

# DETACH: Cross-domain Learning for Long-Horizon Tasks via Mixture of Disentangled Experts

Yutong Shen<sup>1</sup>, Hangxu Liu<sup>2</sup>, Lei Zhang<sup>3†</sup>, Penghui Liu<sup>1</sup>, Ruizhe Xia<sup>1</sup>, Tianyi Yao<sup>1</sup>, Tongtong Feng<sup>4†</sup>

<sup>1</sup> Beijing University of Technology, China <sup>2</sup> Fudan University, China <sup>3</sup> University of Hamburg, Germany <sup>4</sup> Tsinghua University, China

Corresponding authors: fengtongtong@tsinghua.edu.cn, lei.zhang-1@studium.uni-hamburg.de

**Abstract**—Long-Horizon (LH) tasks in Human-Scene Interaction (HSI) are complex multi-step tasks that require continuous planning, sequential decision-making, and extended execution across domains to achieve the final goal. However, existing methods heavily rely on skill chaining by concatenating pre-trained subtasks, with environment observations and self-state tightly coupled, lacking the ability to generalize to new combinations of environments and skills, failing to complete various LH tasks across domains. To solve this problem, this paper presents DETACH, a cross-domain learning framework for LH tasks via biologically inspired dual-stream disentanglement. Inspired by the brain’s “where-what” dual pathway mechanism, DETACH comprises two core modules: i) an environment learning module for spatial understanding, which captures object functions, spatial relationships, and scene semantics, achieving cross-domain transfer through complete environment-self disentanglement; ii) a skill learning module for task execution, which processes self-state information including joint degrees of freedom and motor patterns, enabling cross-skill transfer through independent motor pattern encoding. We conducted extensive experiments on various LH tasks in HSI scenes. Compared with existing methods, DETACH can achieve an average subtasks success rate improvement of 23% and average execution efficiency improvement of 29%. More details can be found at: <https://sites.google.com/view/detach-learning>.

## I. INTRODUCTION

Long-Horizon (LH) tasks in Human-Scene Interaction (HSI) require continuous planning and cross-domain execution, posing challenges due to their complexity and need for environmental adaptation. These tasks have broad applications in robotics [1], medical intervention [2], and smart homes [2], with canonical examples including dexterous hand manipulation [3] and humanoid whole-body control [4]. However, recent benchmarks show that HSI methods achieve low success rates on cross-domain tasks and demand extensive retraining [5]–[7], severely limiting real-world deployment.

Recent large-scale vision-language-action (VLA) models [8], [9] and agent-based manipulation [10] achieve strong results on long-horizon embodied tasks. However, both paradigms typically adopt monolithic or tightly coupled end-to-end designs, where perception and control remain entangled, thereby limiting cross-domain generalization and modular skill reuse.

To bridge these gaps, current approaches [11]–[13] focus on processing self-state information in unified representation spaces, while other solutions [5], [7], [14] further encode self-state information mixed with environmental information. The efficacy of the *decompose-reuse-compose* paradigm has been confirmed by various studies [15]–[18], which also introduced

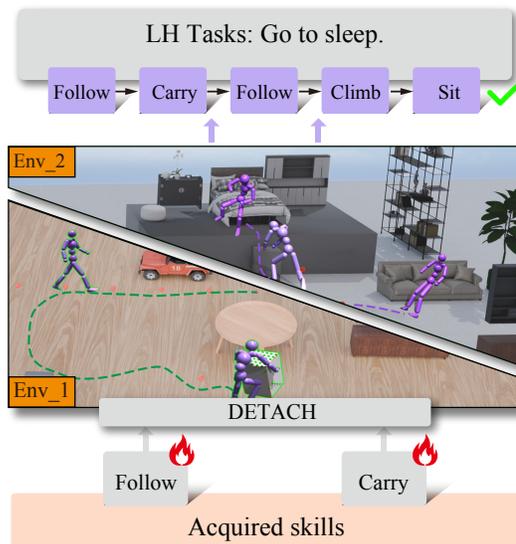


Fig. 1: DETACH achieves generative generalization by learning fundamental subtasks in single environment (Env\_1), enabling it to generalize to novel environments and accomplish Long-Horizon tasks that involve previously unseen subtasks.

a new modular learning paradigm for rapid adaptation to new skills by utilizing skill modules that have already been learned. In particular, CML [16] and TokenHSI [11] have explicitly demonstrated that such modular decomposition significantly outperforms standard end-to-end approaches in multi-task reinforcement learning (RL) and long-horizon HSI, respectively.

Despite their promising performance, these methods suffer from the same architectural flaw: they adopt unified feature representation spaces that tightly couple environmental understanding with self-states. This flaw poses significant challenges in two main aspects: (1) Limited environmental transfer capability: When environmental changes occur (such as shifts from bright laboratory to dim factory settings), these systems cannot effectively separate the effects of environmental changes from self-state changes. This limitation necessitates relearning the entire perception-action mapping [19], significantly constraining their cross-domain generalization capability. (2) Inefficient skill transfer capability: Current methods fail to achieve functional separation between perception and motor control. When encountering novel skills, even those involving similar motor patterns (such as grasping different objects), the system must retrain the entire perception-action network. This

limitation makes it difficult to reuse, prevents effective reuse of learned motor skills, resulting in extremely low knowledge transfer efficiency due to a high risk of skill forgetting [20]. Even advanced modular approaches such as CML [16] and TokenHSI [11] still rely partly on unified feature spaces, thereby inheriting some of these limitations.

To address these challenges, this paper introduces **DETACH**: a biologically inspired functional disentanglement architecture that draws from the dorsal-ventral stream hypothesis in neuroscience [21]. According to this hypothesis, the brain’s ventral *what* pathway specializes in object recognition, while the dorsal *where-how* pathway handles spatial processing and motor control. Unlike existing dual-stream approaches [22] that separate visual modalities, DETACH introduces a functional disentanglement: the **Environmental Encoder** learns scene-invariant spatial relationships [23] while the **Self-Encoder** captures body-schema-specific motor primitives.

Proposed method is extensively evaluated on various self-designed LH-embodied AI tasks, including cross-scene adaptation, novel skill adaptation, and particularly LH control tasks in complex environments. The contributions of this paper can be summarized as follows.

- Proposing the **DETACH disentangled architecture**, the first Embodied AI control framework in HSI based on biologically inspired cognitive principles. This architecture separates traditional unified encoding into specialized parallel processing of environmental perception streams and self-state perception streams.
- Designing **specialized dual-stream encoders**, where the environmental encoder enhances **cross-domain transfer capability**, and the self-encoder achieves **cross-task skill reuse**. Both encoders are independently optimized and flexibly combined.
- Establishing comprehensive benchmark scenarios for LH tasks through designed progressive LH task benchmarks, and validating the effectiveness of DETACH on these benchmarks. Compared to existing methods, DETACH achieves a  $2\times$  **improvement in cross-domain adaptation capability** and a  $1.5\times$  **improvement in skill reuse efficiency**.

## II. RELATED WORKS

### A. Human-Scene Interaction

HSI focuses on enabling embodied agents to interact naturally and effectively with complex 3D environments. Existing approaches include unified representation learning (e.g., Chain of Contacts [14]), which integrates contact and object encoding with LLM-based planning but exhibits limited generalizability due to tight coupling between perception and action components; staged processing (e.g., Dynamic HSI [24]), which uses autoregressive diffusion for disentangled scene understanding and action generation, ensuring temporal coherence but at high computational cost; and end-to-end methods (e.g., TokenHSI [11], and [5], [7]), which synthesize motion from text using pre-trained models and object sensors but are limited

to simple skill composition scenarios. A key limitation shared by these approaches is their tight coupling between perception and control modules, which hinders cross-domain transfer and skill reusability.

### B. Long-Horizon Task

LH tasks in HSI require agents to perform multi-step reasoning and manage long-term dependencies [12]. Current approaches include hierarchical planning (e.g., MLLM-based instruction parsing with visual encoders [25], [26]), which decomposes tasks into subgoals but suffers from low skill prediction accuracy; memory augmentation (e.g., hierarchical memory and knowledge graphs [12]), which models long-term dependencies yet lack dynamic adaptation; and causal modeling [27], which enhances policy learning through observation-action causality but requires high computational resources and relies on limited training data. These methods are limited by their reliance on static representations, which constrains cross-domain transfer, policy reuse, and adaptation to dynamic interaction scenarios.

