AnySkill: Learning Open-Vocabulary Physical Skill for Interactive Agents

Jieming Cui[1,2*], Tengyu Liu[2*], Nian Liu[2,3*], Yaodong Yang[1], Yixin Zhu[1, ✉], Siyuan Huang[2, ✉]

[1] Institute for Artificial Intelligence, Peking University [2] National Key Laboratory of General Artificial Intelligence, BIGAI
[3] School of Artificial Intelligence, Beijing University of Posts and Telecommunications

https://anyskill.github.io

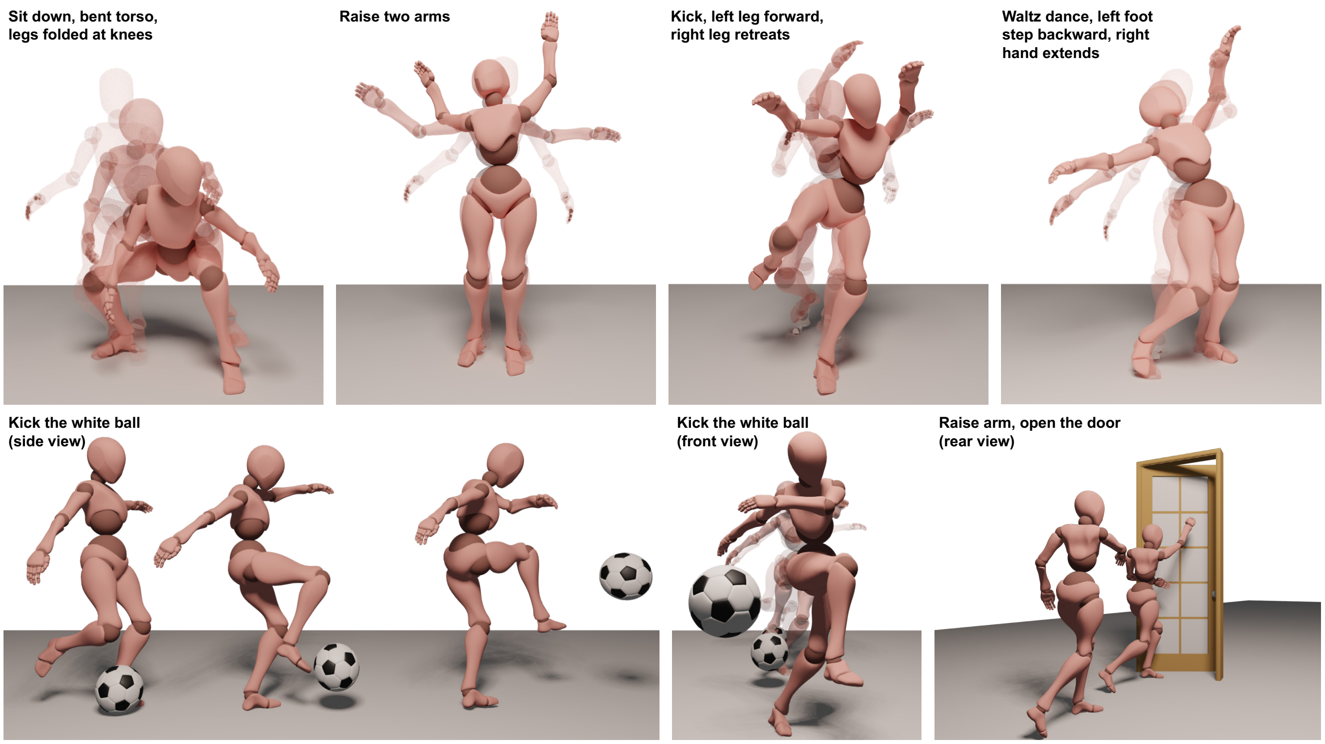


Figure 1. **Diverse motions generated by `AnySkill` conditioned on various instructions.** When provided with an open-vocabulary text description of a motion, `AnySkill` is adept at learning natural and flexible motions that closely align with the description, facilitated by an image-based reward mechanism. Additionally, `AnySkill` demonstrates proficiency in learning interactions with dynamic objects, showcasing its versatile motion generation capabilities.

Abstract

Traditional approaches in physics-based motion generation, centered around imitation learning and reward shaping, often struggle to adapt to new scenarios. To tackle this limitation, we propose `AnySkill`, a novel hierarchical method that learns physically plausible interactions following open-vocabulary instructions. *Our approach begins by developing a set of atomic actions via a low-level controller trained via imitation learning. Upon receiving an open-vocabulary textual instruction, `AnySkill` employs a high-level policy that selects and integrates these atomic actions to maximize the CLIP similarity between the agent's rendered images and the text. An important feature of our method is the use of image-based rewards for the high-level policy, which allows the agent to* learn interactions with objects without manual reward engineering. *We demonstrate `AnySkill`'s capability to generate realistic and natural motion sequences in response to unseen instructions of varying lengths, marking it the first method capable of open-vocabulary physical skill learning for interactive humanoid agents.*

1. Introduction

Confronted with a soccer ball, an individual might engage in various actions such as kicking, dribbling, passing, or shooting. This interaction capability is feasible even for someone who has only observed soccer games, never having played. This ability exemplifies the human aptitude for learning open-vocabulary physical interaction skills from visual experiences and applying these skills to novel objects and actions. Equipping interactive agents with this capability remains a significant challenge.

