Universal Humanoid Robot Pose Learning from Internet Human Videos

Jiageng Mao[1*] Siheng Zhao[1*] Siqi Song[1*†] Tianheng Shi[1] Junjie Ye[1] Mingtong Zhang[1]
Haoran Geng[2] Jitendra Malik[2] Vitor Guizilini[3] Yue Wang[1]

[1]University of Southern California [2]UC Berkeley [3]Toyota Research Institute
https://usc-gvl.github.io/UH-1

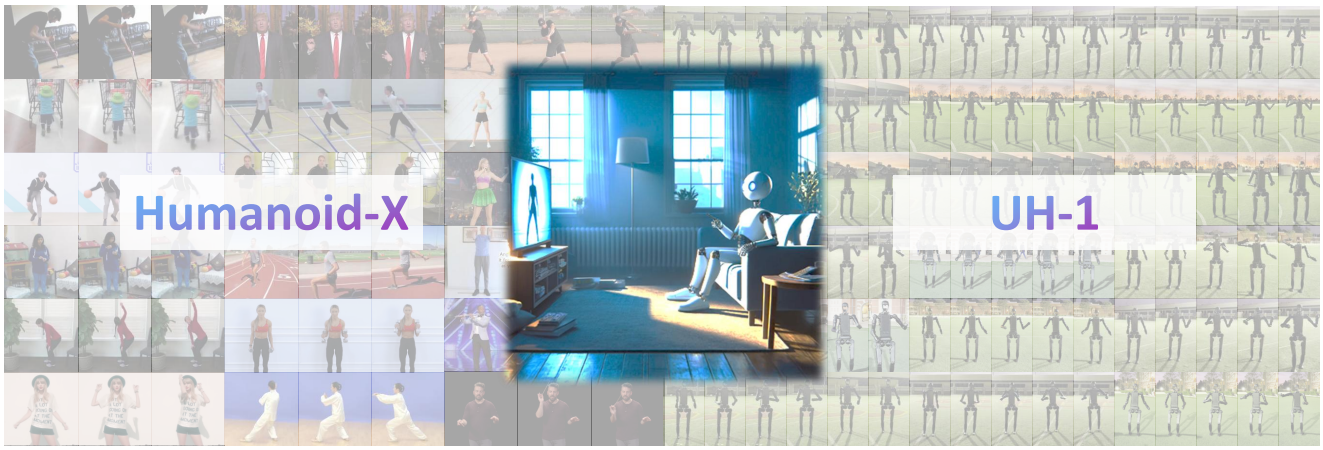


Fig. 1: **Overview.** We introduce *Humanoid-X*, a large-scale dataset to facilitate humanoid robot learning from massive human videos. On top of *Humanoid-X*, we introduce *UH-1*, a large humanoid model for universal language-conditioned pose control of humanoid robots.

***Abstract*— Scalable learning of humanoid robots is crucial for their deployment in real-world applications. While traditional approaches primarily rely on reinforcement learning or teleoperation to achieve whole-body control, they are often limited by the diversity of simulated environments and the high costs of demonstration collection. In contrast, human videos are ubiquitous and present an untapped source of semantic and motion information that could significantly enhance the generalization capabilities of humanoid robots. This paper introduces Humanoid-X, a large-scale dataset of over 20 million humanoid robot poses with corresponding text-based motion descriptions, designed to leverage this abundant data. Humanoid-X is curated through a comprehensive pipeline: data mining from the Internet, video caption generation, motion retargeting of humans to humanoid robots, and policy learning for real-world deployment. With Humanoid-X, we further train a large humanoid model, UH-1, which takes text instructions as input and outputs corresponding actions to control a humanoid robot. Extensive simulated and real-world experiments validate that our scalable training approach leads to superior generalization in text-based humanoid control, marking a significant step toward adaptable, real-world-ready humanoid robots.**

I. Introduction

Scalability is crucial in deep learning. Recent advances in computer vision have demonstrated that scaling up training data leads to more powerful foundation models for visual recognition [1], [2], [3] and generation [4], [5]. In robotics, researchers follow a similar paradigm and build foundation models for robotic manipulation [6], [7], [8], [9] by collecting massive robotic demonstrations. Nevertheless, in contrast to images and videos that are abundant and easily accessible, collecting large-scale robotic demonstrations is expensive and time-consuming, which limits the scalability of current robot learning methods. This raises the question: *Can we use videos as demonstrations to improve the scalability of robot learning?*

To address this challenge, many efforts have been made, such as learning affordances [10], [11], [12], flows [13], [14], and world models [15] from natural videos, which enable more generalizable robotic manipulation. However, when it comes to humanoid robots, learning such action representations from videos remains an open problem. Unlike robotic arms, humanoid robots have distinct kinematic structures and more degrees of freedom (DoFs), making them harder to control. Existing works [16], [17], [18], [19], [20], [21], [22] leverage large-scale reinforcement learning to learn robust humanoid control policies, but they only focus on limited robotic skills such as locomotion or jumping, making them less generalizable for handling everyday tasks. Other works [23], [24], [25], [26] control humanoid robots through teleoperation, but they require human labor to collect robotic data, which is less scalable. In contrast to these previous works, learning a universal action representation

*equal contribution (in alphabetical order). †work done while at USC.

from massive videos will greatly improve the scalability of humanoid robot learning and enable more generalizable humanoid pose control.

To bridge this gap in humanoid robot learning, we introduce Humanoid-X, a large-scale dataset curated from a massive and diverse collection of videos for universal humanoid pose control. Humanoid-X utilizes natural language as an interface to connect human commands and humanoid actions, so humans can talk to their humanoid robots to control their actions. The natural language representations are extracted from videos via captioning tools and are used to describe the actions of humanoid robots. For action representations, Humanoid-X leverages both robotic keypoints for high-level control and robotic target DoF positions for direct position control. To extract humanoid actions from human videos, we first reconstruct 3D humans and their motions from videos. Then, we leverage motion retargeting to transfer motions from 3D humans to humanoid robots, resulting in robotic keypoints for high-level humanoid pose control. Finally, we learn a universal RL-based control policy that maps keypoints to low-level humanoid target DoF positions that can be deployed in real robots. We collect over 160,000 human-centric videos from academic datasets and the Internet, covering diverse action categories. We further transform these videos into text-action pairs, resulting in over 20 million humanoid actions with corresponding text descriptions. Humanoid-X paves the way for developing more generalizable and scalable humanoid robotic control guided by natural language.

On top of the Humanoid-X dataset, we further investigate how to learn a universal humanoid pose control model using large-scale text-action pairs. We introduce **U**niversal **H**umanoid-1 (UH-1), a large humanoid model for universal language-conditioned humanoid pose control. UH-1 leverages the scalability of the Transformer architecture to handle vast amounts of data efficiently. We begin by discretizing 20 million humanoid actions into action tokens, creating a vocabulary of motion primitives. Then, given a text command as input, the Transformer model auto-regressively decodes a sequence of these tokenized humanoid robotic actions. For cases where the action representation involves robotic keypoints, we transform these into robotic DoF positions using an additional action decoder. Finally, we utilize a proportional-derivative (PD) controller to convert the DoF positions into motor torques, enabling us to control humanoid robots and deploy them in the real-world.

To validate the effectiveness of the Humanoid-X dataset and the UH-1 model, we conducted extensive experiments across both simulated and real humanoid platforms. Our results reveal that leveraging vast amounts of video data enables our model to seamlessly translate textual commands into diverse and contextually accurate humanoid actions. Notably, the UH-1 model demonstrates strong robustness, proving reliable in real-world deployment. To summarize, our key contributions are as follows:

· We introduce Humanoid-X, a pioneering large-scale dataset tailored for learning universal humanoid control from massive Internet video data.

· We introduce UH-1, a powerful, scalable model for language-conditioned control of humanoid poses. Our approach supports two flexible control modes that are interchangeable, depending on task requirements. We also provide extensive ablation study for our design choices.

· Our experiments confirm that training on massive video data enables a level of generalizability in humanoid control that was previously unattainable.

II. Related Works

Robot Learning from Internet Data. Many endeavors have been made to learn scalable robot learning policies from non-robotic data, especially Internet videos. The key idea is to learn valuable representations from massive visual data and transfer them to robotic tasks. The learned representations include pre-trained visual features from videos [27], [28], [29], [30] and transferable action representations such as affordances [10], [31] and object-centric flows [13], [14]. Other works [32], [15], [33] attempt to learn world models from Internet videos. However, most of these works focus on robotic manipulation. Since robot arms have totally different kinematic structures from humanoid robots, the learned visual and action representations for robotic manipulation are not transferable to humanoid robot control. In contrast, we investigate how to learn universal pose control for humanoid robots from massive videos.

Humanoid Robot Learning. Extensive work has been dedicated to learning policies that enable robust control of humanoid robots. Some works focus on humanoid locomotion using large-scale reinforcement learning [18], [20], [16], [17], [19] or imitation learning [34], [35]. Other works learn humanoid manipulation via imitation learning [36], [37]. Notably, some works [25], [23], [21], [24], [38] learn humanoid teleoperation by transferring motions from 3D humans to humanoid robots. However, these works mainly focus on accurate motion tracking and control from clean human motions. In contrast, our method focuses on the *generalization* ability of humanoid pose generation, and we explored learning from massive noisy Internet videos for text-conditioned, generalizable humanoid pose control.

3D Human Motion Generation. Many works are attempting to generate diverse 3D human motions via Transformers [39], [40] or diffusion models [41], [42], [43], [44], [45]. Also, some works [46], [47], [48], [49], [50], [51], [52], [53] are trying to generate realistic motions to animate physics-based virtual characters. However, humanoid robots are essentially different from digital humans in many aspects: (1) they have different joint structures and degrees of freedom; (2) humanoid robots cannot access privileged information like linear velocities, which is readily available when controlling virtual humans; (3) humanoid robots have physical constraints such as motor torque limits, whereas 3D virtual humans do not have these limitations. An alternative solution for generalizable humanoid pose control is to first generate 3D human motions and then retarget them to humanoid robots [23], [54]. Compared to these approaches, our

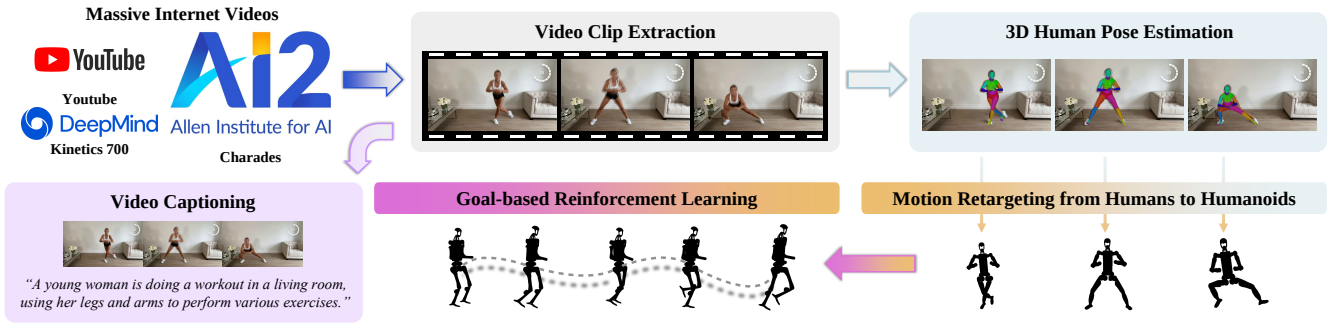


Fig. 2: **Learning Humanoid Pose Control from Massive Videos**. We mine massive human-centric video clips $\mathcal{V}$ from the Internet. We then extract text-based action descriptions $\mathcal{T}$ and 3D human poses $\mathcal{P}_{human}$ from the video clips. Next, we retarget the motions from humans to humanoid robots, resulting in humanoid keypoints $\mathcal{P}_{robot}$ for high-level control. Finally, we employ reinforcement learning to generate physically deployable humanoid actions $\mathcal{A}_{robot}$. In this manner, we collect 163,800 pairs of motion samples $\langle \mathcal{V}, \mathcal{T}, \mathcal{P}_{human}, \mathcal{P}_{robot}, \mathcal{A}_{robot} \rangle$ from Internet videos, which are leveraged to distill a universal humanoid pose control policy.