### C. Disentangled Learning

Disentangled representation learning addresses these limitations by decomposing complex systems into independent, interpretable modules, improving generalization and controllability [28]. Key approaches include mutual information-based disentanglement (e.g., [18]), which minimizes mutual information between skill components but requires domain-specific prior knowledge; factorized representation learning (e.g.,  $\beta$ -VAE framework [29]), which uses disentanglement regularization; and variational disentanglement (e.g., [30]), which optimizes a variational lower bound. An alternative method [31] employs Wasserstein distance for stable disentanglement, though it remains theoretical, while [31] also identifies valuable factors at high computational cost. However, these methods focus on static factor separation, which are ill-suited for the dynamic, continuous interactions and generative adaptation required in LH embodied tasks.

## III. METHODS

DETACH employs a dual-encoder design with environmental encoder  $\Phi_{\text{env}}$  and self-encoder  $\Phi_{\text{self}}$  to disentangle environmental perception from self-state representation. Their outputs are fused via a multi-strategy adaptive mechanism and processed by the shared transformer encoder  $\phi$  to enhance perception-control collaboration.

### A. Observation Space Reconstruction Model

DETACH employs observation disentanglement, modeling unified observation space as a Dual-Stream Separation Process (DSP). The disentanglement objective minimizes mutual information between environmental and self-state representations, quantified as  $D = \sum_{t=0}^T \gamma^t I(\text{obs}_{\text{env}}^t, \text{obs}_{\text{self}}^t)$ , implemented using correlation-based mutual information estimators.

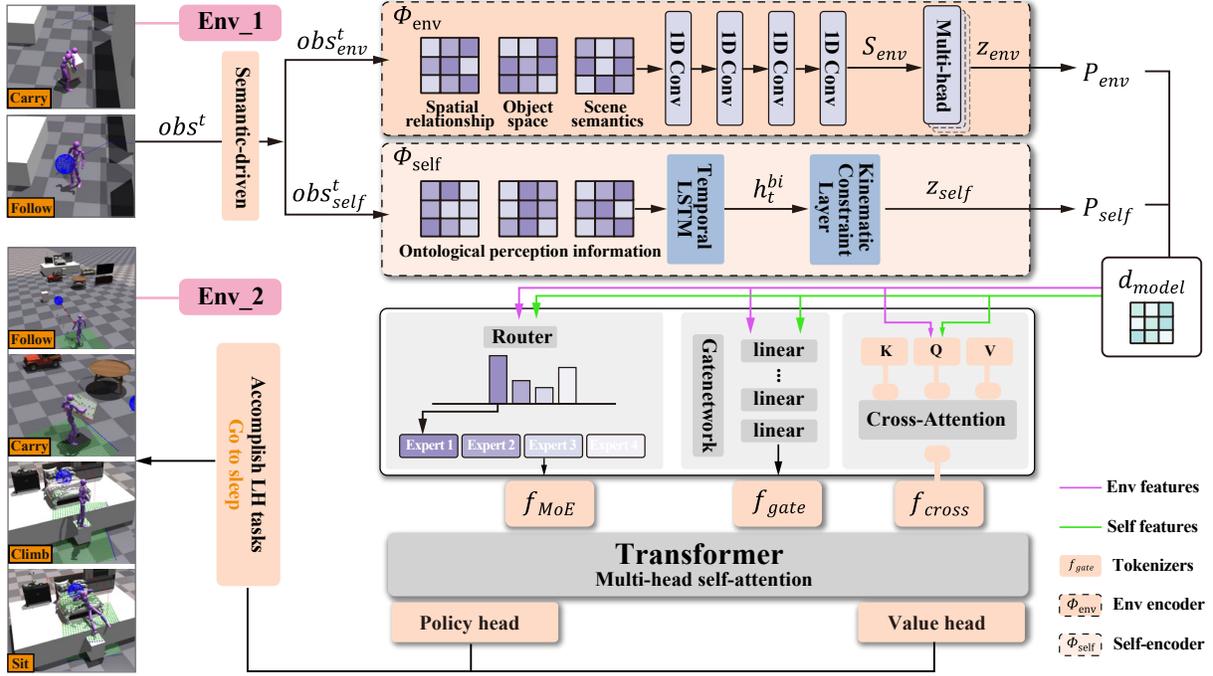


Fig. 2: Illustrating the operational workflow of the DETACH, Raw observation  $obs^t$  is semantically disentangled into environmental  $obs_{env}^t$  and self-state  $obs_{self}^t$  components. Environmental encoder  $\Phi_{env}$  and self-encoder  $\Phi_{self}$  process respective inputs, with projection layers  $P_{env}$  and  $P_{self}$  mapping outputs to unified  $d_{model}$  space. Multi-strategy adaptive fusion integrates features via three components: MoE fusion, gated fusion network, and cross-attention fusion module, producing outputs ( $f_{MoE}$ ,  $f_{gate}$ ,  $f_{cross}$ ). These fused representations undergo Transformer multi-head self-attention before feeding into policy and value heads.

### B. Disentangled Dual-Encoder

DETACH is a biologically inspired, disentangled dual-encoder architecture that separates the traditional unified encoding pathway into two specialized processing streams for environmental perception and self-state representation. Figure 2 illustrates the proposed disentanglement module, which consists of four key components:

*Environmental encoder  $\Phi_{env}$ .* The environmental encoder processes spatial information such as object positions and scene semantics. Given environmental observations  $obs_{env}^t \in \mathbb{R}^{T \times d_{env}}$ , we adopt parallel convolutional layers for feature extraction. Feature extraction is performed as:

$$S_{env} = \text{Concat}[\text{Conv1D}_k(obs_{env}^t)] \quad (1)$$

Features are aggregated through multi-head self-attention for spatial feature aggregation:

$$z_{env} = \text{LayerNorm}(\text{MultiHeadAttn}(S_{env}, S_{env}, S_{env}) + S_{env}) \quad (2)$$

The environmental encoder is paired with decoder  $\text{Decoder}_{env}$  for reconstruction-based pre-training.

*Self-encoder  $\Phi_{self}$ .* The self-encoder processes self-state information  $obs_{self}^t \in \mathbb{R}^{T \times d_{self}}$  including joint angles and velocities. Since accurate self-state understanding requires bidirectional temporal context for accurate motion understanding, the self-encoder employs a recurrent neural architecture

with bidirectional processing capabilities. The model is defined as:

$$h_t^{bi} = [h_t^f; h_t^b] \quad (3)$$

where  $h_t^f$  and  $h_t^b$  represent forward and backward temporal representations, respectively.

The kinematic constraint layer ensures outputs remain within physically feasible ranges through a element-wise soft gating mechanism:

$$z_{self} = h_t^{bi} \odot \sigma(W_k h_t^{bi} + b_k) \quad (4)$$

where  $\sigma$  is the sigmoid function, and  $W_k$  and  $b_k$  are learnable parameters. The kinematic constraint layer ensures the physical feasibility of generated actions.

The self-encoder is paired with a temporal prediction network  $f_{pred}$  for sequence prediction-based pre-training, which learns to predict future self-state representations from current ones. *Feature projection layers  $P_{env}$  and  $P_{self}$ .* Two independent linear layers map the encoder outputs to a unified  $d_{model}$  dimensional space:

$$f_{env} = P_{env}(z_{env}), \quad f_{self} = P_{self}(z_{self}) \quad (5)$$

where  $f_{env}, f_{self} \in \mathbb{R}^{d_{model}}$  are the projected features used for fusion.