Recent physical skill learning methods predominantly rely on imitation learning to acquire realistic physical motions and interactions [29, 31]. However, this approach limits their adaptability to unforeseen scenarios with novel instructions and environments. Furthermore, neglecting physical laws in current models leads to unnatural and unrealistic motions, such as floating, penetration, and foot sliding, despite attempts to integrate physics-based penalties like gravity [59, 66] and collision [13, 58, 68]. Enhancing the generalizability of physically constrained motion generation is essential for decreasing reliance on specific datasets and fostering a more profound comprehension of the world.

On top of generalizability, the ultimate goal is to generate natural and interactive motions from any text input, known as achieving open vocabulary, which significantly increases the complexity of the problem. Several studies have explored open-vocabulary motion generation using large-scale pretrained models [11, 19, 37, 43]. However, these models struggle to produce natural motions, particularly interactive motions that require understanding broader environmental contexts or object interactions [11, 19, 43].

We identify a gap in motion generalizability on novel tasks and interaction capabilities with environments, hypothesizing that this is due to the reliance on improvised state representations and manually crafted reward mechanisms in prior works. Inspired by the human ability to learn new physical skills from visual inputs, we propose utilizing a Vision-Language Model (VLM) to offer flexible and generalizable state representations and image-based rewards for open-vocabulary skill learning. We introduce `AnySkill`, a hierarchical framework designed to equip virtual agents with the ability to learn open-vocabulary physical interaction skills. `AnySkill` combines a shared low-level controller with a high-level policy tailored to each instruction, learning a repertoire of latent atomic actions through generative adversarial imitation learning (GAIL), following CALM [42]. This ensures the naturalness and physical plausibility of each action. Then, for any open-vocabulary textual instruction, a high-level control policy dynamically selects latent atomic actions to optimize the CLIP [35] similarity between the agent's rendered images and the textual instruction. This policy maintains physical plausibility and allows the agent to act according to a broad range of textual instructions. By leveraging CLIP similarity as a flexible and straightforward reward mechanism, our approach overcomes environmental limitations, facilitating interaction with any object. Despite the advances, creating natural and interactive actions for open-vocabulary models remains an ongoing challenge.

Extensive experiments demonstrate `AnySkill`'s ability to execute physical and interactive skills learned from open-vocabulary instructions; Fig. 1 showcases various interactive and non-interactive examples. We further prove that our method outperforms existing open-vocabulary motion generation approaches in creating interaction motions.

To summarize, our contributions are three-fold:

- We introduce `AnySkill`, a hierarchical approach that combines a low-level controller with a high-level policy, specifically designed for the learning of open-vocabulary physical skills.
- We leverage the VLM (*i.e.*, CLIP) to provide a novel means of generating flexible and generalizable image-based rewards. This approach eliminates the need for manually engineered rewards, facilitating the learning of both individual and interactive actions.
- Through extensive experimentation, we demonstrate that our method significantly surpasses existing approaches in both qualitative and quantitative measures. Importantly, `AnySkill` empowers agents with the ability to engage in smooth and natural interactions with dynamic objects across a variety of contexts.

2. Related Work

Physical skills learning emphasizes mastering motions that adhere to physical laws, including gravity, friction, and penetration. This domain has seen approaches that either employ specific loss functions to address constraints like foot-ground penetration [62], body-object interaction [1, 5, 8, 15, 21, 34, 47–50, 53, 61, 65, 67], self-collision [18, 27, 45], and gravity [6, 38, 55], or leverage physics simulators [16, 24, 31, 32, 42, 46, 60] for more dynamic fidelity. Despite these efforts, ensuring fine-grained physical plausibility, especially in complex interactions, remains a challenge. The integration of reinforcement learning (RL) [10, 26, 29] and advanced modeling techniques (*e.g.*, MoE [2, 12, 54], VAE [20, 25], and GAN [10, 41]) alongside CLIP features [19, 37] attempts to improve generalization, yet faces the grand challenge of achieving physical plausibility in open vocabulary. Our method combines a shared low-level controller with a high-level policy tailored to each instruction, ensuring actions are physically realistic and adaptable to diverse instructions.

Open-vocabulary motion generation creates human motions from natural language descriptions outside the training distribution. Leveraging large-scale motion-language datasets [7, 23, 33, 51], generative models have shown promise in motion synthesis [14, 36, 44, 64, 66]. However, these models often struggle with zero-shot generalization or adhering to the laws of physics, limited by their training data scope. Attempts to address these limitations include simplifying complex instructions with Large Language Models [17, 19] and employing pretrained VLMs like CLIP for supervision [11, 22, 43], yet achieving natural and physics-compliant motions remains a significant hurdle. Our method builds upon these foundations, seeking to generate interactive and physically plausible motions from open-vocabulary descriptions, distinguishing itself from approaches like VLM-RMs [37] by modeling motion priors more effectively.

Humanoid object interaction, a relatively uncharted territory in physics-based motion generation, has seen simplifications such as attaching objects to characters' hands to bypass the complexity of modeling physical interactions [29, 57, 63]. For dynamic interactions, encoding object states (positions and velocities) into the agent's observations has facilitated specific tasks like dribbling [30, 31] and interacting with furniture [9], albeit requiring precise, object-specific rewards. This state-based approach is less feasible in open environments with diverse objects. Alternatively, vision-based policies [26] have shown potential for broader applications but are limited by their training domains. Our approach leverages a VLM for a more generalized motion-text alignment, avoiding the intricacies of manual reward crafting for varied interactive tasks.