UH-1 model offers a more streamlined solution by directly mapping text commands into executable humanoid actions without intermediate steps. Furthermore, unlike human motion generation models trained on expensive motion capture data, learning from massive videos significantly enhances the generalization ability of our method.

III. Humanoid-X Dataset

A. Overview

To scale up humanoid robot learning using massive human videos, we introduce Humanoid-X, the largest humanoid robot dataset to date compiled from a vast and diverse collection of videos for universal humanoid pose control. Humanoid-X consists of 163,800 motion samples covering a comprehensive set of action categories. Each motion sample in the dataset contains 5 data modalities: an original video clip $\mathcal{V}$, a text description $\mathcal{T}$ of the action in the video, a sequence of SMPL [55]-based human poses $\mathcal{P}_{human}$ estimated from the video, a sequence of humanoid keypoints $\mathcal{P}_{robot}$ for high-level robotic control, and a sequence of humanoid actions $\mathcal{A}_{robot}$ representing target DoF positions for low-level robotic position control. Humanoid-X encompasses over 20 million frames, totaling approximately 240 hours of data. Beyond its extensive scale across multiple data modalities, which is essential for scalable humanoid policy training, Humanoid-X also features a large and diverse text-based action vocabulary, as shown in fig. 3 (c). This diversity supports universal and text-conditioned humanoid pose control. In the next section, we will discuss how to obtain these motion samples $\langle \mathcal{V}, \mathcal{T}, \mathcal{P}_{human}, \mathcal{P}_{robot}, \mathcal{A}_{robot} \rangle$ from massive videos.

B. Learning from Massive Videos

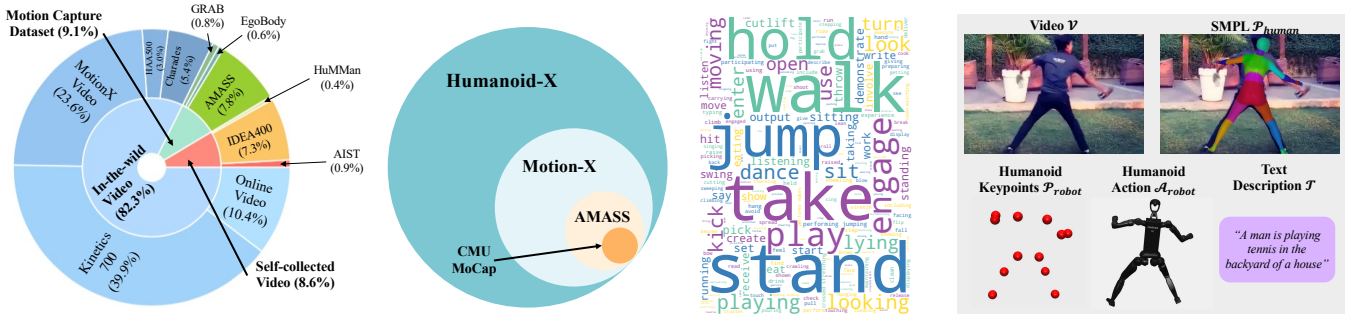
To process large-scale, in-the-wild raw video data, we developed a fully automated data annotation pipeline comprising five modules, as illustrated in fig. 2. The pipeline includes (1) a video processing module that mines and extracts video clips $\mathcal{V}$ from noisy Internet videos, (2) a video captioning model that generates text description of human actions $\mathcal{T}$, (3) a human pose detection module that estimates parametric 3D human poses $\mathcal{P}_{human}$ from video clips, (4) a motion retargeting module to generate humanoid robotic keypoints $\mathcal{P}_{robot}$ by transferring motions from humans to humanoid robots, and (5) a goal-conditioned reinforcement learning policy to learn physically-deployable humanoid actions $\mathcal{A}_{robot}$ by imitating humanoid keypoints.

Video Mining and Processing. The first step of our approach is to collect a large number of human-centric videos that encompass a wide variety of action types. To this end, we mine massive informative video clips from 3 sources: academic datasets for digital human research [56], [57], [58], [59], [60], [61], [62], datasets for video action understanding [63], [64], and Internet videos from YouTube. To collect Internet videos, we designed over 400 unique search terms covering a range of human activities from daily tasks to professional sports, and then utilized the Google Cloud API* to retrieve the top 20 videos for each specified search term.

Original videos are often noisy, including segments with no humans, multiple humans, or a stationary individual, which makes them unsuitable for humanoid control. To obtain meaningful video clips, we begin by downsampling each video to a standardized 20 frames per second (FPS) to ensure consistency across the dataset. Next, we employ an object detector [65] for single-human detection, selecting frames with precisely one visible person. Following detection, we apply motion detection by calculating the pixel-wise grayscale difference between consecutive frames to keep frames showing significant movement. We then compile sequences of at least 64 consecutive frames that satisfy the above single-human motion criterion into video clips, resulting in 163,800 video clips $\mathcal{V}$ in total.

Video captioning. Language bridges human commands and humanoid actions. To associate humanoid actions with semantic meaning and enable language-conditioned humanoid control, we employ a video captioning model [66] to generate

*YouTube Data API v3



(a) Data Distribution (b) Data Scale (c) Vocabulary Diversity (d) Motion Sample

Fig. 3: **Dataset Statistics**. Humanoid-X features extensive scale, diverse sources, a rich action vocabulary, and multiple data modalities.

fine-grained action descriptions $\mathcal{T}$ from videos:

$$\mathcal{T} = F_{caption}(\mathcal{V}), \tag{1}$$

where $F_{caption}$ is the video captioning model. To avoid irrelevant text descriptions, we carefully design prompts to guide the model to describe human actions instead of physical appearance, resulting in action-centric text descriptions.

3D Human Pose Estimation. Humanoid robots inherently share a similar skeleton with humans, which allows for learning control policies for humanoid robots based on human motion data. To this end, we first need to extract human poses from videos. To accurately track and estimate human poses in video clips, we adopt a video-based 3D human parametric model estimator [67], which estimates SMPL [55]-based humans and camera parameters for each frame. We further extract global human motions, *i.e.*, root translations, using the estimated camera parameters. The process can be formulated as:

$$\mathcal{P}_{human}(\beta, \theta, t_{root}) = F_{pose}(\mathcal{V}), \tag{2}$$

where F_{pose} is the human pose estimation model. Finally, we obtain per-frame 3D human pose: $\mathcal{P}_{human}(\beta, \theta, t_{root})$, where β controls the human shapes, θ controls the joint rotations, and t_{root} controls the global root translations.

Motion Retargeting from Humans to Humanoid Robots. Since humans and humanoid robots have similar skeletons, we can track the human joint positions across frames and map them to the corresponding joints in a humanoid robot, resulting in humanoid keypoints $\mathcal{P}_{robot}$ for high-level control. In particular, we chose 12 joints that exist in both humans and humanoid robots: left and right hips, knees, ankles, shoulders, elbows, and wrists. The joint positions $\mathcal{P}_{joints}$ can be obtained via forward kinematics F_{fk}:

$$\mathcal{P}_{joints} = F_{fk}(\mathcal{P}_{human}(\beta, \theta, t_{root})). \tag{3}$$

Since humans have different shapes from humanoid robots, following [25], we first optimize the human shape parameters β to ensure that resized human shapes closely resemble those of a humanoid robot. Specifically, we first obtain joint positions in the humanoid robot under a standard T-shaped pose: $\mathcal{P}^T_{robot}$. Then, under the same T-shaped pose, we optimize β to make human joint positions $\mathcal{P}^T_{joints}$ the same as the corresponding humanoid joint positions $\mathcal{P}^T_{robot}$:

$$\min_{\beta} ||\mathcal{P}^T_{joints} - \mathcal{P}^T_{robot}||_2, \tag{4}$$

$$\text{s.t.} \quad \mathcal{P}^T_{joints} = F_{fk}(\mathcal{P}_{human}(\beta, \theta^T, t_{root})), \tag{5}$$

where θ^T denotes the standard T pose. For each frame of human pose, we replace the original β with the optimal β' in $\mathcal{P}_{human}$, and following Eq. 3 we can obtain the adjusted joint positions $\mathcal{P}'_{joints}$. Finally, we directly set humanoid robotic keypoints as the adjusted human joint positions:

$$\mathcal{P}_{robot} := \mathcal{P}'_{joints}. \tag{6}$$

To effectively control humanoid robots, we also extract the motor DoF positions q_{robot} in the humanoid robot via inverse kinematics F_{ik}:

$$q_{robot} = F_{ik}(\mathcal{P}_{robot}). \tag{7}$$

We use the Adam optimizer [68] to solve the inverse kinematics problem. A smoothing term is added to the optimization to regularize changes in q_{robot} across frames.

Goal-conditioned Humanoid Control Policy. The retargeted humanoid keypoints $\mathcal{P}_{robot}$ and DoF positions q_{robot} accurately reflect humanoid motions, but they cannot be directly deployed to the real robot. This is because they lack the necessary safety guarantees and robustness needed to handle real-world variability and constraints effectively. To address this, we develop a goal-conditioned control policy π that adapts these motions while ensuring safe and reliable deployment on the physical robot:

$$\pi : \mathcal{G} \times \mathcal{O} \mapsto \mathcal{A}_{robot}. \tag{8}$$

The inputs to the policy π include two parts: the goal space $\mathcal{G}$ and the observation space $\mathcal{O}$. The goal space $\mathcal{G}$ contains humanoid keypoints $\mathcal{P}_{robot}$, DoF positions q_{robot}, and root movement goals derived from t_{root}. The observation space $\mathcal{O}$ contains robot proprioception information such as root orientation, angular velocity, and current motor DoF positions. The output action space $\mathcal{A}_{robot}$ are target DoF positions of each joint for controling the humanoid robot, which can be further transformed into motor torque signals through a proportional-derivative (PD) controller.

