the final content of this work.

## ETHICS STATEMENT

This work uses publicly available datasets and open-weight models and does not collect private or sensitive personal information. Human review is limited to validating the semantic consistency and correctness of controlled queries. We acknowledge potential biases in the source datasets, evaluated models, and automated evaluators.

## REFERENCES

Josh Abramson, Arun Ahuja, Federico Carnevale, Petko Georgiev, Alex Goldin, Alden Hung, Jessica Landon, Jirka Lhotka, Timothy Lillicrap, Alistair Muldal, et al. Improving multimodal interactive agents with reinforcement learning from human feedback. *arXiv preprint arXiv:2211.11602*, 2022.

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.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

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.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Changan Chen, Juze Zhang, Shrinidhi K Lakshmikanth, Yusu Fang, Ruizhi Shao, Gordon Wetzstein, Li Fei-Fei, and Ehsan Adeli. The language of motion: Unifying verbal and non-verbal language of 3d human motion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 6200–6211, 2025.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

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.

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.

Haoran Jiang, Jin Chen, Qingwen Bu, Li Chen, Modi Shi, Yanjie Zhang, Delong Li, Chuanzhe Suo, Chuang Wang, Zhihui Peng, et al. Wholebodyvla: Towards unified latent vla for whole-body loco-manipulation control. *arXiv preprint arXiv:2512.11047*, 2025a.

Nan Jiang, Zimo He, Wanhe Yu, Lexi Pang, Yunhao Li, Hongjie Li, Jieming Cui, Yuhan Li, Yizhou Wang, Yixin Zhu, et al. Uniact: Unified motion generation and action streaming for humanoid robots. *arXiv preprint arXiv:2512.24321*, 2025b.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Scott Kuindersma, Robin Deits, Maurice Fallon, Andrés Valenzuela, Hongkai Dai, Frank Permenter, Twan Koolen, Pat Marion, and Russ Tedrake. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous robots*, 40(3):429–455, 2016.

Yitang Li, Zhengyi Luo, Tonghe Zhang, Cunxi Dai, Anssi Kanervisto, Andrea Tirinzoni, Haoyang Weng, Kris Kitani, Mateusz Guzek, Ahmed Touati, et al. Bfm-zero: A promptable behavioral foundation model for humanoid control using unsupervised reinforcement learning. *arXiv preprint arXiv:2511.04131*, 2025a.

Zhe Li, Cheng Chi, Yangyang Wei, Boan Zhu, Tao Huang, Zhenguo Sun, Yibo Peng, Pengwei Wang, Zhongyuan Wang, Fangzhou Liu, et al. Do you have freestyle? expressive humanoid locomotion via audio control. *arXiv preprint arXiv:2512.23650*, 2025b.

Zhe Li, Cheng Chi, Yangyang Wei, Boan Zhu, Yibo Peng, Tao Huang, Pengwei Wang, Zhongyuan Wang, Shanghang Zhang, and Chang Xu. From language to locomotion: Retargeting-free humanoid control via motion latent guidance. *arXiv preprint arXiv:2510.14952*, 2025c.

Zhe Li, Weihao Yuan, Weichao Shen, Siyu Zhu, Zilong Dong, and Chang Xu. Omnimotion: Multimodal motion generation with continuous masked autoregression. *arXiv preprint arXiv:2510.14954*, 2025d.

Jinying Lin, Zhen Ma, Randy Gomez, Keisuke Nakamura, Bo He, and Guangliang Li. A review on interactive reinforcement learning from human social feedback. *IEEE Access*, 8:120757–120765, 2020.

Thomas Lucas, Fabien Baradel, Philippe Weinzaepfel, and Grégory Rogez. Posegpt: Quantization-based 3d human motion generation and forecasting. In *European Conference on Computer Vision*, pp. 417–435. Springer, 2022.

Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Sirui Chen, Fernando Castaneda, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, et al. Sonic: Supersizing motion tracking for natural humanoid whole-body control. *arXiv preprint arXiv:2511.07820*, 2025.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024.

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.