3. AnySkill

AnySkill consists of two core components: the **low-level controller** and the **high-level policy**, illustrated in Fig. 2. Initially, we train a shared low-level controller, π^L, using unlabeled motion clips to distill a latent representation of atomic actions. This process utilizes GAIL [10], guaranteeing that the atomic actions are physically plausible.

Subsequently, for each open-vocabulary textual instruction, we train a high-level policy, π^H, tasked with composing atomic actions derived from low-level controllers. This high-level policy leverages a **flexible and generalizable image-based reward** via a VLM. This design facilitates the learning of physical interactions with dynamic objects, obviating the need for handcrafted reward engineering.

3.1. Low-Level Controller

The low-level controller, inspired by CALM [42], enables the physically simulated humanoid agent to learn a diverse set of atomic actions. Formally, given an unlabeled motion dataset $\mathcal{M}$, we simultaneously train a motion encoder E, a discriminator D, and a controller $\pi^L(a|s,z)$. Here, a denotes the action, s the state, and $z \in \mathcal{Z}$ the latent motion representation. The state s comprises the agent's current root position, orientation, joint positions, and velocities, while the action a specifies the next target joint rotations.

Training proceeds as follows: A motion clip M from $\mathcal{M}$ is encoded by E to yield the latent representation $z = E(M)$. The controller $\pi^L(a|s,z)$ generates an action a based on the current state s and latent z. The agent then executes the

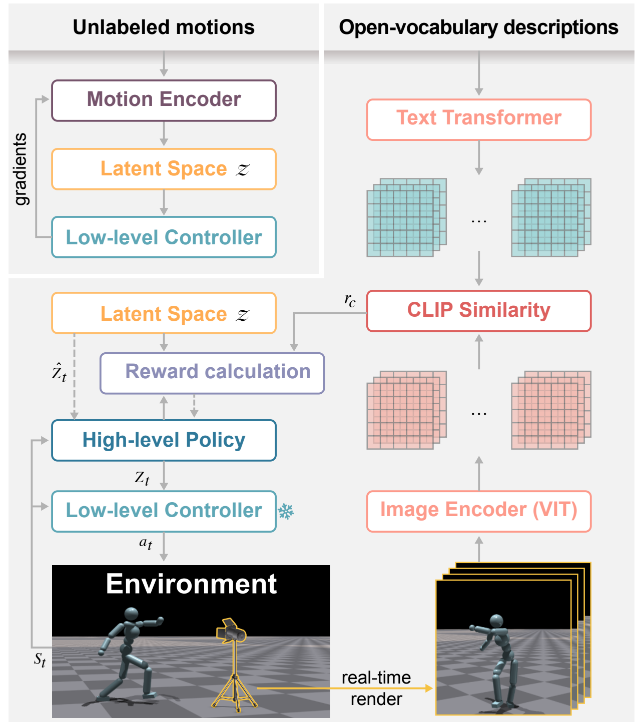


Figure 2. **The hierarchical structure of AnySkill.** Initially, the low-level controller (top-left) is trained to encode unlabeled motions into a shared latent space $\mathcal{Z}$. Subsequently, for each open-vocabulary text description, a high-level policy is trained. This policy orchestrates low-level actions to optimize the CLIP similarity between rendered images and the provided text, effectively composing actions that align with the textual instructions.

action a in the physics-based simulator with a PD controller, resulting in a new state s'.

The discriminator D distinguishes whether the given (s,s') originates from the motion M corresponding to z, is produced by the controller π^L following the latent code z, or is produced by π^L following another latent code $z' \sim \mathcal{Z}$. We train D with a ternary adversarial loss:

$$\begin{aligned}\mathcal{L}_{\mathcal{D}} = -\mathbb{E}_{M\in\mathcal{M}}\Big(&\mathbb{E}_{d^{\pi}(s,s'|z)}\left[\log\left(1-\mathcal{D}(s,s'|z)\right)\right] \\ &+\mathbb{E}_{d^{M}(s,s')}\left[\log\mathcal{D}(s,s'|z)+\log\left(1-\mathcal{D}(s,s'|z'\sim\mathcal{Z})\right)\right] \\ &+w_{\text{gp}}\mathbb{E}_{d^{\mathcal{M}}(s,s')}\left[||\nabla_{\theta}\mathcal{D}(\theta)|_{\theta=(s,s'|\hat{z})}||^2\right]\Big|\hat{z}=\text{sg}(E(M))\Big),\end{aligned} \tag{1}$$

incorporating a gradient penalty with coefficient w_{gp} for stability, where sg$(\cdot)$ denotes the stop gradient operator.

The encoder E is refined with both alignment and uniformity losses to ensure that embeddings of similar motions are closely aligned in the latent space, while dissimilar ones remain distinct [52], thus structuring $\mathcal{Z}$ effectively.

The controller π^L aims to maximize the GAIL reward from D, calculated as

$$r^L(s,s',z) = -\log\left(1-\mathcal{D}\left(s,s'|z\right)\right), \tag{2}$$

encouraging the generation of motions that closely resemble the original motion M associated with latent code z.