We train the control policy, π, using large-scale reinforcement learning with PPO [69] for policy optimization. The

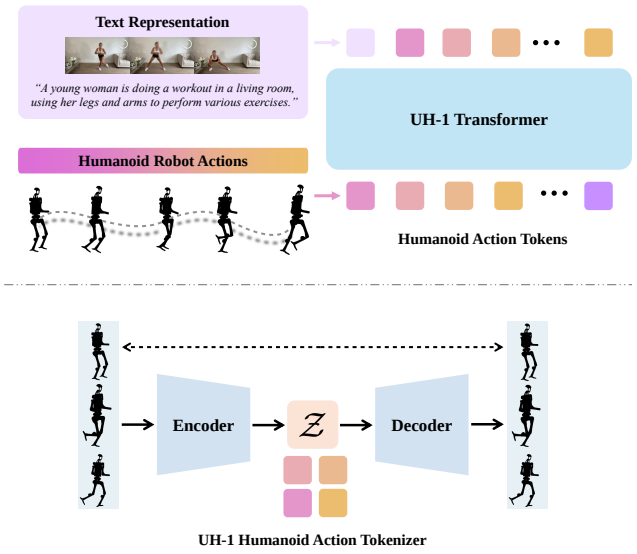


Fig. 4: **UH-1 Model Architecture**. UH-1 leverages the Transformer for scalable learning. Humanoid actions are first tokenized into discrete action tokens. Then, we train the UH-1 Transformer that takes text commands as inputs and auto-regressively generates the corresponding humanoid action tokens.

reward function includes multiple terms: motion rewards to encourage imitation of the retargeted humanoid keypoints $\mathcal{P}_{robot}$ and DoF positions q_{robot}; root tracking rewards to follow target root orientations and linear velocities from t_{root}; and stability rewards to help the robot maintain balance and prevent falls during movement. The resulting policy π and robotic actions $\mathcal{A}_{robot}$ enable the humanoid robot to operate safely in the physical world while maintaining the desired motions.

Finally, we collect a large number of motion samples $\langle \mathcal{V}, \mathcal{T}, \mathcal{P}_{human}, \mathcal{P}_{robot}, \mathcal{A}_{robot} \rangle$ from massive videos. In the next section, we investigate how to train a universal humanoid pose control policy using massive motion samples.

IV. UH-1 for Universal Humanoid Pose Control

Learning from massive videos enables us to distill a universal humanoid pose control policy from large-scale motion samples $\langle \mathcal{V}, \mathcal{T}, \mathcal{P}_{human}, \mathcal{P}_{robot}, \mathcal{A}_{robot} \rangle$. We introduce UH-1, a large language-conditioned humanoid model that takes natural language commands $\mathcal{T}$ and generates corresponding humanoid robotic actions $\{\mathcal{P}_{robot}, \mathcal{A}_{robot}\}$:

$$\pi_{UH\text{-}1} : \mathcal{T} \mapsto \{\mathcal{P}_{robot}, \mathcal{A}_{robot}\}, \tag{9}$$

where $\pi_{UH\text{-}1}$ denotes the UH-1 model. Notably, as illustrated in Fig. 5, our model can either generate high-level humanoid keypoints $\mathcal{P}_{robot}$, which are then fed into the goal-conditioned policy π to control the humanoid robot in closed-loop, or generate robotic actions $\mathcal{A}_{robot}$ for direct open-loop control. Our model bridges the gap between semantic language commands and physically deployable robotic actions,

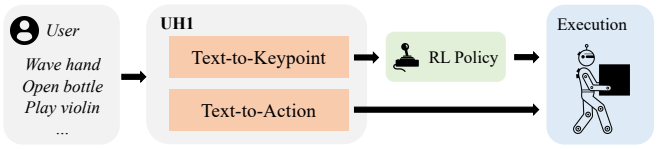


Fig. 5: **Text-to-keypoint and text-to-action control modes**. UH-1 can either generate high-level humanoid keypoints (text-to-keypoint) for the goal-conditioned policy π to control the humanoid robot in closed-loop, or generate robotic actions $\mathcal{A}_{robot}$ for direct open-loop control (text-to-action).

enabling more generalizable humanoid robotic control using text instructions. For simplicity, in the following section, we use $\mathcal{A}_{robot}$ as an example to illustrate our method; $\mathcal{P}_{robot}$ can be generated in the same manner.

We adopt the Transformer [70] as our main model architecture due to its scalability to large-scale data. As shown in fig. 4, to enable efficient learning, we first train an action tokenizer using [71] to discretize humanoid motions into a vocabulary of action tokens. Then, we train the Transformer to auto-regressively decode action tokens, resulting in executable humanoid actions.

UH-1 Action Tokenizer. We follow [71] and map T frames of actions $\mathcal{A}_{robot} = [a_1, \ldots, a_T]$ into a sequence of discrete action tokens $\mathcal{Z}_{token} = [z_1, \ldots, z_{T/K}]$ via an encoder F_{encode} and quantization F_{quant}:

$$\mathcal{Z}_{token} = F_{quant}(F_{encode}(\mathcal{A}_{robot})), \tag{10}$$

where F_{encode} and F_{quant} are standard operations in [71]. The action tokens $\mathcal{Z}_{token}$ come from a shared action vocabulary, and each token can be viewed as a motion primitive that is learned and shared across all data samples. Notably, different from language tokenization, humanoid actions won't change much in adjacent frames. To maintain the temporal smoothness in humanoid actions, we encode a short clip with K frames of actions $[a_{iK}, \ldots a_{(i+1)K}]$ into a single action token z_i, rather than encoding each frame individually. This approach not only preserves smooth transitions but also eases the learning process.

The decoder of VQ-VAE F_{decode} tries to reconstruct the original action sequence with the latent embeddings associated with the action tokens:

$$\mathcal{A}'_{robot} = F_{decode}(\mathcal{Z}_{token}). \tag{11}$$

We denote the reconstructed action sequence as $\mathcal{A}'_{robot} = [a'_1, \ldots, a'_T]$. The reconstruction loss is formulated as

$$L_{recon} = \sum_{i}^{T} (|a'_i - a_i| + |(a'_{i+1} - a'_i) - (a_{i+1} - a_i)|), \tag{12}$$

where the first term is the L_1 reconstruction loss in [71] and the second term encourages the first-order similarity of original and reconstructed action sequences. Additionally, we add regularization terms on latent embeddings as in [71].

UH-1 Transformer. We formulate the task of language-conditioned humanoid pose control as auto-regressively decoding action tokens $\mathcal{Z}_{token}$ conditioning on text commands $\mathcal{T}$. Formally, let $\mathcal{Z}_{token} = [z_1, \ldots z_{T/K}]$ denote the target

Methods	FID ↓	MM Dist ↓	Diversity ↑	R Precision ↑
Oracle	$0.005^{\pm.001}$	$3.140^{\pm.010}$	$9.846^{\pm.062}$	$0.780^{\pm.003}$
MDM [73]	$0.582^{\pm.051}$	$5.921^{\pm.034}$	$10.122^{\pm.078}$	$0.617^{\pm.007}$
T2M-GPT [74]	$0.667^{\pm.109}$	$3.401^{\pm.017}$	$\mathbf{10.328}^{\pm.099}$	$0.734^{\pm.004}$
UH-1 (ours)	$\mathbf{0.445}^{\pm.078}$	$\mathbf{3.249}^{\pm.016}$	$10.157^{\pm.106}$	$\mathbf{0.761}^{\pm.003}$

TABLE I: **Comparisons of model performances on the HumanoidML3D benchmark**. We calculate standard metrics following [57], repeating each evaluation 20 times and reporting the average along with the 95% confidence interval. The results indicate that UH-1 attains the highest performance across most metrics and achieves comparable performance on the *Diversity* metric.

Dataset	FID ↓	MM Dist ↓	Diversity ↑	R Precision ↑
Oracle	$0.005^{\pm.001}$	$3.140^{\pm.010}$	$9.846^{\pm.062}$	$0.780^{\pm.003}$
HumanoidML3D	$0.445^{\pm.078}$	$3.249^{\pm.016}$	$10.157^{\pm.106}$	$0.760^{\pm.003}$
Humanoid-X	$\mathbf{0.379}^{\pm.046}$	$\mathbf{3.232}^{\pm.008}$	$\mathbf{10.221}^{\pm.100}$	$\mathbf{0.761}^{\pm.003}$

TABLE II: **Dataset quality evaluation.**. Training on the Humanoid-X dataset greatly improves the quality and reliability of humanoid actions, compared to training on the HumanoidML3D dataset.

action token sequence, where z_i is the current step to predict, and $z_{1:i-1}$ represent the preceding context of action tokens, and l denote the text embedding by encoding the text command $\mathcal{T}$ with the CLIP [72] encoder. The UH-1 Transformer is then trained to model the conditional probability distribution $P(z_i|z_{1:i-1}, l)$. A special [End] token is incorporated into the vocabulary to signal the termination of sequence generation. During training, we first tokenize each $\mathcal{A}_{robot}$ into $\mathcal{Z}_{token}$ using Eq. 3. Then, we feed the language embedding l into the UH-1 transformer, and the transformer auto-regressively decodes action tokens. The learning objective is to minimize the negative log-likelihood over the whole training dataset $\mathcal{D}$:

$$\mathcal{L}_{learn} = -\sum_{\mathcal{Z}\in\mathcal{D}} \log \prod_{i=1}^{|\mathcal{Z}|} p(z_i|z_{1:i-1}, l). \tag{13}$$

During inference, using Eq. 11, the generated action tokens are decoded into $\mathcal{A}_{robot}$ for controlling the humanoid robot. The Transformer architecture and auto-regressive modeling ensure scalable learning of humanoid robot pose control.

V. Experiments

In this section, we conduct extensive experiments to investigate the following research questions: (1) *Universal Pose Control with UH-1*: Does our UH-1 model enable universal humanoid robot pose control based on text commands? (2) *Scalability and Generalization with Humanoid-X*: Does the large-scale Humanoid-X dataset facilitate scalable training and improve the generalization ability of our UH-1 model? (3) *Real-World Deployment of UH-1*: Can our UH-1 model be deployed on real humanoid robots to enable reliable robotic control in real-world environments?

If not specially mentioned in our experiments, we use *whole-body control* for the humanoid robot by default.

A. Universal Humanoid Pose Control with UH-1

We conduct extensive experiments to validate the generalization ability of the UH-1 model. An alternative solution to text-to-humanoid action generation is a two-stage pipeline: generating 3D human motions first and then retargeting the human motions to humanoid robots. To this end, we compare our method with two important baselines for text-to-human motion generation: Motion Diffusion Model (MDM) [73] and Text-to-Motion GPT (T2M-GPT) [74]. For fair comparisons, We choose the commonly used HumanML3D [57] benchmarks and transform the humans in this dataset into humanoid robots, resulting in a new benchmark called HumanoidML3D. Similarly, we adopt the same motion retargeting method as in this paper to transform the human motions generated by the baselines into humanoid actions. We adopt the metrics in [57] to evaluate the humanoid motions from different aspects: (1) Quality: The *Frechet Inception Distance (FID)* evaluates the dissimilarity between feature distributions of generated and ground truth humanoid poses. (2) Diversity: The *Diversity* metric evaluates the variability within the generated humanoid pose distribution, calculated as the average Euclidean distance between 300 randomly sampled pairs of humanoid poses. (3) Reliability: The *Multimodal Distance (MM Dist)* measures the Euclidean distance between motions and corresponding texts, and the *R Precision* assesses the accuracy of text and humanoid pose matches in the Top 3 rankings.

table I shows the results of our UH-1 model compared against the baselines. The results indicate that UH-1 attains the highest performance across nearly all metrics, showing an over 23% reduction in the critical *FID* metric, while also maintaining comparable performance on the *Diversity* metric. The first-order similarity loss proposed in this paper greatly enhances the quality and reliability of the generated outputs. The results suggest that UH-1 is a streamlined model and performs better than the two-stage methods.

B. Scalable Learning with Humanoid-X

In this section, we investigate whether scaling up training data with the large-scale Humanoid-X dataset can improve the generalization ability of our model. To explore this, we first pre-trained our UH-1 model on the Humanoid-X dataset and then finetuned and evaluated the performance on the HumanoidML3D benchmark. table II shows the performance comparison with training only on HumanoidML3D. We found that pre-training on the Humanoid-X dataset greatly improves the quality, reliability, and diversity of humanoid actions, with an *FID* improvement from 0.445 to 0.379, a *MM Dist* score improvement from 3.249 to 3.232, and a *Diversity* improvement from 10.157 to 10.221.