Ram Ramrakhya, Matthew Chang, Xavier Puig, Ruta Desai, Zsolt Kira, and Roozbeh Mottaghi. Grounding multimodal llms to embodied agents that ask for help with reinforcement learning. *arXiv preprint arXiv:2504.00907*, 2025.

Jonah Siekmann, Yesh Godse, Alan Fern, and Jonathan Hurst. Sim-to-real learning of all common bipedal gaits via periodic reward composition. In *2021 IEEE international conference on robotics and automation (ICRA)*, pp. 7309–7315. IEEE, 2021.

Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, et al. Kimi-vl technical report. *arXiv preprint arXiv:2504.07491*, 2025.

Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022.

Siyin Wang, Jinlan Fu, Feihong Liu, Xinzhe He, Huangxuan Wu, Junhao Shi, Kexin Huang, Zhaoye Fei, Jingjing Gong, Zuxuan Wu, et al. Roboomni: Proactive robot manipulation in omni-modal context. *arXiv preprint arXiv:2510.23763*, 2025.

Songlin Wei, Hongyi Jing, Boqian Li, Zhenyu Zhao, Jiageng Mao, Zhenhao Ni, Sicheng He, Jie Liu, Xiawei Liu, Kaidi Kang, et al. Psi_0: An open foundation model towards universal humanoid loco-manipulation. *arXiv preprint arXiv:2603.12263*, 2026.

Alexander W Winkler, C Dario Bellicoso, Marco Hutter, and Jonas Buchli. Gait and trajectory optimization for legged systems through phase-based end-effector parameterization. *IEEE Robotics and Automation Letters*, 3(3):1560–1567, 2018.

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.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-omni technical report. *arXiv preprint arXiv:2503.20215*, 2025.

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Yong Zhang, Hongwei Zhao, Hongtao Lu, Xi Shen, and Ying Shan. Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14730–14740, 2023.

Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *IEEE transactions on pattern analysis and machine intelligence*, 46(6):4115–4128, 2024.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023.

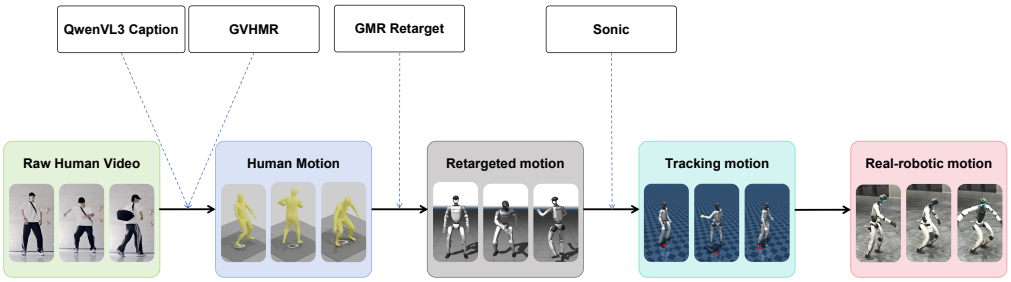



Figure 6: From internet videos to structured data tasks: our pipeline for collecting, processing, and building training data.

# A Data Collection Framework

To construct the training data used in this work, we start from large-scale human videos and progressively convert them into robot-executable motion data. The data processing pipeline consists of five main stages: video segmentation, video quality filtering, video information extraction, motion data extraction, and motion post-processing.

In the video segmentation stage, large-scale videos collected from the Internet are divided into multiple clips according to scene transitions, ensuring that no further transitions occur within each individual clip. In the video quality filtering stage, each clip is annotated with video quality-related labels, including camera shake estimation, aesthetic scoring, and human detection. Based on these annotations, we retain clips that contain a single person, have stable camera motion, and exhibit high aesthetic quality.

In the video information extraction stage, captions are generated for each video clip. In addition, human-related attributes, such as age and body weight, are annotated. For motion data extraction, we use GVHMR to recover SMPL motion data from the videos, and render the recovered motion as a yellow humanoid video for visualization. Finally, in the motion post-processing stage, we evaluate the quality of the extracted motions, filter out low-quality motion sequences according to their quality scores, convert the human motion data into G1-compatible motion data, and further use Sonic-based tracking to filter out motions that cannot be successfully executed by the robot.