3.2. High-Level Policy

Building upon the atomic action repository created by the low-level controller, the high-level policy's objective is to compose these actions, via the control of latent representation z, to generate motions that align with given text descriptions. With the low-level controller π^L fixed, we train a high-level policy π^H for each specific textual instruction, ensuring that the combined operation of both policy levels results in motions congruent with the text. The training process for the high-level policy is outlined in Algorithm 1.

Algorithm 1: Training of the high-level policy

Input: Reference motion dataset $\mathcal{M}$, frozen low-level controller π^L, frozen motion encoder E, simulation environment $\textsc{Env}$, renderer image $\mathcal{I}$, CLIP feature of the description text f_d

$\mathcal{Z} = E(\mathcal{M})$ initialize motion latent space
while *not converged* **do**
  $\mathcal{B} \leftarrow \emptyset; p \leftarrow 0$ initialize
  for *horzion_length* $= 1, ..., n$ **do**
    sample $\hat{z}$ from $\mathcal{Z}$
    if *horzion_length* $= 1$ **then**
      $s \leftarrow$ initialize; $z \leftarrow \hat{z}$
    else
      $s \leftarrow \textsc{Env}(s, a); z \leftarrow \pi^H(s)$
    end
    for *llc_steps* $= 1, ..., t$ **do**
      $s \leftarrow \textsc{Env}(s, \pi^L(s, z))$ step simulation
      $r^H \leftarrow$ calculate reward with Eq. (3)
      if HEAD_HEIGHT < 0.15 **then**
        $s, p \leftarrow 0$ reset agent and counter
      end
      if similarity is less than last step **then**
        $p \leftarrow p + 1$ increment counter
        if $p \geqslant 8$ **then**
          $p \leftarrow 0$ reset counter
          reset s with 80% probability
        end
      end
    end
    update $\mathcal{B}$ and π^H according to PPO
  end
end

The high-level policy π^H is implemented as an MLP, taking the agent's state s as input and outputting a latent representation z close to the low-level controller's latent space $\mathcal{Z}$. It is trained using a composite reward of image-based similarity and latent-representation alignment. Given state s and text description d, we render the agent's image $\mathcal{I}(s)$ and encode it along with the text using a pretrained, frozen CLIP model to obtain features $f_{\mathcal{I}}$ and f_d. The similarity reward is computed as the cosine similarity between $f_{\mathcal{I}}$ and f_d, with an additional latent-representation alignment reward to draw z nearer to the latent distribution of $\mathcal{M}$. The combined reward is given by:

$$r^H = \omega_c \cdot \frac{f_{\mathcal{I}} \cdot f_d}{|f_{\mathcal{I}}||f_d|} + \omega_s \cdot \exp(-4\|z - \hat{z}\|_2), \tag{3}$$

where ω_c, ω_s are weighting factors, and $\hat{z}$ is a sample from $\mathcal{Z}$. This image-based reward mechanism enables `AnySkill` to achieve text-to-motion alignment for open-vocabulary instructions. In addition, the image-based representation naturally encodes the entire environment around the agent, thus facilitating object interactions without modifying the encoding or architecture.

3.3. Implementation Details

Low-level controller The architecture of the encoder, low-level control policy, and discriminator comprises MLPs with hidden layers sized [1024, 1024, 512]. The latent space $\mathcal{Z}$ is 64-dimensional. The alignment loss is set to 0.1, uniformity loss to 0.05, and gradient penalty to 5. The low-level controller is optimized using PPO [39] in IsaacGym. The training process is conducted on a single A100 GPU, operating at a 120Hz simulation frequency, and spans four days to cover a dataset comprising 93 unique motion patterns. Detailed hyperparameter settings of the low-level controller can be found in Tab. A1.

High-level policy The high-level policy, implemented as a two-layer MLP with hidden units of [1024, 512], outputs a 64-dimensional vector and is optimized using PPO. Training is conducted on an NVIDIA RTX3090 GPU, taking approximately 2.2 hours. Operationally, the high-level policy executes at a frequency of 6Hz, in contrast to the low-level policy, which operates at a more rapid 30Hz. This discrepancy in execution rates is strategic; the high-level policy is invoked every five timesteps, granting the low-level controller sufficient time to act on a given stable latent representation z and execute a complete atomic action. Such a setup is crucial for preventing the emergence of unnatural motion sequences by ensuring that each selected atomic action is fully realized before transitioning. Detailed hyperparameters of the high-level policy can be found in Tab. A2.

To further refine the training process and motion quality, an early termination strategy is employed to circumvent potential pitfalls of the high-level policy becoming trapped in suboptimal local minima. Specifically, the environment is reset with an 80% probability following eight successive reductions in CLIP similarity, or deterministically if the agent's head height falls below 15cm. This approach significantly enhances training efficiency and the fidelity of the generated motions, ensuring a balance between exploration and the avoidance of poor performance traps.

Rendering We use IsaacGym's default renderer, positioning the camera at (3m, 0m, 1m) while the agent is initialized at the origin. To maintain the agent at the focus of our visual feedback, we dynamically adjust the camera's orientation each timestep to align with the agent's pelvis joint. To encode the rendered images into a feature space compatible with our learning objectives, we employ the `CLIP-ViT-B/32` model checkpoint from OpenCLIP [3], leveraging its robust representational capabilities.

State projection Given the computational demands of rendering images and extracting their CLIP features, we streamline the training process by introducing an MLP that projects the agent's state vectors s directly to CLIP image features. This projection MLP is fine-tuned with an MSE loss against 104 million agent states accumulated during the high-level policy training. By substituting the render-and-encode steps with this MLP, we achieve a significant speedup, enhancing training efficiency by approximately 10.4 times, thereby mitigating the bottleneck associated with real-time image rendering and feature extraction.