In addition, we also study how scaling up training data affects the model performance. To this end, we train our UH-1 model on varying proportions of the Humanoid-X dataset, specifically 1%, 10%, 25%, 50%, 75%, and 100%. The results shown in fig. 7 indicate that scaling up training

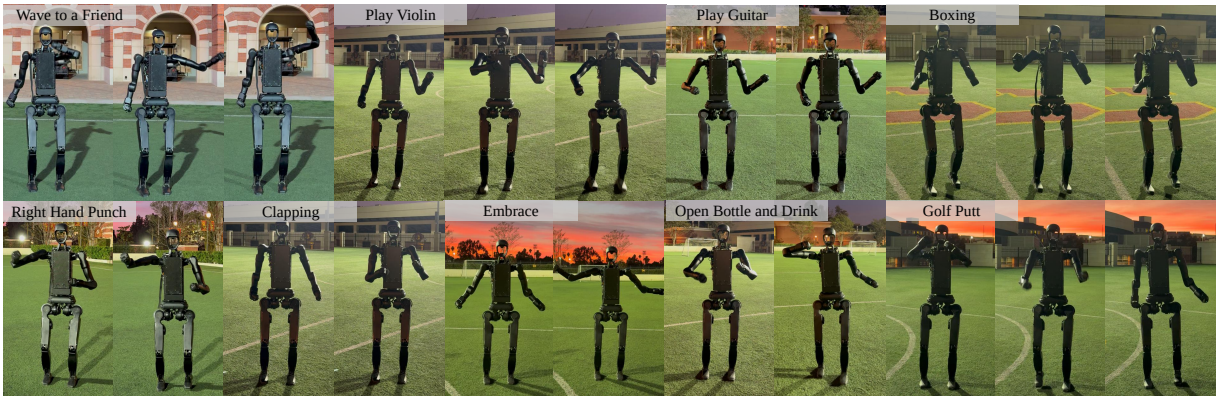
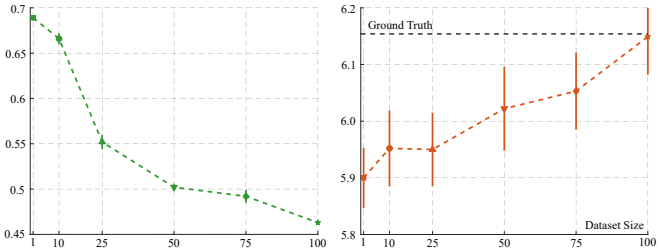


Fig. 6: **Real robot experiment**. UH-1 model can be reliably deployed on the real humanoid robot with a nearly 100% success rate.



(a) FID ↓ (b) Diversity ↑

Fig. 7: **Effectiveness of scaling up training data**. Points indicate the mean values, and error bars indicate the 95% confidence interval. Increasing the dataset size from 1% to 100% leads to significant improvements in both *FID* and *Diversity* metric.

data from 1% to 100% leads to a significant performance improvement in all metrics (*FID* from 0.689 to 0.463 and *Diversity* from 5.900 to 6.149). This suggests that by learning from massive videos, we successfully scale up the training data of humanoid robots and attain better performance.

C. Real-World Deployment of UH-1

To investigate whether our UH-1 model, trained on the Humanoid-X dataset, can generate reliable humanoid actions that are physically deployable on humanoid robots, we designed 12 distinct language commands, as shown in table III, and evaluated them on a real humanoid robot. We use Unitree H1-2 as our test embodiment. For the experiments, we evaluated each language command 10 times and controlled the robot in different places. The success criteria is in Appendix E.1. Notably, for text-to-humanoid actions, we found that open-loop control can only work for upper-body control, so in this control mode, we use a pre-trained locomotion policy for controlling the lower-body of the humanoid robot. fig. 6 shows the demos of real-robot experiments. table III measures the task success rate for each language command. Our experimental results demonstrate that our UH-1 model can be reliably deployed on the real humanoid robot, achieving a success rate of nearly 100% across all evaluated language instructions.

Instruction	Text-to-Keypoint	Text-to-Action
Boxing	90%	70%
Clapping	100%	100%
Cross Arms	80%	80%
Embrace	100%	100%
Golf Putt	90%	100%
Open Bottle & Drink	100%	100%
Play Guitar	100%	100%
Play Violin	100%	80%
Pray	100%	100%
Left Hand Punch	100%	100%
Right Hand Punch	100%	90%
Wave to Friend	100%	100%

TABLE III: **Task success rate on a real humanoid robot**. Both *Text-to-Keypoint* and *Text-to-Action* modes can reach a success rate of nearly 100% across all evaluated language instructions.

D. Empirical Studies

Analysis of two control modes. UH-1 can either produce high-level humanoid keypoints for a goal-conditioned, closed-loop policy or directly generate robotic actions for open-loop control. To investigate the effectiveness of these two control modes, we randomly generate 100 keypoint sequences and 100 action sequences for each task, as illustrated in fig. 8, and apply them in simulated robot control. The findings indicate that both modes can achieve an average success rate exceeding 89%, suggesting that text-to-action open-loop control with a separate locomotion policy is sufficient for most tasks. Moreover, the text-to-keypoint control mode, benefiting from the whole-body control policy, demonstrates slightly better robustness.

Ablation study on the action tokenizer. We conduct an ablation study to investigate the impact of different vocabulary sizes of the UH-1 action tokenizer on model training. We selected the vocabulary sizes of 512, 1024, and 2048, and reported the model performances on the Humanoid-X dataset. As illustrated in fig. 9, increasing the vocabulary size up to 2048 leads to an improvement in *FID* metric from 0.539 to 0.463 and brings an improvement in *Diversity* metric from 6.050 to 6.149. This indicates that increasing the number of motion primitives learned in the action tokenizer results in more diverse humanoid motion generation. Due to

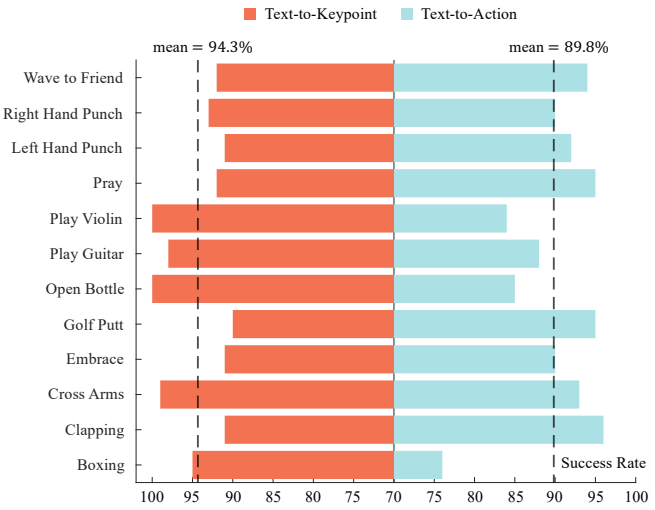


Fig. 8: **Simulated experiments on the UH-1 control modes**. Bars indicate success rates for specific commands and dash lines show the mean success rate on 12 different text instructions. While *Text-to-Action* mode with a separate locomotion policy is sufficient for most tasks, *Text-to-Keypoint* mode shows greater robustness.

Methods	FID ↓	MM Dist ↓	Diversity ↑	R Precision ↑
Oracle	$0.005^{\pm.001}$	$3.140^{\pm.010}$	$9.846^{\pm.062}$	$0.780^{\pm.003}$
Diffusion model	$0.624^{\pm.074}$	$5.536^{\pm.029}$	$\mathbf{10.281}^{\pm.096}$	$0.630^{\pm.007}$
Transformer	$\mathbf{0.379}^{\pm.046}$	$\mathbf{3.232}^{\pm.008}$	$10.221^{\pm.100}$	$\mathbf{0.761}^{\pm.003}$

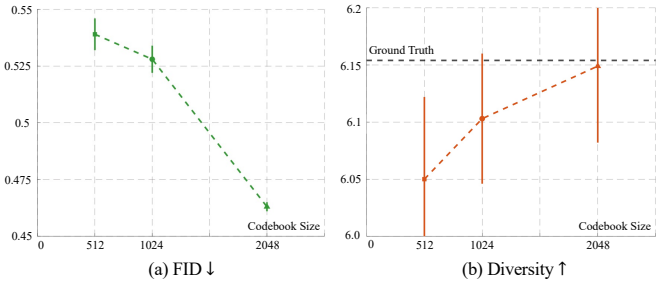
TABLE IV: **Diffusion model vs. Transformer as the UH-1 model.** We found that the Transformer architecture is more scalable to large-scale training data and exhibits better performance.

the limited computational resources, we didn't try a larger vocabulary. We will leave this for future works.

Ablation study on the model architecture. A key consideration for generation tasks is selecting the appropriate model architecture, such as the Transformer or diffusion model. To explore this, we trained a text-controlled humanoid motion diffusion model on the Humanoid-X dataset and compared its performance with the original Transformer-based UH-1 model. The results in table IV show that the Transformer architecture used in UH-1 is more scalable to large-scale training data and achieves better performance, with a lower *FID* and *MM Dist* score compared to the diffusion-based model.

VI. Conclusion

We introduce Humanoid-X, a large-scale dataset that facilitates scalable humanoid robot learning from massive videos. On top of Humanoid-X, we trained a large humanoid model, UH-1, for generalizable humanoid pose control based on language commands. Extensive experiments demonstrate that scalable training enables UH-1 to generate generalizable and reliable humanoid actions following language commands, and the UH-1 model can be effectively deployed on the real humanoid robot.



(a) FID ↓ (b) Diversity ↑

Fig. 9: **Ablation on the vocabulary sizes of the UH-1 action tokenizer**. Increasing the vocabulary size of the action tokenizer provides more motion primitives for humanoid robots and thus leads to an improvement in both *FID* and *Diversity* metric.

Limitations. In this paper, we only study the humanoid pose control. Humanoid manipulation is not in the scope of this paper. In future works, we plan to investigate learning humanoid loco-manipulation from Internet videos.

References

[1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*. PMLR, 2021.

[2] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *ICCV*, 2023.

[3] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *TMLR*, 2023.

[4] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *CVPR*, 2022.

[5] A. Blattmann, T. Dockhorn, S. Kulal, D. Mendelevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English, V. Voleti, A. Letts *et al.*, "Stable video diffusion: Scaling latent video diffusion models to large datasets," *arXiv preprint arXiv:2311.15127*, 2023.

[6] 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," *CoRL*, 2024.

[7] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, "Rt-1: Robotics transformer for real-world control at scale," in *RSS*, 2023.

[8] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *CoRL*, 2023.

[9] A. O'Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models," in *ICRA*, 2023.

[10] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak, "Affordances from human videos as a versatile representation for robotics," in *CVPR*, 2023.

[11] Y. Kuang, J. Ye, H. Geng, J. Mao, C. Deng, L. Guibas, H. Wang, and Y. Wang, "Ram: Retrieval-based affordance transfer for generalizable zero-shot robotic manipulation," *CoRL*, 2024.

[12] K. Grauman, A. Westbury, L. Torresani, K. Kitani, J. Malik, T. Afouras, K. Ashutosh, V. Baiyya, S. Bansal, B. Boote *et al.*, "Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives," in *CVPR*, 2024.