### C. Multi-Strategy Adaptive Fusion Mechanism

To effectively integrate heterogeneous features from the environmental encoder and self-encoder, **DETACH** incorporates a multi-strategy adaptive fusion mechanism that combines three complementary fusion strategies. According to Figure 2, the adopted fusion mechanism comprises three core components:

*Cross-attention fusion module.* This module [32] is selected for its capability to enable internal states to actively query key information from the environment, thereby achieving state-driven dynamic feature alignment. It uses self-state features  $f_{self} \in \mathbb{R}^{d_{model}}$  as Query, and environment features  $f_{env} \in \mathbb{R}^{d_{model}}$  as Key and Value, achieving dynamic weight allocation through a multi-head attention mechanism:

$$\begin{aligned} f_{cross} &= \text{MultiHead}(f_{self}, f_{env}, f_{env}) \\ &= \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \end{aligned} \quad (6)$$

where each attention head:

$$\text{head}_i = \text{Attention}(f_{self}W_i^Q, f_{env}W_i^K, f_{env}W_i^V) \quad (7)$$

*Gated Fusion Network.* This module dynamically modulates contribution weights between environmental perception and self-state features to prevent imbalance, using learnable gating units. It is implemented via a multi-layer MLP [33] with decreasing hidden units, matching the fused feature dimension. The gated fusion strategy is defined as:

$$\begin{aligned} f_{gate} &= \sigma(W_g[f_{env}; f_{self}] + b_g) \odot f_{env} \\ &+ (1 - \sigma(W_g[f_{env}; f_{self}] + b_g)) \odot f_{self} \end{aligned} \quad (8)$$

*Mixture of Experts (MoE) fusion module.* This module is adopted for its capacity to dynamically select optimal fusion experts based on task characteristics and environmental complexity, enabling adaptive feature integration. It designs multiple specialized fusion experts [34], each modeled by a multi-layer MLP network with a hierarchical structure, dynamically selecting the most suitable expert for feature fusion through a routing network. The mixture of experts' fusion is represented as:

$$f_{moe} = \sum_{i=1}^4 w_i \cdot E_i(f_{env}, f_{self}) \quad (9)$$

where the routing weights are

$$w_i = \text{Softmax}(W_r[f_{env}; f_{self}] + b_r)_i \quad (10)$$

The three fusion strategies are combined through a learnable weighted combination:

$$f_{fused} = \alpha \cdot f_{cross} + \beta \cdot f_{gate} + \gamma \cdot f_{moe} \quad (11)$$

where  $\alpha, \beta, \gamma$  are learnable parameters that balance the contributions of different fusion strategies.

*Shared Transformer Encoder.* The fused features are processed by a shared transformer encoder  $\phi$  to enhance perception-control collaboration:

$$h_{transformer} = \phi(f_{fused}) \quad (12)$$

where  $\phi$  consists of multiple transformer layers with self-attention mechanisms to capture long-range dependencies and temporal relationships.

*Policy and Value Heads.* The transformer output is fed into separate policy and value heads for action prediction and value estimation:

$$\begin{aligned} \pi(a|s) &= \text{PolicyHead}(h_{transformer}) \\ V(s) &= \text{ValueHead}(h_{transformer}) \end{aligned} \quad (13)$$

where  $\pi(a|s)$  represents the action probability distribution and  $V(s)$  represents the state value function.

### D. Progressive Training Protocol

To fully leverage the advantages of disentangled architecture and ensure the specialized characteristics of each module, **DETACH** designed a comprehensive progressive training protocol and specialized regularization mechanisms. The adopted training protocol involves three progressive stages, each with clear training objectives and parameter update strategies:

*Independent Pre-training Stage.* In this stage, the environmental encoder  $\Phi_{env}$  and self-encoder  $\Phi_{self}$  are trained independently to establish their respective feature representation capabilities. The environmental encoder is pre-trained through the scene reconstruction loss:

$$\mathcal{L}_{env} = \|\text{Decoder}_{env}(\Phi_{env}(obs_{env}^t)) - obs_{env}^t\|_2^2 \quad (14)$$

The self-encoder is pre-trained through action sequence prediction tasks:

$$\mathcal{L}_{self} = \sum_{t=1}^{T-1} \|\Phi_{self}(obs_{self}^{t+1}) - f_{pred}(\Phi_{self}(obs_{self}^t))\|_2^2 \quad (15)$$

where  $f_{pred}$  is the temporal prediction network. This stage establishes domain-specific representation foundations.

*Fusion Layer Optimization Stage.* In this stage, the pre-trained encoder parameters  $\theta_{env}, \theta_{self}$  are frozen to preserve learned representations, focusing on training the feature fusion layer and Transformer encoder  $\phi$ :

$$\mathcal{L}_{fusion} = \mathcal{L}_{task} + \lambda_{quality} \mathcal{L}_{fusion\_quality} \quad (16)$$

where  $\mathcal{L}_{task}$  represents the standard reinforcement learning objective (e.g., policy gradient loss for PPO), which guides the agent to maximize expected cumulative rewards.

where the fusion quality loss is defined as:

$$\begin{aligned} \mathcal{L}_{fusion\_quality} &= \|f_{cross} - (f_{env} + f_{self})\|_2^2 \\ &+ \lambda_{disentangle} \cdot I(z_{env}, z_{self}) \end{aligned} \quad (17)$$

where  $I(z_{env}, z_{self})$  represents the mutual information between environmental and self-state features. The first term ensures fusion consistency, while the second term maintains disentanglement by minimizing mutual information between representations.

*End-to-End Joint Optimization Stage.* In the end-to-end joint optimization stage, all network parameters are unfrozen for

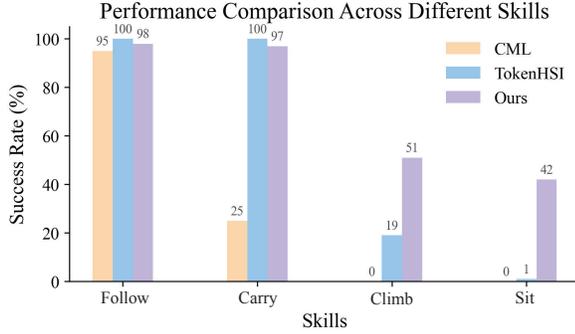


Fig. 3: Success rate comparison across different skills among CML, TokenHSI, and our DETACH framework.

| Method        | Follow | Carry | Climb       | Sit         | LH1         |
|---------------|--------|-------|-------------|-------------|-------------|
| CML [17]      | 0.95   | 0.25  | 0.00        | 0.00        | 0.30        |
| TokenHSI [11] | 1.00   | 1.00  | 0.19        | 0.01        | 0.55        |
| Ours          | 0.98   | 0.97  | <b>0.51</b> | <b>0.42</b> | <b>0.72</b> |

TABLE I: Success rates for foundational skills and composite task completion.

end-to-end joint optimization, while introducing specialized preservation regularization:

$$\mathcal{L}_{total} = \mathcal{L}_{task} + \lambda_{disentangle} \cdot I(z_{env}, z_{self}) + \sum_i \lambda_i \mathcal{R}_i \quad (18)$$

where the regularization terms include:

$$\mathcal{R}_1 = \|\theta_{env} - \theta_{env}^*\|_2^2 \quad (\text{encoder preservation}) \quad (19)$$

$$\mathcal{R}_2 = \|\theta_{self} - \theta_{self}^*\|_2^2 \quad (\text{encoder preservation}) \quad (20)$$

$$\mathcal{R}_3 = \sum_{i \neq j} \|f_i - f_j\|_2^2 \quad (\text{fusion diversity}) \quad (21)$$

where  $f_i, f_j \in \{f_{cross}, f_{gate}, f_{moe}\}$  and  $\mathcal{R}_3$  encourages different fusion strategies to learn complementary representations. The regularization weights  $\lambda_i$  (where  $i \in \{1, 2, 3\}$ ) are hyperparameters that balance the contributions of different regularization terms.  $\theta_{env}^*$  and  $\theta_{self}^*$  are the pre-trained encoder parameters, and the disentanglement regularization term  $I(z_{env}, z_{self})$  ensures continued minimization of mutual information between environmental and self-state representations throughout the training process.

#### IV. EXPERIMENT

We conduct comprehensive experiments to evaluate our method across foundational skill learning and Long-Horizon(LH) task execution. Section IV-A assesses the robustness of foundational skill acquisition and task completion. Section IV-B provides ablation studies on each component of the framework. Section IV-C illustrates the model’s generalization to complex LH tasks with diverse skill and environment compositions.

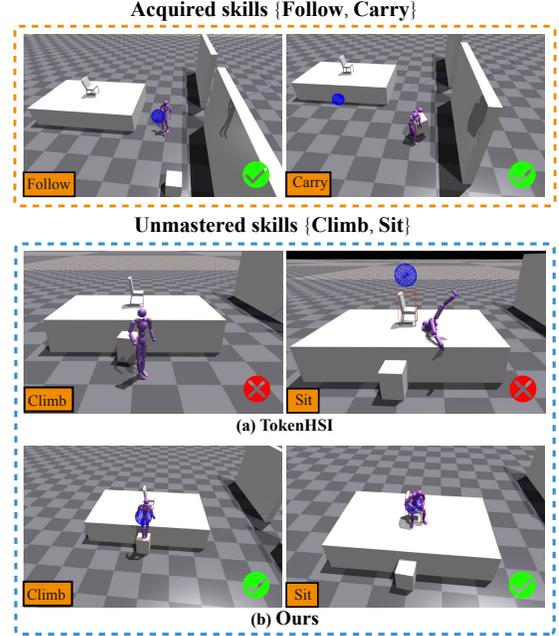


Fig. 4: Skill acquisition performance comparison between Detach and TokenHSI. The orange box represents the skills learned in pre-training, and the blue boxes represent new generalized skills.