4. Experiments

In this section, we detail the motion dataset curation for `AnySkill`'s low-level controller training (Sec. 4.1), evaluate `AnySkill`'s open-vocabulary motion generation against others (Sec. 4.2), analyze the text enhancement impact on effectiveness (Sec. 4.3), showcase physical interaction examples (Sec. 4.4), and compare our reward design with existing formulations (Sec. 4.5).

4.1. Training of Low-Level Controller

Dataset To enrich the low-level controller with diverse atomic actions, we assembled a dataset of 93 distinct motion records, primarily sourced from the CMU Graphics Lab Motion Capture Database [4] and SFU Motion Capture Database [40]. This collection spans various action categories, including locomotion (*e.g.*, walking, running, jumping), dance (*e.g.*, jazz, ballet), acrobatics (*e.g.*, roundhouse kicks), and interactive gestures (*e.g.*, pushing, greeting), all retargeted to a humanoid skeleton with 15 bones. We also adjusted any motions that lacked physical plausibility, ensuring the dataset's fidelity for effective imitation learning.

Training stabilization Adversarial imitation learning's instability, influenced by the volume and distribution of training data, can skew the density distribution in latent space, limiting the diversity of atomic actions for high-level policy selection. To mitigate this, we categorized motion records into 3 primary and 4 secondary groups by action scale and involved limbs. Details of the category division are described in Appendix A.2. By adjusting training data weights, we increased the likelihood of less frequent action groups, ensuring the variety of learned atomic actions; see also Figs. 3 and A7.

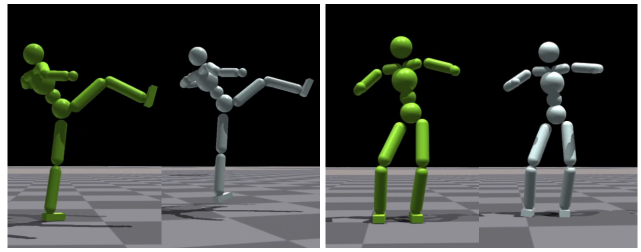


Figure 3. **Atomic actions from the trained low-level controller.** Each subfigure depicts the green agent demonstrating the reference motion from the dataset, while the white agent illustrates the corresponding learned atomic action.

4.2. `AnySkill` Evaluation

Given the nascent field of open-vocabulary physical skill learning, we benchmark `AnySkill` against the two foremost similar methods in open-vocabulary motion generation: MotionCLIP [43] and AvatarCLIP [11], which also utilize CLIP similarity for generating human motions. To further understand the efficacy of our approach, we introduce a variant of our method, "Ours (no ET)," which operates without