# B Training Details

## B.1 Implementation Framework

Our implementation is built on top of the open-source Psi0 Wei et al. (2026) framework. Rather than re-implementing the entire training stack, we reuse the Psi0 training and deployment interface and extend it for full-body motion generation.

## B.2 Model and Motion Representation

Within the Psi0 framework, we replace the original vision-language backbone with Qwen2.5-Omni-3B and retain the Psi0 flow-based action expert as the motion generator. The resulting model has two main components. The first component is the frozen Qwen2.5-Omni-3B multimodal backbone, which encodes text, audio, and video conditions into a shared hidden space. All backbone parameters are frozen during post-training.

The second component is the Psi0 action expert. It is a 6-block flow-matching transformer with hidden size 1536, 24 attention heads, GELU activation, and a 2048-dimensional feed-forward layer. It receives the 2048-dimensional multimodal condition tokens from Qwen2.5-Omni-3B together with the robot state tokens, and predicts a 30-step future action chunk in the 70D TextOp-style action space. The predicted 70D action sequence is finally decoded back to the 36D absolute robot-motion space used by the downstream controller.

We represent each robot state in a 36-dimensional absolute G1 motion format. Each frame contains 29 joint values, a 3D root translation, and a 4D root orientation quaternion in `wxyz` order. In the current training configuration, the model receives the current state together with the previous three states, resulting in an observation horizon of four motion frames.

The supervised prediction target is a 70-dimensional TextOp-style action representation. For each future frame, the 70 dimensions are decomposed into: 29D absolute joint values, 29D joint deltas with respect to the previous pose, 3D root translation deltas expressed in the previous local frame, 6D root rotation deltas, 1D root height, and 2D estimated foot-contact indicators. This representation preserves absolute pose information while exposing local motion transitions that are easier for the action expert to model. During training, actions are padded to 70 dimensions and states are padded to 36 dimensions. At inference time, the predicted 70D action chunk is decoded back to the 36D absolute motion sequence.

In all three condition settings, the model predicts a chunk of 30 future action steps. The downstream runtime interface is configured with an action execution horizon of 20 steps.

### B.3 Multimodal Training Task

For text-conditioned training, the raw language input is not a single fixed template. Instead, the dataset collects candidate descriptions from seven metadata fields, including short descriptions, technical descriptions, and natural-language descriptions. For each training sample, one description is randomly selected and wrapped by a lightweight instruction template, such as a system-first prompt or a description-first prompt. The resulting natural language instruction is then directly passed to the Qwen2.5-Omni-3B processor as the text condition. One example template is: `Generate a motion sequence that matches the following description. Description: {desc}`. If the selected description is `a person walks forward and waves`, the final model input becomes: `Generate a motion sequence that matches the following description.`
`Description: a person walks forward and waves.`

For audio-conditioned training, each motion clip is paired with a matched `.mp4` file, and the entire audio segment is used as the model input for each training sample. The audio stream is decoded as a mono waveform at 16 kHz and passed to Qwen2.5-Omni-3B together with an auxiliary audio instruction. The instruction is sampled from a small prompt pool. A typical example is: `Generate a dance motion that follows the rhythm and energy of this music.` Other prompts describe the same task in equivalent language, such as creating a full-body dance sequence that matches the beat, tempo, and mood of the audio.