Our experiments are conducted entirely on three LH tasks that we designed:

**LH1: “Sit on Chair!”** This LH task comprises a sequence of four fundamental skills: *Follow*, *Carry*, *Climb*, and *Sit*, where the target object for *Sit* is a Chair.

**LH2: “Sit on Sofa!”** This LH task similarly comprises a sequence of four fundamental skills: *Follow*, *Carry*, *Follow*, and *Sit*, where the target object for *Sit* is a Sofa.

**LH3: “Go to Bed!”** This LH task comprises a sequence of five fundamental skills: *Follow*, *Carry*, *Follow*, *Climb*, and *Sit*, where the target object for *Sit* is a Bed.

Our object assets are sourced from the 3D-FRONT dataset [35], while the motion data is inherited from TokenHSI [11].

##### A. Evaluation on Foundational Skill Learning and Task Completion

**Experimental Setup.** To evaluate the robustness and universality of our disentangled architecture, we employ a progressive learning protocol where foundational skills *Follow* and *Carry* are established through comprehensive training, while *Climb* and *Sit* skills are acquired through compositional learning. This approach enables systematic assessment of skill generalization capabilities and adaptation to diverse environments. The training procedure uses large-scale parallelization in 4,096 environments, employing PPO [36] with 10k iterative updates. We conducted 100 independent experimental trials to ensure statistical reliability, quantifying robustness and universality through the success rate means of all skills and LH tasks. This rigorous evaluation framework provides a compre-

| Experiment | Method   | Follow      | Carry       | Follow      | Climb       | Sit         | Time(s)      | LH.         | SGR.        | EGR.        |
|------------|----------|-------------|-------------|-------------|-------------|-------------|--------------|-------------|-------------|-------------|
| LH2        | TokenHSI | 1.00        | 0.56        | 0.13        | -           | 0.01        | 99.00        | 0.42        | 0.01        | 0.76        |
|            | Ours     | <b>1.00</b> | <b>0.96</b> | <b>0.67</b> | -           | <b>0.16</b> | <b>85.00</b> | <b>0.70</b> | <b>0.08</b> | <b>0.97</b> |
| LH3        | TokenHSI | 1.00        | 0.50        | 0.21        | 0.20        | 0.00        | 102.90       | 0.38        | 0.67        | 0.69        |
|            | Ours     | <b>1.00</b> | <b>0.95</b> | <b>0.50</b> | <b>0.40</b> | <b>0.10</b> | <b>97.60</b> | <b>0.59</b> | <b>0.13</b> | <b>0.81</b> |

TABLE II: Comparison of generalization performance between TokenHSI and DETACH on LH tasks.

hensive assessment of the architecture’s performance through systematic skill composition and environmental adaptation.

**Baselines.** We train TokenHSI from scratch using our custom dataset. TokenHSI is a state-of-the-art full-body humanoid controller that learns a set of foundational skills comparable to ours. We also include CML [17], a composite motion learning baseline commonly used alongside TokenHSI, as an additional point of reference.

**Follow and Carry.** The success rate of *Follow* is defined as maintaining the pelvis within a 30cm distance threshold from the target path in the XY plane. For the *Carry* task, which can be decomposed into ‘grasp’ and ‘transport’ components, achieving only the grasp phase without successful transport to the designated target location is considered 0.5 task completion. *Follow* task training utilized procedurally generated trajectories, while *Carry* task training employed 9 boxes of varying dimensions. Subsequently, we trained on the compositional LH1 task combining these two primitives and evaluated performance on identical task compositions.

**Climb and Sit.** The success of *Climb* is defined as reaching the target object with the pelvis positioned at or above the target elevation. Success of *Sit* requires the pelvis to be positioned on the upper surface of the target object.

**LH1 task.** The Success rate for LH1 is the success rate of the sub-skill sequence. Due to the sequential nature of LH tasks, where skills must be executed in order, failure in a preceding task prevents the execution of subsequent tasks. Therefore, the skill sequence success rate serves as an excellent metric for evaluating the success rate of LH tasks.

**Results.** Table I presents the quantitative analysis results, where we evaluated the effectiveness of three methods: CML, TokenHSI, and DETACH. While all methods demonstrate comparable performance on pre-trained skills such as *Follow* and *Carry*, Figure 3 and Figure 4 reveal that, compared to methods with limited generalization capabilities like CML and TokenHSI, our DETACH method maintains high success rates for pre-trained skills while achieving success rates of 51% and 42% on two additional tasks, *Climb* and *Sit*, respectively, significantly outperforming the other two approaches. In contrast, TokenHSI and CML exhibit limited generalization on these tasks, resulting in success rates approaching zero. Furthermore, in terms of overall task success rate, DETACH achieves 72%, surpassing CML and TokenHSI by 42% and 17%, respectively. These results highlight DETACH’s stability in executing existing skills and its versatility in handling novel tasks, demonstrating its superior performance capabilities.

### B. Long-Horizon Task Completion

This section evaluates the DETACH framework’s performance on Long-Horizon (LH) tasks, designed to test generalization across skills and environments. We focus on **skill generalization** and **environment generalization**, using LH2 and LH3 for assessment, which target adaptation to novel environments and task compositions. Generalization is evaluated over 100 test runs per task, measuring subtask success rates in diverse, unseen scenes to assess robustness.

**Task Execution Times.** Task execution time refers to the duration from the start of the current LH task to the initiation of the next LH task. The criteria for determining the execution of the next task include the occurrence of errors (such as falling) or exceeding the threshold time for task execution.

**Experiment setup.** As described in Section IV-A, to validate the environment generalization capability in this section, we employ the same progressive learning protocol on the LH1 task and directly evaluate generalization on the LH2 and LH3 tasks. This approach allows us to observe the environment generalization capability of our DETACH framework more intuitively. Since we similarly establish foundational skills *Follow* and *Carry* through comprehensive training, we can also assess skill generalization rates through the completion performance on *Climb* and *Sit*.

**Generalization Rate Definition.** Based on our experimental data, we formally define the Environment Generalization Rate (EGR) and Skill Generalization Rate (SGR) as follows:

$$EGR = \frac{S_{Li}}{S_{L1}}, i \in 2, 3 \quad (22)$$

$$SGR = \frac{(S_{climb} + S_{sit})/2}{(S_{follow} + S_{carry})/2} \quad (23)$$

where  $S_{Li}, i \in \{1, 2, 3\}$  represents the success rate of LH tasks, and the testing on LH2 and LH3 involves direct transfer from the LH1 environment training, demonstrating its rationality. Similarly,  $S_{climb}$  etc. represent the success rates of skills, where *Climb* and *Sit* are composed from foundational *Follow* and *Carry* skills; therefore, we define the skill generalization rate using this formula.

**Results.** Figure 5 visually highlights DETACH’s superior skill composition over TokenHSI. Table II compares subtask success rates, LH task success rates, execution times, and environment/skill generalization rates, showing DETACH’s consistent outperformance. Specifically, DETACH reduces average execution times by 14s (LH2) and 5s (LH3) compared