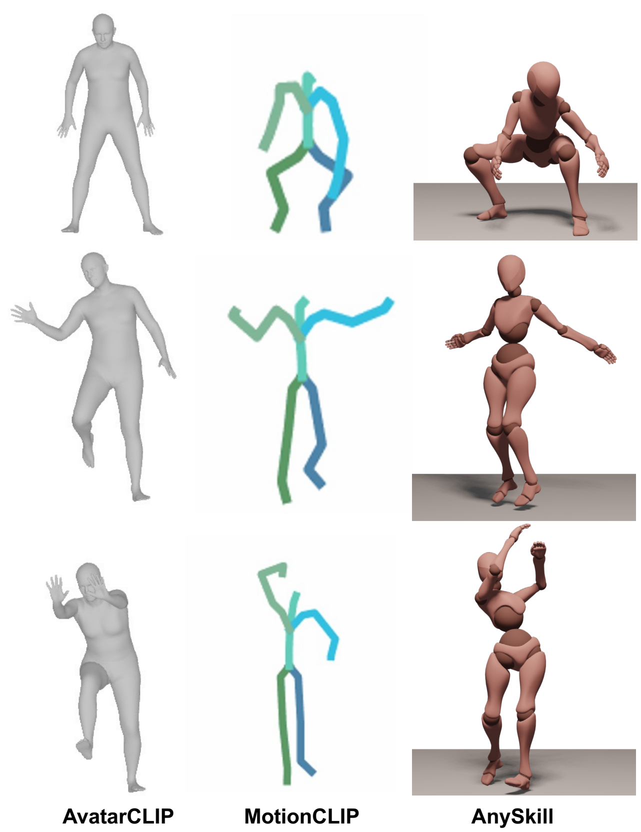
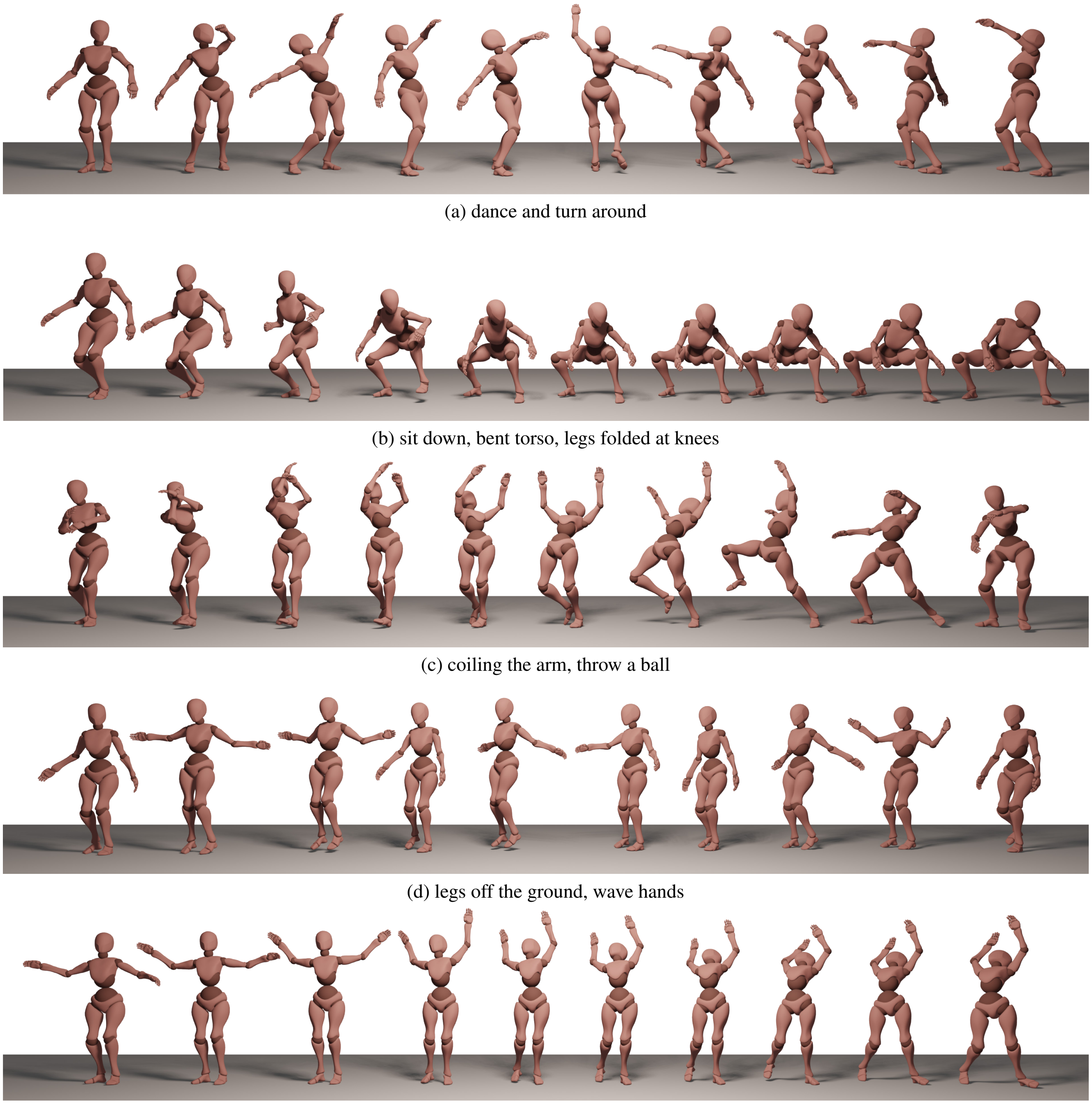


Figure 4. **Qualitative comparisons on open-vocabulary motion generation.** From top to bottom, the descriptions are *"sit down, bent torso, legs folded at knees"*, *"legs off the ground, wave hands"*, and *"coiling the arm, throw a ball"*. We showcase the most representative frames that best align with the descriptions.



(a) dance and turn around

(b) sit down, bent torso, legs folded at knees

(c) coiling the arm, throw a ball

(d) legs off the ground, wave hands

(e) raise two arms

Figure 5. **Qualitative results of generated motion by `AnySkill`.** Displayed are specific text descriptions and the corresponding motions generated by `AnySkill`, as evaluated in the user study. Motion sequences progress from left to right.

the early termination strategy.

For this evaluation, we selected 5 open-vocabulary text descriptions requiring comprehensive body movement and not covered in `AnySkill`'s training data. To assess the generated motions, we engaged 24 MTurk workers to rate them on task completion, smoothness, naturalness, and physical plausibility, using a scale from 0 to 10. Moreover, we computed the CLIP similarity score between the rendered images and the text descriptions for each method as an objective measure. The motions generated by each method, including qualitative comparisons, are showcased in Fig. 4, with an in-depth look at `AnySkill`'s outputs presented in Fig. 5. Beyond the five actions presented, additional actions are shown in Appendix B.4

Table 1. **Quantitative evaluation of high-level policy.**

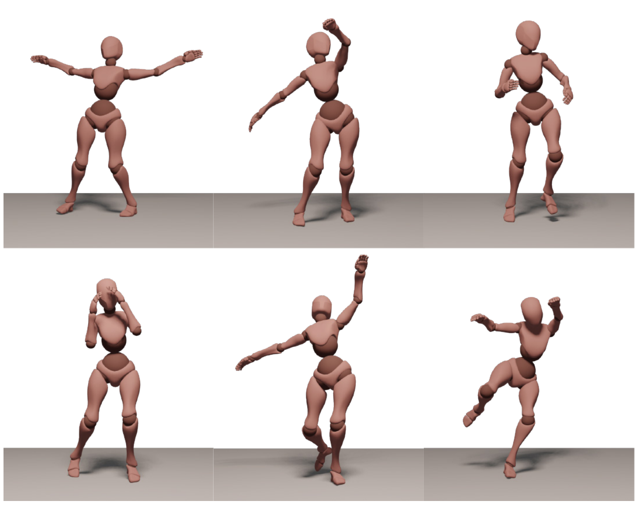
	Success ↑	Natural ↑	Smooth↑	Physics↑	CLIP_S↑
AvatarCLIP [11]	4.29	4.74	5.79	5.74	21.11
MotionCLIP [43]	3.16	4.93	5.72	5.83	21.16
Ours (w/o ET)	5.05	4.88	5.68	5.31	21.89
Ours (w/o text-enhance)	3.06	4.48	5.19	5.96	20.76
Ours (w/ VideoCLIP [56])	2.37	4.90	5.65	6.41	21.35
Ours (full)	**6.16**	**6.23**	**6.51**	**6.93**	**24.18**