For video-conditioned training, each motion clip is paired with a matched video file. The current configuration uses an aligned video segment instead of the full original video, and the segment boundaries are resolved from the annotation metadata when available. The motion stream is indexed at 50 Hz with sample stride 1, while the visual condition is sampled as 8 frames at 2 FPS. The processor is additionally constrained by a maximum visual budget of `200704` pixels. As in the audio branch, an auxiliary instruction prompt is enabled in the video-conditioned setup. A representative prompt is: `Generate a full-body motion sequence that matches the person's movement in this video.`

All modalities share the same robot-side interface: the same 36D motion state space, the same 70D action target space, the same 30-step prediction horizon, and the same Qwen2.5-Omni-3B backbone family. Modality differences are introduced only through the conditioning interface, namely natural-language instruction, full-audio waveform, or sampled video frames. For multimodal joint training, we use a batch-mixture strategy over text, audio, and video samples. The three modalities are sampled with equal weights: $1/3$ text-conditioned samples, $1/3$ audio-conditioned samples, and $1/3$ video-conditioned samples. Each sample is converted to the same motion prediction problem after modality-specific preprocessing. Therefore, all modality batches update the same Qwen2.5-Omni-3B-conditioned action expert and share the same 70D TextOp supervision space.

### B.4 Optimization and Training Budget

The joint multimodal model is trained on 8 NVIDIA A100 GPUs. We use AdamW with a cosine-with-minimum-learning-rate scheduler, a base learning rate of $1 \times 10^{-4}$, and bf16 mixed-precision training. The per-GPU batch size is 128, giving a global batch size of 1024 with gradient accumulation set to 1. Gradient-norm clipping is set to 1.0, and non-finite gradient updates are skipped. The minimum learning rate is set to $5 \times 10^{-7}$.

We train the joint text-audio-video model for 300000 optimization steps. The first 1000 steps are used for linear warmup, followed by cosine decay to the minimum learning rate. Validation is performed every 1000 steps, and model checkpoints are saved every 1000 steps. Each validation pass uses the same three-way modality mixture as training so that text, audio, and video conditioning are monitored under a shared evaluation protocol.

States are normalized in the 36D motion space, and actions are normalized in the 70D TextOp space using the precomputed `bounds_q99` statistics. This normalization is shared with the corresponding modality-specific training runs.

## C Inference Configuration

### C.1 Runtime Setup

All inference experiments are conducted on a single NVIDIA A100 GPU using the trained checkpoint. The Qwen2.5-Omni-3B backbone remains unchanged from the trained checkpoint. The three input modalities share the same inference pipeline after conditioning: the model encodes the text, audio, or video input, samples a future action chunk with the flow-matching action expert, decodes the action chunk into absolute G1 motion states, and appends the executed states to the rollout history.

### C.2 Modality-Specific Inputs

For text-conditioned inference, the model takes a natural language motion instruction as input. For audio-conditioned inference, the full task-level audio segment is provided to the model, while multimodal conditions are used according to the temporal schedule of the interaction. For video-conditioned inference, we sample 8 frames at 2 FPS with a maximum pixel budget of 200,704 pixels, following the same video configuration used during training.

### C.3 Flow-Matching Sampling and Rollout

The action expert uses the same flow-matching sampler as in training. At inference time, we use 10 denoising steps for each policy call. For quantitative evaluation, the random seed is fixed so that the stochastic sampling process is reproducible. Unless otherwise specified, the same sampling configuration is used for text-, audio-, and video-conditioned inference.

The rollout starts from a natural static standing pose, which provides the initial robot state. For each policy call, the model predicts a 30-step future action chunk. We adopt a receding-horizon rollout strategy with a 20-step commitment window, where only the first 20 steps are committed to the generated sequence. Subsequent policy predictions are launched asynchronously while the committed trajectory is being executed, allowing policy inference to overlap with ongoing motion execution. The newly generated states available at each policy update are incorporated into the motion history, providing updated observation context for the next action chunk. This process is repeated until the required sequence length is reached.

### C.4 Action Decoding and Motion Export

The sampled action chunk is first denormalized using the same action statistics used during training. The denormalized 70D TextOp-style action is then decoded into the 36D absolute G1 motion state. During decoding, the joint positions are read from the absolute joint component, while the root trajectory is reconstructed by integrating the local root translation delta and converting the predicted

6D root rotation update back to a quaternion. The decoded frame therefore contains 29 joint values, 3D root translation, and 4D root orientation.

For offline evaluation, the generated motion is exported as frame-wise G1 motion states and can be stored as a 36D motion array or as the corresponding `joint_pos`, `body_pos`, and `body_quat` files. This is the same motion format consumed by the TMR evaluator described below.