[13] M. Xu, Z. Xu, Y. Xu, C. Chi, G. Wetzstein, M. Veloso, and S. Song, "Flow as the cross-domain manipulation interface," *CoRL*, 2024.

[14] C. Yuan, C. Wen, T. Zhang, and Y. Gao, "General flow as foundation affordance for scalable robot learning," *arXiv preprint arXiv:2401.11439*, 2024.

[15] M. Yang, Y. Du, K. Ghasemipour, J. Tompson, D. Schuurmans, and P. Abbeel, "Learning interactive real-world simulators," in *ICLR*, 2024.

[16] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Real-world humanoid locomotion with reinforcement learning," *Science Robotics*, vol. 9, no. 89, 2024.

[17] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Robust and versatile bipedal jumping control through reinforcement learning," in *RSS*, 2023.

[18] X. Gu, Y.-J. Wang, and J. Chen, "Humanoid-gym: Reinforcement learning for humanoid robot with zero-shot sim2real transfer," *arXiv preprint arXiv:2404.05695*, 2024.

[19] X. Gu, Y.-J. Wang, X. Zhu, C. Shi, Y. Guo, Y. Liu, and J. Chen, "Advancing humanoid locomotion: Mastering challenging terrains with denoising world model learning," *RSS*, 2024.

[20] Z. Chen, X. He, Y.-J. Wang, Q. Liao, Y. Ze, Z. Li, S. S. Sastry, J. Wu, K. Sreenath, S. Gupta *et al.*, "Learning smooth humanoid locomotion through lipschitz-constrained policies," *arXiv preprint arXiv:2410.11825*, 2024.

[21] X. Cheng, Y. Ji, J. Chen, R. Yang, G. Yang, and X. Wang, "Expressive whole-body control for humanoid robots," *RSS*, 2024.

[22] I. Radosavovic, S. Kamat, T. Darrell, and J. Malik, "Learning humanoid locomotion over challenging terrain," *arXiv preprint arXiv:2410.03654*, 2024.

[23] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. Kitani, C. Liu, and G. Shi, "Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning," *CoRL*, 2024.

[24] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn, "Humanplus: Humanoid shadowing and imitation from humans," *CoRL*, 2024.

[25] T. He, Z. Luo, W. Xiao, C. Zhang, K. Kitani, C. Liu, and G. Shi, "Learning human-to-humanoid real-time whole-body teleoperation," *IROS*, 2024.

[26] M. Seo, S. Han, K. Sim, S. H. Bang, C. Gonzalez, L. Sentis, and Y. Zhu, "Deep imitation learning for humanoid loco-manipulation through human teleoperation," in *Humanoids*. IEEE, 2023.

[27] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," in *CoRL*, 2022.

[28] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik, "Masked visual pre-training for motor control," *arXiv preprint arXiv:2203.06173*, 2022.

[29] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang, "Vip: Towards universal visual reward and representation via value-implicit pre-training," in *ICLR*, 2023.

[30] I. Radosavovic, B. Shi, L. Fu, K. Goldberg, T. Darrell, and J. Malik, "Robot learning with sensorimotor pre-training," in *CoRL*. PMLR, 2023.

[31] S. Bahl, A. Gupta, and D. Pathak, "Human-to-robot imitation in the wild," in *RSS*, 2022.

[32] R. Mendonca, S. Bahl, and D. Pathak, "Structured world models from human videos," in *RSS*, 2023.

[33] Y. Du, M. Yang, P. Florence, F. Xia, A. Wahid, B. Ichter, P. Sermanet, T. Yu, P. Abbeel, J. B. Tenenbaum *et al.*, "Video language planning," in *ICLR*, 2024.

[34] I. Radosavovic, B. Zhang, B. Shi, J. Rajasegaran, S. Kamat, T. Darrell, K. Sreenath, and J. Malik, "Humanoid locomotion as next token prediction," *NeurIPS*, 2024.

[35] A. Tang, T. Hiraoka, N. Hiraoka, F. Shi, K. Kawaharazuka, K. Kojima, K. Okada, and M. Inaba, "Humanmimic: Learning natural locomotion and transitions for humanoid robot via wasserstein adversarial imitation," in *ICRA*. IEEE, 2024.

[36] J. Li, Y. Zhu, Y. Xie, Z. Jiang, M. Seo, G. Pavlakos, and Y. Zhu, "Okami: Teaching humanoid robots manipulation skills through single video imitation," *CoRL*, 2024.

[37] Y. Ze, Z. Chen, W. Wang, T. Chen, X. He, Y. Yuan, X. B. Peng, and J. Wu, "Generalizable humanoid manipulation with improved 3d diffusion policies," *RSS*, 2024.

[38] T. He, W. Xiao, T. Lin, Z. Luo, Z. Xu, Z. Jiang, J. Kautz, C. Liu, G. Shi, X. Wang *et al.*, "Hover: Versatile neural whole-body controller for humanoid robots," *ICRA*, 2025.

[39] J. Zhang, Y. Zhang, X. Cun, Y. Zhang, H. Zhao, H. Lu, X. Shen, and Y. Shan, "Generating human motion from textual descriptions with discrete representations," in *CVPR*, 2023.

[40] B. Jiang, X. Chen, W. Liu, J. Yu, G. Yu, and T. Chen, "Motiongpt: Human motion as a foreign language," in *NeurIPS*, vol. 36, 2023.

[41] G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-Or, and A. H. Bermano, "Human motion diffusion model," in *ICLR*, 2023.

[42] M. Zhang, Z. Cai, L. Pan, F. Hong, X. Guo, L. Yang, and Z. Liu, "Motiondiffuse: Text-driven human motion generation with diffusion model," *PAMI*, 2024.

[43] Y. Shafir, G. Tevet, R. Kapon, and A. H. Bermano, "Human motion diffusion as a generative prior," in *ICLR*, 2024.

[44] H. Liang, W. Zhang, W. Li, J. Yu, and L. Xu, "Intergen: Diffusion-based multi-human motion generation under complex interactions," *IJCV*, 2024.

[45] Y. Xie, V. Jampani, L. Zhong, D. Sun, and H. Jiang, "Omnicontrol: Control any joint at any time for human motion generation," in *ICLR*, 2023.

[46] Y. Yuan, J. Song, U. Iqbal, A. Vahdat, and J. Kautz, "Physdiff: Physics-guided human motion diffusion model," in *ICCV*, 2023.

[47] Z. Luo, J. Cao, J. Merel, A. Winkler, J. Huang, K. Kitani, and W. Xu, "Universal humanoid motion representations for physics-based control," in *ICLR*, 2024.

[48] Z. Luo, J. Cao, K. Kitani, W. Xu *et al.*, "Perpetual humanoid control for real-time simulated avatars," in *ICCV*, 2023.

[49] X. B. Peng, Y. Guo, L. Halper, S. Levine, and S. Fidler, "Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters," *TOG*, vol. 41, no. 4, 2022.

[50] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, "Amp: Adversarial motion priors for stylized physics-based character control," *TOG*, vol. 40, no. 4, 2021.

[51] C. Tessler, Y. Kasten, Y. Guo, S. Mannor, G. Chechik, and X. B. Peng, "Calm: Conditional adversarial latent models for directable virtual characters," in *ACM SIGGRAPH 2023 Conference Proceedings*, 2023.

[52] J. Won, D. Gopinath, and J. Hodgins, "A scalable approach to control diverse behaviors for physically simulated characters," *TOG*, vol. 39, no. 4, 2020.

[53] S. Goel, G. Pavlakos, J. Rajasegaran, A. Kanazawa, and J. Malik, "Humans in 4d: Reconstructing and tracking humans with transformers," in *ICCV*, 2023.

[54] Z. Jiang, Y. Xie, J. Li, Y. Yuan, Y. Zhu, and Y. Zhu, "Harmon: Whole-body motion generation of humanoid robots from language descriptions," *CoRL*, 2024.

[55] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl: a skinned multi-person linear model," *TOG*, vol. 34, no. 6, 2015.

[56] S. Tsuchida, S. Fukayama, M. Hamasaki, and M. Goto, "Aist dance video database: Multi-genre, multi-dancer, and multi-camera database for dance information processing." in *ISMIR*, vol. 1, no. 5, 2019.

[57] C. Guo, S. Zou, X. Zuo, S. Wang, W. Ji, X. Li, and L. Cheng, "Generating diverse and natural 3d human motions from text," in *CVPR*, 2022.

[58] S. Zhang, Q. Ma, Y. Zhang, Z. Qian, T. Kwon, M. Pollefeys, F. Bogo, and S. Tang, "Egobody: Human body shape and motion of interacting people from head-mounted devices," in *ECCV*. Springer, 2022.

[59] O. Taheri, N. Ghorbani, M. J. Black, and D. Tzionas, "Grab: A dataset of whole-body human grasping of objects," in *ECCV*. Springer, 2020.

[60] J. Chung, C.-h. Wuu, H.-r. Yang, Y.-W. Tai, and C.-K. Tang, "Haa500: Human-centric atomic action dataset with curated videos," in *ICCV*, 2021.

[61] Z. Cai, D. Ren, A. Zeng, Z. Lin, T. Yu, W. Wang, X. Fan, Y. Gao, Y. Yu, L. Pan *et al.*, "Humman: Multi-modal 4d human dataset for versatile sensing and modeling," in *ECCV*. Springer, 2022.

[62] J. Lin, A. Zeng, S. Lu, Y. Cai, R. Zhang, H. Wang, and L. Zhang, "Motion-x: A large-scale 3d expressive whole-body human motion dataset," in *NeurIPS*, vol. 36, 2024.

[63] G. A. Sigurdsson, G. Varol, X. Wang, I. Laptev, A. Farhadi, and A. Gupta, "Hollywood in homes: Crowdsourcing data collection for activity understanding," in *ECCV*, 2016. [Online]. Available: http://arxiv.org/abs/1604.01753

[64] J. Carreira, E. Noland, C. Hillier, and A. Zisserman, "A short note on the kinetics-700 human action dataset," *arXiv preprint arXiv:1907.06987*, 2019.

[65] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-time flying object detection with yolov8," *arXiv preprint arXiv:2305.09972*, 2023.

[66] Z. Cheng, S. Leng, H. Zhang, Y. Xin, X. Li, G. Chen, Y. Zhu, W. Zhang, Z. Luo, D. Zhao *et al.*, "Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms," *arXiv preprint arXiv:2406.07476*, 2024.

[67] M. Kocabas, N. Athanasiou, and M. J. Black, "Vibe: Video inference for human body pose and shape estimation," in *CVPR*, 2020.
[68] D. P. Kingma, "Adam: A method for stochastic optimization," in *ICLR*, 2014.
[69] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
[70] A. Vaswani, "Attention is all you need," in *NeurIPS*, 2017.
[71] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," in *NeurIPS*, vol. 30, 2017.
[72] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*. PMLR, 2021.
[73] G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-or, and A. H. Bermano, "Human motion diffusion model," in *ICLR*, 2023. [Online]. Available: https://openreview.net/forum?id=SJ1kSyO2jwu
[74] J. Zhang, Y. Zhang, X. Cun, Y. Zhang, H. Zhao, H. Lu, X. Shen, and Y. Shan, "Generating human motion from textual descriptions with discrete representations," in *CVPR*, 2023.
[75] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "Amass: Archive of motion capture as surface shapes," in *ICCV*, Oct 2019. [Online]. Available: https://amass.is.tue.mpg.de
[76] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+ d: A large scale dataset for 3d human activity analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1010–1019.