We present the results of the human study and quantitative metrics in Tab. 1, demonstrating that AnySkill significantly surpasses current methods across all evaluated metrics. The ablation study underscores the importance of incorporating early termination into the training process. For additional comparative and qualitative results, see Fig. A5.

4.3. Text Enhancement

AnySkill excels at open-vocabulary skill acquisition, outperforming existing models. Its performance, however, is contingent on the specificity and scope of text descriptions. Performance drops with vague descriptions or for tasks requiring prolonged execution due to reliance on image-based similarity for rewards. For example, "do yoga" encompasses a broad range of poses, complicating convergence on a specific action. Similarly, for extended actions like "walk in a circle," the model may not fully complete the task, as image-based rewards provide insufficient directional guidance.

To counteract these limitations, we introduced an automated script utilizing GPT-4 [28] to refine and clarify textual instructions, enhancing specificity and reducing potential



(a) (b) (c)

Figure 6. **Qualitative evaluation of text description enhancement.** We compare motions generated with original HumanML3D [7] descriptions (top row) against those from our enhanced descriptions (bottom row). Text descriptions are (a) *"wave hi" and "raised arm bent at the elbow"*; (b) *"Waltz dance" and "left foot step backward, right hand extends"*; (c) *"kick" and "left leg forward, right leg retreats"*.

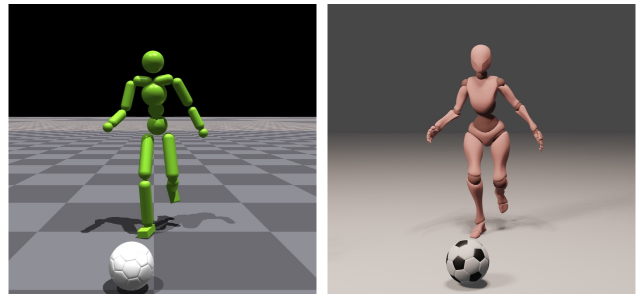


Figure 7. **Agent and rendered mesh.** The simulation of our agent and the interacting object (left) alongside their visualization (right).

motion interpretation ambiguity. This refinement process significantly improves AnySkill's execution accuracy. Fig. 6 compares the original and refined texts alongside their generated motions; see Appendix B.1 for more qualitative results.

Moreover, we refined text descriptions from the HumanML3D [7] and BABEL [33] databases, amassing 1,896 unique, enhanced text instructions. For comprehensive details on the refined texts and their impact on motion generation, refer to Appendix A.1.

4.4. Interaction Motions

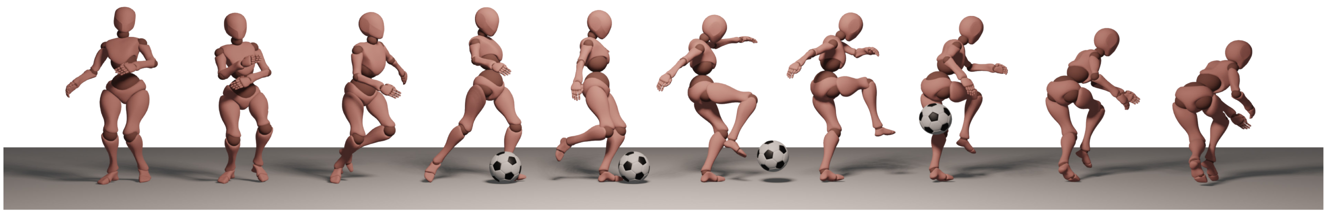
AnySkill demonstrates the superb capability to interact with dynamic objects, for instance, a *soccer ball* and a *door*. To capture these interactions accurately during training, we manually adjust the camera positions, focusing on the door and soccer ball. The alignment between the simulation environment and the rendered visualizations is showcased in Fig. 7. The qualitative assessments, as seen in Fig. 8, along with the quantitative evaluations in Tab. 2, confirm that AnySkill efficiently learns to interact with a variety of objects without necessitating any modifications to its learning algorithm or reward design. Our tests primarily involve interactions with a single object, yet extending AnySkill to engage with multiple objects concurrently is anticipated to be straightforward. Further interactive motions with various objects are available in Appendix B.4 and Fig. A8.