### C.5 Real-Robot Execution

For real-robot deployment, the generated motion commands are passed to the GEAR-SONIC whole-body control stack in GR00T-WholeBodyControl. GEAR-SONIC tracks the generated full-body motion on the robot while handling the low-level whole-body control interface.

## D Evaluation Protocol

### D.1 TMR Evaluator Training

We train a standalone Text-Motion Retrieval (TMR) evaluator to compute FID, Diversity, MM-dist, and R-precision. The evaluator is trained only on paired text-motion data and is frozen during all downstream evaluations.

The motion input to TMR uses the same 36D G1 motion representation as our model, consisting of 29 joint positions, 3D root position, and 4D root quaternion. Before TMR training, raw motion sequences are converted from CSV files, resampled from 50 Hz to 20 Hz, uniformly compressed to at most 320 frames, and normalized with mean and standard deviation estimated from the TMR training split. The current split contains 113,770 training sequences, 14,221 validation sequences, and 14,221 test sequences.

For text, we use the available motion descriptions from the metadata as text candidates. During training, one candidate description is randomly sampled for each motion sequence, while evaluation uses the primary text description. Token-level text features are extracted using DistilBERT, resulting in 768D token embeddings. We additionally cache sentence-level MPNet embeddings to identify near-duplicate text descriptions when computing the contrastive loss and retrieval metrics.

The TMR model follows an ACTOR-style Transformer VAE design. It contains a text encoder, a motion encoder, and a motion decoder. Both encoders use 6 Transformer layers, 4 attention heads, a latent dimension of 256, a feed-forward dimension of 1024, GELU activation, and dropout of 0.1. The motion encoder maps 36D motion sequences into the shared latent space, while the text encoder maps 768D token embeddings into the same space. The model is trained with a combination of motion reconstruction loss, text-motion latent alignment loss, KL regularization, and bidirectional InfoNCE contrastive loss. We train the evaluator for 10 epochs with AdamW, learning rate $1 \times 10^{-4}$, weight decay $1 \times 10^{-4}$, and batch size 128.

After training, the TMR evaluator is frozen. For text-conditioned generation, text embeddings and generated motion embeddings are compared directly in the learned TMR space. FID and Diversity are computed using motion embeddings, while MM-dist and R-precision are computed between the text embedding and the generated motion embedding.

### D.2 UniAct-Style Evaluation for Audio and Video

The trained TMR evaluator is a text-motion model and does not contain audio or video encoders. Therefore, for audio- and video-conditioned generation, we follow a UniAct-style protocol and convert the evaluation into motion-to-motion comparison.

For each audio or video input, the paired ground-truth motion is encoded by the frozen TMR motion encoder and used as the reference embedding. The generated motion from the same input is also encoded by the same motion encoder. MM-dist is then computed as the distance between the generated motion embedding and its paired ground-truth motion embedding. R-precision is computed by using the generated motion embedding as the query and retrieving the corresponding paired ground-truth motion from a candidate pool. The candidate pool is built from the same evaluation split: for each query, the paired ground-truth motion is treated as the positive target, and all other

ground-truth motions in the pool are treated as mismatched negatives. We report top-1, top-2, and top-3 retrieval accuracy. FID is computed between the distribution of generated motion embeddings and ground-truth motion embeddings, and Diversity is computed as the average pairwise distance among generated motion embeddings.

This protocol avoids training separate audio-motion or video-motion retrieval models. It keeps all modalities evaluated with the same frozen motion encoder, although the retrieval target differs between text-conditioned and audio/video-conditioned evaluation.

### D.3 Metric Computation Details

All generated and ground-truth motions are converted to the same 36D motion format before evaluation and processed by the same TMR preprocessing pipeline, including temporal resampling, length compression, and normalization. During evaluation, we use the deterministic latent mean from the frozen TMR encoder rather than sampling from the VAE latent distribution.