Appendix

This paper presents Humanoid-X, a large-scale dataset that facilitates scalable humanoid robot learning from massive videos, and UH-1, a large humanoid model for generalizable humanoid pose control based on language commands. The Internet videos that Humanoid-X and UH-1 involve in the dataset and the pipeline are strictly for academic research and are not intended for commercial use. On the privacy protection side, we apply face anonymization to all human subjects in the Internet videos involved in Humanoid-X and UH-1, making sure that the videos do not include any personal information. In addition, we will not release the original Internet videos to protect copyright. In summary, we believe that Humanoid-X and UH-1 do not raise ethical concerns.

In this section, we will introduce more details on the whole data collection pipeline of the Humanoid-X dataset, including data source distribution, video mining and processing, video captioning, 3D human pose estimation, motion retargeting from humans to humanoid robots, and the goal-conditioned humanoid control policy.

A. Data Source Distribution

Humanoid-X consists of massive motion samples with diverse sources, and the detailed source of the data in our Humanoid-X dataset is shown in table I. Humanoid-X consists of 163.8K motion samples, spanning 240.3 hours of video footage, containing 20.7M frames of human and robotic motion data, with a vocabulary size of 3206 words. Each motion video sample is expanded to the 5 data modalities $\langle \mathcal{V}, \mathcal{T}, \mathcal{P}_{human}, \mathcal{P}_{robot}, \mathcal{A}_{robot} \rangle$ of the motion sample in our Humanoid-X dataset. The subsections below introduce details on the dataset building and data processing pipeline.

B. Video Mining and Processing

To collect a dataset of videos featuring single-person movements, we first designed specific motion categories and then generated search prompts based on these categories. Using the phrase "single person" in searches often produced irrelevant results since the majority of the video titles would not specify whether the video is single person using the exact word "single person". So, activity-based terms were created to ensure relevant data retrieval. These categories included martial arts tutorials, fitness and exercise drills, sports techniques, dance practice, music performance tutorials, everyday movement patterns, animal-inspired movements, and rehabilitation exercises.

Martial arts tutorials included search terms for techniques, drills, and demonstrations across disciplines like Wushu, Taekwondo, Karate, and Kung Fu. Examples of generated terms are "karate front kick training," "taekwondo spinning hook kick demonstration," and "wushu staff spin practice." Fitness and exercise drills focused on isolated movements like "yoga handstand practice," and "calisthenics planche progression tutorial,".

Sports techniques targeted individual actions in activities like baseball, tennis, archery, running, and parkour, with examples including "tennis serve technique tutorial" and

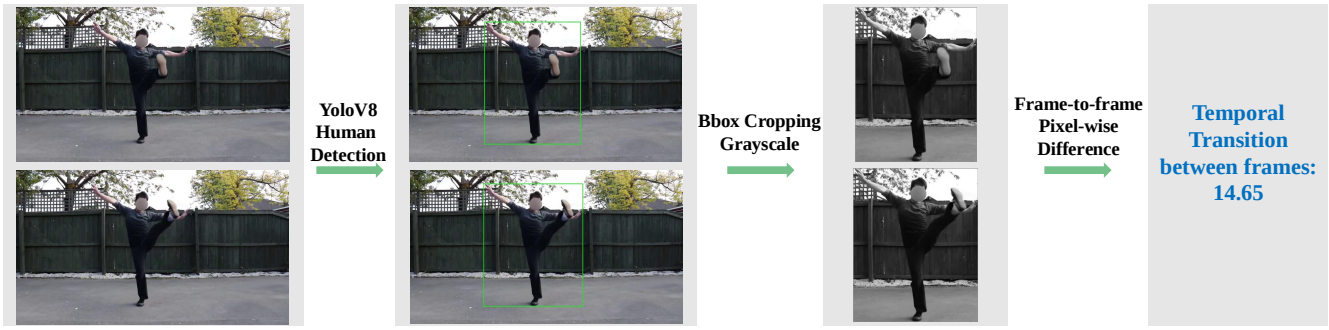


Fig. 1: **Video Processing Pipeline**.

"running stride form analysis." Dance practice emphasized solo routines in styles such as salsa, hip hop, ballet, modern dance, and improvisation, using terms like "salsa basic turn solo" and "ballet arabesque demonstration." Music performance tutorials captured movements involved in playing instruments such as guitar, violin, piano, and drums, with terms like "guitar strumming while standing solo" and "violin bowing technique while standing demonstration."

Everyday movement patterns focus on practical motions during daily activities, using terms like "picking up an object while balancing," "loading a dishwasher with proper form," and "squatting to tie shoelaces,". Animal-inspired movements were included to capture dynamic motion patterns with terms like "bear crawl coordination movement," "frog jump exercise," and "flamingo balance on one leg." Rehabilitation and mobility exercises targeted balance, flexibility, and strength, focusing on slow and deliberate movements such as "dynamic torso twist warm-up" and "hip flexor stretch technique breakdown."

By designing categories and generating search terms from these, we ensured the collected videos focused on single-person movements while covering a wide range of activities.

After collection of videos from the designed searching prompts, we designed a pipeline for detecting and extracting video segments featuring single-person movements. The process begins with the YOLOv8 model [65], which detects objects in each frame and identifies detected humans based on the class label corresponding to "person". Frames containing exactly one detected person are selected, ensuring the focus remains solely on single-person actions. Once a single-person frame is identified, a region of interest (ROI) is extracted using the bounding box of the detected individual from YOLOv8 detection result. To determine existence of motion, the pipeline calculates frame-to-frame differences in the grayscale ROI, assessing movement levels using predefined thresholds. This ensures only frames with significant motion are retained, while static or irrelevant segments are excluded.

To further refine the selection regarding motion, the pipeline employs a batch-based filtering process, analyzing sequences of frames to identify consistent motion patterns over time. Small movement threshold is applied to frame-to-frame and a larger threshold is applied to the frame batch, enabling the detection of subtle and significant activities by allowing relative small motions for several frames as long as large motion is detected in frames' batch. Such design would benefit continuity of the clips by keeping frames in between large motions. Frames that meet these criteria are grouped into chunks representing continuous motion, and only chunks exceeding a minimum duration are considered for clip generation.

The output clips are processed to maintain consistent quality and playback speed. Frames within each chunk are downsampled for efficiency, interpolated for smooth transitions, and standardized to 20 FPS. The resulting clips focus exclusively on single-person actions, discarding distractions such as multiple individuals or irrelevant frames. This approach ensures a precise and diverse dataset of single-person motion segments, suitable for applications in motion analysis, action recognition, and training of computer vision models. By integrating object detection, motion analysis, and sequence processing, the pipeline achieves high accuracy and relevance in isolating meaningful single-person movements.

Data Source	# of Clips	# of Frames	# of Hours	Vocab. Size
AIST	1.5K	0.3M	3.2	590
AMASS	13.4K	2.0M	27.4	3942
Charades	9.3K	1.0M	1.0	813
EgoBody	1.0K	0.4M	4.0	367
GRAB	1.3K	0.4M	3.8	565
HAA500	5.2K	0.3M	2.9	1754
HuMMan	0.7K	0.1M	1.0	980
IDEA400	12.5K	2.6M	24.0	1715
Kinetics700	68.6K	5.2M	72.4	3360
MotionX Video	40.6K	7.9M	72.9	4021
Online Video	17.8K	2.3M	32.6	2040
Total	**163.8K**	**20.7M**	**240.3**	**11897**

TABLE I: **Dataset statistics**. Compiled from diverse data sources, Humanoid-X possesses an extensive scale of data modalities and a massive action vocabulary.

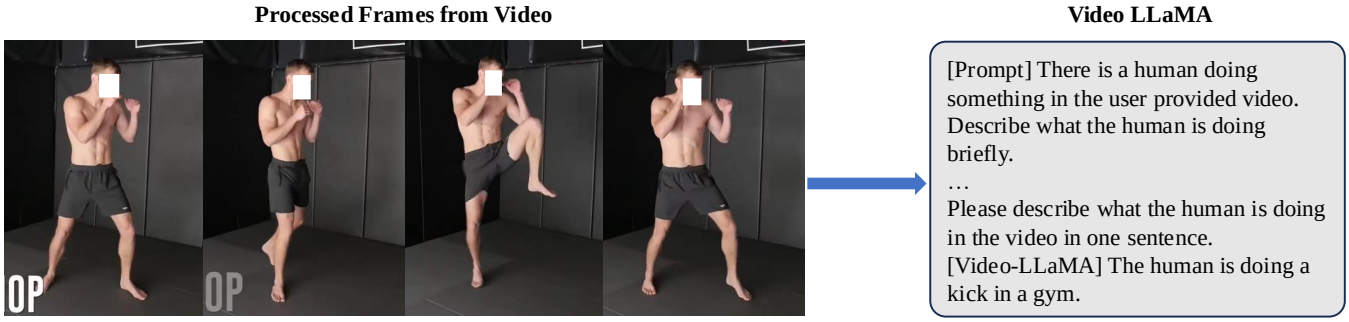


Fig. 2: **Video captioning example by using Video-LLaMA**.

C. Video Captioning

Video-LLaMA Prompt

There is a human doing something in the user provided video. Describe what the human is doing briefly.
You must follow the following rules:
1. Do not describe the appearance of the human.
2. You must at least answer "a man/woman doing something [adverb]"
3. If applicable, you should describe the [item] the human is interacting with, the [body part] the human is using, or the [location] the human is in.
4. Your answer must be within one sentence, and do not begin with "in the video".
Please describe what the human is doing in the video in one sentence.

For video captioning, we implemented a video captioning pipeline using Video LLaMA [66], with a video processing framework which extracts visual information from input videos by sampling a fixed number of eight frames at regular intervals.

The prompts used for video captioning are designed to produce concise and action-focused descriptions. The main prompt directs the model to describe the actions of a person in the video in a single sentence, explicitly avoiding mentions of the person's appearance. We used the query "Please describe what the human is doing in the video in one sentence." with guidance of rules shown above. Such a query would guarantee a concise description of motion without any irrelevant information being collected. An example of such interaction with Video-LLaMA is shown in fig. 2.

D. 3D Human Pose Estimation

The SMPL generation pipeline is designed to estimate 3D human pose and shape parameters from video frames. This process involves several key steps, including detecting the subject in video frames, estimating pose and shape parameters, and generating a 3D mesh representation. VIBE model [67] is used to infer SMPL parameters, such as body pose, global orientation, and shape coefficients, from video sequences. Bounding boxes are first detected for the subject, and these are used to crop and process the frames for subsequent steps. The final output includes SMPL parameters, root translations, and optional visualizations of the 3D mesh overlaid on the video frames.

The VIBE-based mesh regression model is used as video-based inference, which benefits from temporal consistency across frames. For the detected person in a video, the pipeline extracts bounding boxes and sequences of features from the video frames. VIBE processes these sequences to estimate the SMPL parameters, including pose rotations, shape coefficients, and camera parameters. The extracted parameters are then stored for further use in 3D visualization or downstream tasks. An example of SMPL visualization is shown in fig. 3.

To compute the root translation of the subject in 3D space, the bounding boxes and camera parameters from the mesh regression step are combined. The bounding box coordinates are converted to the original image coordinate system, accounting for resolution and aspect ratio. Using the weak-perspective camera parameters, including scale s and 2D translation $\mathbf{t} = (t_x, t_y)$, the depth t_z is estimated based on a predefined focal length f. The depth is computed as:

$$t_z = \frac{f}{s \cdot 0.5 \cdot W_{\text{img}}}, \tag{1}$$

where W_{img} represents the width of the input image. The root translation vector $\mathbf{T}_{\text{root}}$ is then formed as:

$$\mathbf{T}_{\text{root}} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}, \tag{2}$$

where $\mathbf{t} = (t_x, t_y)$ corresponds to the 2D translations from the camera parameters, and t_z is the computed depth.