Table 2. **Quantitative evaluation of interaction motions.**

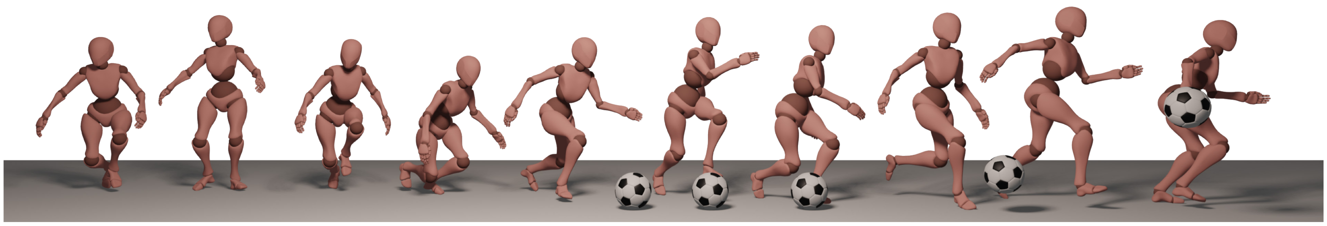
	Success ↑	Natural ↑	Smooth↑	Physics↑	CLIP_S↑
Interaction w. object	5.42 -0.74	5.62 -0.61	5.34 -1.17	5.45 -1.48	24.49 +0.35
Interaction w. scene	4.53 -1.63	4.47 -1.76	5.01 -1.50	5.41 -1.52	22.41 -1.73

4.5. Reward Function Analysis

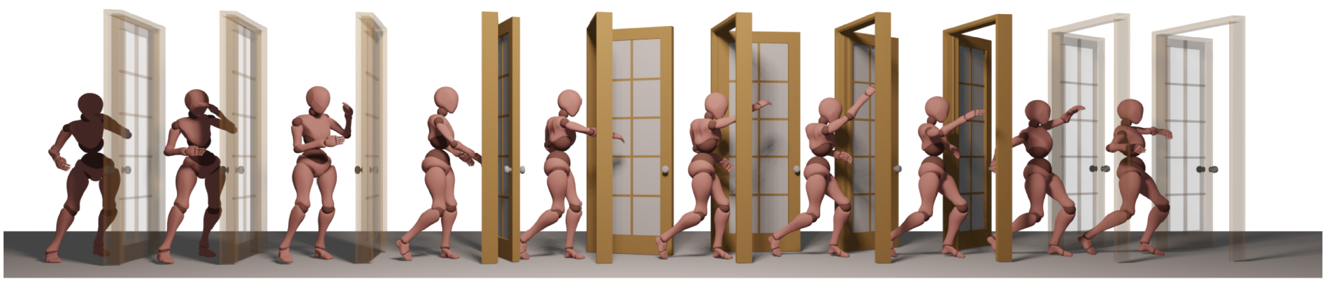
We evaluate 4 recent reward functions image- and physics-based RL and compare them with ours using cosine similarity. These include VLM-RMs [37], which adjusts the CLIP feature of text to exclude agent-specific details; CLIP-S [69], applying a modified CLIP similarity as the reward; VideoCLIP [56], calculating mean-pooled CLIP features across frames for temporal coherence; and ASE [32], adding a velocity reward for desired agent movement.



(a) kick the white ball



(b) move the white ball



(c) raise arm, open the door

Figure 8. **Interaction motions generated by `AnySkill`.** Displayed are interaction sequences by `AnySkill`: two with a soccer ball (a-b) and one with a door (c), progressing from left to right.

Using these rewards, we train `AnySkill` on identical descriptions and assess motion quality via a user study similar to the one described in Sec. 4.2, with results presented in Tab. 3 and Appendix B.3. Our approach surpasses the baseline methods in most metrics, demonstrating the effectiveness of our reward function. Notably, AvgPool scores highly in smoothness, benefiting from averaging alignment scores over time.

Table 3. **Comparisons of the reward design.**

	Success↑	Natural↑	Smooth↑	Physics↑	CLIP_S↑
VLM-RMs [37]	3.15	4.36	5.35	5.17	19.46
CLIP-S [69]	3.80	5.41	5.98	6.21	19.78
AvgPool [56]	5.09	5.96	**6.55**	6.70	20.25
+ vel. rew. [32]	2.73	4.42	5.35	5.22	18.39
Ours	**6.16**	**6.23**	6.51	**6.93**	**24.18**

5. Conclusion

This work introduced `AnySkill`, a novel hierarchical approach for mastering open-vocabulary physical interaction skills, leveraging a low-level controller grounded in imitation learning for motion fidelity and a novel, effective image-based reward for flexible skill acquisition. Through comprehensive evaluations, `AnySkill` has proven its unique ability to generalize across unseen tasks and interact with novel objects, marking a significant advancement in motion generation for interactive agents.

Future directions `AnySkill`'s next steps involve addressing its reliance on the CLIP model and image-based rewards, which currently limit handling complex scenarios with prolonged actions or visual ambiguity. Future directions include better temporal understanding, advanced multimodal alignment, and interactive feedback integration. Additionally, we aim to streamline the training process, moving towards a generalized framework that reduces the computational demand for learning new skills, broadening `AnySkill`'s applicability in creating interactive virtual agents.

Acknowledgement The authors would like to thank Ms. Zhen Chen (BIGAI) for her exceptional contribution to the figure designs, Yanran Zhang and Jiale Yu (Tsinghua University) for their invaluable assistance in the experiments and prompt design, and Huiying Li (BIGAI) for crafting the agent's appearance. We also thank NVIDIA for generously providing the necessary GPUs and hardware support. This work is supported in part by the National Science and Technology Major Project (2022ZD0114900), an NSFC fund (62376009), and the Beijing Nova Program.