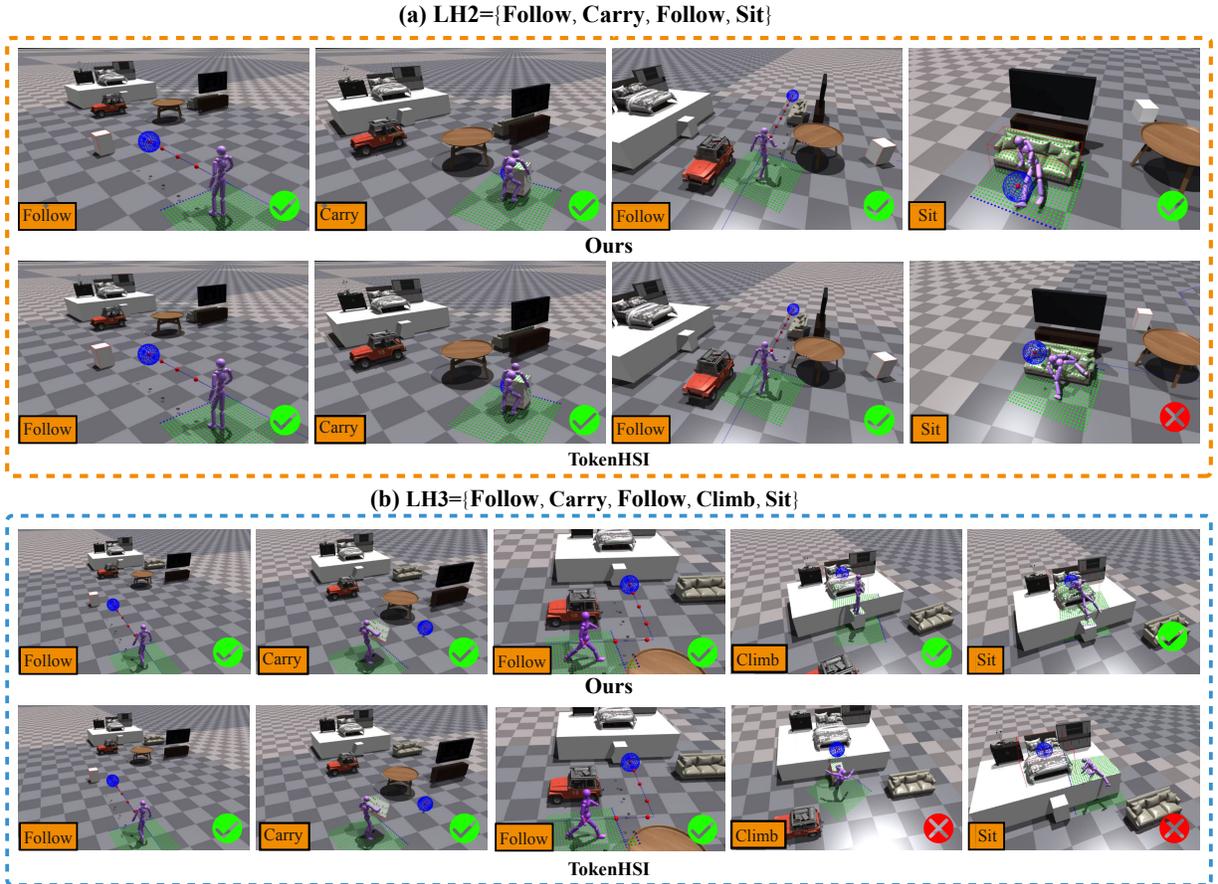


Fig. 5: Generalization comparison between DETACH and TokenHSI on LH tasks, where (a) and (b) represent tasks composed of sequences of four and five foundational skills, respectively. We only pre-trained the first two actions, *Follow* and *carry* on LH1 tasks, and tested skill generalization and environmental generalization in new scenarios.

| ID   | Configuration                  | EGR.        | SGR.        | LH.         |
|------|--------------------------------|-------------|-------------|-------------|
| Full | All modules enabled            | <b>0.81</b> | <b>0.13</b> | <b>0.58</b> |
| A1   | w/o Env Encoder $\Phi_{env}$   | <b>0.64</b> | 0.12        | <b>0.41</b> |
| A2   | w/o Self Encoder $\Phi_{self}$ | 0.74        | <b>0.05</b> | <b>0.38</b> |

TABLE III: Ablation study results. Each variant disables one key module. Bold indicates best performance.

to TokenHSI, enhancing efficiency in human body control. Task success rates improve by 28% (LH2) and 21% (LH3), balancing efficiency with success. Notably, environment generalization rates reach 0.08 (LH2) and 0.13 (LH3), with skill generalization rates of 0.97 (LH2) and 0.81 (LH3), significantly exceeding TokenHSI. These results underscore DETACH’s enhanced composition capabilities for long-horizon HSI tasks.

### C. Ablation Experiment

To evaluate the individual contributions of key modules in our DETACH framework, we perform a comprehensive ablation study. Each variant is constructed by disabling or

removing a specific component from the full model while keeping all other settings fixed. The experiments are conducted on LH3 tasks composed of foundational skill primitives (e.g., *follow*, *carry*, *climb*, *sit*) in diverse environments.

**Experimental Setup.** We recorded three key metrics: environment generalization success rate, skill generalization success rate, and overall LH task success rate. Each model was trained under the same progressive learning protocol as described in Section IV-A and evaluated by executing LH3, with results shown in Table III below.

**Results.** As shown in Table III, removing the environmental encoder (A1) causes the environment generalization rate to drop from 0.81 to 0.64, while removing the self encoder (A2) leads to a decrease in skill generalization rate from 0.127 to 0.045. Removing any component results in either substantial or moderate degradation across all metrics. This confirms that each encoder in our framework serves a distinct function and is indispensable to the overall architecture.

## V. CONCLUSION AND FUTURE WORK

In this work, we presented **DETACH**, a biologically inspired dual-stream disentanglement framework that explicitly

separates environment understanding from self-state encoding. This design enables **cross-domain transfer, modular skill reuse, and efficient long-horizon task composition**. Extensive experiments on diverse HSI scenarios demonstrate that DETACH achieves substantial improvements of 23% in sub-task success rate and 29% in execution efficiency, along with stronger generalization over state-of-the-art modular baselines.

While our current implementation relies on a pre-defined skill set, future work will explore open-ended skill discovery from unlabeled data and real-world deployment under dynamic environments. We believe DETACH provides a promising step toward scalable, generalizable embodied intelligence in complex human-scene interactions.