E. Motion Retargeting

Our motion retargeting process mainly consists of two tasks: the optimization of human shape parameters β to fit human shapes to those of a humanoid robot, and solve the humanoid motor DoF positions q_{robot} from adjusted human joint positions with inverse kinematics.

Optimization of human shape parameters β. Given the forward kinematics of human body models in eq. (3), we

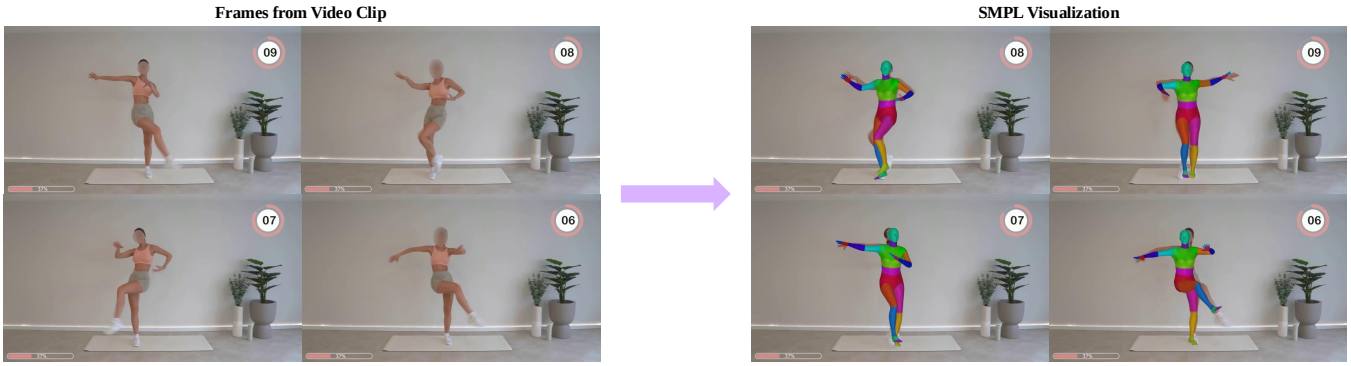


Fig. 3: **SMPL 3D human model estimation example**.

optimize the human shape parameters β with the Adam optimizer [68], using the loss $\mathcal{L}(\beta)$:

$$\mathcal{L}(\beta) = ||\mathcal{P}^T_{joints} - \mathcal{P}^T_{robot}||_2, \tag{3}$$

$$\text{s.t.} \quad \mathcal{P}^T_{joints} = F_{fk}(\mathcal{P}_{human}(\beta, \theta^T, t_{root})). \tag{4}$$

To avoid overfitting on $\mathcal{P}^T_{robot}$ which leads to too much deformation on the human model T-shaped pose, we set a limit to the human shape parameters β:

$$\forall i \in \{1, 2, \ldots, n\}, \beta = (\beta_1, \beta_2, \ldots, \beta_n), |\beta_i| < 5, \tag{5}$$

where n denotes the size of the human shape parameters β.

Solving humanoid motor DoF positions q_{robot}. With the optimal β and eq. (6), we need to extract the motor DoF positions q_{robot} through inverse kinematics in eq. (7). The inverse kinematics problem is solved by optimimzation with the loss $\mathcal{L}_{ik}$:

$$\mathcal{L}_{ik} = \mathcal{L}_r + \lambda \mathcal{L}_s. \tag{6}$$

In eq. (6), the retarget loss $\mathcal{L}_r$:

$$\mathcal{L}_r(q_{robot}, s_{root}) = ||F_{rk}(q_{robot}, s_{root}) - \mathcal{P}_{robot}||_1, \tag{7}$$

where s_{root} denotes robot root states including root translation and root orientation, F_{rk} denotes robot forward kinematics which maps from q_{robot}, s_{root} to humanoid robot keypoint positions. Also in eq. (6), the smoothing term $\mathcal{L}_s$:

$$\mathcal{L}_s(q_{robot}) = \sum_{i=1}^{n-2}(2q_{robot}[i] - q_{robot}[i-1] - q_{robot}[i+1]), \tag{8}$$

where n is the number of frames of one motion sample trajectory, with the index ranging from 0 to $n-1$. We use the Adam optimizer [68] to solve the inverse kinematics problem, where the weight of smoothing term $\lambda = 0.05$.

F. Goal-conditioned Control Policy

We use massively parallel simulation to train our goal-conditioned humanoid RL control policy with Isaac Gym. In this subsection, we will introduce our training data, our policy, our training rewards and training parameters.

Training Data. We selectively used a portion of the CMU MoCap dataset in AMASS [75], in the form of SMPL models. We exclude motions that involve physical interactions with others, heavy objects, or rough terrain. We

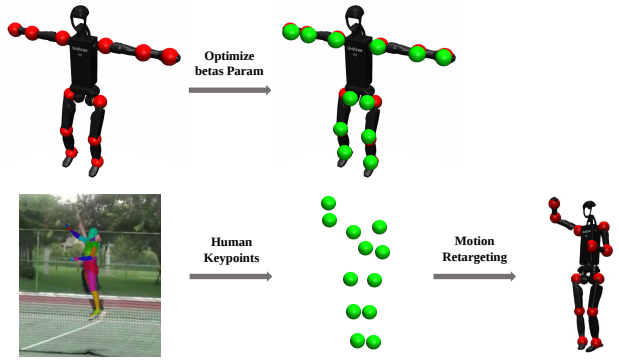


Fig. 4: **Motion Retargeting**, including optimization of human shape parameters and solving humanoid motor DoF positions.

Term	Reward Expression	Weight
DoF Position	$\exp(-0.7\|\mathbf{q}_{\text{tar}} - \mathbf{q}\|)$	3.0
Keypoint Position	$\exp(-\|\mathbf{t}_{\text{tar}} - \mathbf{t}\|)$	2.0
Root Linear Velocity	$\exp(-4.0\|\mathbf{v}_{\text{tar}} - \mathbf{v}\|)$	6.0
Root Roll & Pitch	$\exp(-\|\Omega^{\phi\theta}_{\text{tar}} - \Omega^{\phi\theta}\|)$	1.0
Root Yaw	$\exp(-\|\Delta y\|)$	1.0

TABLE II: **Imitation Rewards**.

retarget from the training data to humanoid robot motion with the method introduced above, including humanoid keypoint joint positions $\mathcal{P}_{robot}$, humanoid robot DoF positions q_{robot} and humanoid robot root states s_{root}. We can estimate the corresponding linear or angular velocities of humanoid DoFs and humanoid root joint from the humanoid motion data across frames.

RL Control Policy. Our goal is to track the root movement goal for the whole body and the target expression goal for upper body, and our training data is introduced above. The humanoid control policy is defined with eq. (8). The goal space can be formulated as $\mathcal{G} = \mathcal{G}^e \times \mathcal{G}^m$, where $\mathcal{G}^e$ includes joint angles and keypoint translations from the retargeting process above and the goal space for robot movement control $\mathcal{G}^m = \langle \mathbf{v}, rpy, h \rangle$ where $\mathbf{v} \in \mathbb{R}^3$ is the linear velocity, $rpy \in \mathbb{R}^3$ is the robot pose in terms of row/pitch/yaw and h is the body height. The observation $\mathcal{O}$ includes robot proprioception information $o_t = [\omega_t, r_t, p_t, \Delta y, q_t, \dot{q}_t, \mathbf{a}_{t-1}]^T$ where

Term	Reward Expression	Weight
Height	$\max(\lvert\mathbf{h}_{\text{feet}}\rvert - 0.2, 0)$	2.0
Time in Air	$\sum t_i^{\text{air}} \times 1_{\text{new contact}}$	10.0
Drag	$\sum \lvert\mathbf{v}_i^{\text{foot}}\rvert \times 1_{\text{new contact}}$	-0.1
Contact Force	$1\{\lvert F_i^z\rvert \geq F_{\text{th}}\} \times (\lvert F_i^z\rvert - F_{\text{th}})$	-3e-3
Stumble	$1\{\exists i, \lvert\mathbf{F}_i^{xy}\rvert > 4\lvert F_i^z\rvert\}$	-2.0
DoF Acceleration	$\lvert\ddot{\mathbf{q}}\rvert^2$	-3e-7
Action Rate	$\lvert\mathbf{a}_{t-1} - \mathbf{a}_t\rvert$	-0.1
Energy	$\lvert\dot{\mathbf{q}}\rvert^2$	-1e-3
Collision	$1_{\text{collision}}$	-10.0
DoF Limit Violation	$1_{q_i > q_{\max} \lvert\rvert q_i < q_{\min}}$	-0.1
DoF Deviation	$\lvert\mathbf{q}_{\text{default}}^{\text{low}} - \mathbf{q}^{\text{low}}\rvert^2$	-10.0
Vertical Linear Velocity	v_z^2	-1.0
Horizontal Angular Velocity	$\lvert\omega_{\text{xy}}\rvert^2$	-0.4
Projected Gravity	$\lvert\mathbf{g}_{\text{xy}}\rvert^2$	-2.0

TABLE III: **Regularization Rewards**.

Hyperparameter	Value
Discount Factor	0.99
GAE parameter	0.95
Timesteps per Rollout	21
Epochs per Rollout	5
Minibatches per Epoch	4
Extropy Bonus (α_2)	0.01
Value Loss Coefficient (α_1)	1.0
Clip Range	0.2
Reward Normalization	Yes
Learning Rate	1e-3
# Environments	6192
Optimizer	Adam

TABLE IV: **Training Parameters**.

ω_t is robot root angular velocity, r_t, p_t is roll and pitch, $\Delta y = y_t - y$ is the difference between current and desired yaw angle, q_t and $\dot{q}_t$ is the joint position and angular velocity and $\mathbf{a}_t \in \mathbb{R}^{27}$ is the target position of the joint proportional-derivative (PD) controllers.

Training Rewards. In each step, the reward from the environment consists of motion rewards, root tracking rewards and regularization terms. To protect the fragile ankle roll joints on the robot hardware, we set the actions of the two joints to zero every simulation step. Motion rewards include DoF position reward and keypoint position reward, and root tracking rewards include root linear velocity reward, root roll & pitch reward and root yaw reward.

The imitation rewards, including motion rewards and root tracking rewards, are listed in table II, where $\mathbf{q}_{\text{tar}}, \mathbf{q} \in \mathbb{R}^9$ are the target and actual upper body DoF positions, $\mathbf{t}_{\text{tar}}, \mathbf{t} \in \mathbb{R}^{18}$ are the target and actual upper body keypoint positions, $\mathbf{v}_{\text{tar}}, \mathbf{v} \in \mathbb{R}$ are the target and actual root velocity, $\Omega_{\text{tar}}^{\phi\theta}, \Omega^{\phi\theta}$ are the target and actual body roll and pitch.

The regularization rewards are listed in table III, where h_{feet} is feet height, t_i^{air} denotes the duration for which each foot remains in the air, $1_{\text{new contact}}$ means new foot contact with the ground, $\mathbf{F}_i^{xy}, F_i^z, F_{\text{th}}$ are foot contact force in horizontal plane and along the z-axis, and the contact force threshold respectively, $\dot{\mathbf{q}}, \ddot{\mathbf{q}}$ are joint velocity and acceleration, $\mathbf{a}_t$ is action at timestep t, $1_{\text{collision}}$ denotes self-collision, $q_{\max}, q_{\min}$ are limits for joint positions, and $\mathbf{g}_{\text{xy}}$ is gravity vector projected on horizontal plane.