For retrieval-based metrics, similarities are computed as cosine similarities between L2-normalized embeddings. MM-dist is computed as the Euclidean distance between the corresponding unnormalized latent means. FID is computed on unnormalized motion latent means by matching the mean and covariance of generated and ground-truth motion features. Diversity is computed as the Euclidean distance between randomly sampled pairs of generated motion embeddings, using at most 300 pairs with a fixed random seed.

## E Additional details regarding Figure 5

For the video demonstration, we additionally provide five examples in the supplementary materials, each illustrating the complete pipeline from the original video to deployment on a real robot.

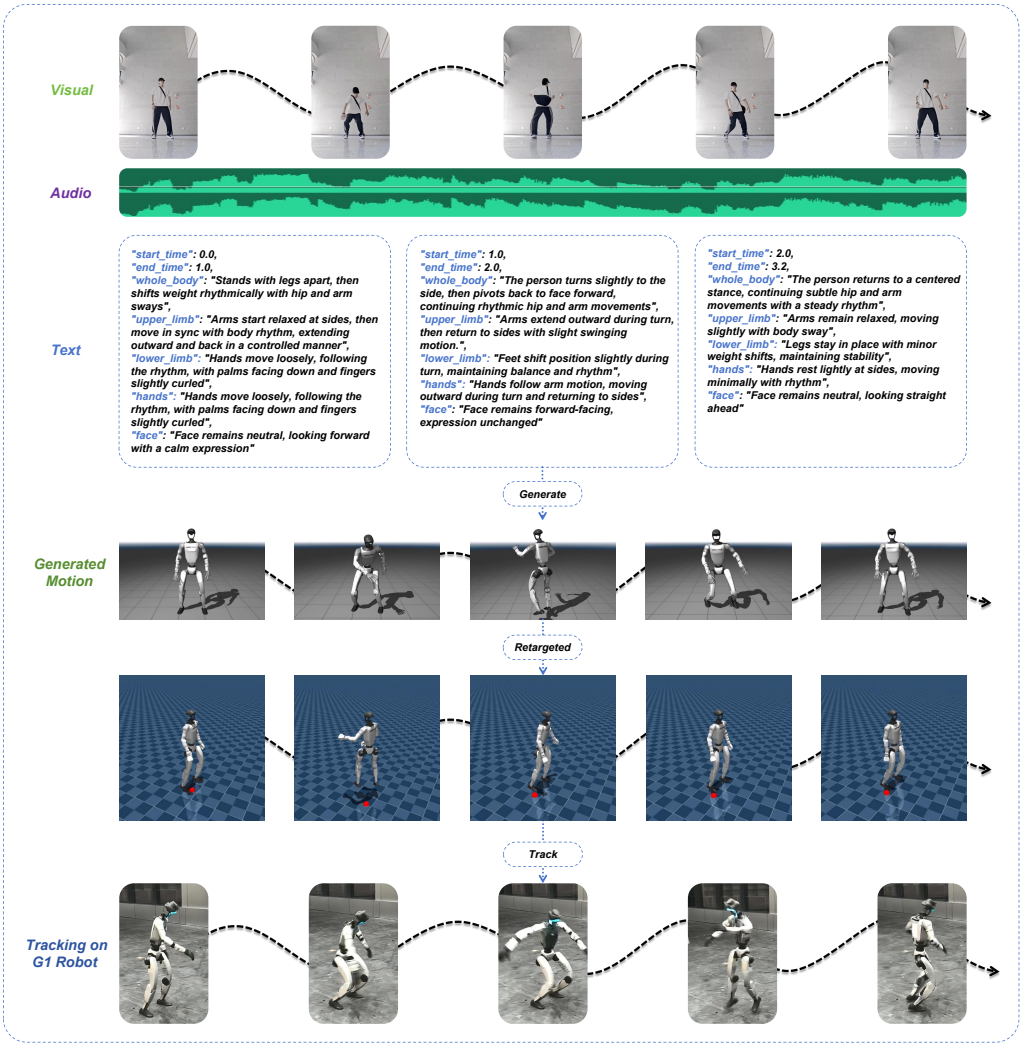



Figure 7: A detailed illustration of the workflow shown in Figure 5.