## REFERENCES

- [1] R.-Z. Qiu, Y. Hu, Y. Song, G. Yang, Y. Fu, J. Ye, J. Mu, R. Yang, N. Atanasov, S. Scherer *et al.*, “Learning generalizable feature fields for mobile manipulation,” *arXiv preprint arXiv:2403.07563*, 2024.
- [2] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi *et al.*, “Openvla: An open-source vision-language-action model,” *arXiv preprint arXiv:2406.09246*, 2024.
- [3] L. Zhang, K. Bai, G. Huang, Z. Bing, Z. Chen, A. Knoll, and J. Zhang, “Contactdexnet: Multi-fingered robotic hand grasping in cluttered environments through hand-object contact semantic mapping,” *arXiv preprint arXiv:2404.08844*, 2024.
- [4] C. Sferazza, D.-M. Huang, X. Lin, Y. Lee, and P. Abbeel, “Humanoid-bench: Simulated humanoid benchmark for whole-body locomotion and manipulation,” *arXiv preprint arXiv:2403.10506*, 2024.
- [5] J. Zhang, Y. Chen, Z. Wang, J. Yang, Y. Wang, and S. Huang, “Interactanything: Zero-shot human object interaction synthesis via llm feedback and object affordance parsing,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 7015–7025.
- [6] T. Wang, X. Mao, C. Zhu, R. Xu, R. Lyu, P. Li, X. Chen, W. Zhang, K. Chen, T. Xue *et al.*, “Embodiedscan: A holistic multi-modal 3d perception suite towards embodied ai,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 757–19 767.
- [7] S. Xu, Y.-X. Wang, L. Gui *et al.*, “Interdreamer: Zero-shot text to 3d dynamic human-object interaction,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 52 858–52 890, 2024.
- [8] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky, “ $\pi_0$ : A vision-language-action flow model for general robot control,” 2024. [Online]. Available: <https://arxiv.org/abs/2410.24164>
- [9] G. R. Team, S. Abeyruwan, J. Ainslie, J.-B. Alayrac, M. G. Arenas, T. Armstrong, A. Balakrishna, R. Baruch, M. Bauza, M. Blokzijl *et al.*, “Gemini robotics: Bringing ai into the physical world,” *arXiv preprint arXiv:2503.20020*, 2025.
- [10] M. Ni, L. Zhang, Z. Chen, K. Bai, Z. Chen, J. Zhang, and W. Zuo, “Don’t let your robot be harmful: Responsible robotic manipulation via safety-as-policy,” *arXiv preprint arXiv:2411.18289*, 2024.
- [11] L. Pan, Z. Yang, Z. Dou, W. Wang, B. Huang, B. Dai, T. Komura, and J. Wang, “Tokenhsi: Unified synthesis of physical human-scene interactions through task tokenization,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 5379–5391.
- [12] Z. Li, Y. Xie, R. Shao, G. Chen, D. Jiang, and L. Nie, “Optimus-1: Hybrid multimodal memory empowered agents excel in long-horizon tasks,” *Advances in neural information processing systems*, vol. 37, pp. 49 881–49 913, 2024.
- [13] C. F. Park, A. Lee, E. S. Lubana, Y. Yang, M. Okawa, K. Nishi, M. Wattenberg, and H. Tanaka, “Iclr: In-context learning of representations,” *arXiv preprint arXiv:2501.00070*, 2024.
- [14] Z. Xiao, T. Wang, J. Wang, J. Cao, W. Zhang, B. Dai, D. Lin, and J. Pang, “Unified human-scene interaction via prompted chain-of-contacts,” *arXiv preprint arXiv:2309.07918*, 2023.
- [15] W. Huang, I. Mordatch, and D. Pathak, “One policy to control them all: Shared modular policies for agent-agnostic control,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 4455–4464.
- [16] S. Lan, R. Zhang, Q. Yi, J. Guo, S. Peng, Y. Gao, F. Wu, R. Chen, Z. Du, X. Hu *et al.*, “Contrastive modules with temporal attention for multi-task reinforcement learning,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 36 507–36 523, 2023.
- [17] P. Xu, X. Shang, V. Zordan, and I. Karamouzas, “Composite motion learning with task control,” *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1–16, 2023.
- [18] J. Hu, Z. Wang, P. Stone, and R. Martín-Martín, “Disentangled unsupervised skill discovery for efficient hierarchical reinforcement learning,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 76 529–76 552, 2024.
- [19] S. L. Li, A. Zhang, B. Chen, H. Matusik, C. Liu, D. Rus, and V. Sitzmann, “Controlling diverse robots by inferring jacobian fields with deep networks,” *Nature*, pp. 1–7, 2025.
- [20] G. M. van de Ven, N. Soares, and D. Kudithipudi, “Continual learning and catastrophic forgetting,” *arXiv preprint arXiv:2403.05175*, 2024.
- [21] L. G. Ungerleider, “Two cortical visual systems,” *Analysis of visual behavior*, vol. 549, pp. chapter–18, 1982.
- [22] T. Ibrayev, A. Mukherjee, S. A. Aketi, and K. Roy, “Toward two-stream foveation-based active vision learning,” *IEEE Transactions on Cognitive and Developmental Systems*, vol. 16, no. 5, pp. 1843–1860, 2024.
- [23] D. Arkhangelsky and G. Imbens, “Causal models for longitudinal and panel data: A survey,” *The Econometrics Journal*, vol. 27, no. 3, pp. C1–C61, 2024.
- [24] N. Jiang, Z. Zhang, H. Li, X. Ma, Z. Wang, Y. Chen, T. Liu, Y. Zhu, and S. Huang, “Scaling up dynamic human-scene interaction modeling,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 1737–1747.
- [25] Z. Li, Y. Xie, R. Shao, G. Chen, D. Jiang, and L. Nie, “Optimus-2: Multimodal minecraft agent with goal-observation-action conditioned policy,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 9039–9049.
- [26] S. Zheng, J. Liu, Y. Feng, and Z. Lu, “Steve-eye: Equipping llm-based embodied agents with visual perception in open worlds,” *arXiv preprint arXiv:2310.13255*, 2023.
- [27] Z. Li, Y. Xie, R. Shao, G. Chen, W. Guan, D. Jiang, and L. Nie, “Optimus-3: Towards generalist multimodal minecraft agents with scalable task experts,” *arXiv preprint arXiv:2506.10357*, 2025.
- [28] S. E. Ada, E. Oztop, and E. Ugur, “Diffusion policies for out-of-distribution generalization in offline reinforcement learning,” *IEEE Robotics and Automation Letters*, vol. 9, no. 4, pp. 3116–3123, 2024.
- [29] A. Uppal, Y. Takida, C.-H. Lai, and Y. Mitsufuji, “Denoising multi-beta vae: Representation learning for disentanglement and generation,” *arXiv preprint arXiv:2507.06613*, 2025.
- [30] P. Bhowal, A. Soni, and S. Rambhatla, “Why do variational autoencoders really promote disentanglement?” in *Proceedings of the 41st International Conference on Machine Learning*, vol. 235, 2024, pp. 3817–3849.
- [31] Y. Yang, T. Zhou, Q. He, L. Han, M. Pechenizkiy, and M. Fang, “Task adaptation from skills: Information geometry, disentanglement, and new objectives for unsupervised reinforcement learning,” *arXiv preprint arXiv:2506.10629*, 2025.
- [32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [33] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit *et al.*, “Mlp-mixer: An all-mlp architecture for vision,” *Advances in neural information processing systems*, vol. 34, pp. 24 261–24 272, 2021.
- [34] T. Zadouri, A. Üstün, A. Ahmadian, B. Ermiş, A. Locatelli, and S. Hooker, “Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning,” *arXiv preprint arXiv:2309.05444*, 2023.
- [35] H. Fu, B. Cai, L. Gao, L.-X. Zhang, J. Wang, C. Li, Q. Zeng, C. Sun, R. Jia, B. Zhao *et al.*, “3d-front: 3d furnished rooms with layouts and semantics,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 933–10 942.
- [36] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.

### A. Simulated Character

Our humanoid dataset builds upon TokenHSI [11].

TokenHSI develops a customized character model with 32 degrees of freedom based on the AMP system, comprising 15 rigid bodies, 12 controllable joints, and 32 degrees of freedom. To address the mismatch between the SMPL parameters used in reference motion datasets and the kinematic structure of the AMP character model, TokenHSI implements three key improvements:

- 1) Adjustment of the 3D positions of lower limb joints (hip, knee, and ankle joints) to align with the SMPL human body model;
- 2) Adoption the SimPoE method to transform foot collision shapes from rectangular boxes to realistic foot meshes;
- 3) Upgrading knee joints from 1-DOF rotational joints to 3-DOF spherical joints.

These improvements aim to reduce motion retargeting errors and enhance the naturalness and fluidity of character movements. The improved model is applicable to most scenarios.

### B. Environment Construction

We utilized object assets from the 3D-Front scene dataset to construct three long-horizon task environments, each designed for distinct experimental scenarios. Environment LH1 is composed of walls, boxes, platforms, and chairs. Environment LH2 incorporates boxes, tables, toy cars, sofas, and television cabinets. Environment LH3 encompasses boxes, tables, toy cars, platforms, nightstands, beds, cushions, and wardrobes. Each environment is configured to support different task specifications.

We implement standardized transformation and multi-dimensional data extraction for 3D mesh models, providing comprehensive object descriptions for robotic interaction tasks.

**In the mesh standardization processing stage:** We first load the original normalized models, then apply a specific 90-degree rotation transformation matrix to adjust the coordinate system orientation, followed by calculating the bounding box and repositioning the model at the origin, ensuring all processed objects have a unified spatial reference framework and standardized geometric representation. **In the multi-dimensional feature data generation stage:** We perform bounding box dimension calculations and set standard orientation vectors based on the standardization, use ray tracing techniques to generate  $128 \times 128$  resolution height maps to capture fine geometric information of object surfaces, and simultaneously calculate two key target interaction positions based on the height map data we set the sitting position 0.1 units above the surface and the climbing position directly at the surface height. **In the complete data output stage:** We generate four types of output data: processed standardized meshes, JSON configurations containing bounding box dimensions and target positions, numerical data and visualization images of height maps, and comprehensive visualization

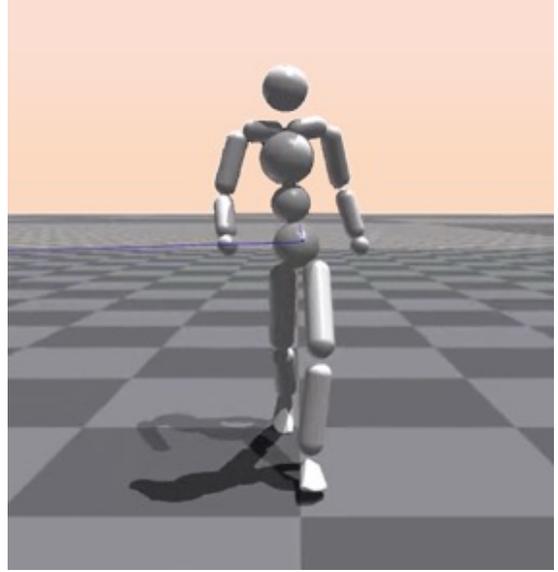


Fig. 6: The humanoid we use.

models integrating original models, position marker spheres, bounding box wireframes, and ground planes.

Through the entire processing pipeline, we convert original 3D models into standardized, structured data formats suitable for robotic sitting planning, object climbing, or other spatial interaction tasks, providing a complete geometric and semantic information foundation for our subsequent algorithm development and simulation.

### C. Task Configuration

Based on our established long-horizon task scenarios, we designed three tasks with increasing complexity to progressively validate the effectiveness of multi-skill composition.

1) *LH1: Basic Four-Skill Composition:* . LH1 represents a relatively simple long-horizon task designed to validate fundamental four-skill combinations. The scene consists of five core objects: a carryable box positioned at  $[4.5, -4.0, 0.35]$ , a static chair at  $[0, 0, 1.46]$  with  $-1.57$  radians rotation, a base platform, and two boundary walls located at  $[6, -3.25, 1.5]$  and  $[6, 3.25, 1.5]$  respectively. The execution plan follows [“traj”, “carry”, “climb”, “sit”], requiring the agent to navigate along a predefined trajectory from the starting point  $[10.0, 4.0]$  to an intermediate position, pick up and carry the box to the target position  $[0, -1.8, 0.35]$ , climb to a specified height, and finally sit on the chair to complete the task. We configure target object indices as  $[0, 0, 0, 1]$  with sampling sources [“traj\_0”, “tarpos\_0”, “scene\_0”, “scene\_1”].

2) *LH2: Complex Indoor Scene Navigation:* LH2 constitutes a sophisticated indoor scenario that simulates realistic residential interactions. We construct a rich indoor environment with ten objects: two carryable boxes positioned at  $[11, 11, 0.5]$  and  $[15, 18.2, 0.5]$ , furniture including a bed ( $[15, 20, 1.5]$ ), a table ( $[15, 13, 0.3]$ ), nightstands ( $[13.7, 21, 1.7]$  and  $[16.5, 21, 1.65]$ ), sofa ( $[18, 17, 0.35]$ ), TV stand ( $[18, 13, 0.35]$ ), television ( $[18, 13, 1.5]$ ), along with a car ( $[13.2,$



Fig. 7: Object assets extracted from 3D-FRONT (3D Furnished Rooms with layOuts and semaNTics)

TABLE IV: Task Configuration Overview

| Task | Scene Objects        |       |           |             | Skills           |         |          |          |                 | Complexity |          |         |
|------|----------------------|-------|-----------|-------------|------------------|---------|----------|----------|-----------------|------------|----------|---------|
|      | Total                | Boxes | Furniture | Dynamic     | Traj             | Carry   | Climb    | Sit      | Multi-Traj      | Skills     | Traj Pts | Targets |
| LH1  | 5                    | 1     | 1         | 1           | ✓                | ✓       | ✓        | ✓        | ✗               | 4          | 4        | 2       |
| LH2  | 10                   | 2     | 7         | 3           | ✓                | ✓       | ✗        | ✓        | ✓               | 4          | 12       | 2       |
| LH3  | 11                   | 2     | 8         | 3           | ✓                | ✓       | ✓        | ✓        | ✓               | 5          | 16       | 3       |
| Task | Environment Features |       |           |             | Object Positions |         |          |          | Sampling Config |            |          |         |
|      | Rotation             | Rooms | Vehicle   | Electronics | Start X          | Start Y | Target X | Target Y | Sources         | Obj Idx    | Trajs    |         |
| LH1  | ✓                    | ✗     | ✗         | ✗           | 10.0             | 4.0     | 0        | -1.8     | 4               | 2          | 1        |         |
| LH2  | ✓                    | ✓     | ✓         | ✓           | 11.5             | 0.0     | 15       | 11       | 4               | 2          | 2        |         |
| LH3  | ✓                    | ✓     | ✓         | ✓           | 11.5             | 0.0     | 15       | 11       | 5               | 3          | 2        |         |

16, 0.5]) and a support platform. The execution plan [“traj”, “carry”, “traj”, “sit”] incorporates dual-trajectory navigation: traversing trajectory 0 from [11.5, 0.0] to [11.5, 8.0], carrying the first box to target position [15, 11, 0.35], navigating trajectory 1 from [14, 11.0] to [18.0, 14.4], and sitting on the sofa to complete the task. Target object indices are [0, 0, 0, 8] with sampling sources [“traj\_0”, “scene\_0”, “traj\_1”, “scene\_8”].

3) *LH3: Comprehensive Five-Skill Integration*: LH3 represents the most complex indoor scenario designed to validate comprehensive five-skill composition capabilities. We extend the environment to eleven objects by adding a second box at [15, 18.2, 0.43] to LH2’s configuration, creating a more intricate interaction scenario. The execution plan [“traj”, “carry”, “traj”, “climb”, “sit”] implements the complete skill sequence: initial trajectory navigation, object manipulation, complex path navigation (trajectory 1 contains twelve waypoints from [14, 11.0] to [15.0, 16.0]), climbing skill execution, and final sitting completion. Target object indices are configured as [0, 0, 0, 1, 2] with sampling sources [“traj\_0”, “tarpos\_0”, “traj\_1”, “scene\_1”, “scene\_2”], supporting more sophisticated multi-object interactions.

These three tasks follow a progressive complexity pattern, systematically validating multi-skill composition effectiveness in long-horizon tasks from LH1’s basic four-skill verification through LH2’s complex scene navigation to LH3’s comprehensive five-skill integration.

4) *Task Metrics*: Based on the task configuration table, the three long-horizon tasks (LH1-LH3) demonstrate a progressive complexity design for validating multi-skill composition effectiveness.

**Scene Complexity**: The environments progress from LH1’s basic 5-object setup to LH3’s comprehensive 11-object indoor scenario, incorporating carryable boxes, various furniture pieces (chairs, beds, tables, sofas, TV stands), and dynamic elements like vehicles.

**Skill Integration**: The progression is systematic. LH1 validates fundamental four-skill composition (trajectory, carry, climb, sit), LH2 introduces complex indoor navigation with dual-trajectory execution, and LH3 achieves complete five-skill integration, including multi-trajectory navigation capabilities.

**Task Complexity Metrics**: This reflects the progression: LH1 requires 4 skills across 4 trajectory points targeting 2 objects, while LH3 requires 5 skills across 16 trajectory points targeting 3 objects.

**Environmental Features**: The sophistication advances from LH1’s simple boundary-wall setup to LH2 and LH3’s realistic residential environments featuring room structures, vehicles, and electronics.

**Execution Plans**: These follow increasingly complex sequences - LH1’s basic “traj-carry-climb-sit” pattern, LH2’s dual-trajectory indoor navigation simulation, and LH3’s comprehensive skill sequence validation, creating a systematic framework that progresses from basic skill verification through

complex scene navigation to complete multi-skill integration testing.

TABLE V: Environment & Simulation Parameters

| Parameter Name        | Value            |
|-----------------------|------------------|
| Parallel Environments | 4096             |
| Episode Length        | 1200 steps (40s) |
| Control Frequency     | 30Hz             |
| Sub-steps             | 2 steps          |
| Environment Spacing   | 5m               |
| Physics Engine        | PhysX            |
| Solver Type           | TGS (Type 1)     |
| Position Iterations   | 4 iter           |
| Contact Offset        | 0.02m            |
| Static Friction       | 1.0              |
| Dynamic Friction      | 1.0              |

TABLE VI: Neural Network Parameters

| Parameter Name         | Value                |
|------------------------|----------------------|
| Transformer Layers     | 4 layers             |
| Attention Heads        | 8 heads              |
| Base Feature Dimension | 64D                  |
| Task Observation Space | [128, 96, 112, 144]D |
| Adapter Units          | [1024, 512]D         |

#### D. Parameter configuration

1) *Environment and Simulation Parameters* : DETACH controls the fundamental simulation setup with 4096 parallel environments, each episode of 1200 steps (40 seconds). Control commands are executed at a 30Hz frequency with 2 physics simulation substeps between controls. We spatially separate environment instances by 5 meters. The PhysX [?] physics engine uses TGS (Type 1) solver with 4 position constraint iterations, 0.02m contact detection offset, and both static and dynamic friction coefficients set to 1.0.

2) *Neural Network Parameters*: The neural network architecture employs a 4-layer Transformer with an 8-head multi-head attention mechanism. Base feature vectors are 64-dimensional, while different tasks have varying observation space dimensions: [128, 96, 112, 144]. Network adapters use hidden layer dimensions [1024, 512] to handle task-specific adaptations.

3) *Training and Reward Parameters*: The training strategy utilizes Adversarial Motion Priors (AMP) with 10-step historical observations and skill selection based on specified probability distribution [0.1, 0.1, 0.2, 0.1, 0.1, 0.05, 0.0, 0.05, 0.1, 0.1, 0.1]. We adopt the mixed initialization strategy with 0.5 probability using the default state initialization mode. Task transitions allow a maximum of 60 steps during training and 20 steps during testing, with success determined by a 0.3 m distance threshold. Interactive Early Termination (IET) is enabled only for the final subtask with task-specific discrimination activated. The reward function includes power consumption weighting (coefficient 0.0005), trajectory tracking failure at 4.0m distance, episode termination at 0.15m height, dynamic object speed penalty (coefficient 1.0, threshold 1.5), and decoupling mask strength of 0.3

TABLE VII: Training & Reward Parameters

| Parameter Name               | Value                      |
|------------------------------|----------------------------|
| <b>Training Strategy</b>     |                            |
| AMP Observation Steps        | 10 steps                   |
| Mixed Initialization Prob    | 0.5                        |
| State Initialization         | Default mode               |
| Max Transition Steps         | Train: 60, Test: 20        |
| Success Threshold            | 0.3m                       |
| IET Enable                   | Last subtask only          |
| Task-specific Discrimination | Enabled                    |
| <b>Reward Function</b>       |                            |
| Power Reward                 | Enabled, coeff: 0.0005     |
| Trajectory Failure Distance  | 4.0m                       |
| Termination Height           | 0.15m                      |
| Dynamic Object Speed Penalty | Coeff: 1.0, Threshold: 1.5 |
| Decoupling Mask Strength     | 0.3                        |

TABLE VIII: Data Generation Parameters

| Parameter Name               | Value                |
|------------------------------|----------------------|
| <b>Height Map Perception</b> |                      |
| Use Height Map               | Enabled              |
| Cube Side Length             | 2.0m                 |
| Grid Points                  | 25 × 25              |
| Grid Spacing                 | 0.1m                 |
| Field of View Spacing        | 1.0m                 |
| Camera Height                | 10.0m                |
| <b>Trajectory Generation</b> |                      |
| Trajectory Sample Points     | 10 points            |
| Sampling Time Step           | 0.5s (10Hz)          |
| Velocity Range               | 1.4-1.5 m/s          |
| Maximum Acceleration         | 2.0 m/s <sup>2</sup> |
| Sharp Turn Probability       | 0.02                 |
| Sharp Turn Angle             | 1.57 rad (90°)       |

4) *Data Generation Parameters*: Height map perception is enabled with a 2.0m cube coverage area using a 2525 grid resolution (0.1m spacing between adjacent points). The field of view is set to 1.0m with the camera positioned at 10.0m height for data acquisition. Reference trajectories are generated with 10 sampling points at 0.5s intervals (2Hz frequency). Trajectory velocity ranges from 1.4 to 1.5 m/s with a maximum acceleration of 2.0 m/s. Sharp turns occur with 0.02 probability, creating 1.57 radian (90) angle changes when triggered.

#### E. Failure Analysis

We conducted a comprehensive analysis of failure modes across different methods, summarized by task categories below:

**LH1 Task** The humanoid agent exhibits failures across multiple execution phases. Initially, spawn point configurations present challenges due to random orientation initialization—specifically, when the humanoid initializes at  $-90$ , the 180 angular deviation from the target trajectory causes instability during turning maneuvers, frequently resulting in falls. During object manipulation phases, the agent fails to successfully grasp and lift boxes due to insufficient grip strength or improper contact dynamics. In addition, climbing sequences are prone to failure due to foot placement errors that lead to missed footholds and subsequent falls.

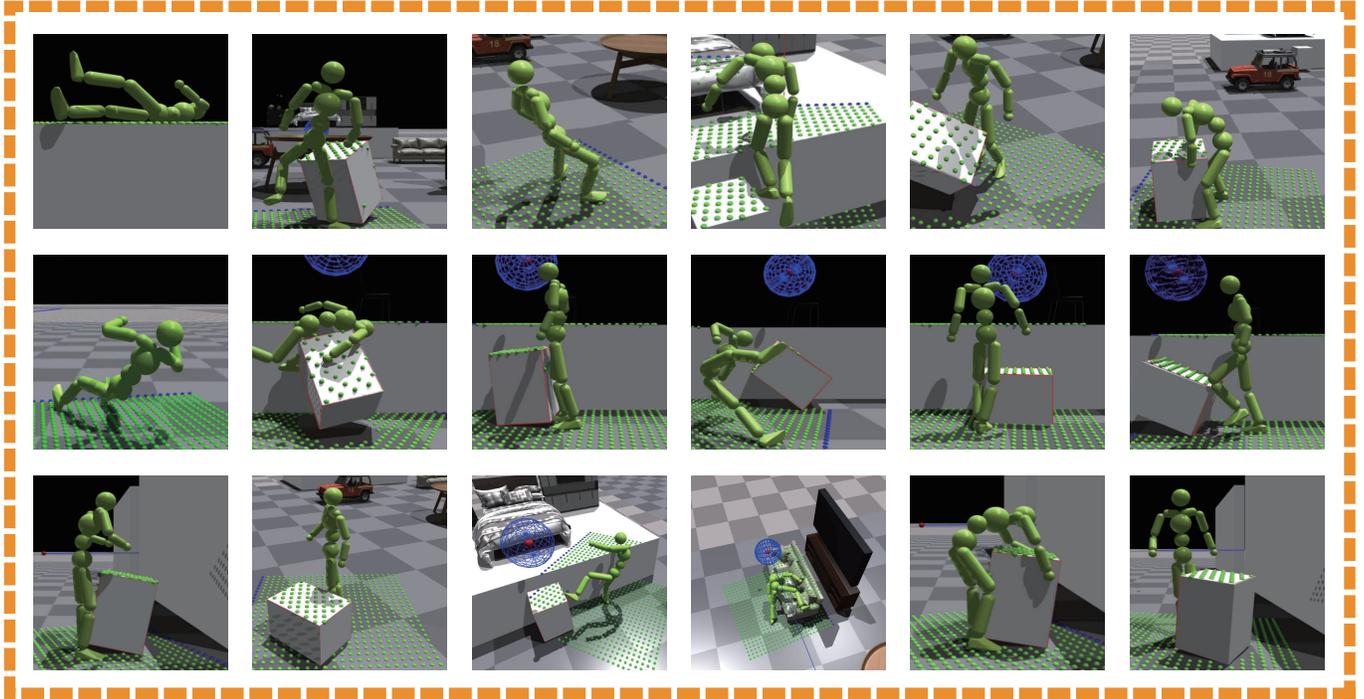


Fig. 8: Negative Results Presentation

**LH2 Task** Box carrying operations frequently fail due to inadequate lifting capabilities, causing the agent to remain stationary. Subsequently, detection failures occur when the box is not properly positioned within the target zone, preventing trigger activation for subsequent actions and resulting in deadlock states. During locomotion phases, collisions with static environmental objects can cause destabilization and falls. Chair sitting behaviors are compromised by scale mismatches in the furniture models, leading to improper seating postures and misalignment.

**LH3 Task** Climbing maneuvers exhibit similar failure patterns to LH1, with foot placement inaccuracies resulting in falls. Bed-sitting tasks present unique challenges due to the geometric constraints and physical properties of the bed model, preventing successful completion of proper sitting behaviors in the conventional sense.

These failure modes highlight the critical need for improved contact dynamics, environmental awareness, and robust motion planning in humanoid robot control systems.

### F. Methodological Supplement

In this section, we supplement the main methodology with additional details that were omitted from the primary exposition.

1) *Progressive Training Protocol*: To maintain the specialized characteristics of modules during joint training, we designed three key regularization constraints:

**Feature Decoupling Regularization** Reinforces independence by minimizing mutual information between environ-

mental features and self-features:

$$\mathcal{R}_{decouple} = \|\text{Corr}(z_{env}, z_{self})\|_F^2 \quad (24)$$

where  $\text{Corr}(\cdot, \cdot)$  computes the feature correlation matrix, and  $\|\cdot\|_F$  denotes the Frobenius norm.

**Temporal Consistency Constraint** Applies temporal smoothness constraints to the output of the self-state encoder:

$$\mathcal{R}_{temporal} = \sum_{t=1}^{T-1} \|z_{self}^{t+1} - z_{self}^t\|_2^2 \quad (25)$$

**Semantic Preservation Constraint** Ensures that encoded features maintain original semantic information through reconstruction loss:

$$\mathcal{R}_{semantic} = \alpha \|\text{Decoder}_{env}(z_{env}) - obs_{env}^t\|_2^2 + \beta \|\text{Decoder}_{self}(z_{self}) - obs_{self}^t\|_2^2 \quad (26)$$