Training Parameters. We use PPO with hyperparameters listed in table IV to train the policy.

In this section, we will introduce the Humanoid-X dataset. We will introduce the data format and structure and show several examples of the dataset.

G. Data Format and Structure

For each motion sample in Humanoid-X, we expand them to the 5 data modalities introduced in Sec. 3.1, where they are described with $\langle \mathcal{V}, \mathcal{T}, \mathcal{P}_{human}, \mathcal{P}_{robot}, \mathcal{A}_{robot} \rangle$. Visualization of part of the data samples in the dataset will be shown in appendix J.

Motion Video Clip $\mathcal{V}$. The video clips are collected in MP4 format at a frame rate of 20 frames per second (fps).

Text Description $\mathcal{T}$. The text descriptions are stored in plain text (*.txt*) format.

Human Poses $\mathcal{P}_{human}$. The human poses are sequences of SMPL model parameters with a frame rate of 20 fps. We stored the collected data for each motion sample in a NumPy (*.npy*) file.

Humanoid Keypoints $\mathcal{P}_{robot}$. The humanoid keypoints include humanoid robot DoFs q_{robot} and humanoid robot root states s_{root}. Each frame of the data contains 27 DoFs of the robot configuration and a 7-dimensional root state vector, consisting of 3-DoF root translation and 4-DoF quaternion representation for root orientation. The humanoid keypoints are recorded with a frame rate of 20 fps. We stored the collected data (27 robot DoFs and 7-DoF root state) for each motion sample in a NumPy (*.npy*) file for efficient data management and processing.

Humanoid Actions $\mathcal{A}_{robot}$. The humanoid actions are sequences of target DoF positions. The data is collected and stored at 50 fps, with each frame containing 27 robot DoFs that correspond to the robot's physical configuration. We stored the collected data for each motion sample in a NumPy (*.npy*) file.

H. Data Statistics

Comparison with other datasets. We compare Humanoid-X with other similar datasets at table V, including human action recognition, motion capture, and human motion dataset.

Sequence Length Analysis. We conduct comprehensive statistical analysis on both video sequence durations and their corresponding caption lengths, as illustrated in fig. 5 and fig. 6. The analysis reveals that the majority of video clips are relatively short, with durations less than 10 seconds. This distribution pattern stems from our video segmentation strategy, where clips are specifically extracted when significant or meaningful motion patterns are detected within the continuous recordings. This approach naturally results in shorter, more focused segments, making longer clips relatively rare in our dataset. Regarding the textual descriptions, the distribution of caption lengths shows that most sentences contain fewer than 20 words. This concise nature of captions

	Video (Single Human)	Language Description	3D Human Model	Robot Pose	Robot Action
Kinetics [64]	✓(✗)	✗	✗	✗	✗
NTU RGB+D [76]	✓(✗)	✗	✓	✗	✗
AMASS [75]	✗	✗	✓	✗	✗
HumanML3D [57]	✗	✓	✓	✗	✗
Motion-X [62]	✗	✓	✓	✗	✗
Humanoid-X (Ours)	✓(✓)	✓	✓	✓	✓

TABLE V: Comparison of Humanoid-X and other human datasets.

Part of Speech	Vocabulary Size
Verbs	3206
Nouns	6048
Adjectives	1526
Adverbs	590
Others	527
Total	11897

TABLE VI: Vocabulary Sizes for Each Part of Speech.

aligns with our guidelines, which emphasized brevity while maintaining descriptive accuracy.

Vocabulary Analysis. To gain deeper insights into the linguistic composition of video captions, we conduct a comprehensive analysis of different parts of speech, focusing on nouns, verbs, adjectives, and adverbs. This grammatical categorization helps understand how motions and actions are described in our dataset. table VI presents the vocabulary size distribution across these grammatical categories, providing a quantitative view of the linguistic diversity in our annotations. The analysis reveals the richness of descriptive elements used in capturing robot motions and their contextual information.

For verbs, the word cloud and the top-40 frequent words are shown in fig. 8. It can be seen that verbs like "doing", "standing", "playing", "holding" and "performing" occur with a relatively high frequency. This implicitly matched the expectation since these words are heavily used as the prompt for video collection.

For nouns, the word cloud and the top-40 frequent words are shown in fig. 9. The top 3 frequent words occurred are "man", "person" and "woman". This is also expected given the prompt for caption since it is specifically mentioned that the description should indicate what the man or woman is doing.

For adjectives and adverbs, their word cloud and top-40 frequent words are shown in fig. 10 and fig. 11. It can been seen that most frequent adverbs are mostly about direction of motions and most frequent adjectives are mostly above color. This is cause by the fact that we explicitly instruct the Video-LLaMA to be concise so that there would not be redundant words for non-motion-related contents.

I. Data Preparation and Release

We will fully release our data and code in the future, without violating the ethics concerns stated in appendix .

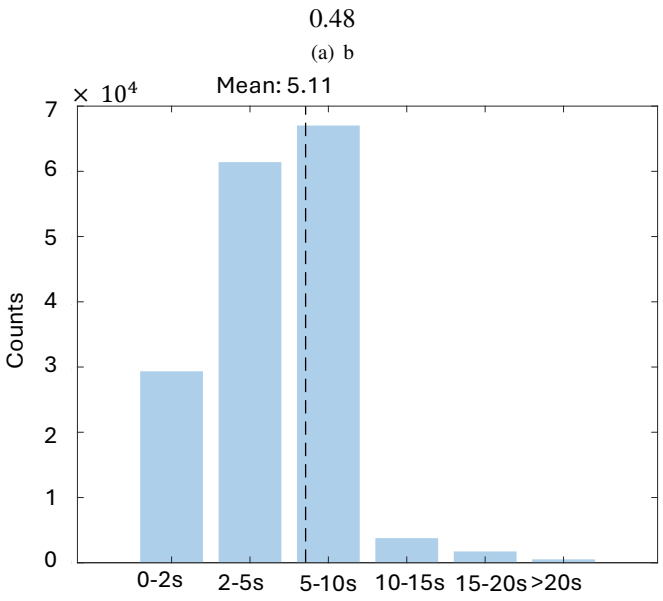


Fig. 5: Video Length

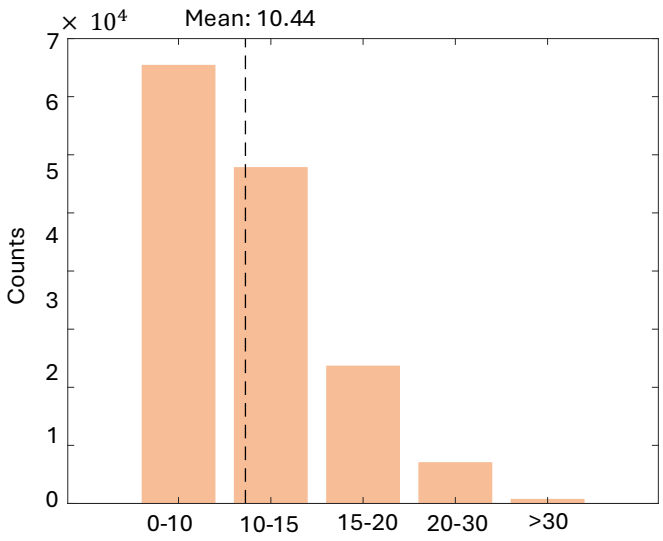
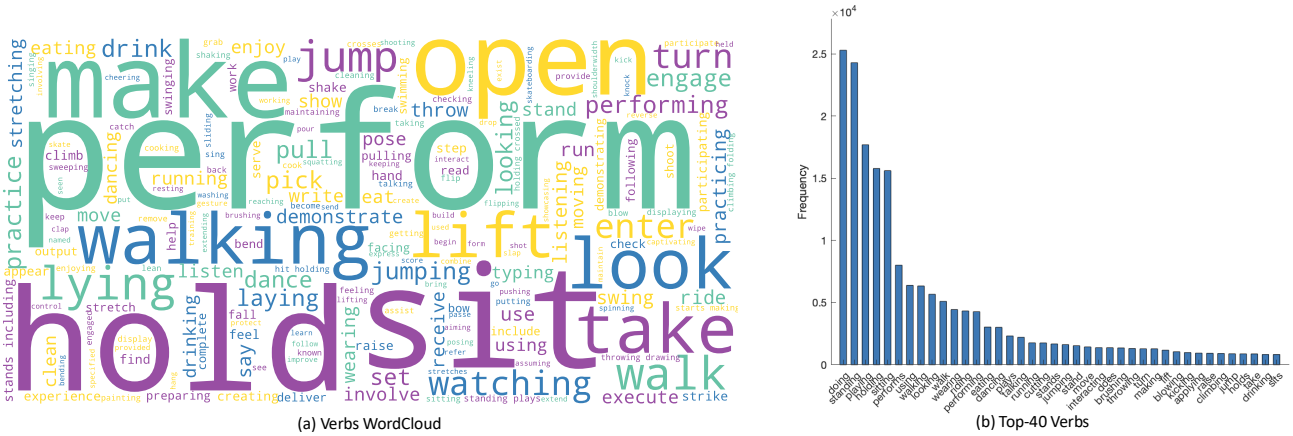


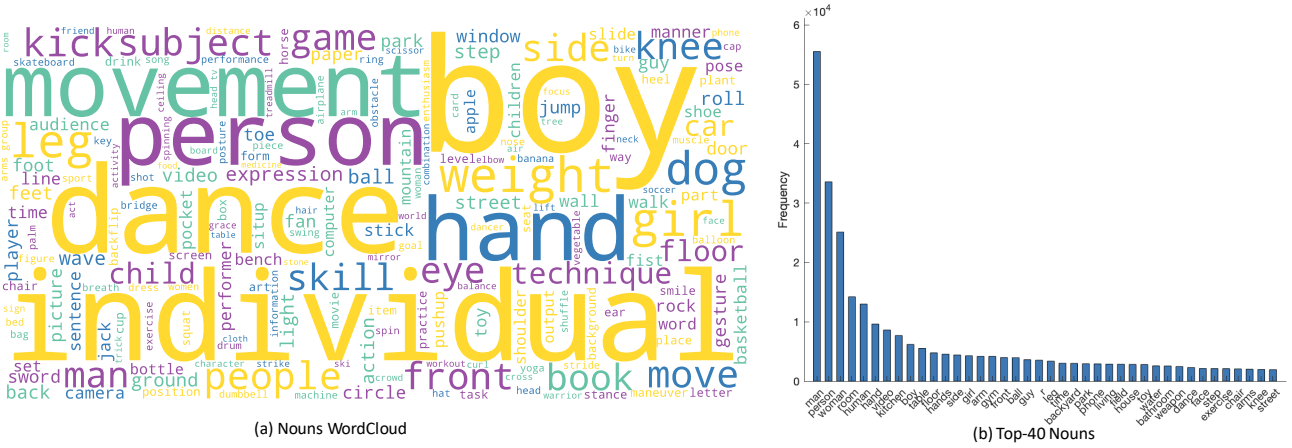
Fig. 6: Captioning Sentence Length

Fig. 7: Distribution of video length (in seconds) and captioning sentence length (in words), with the dotted line representing the average length.



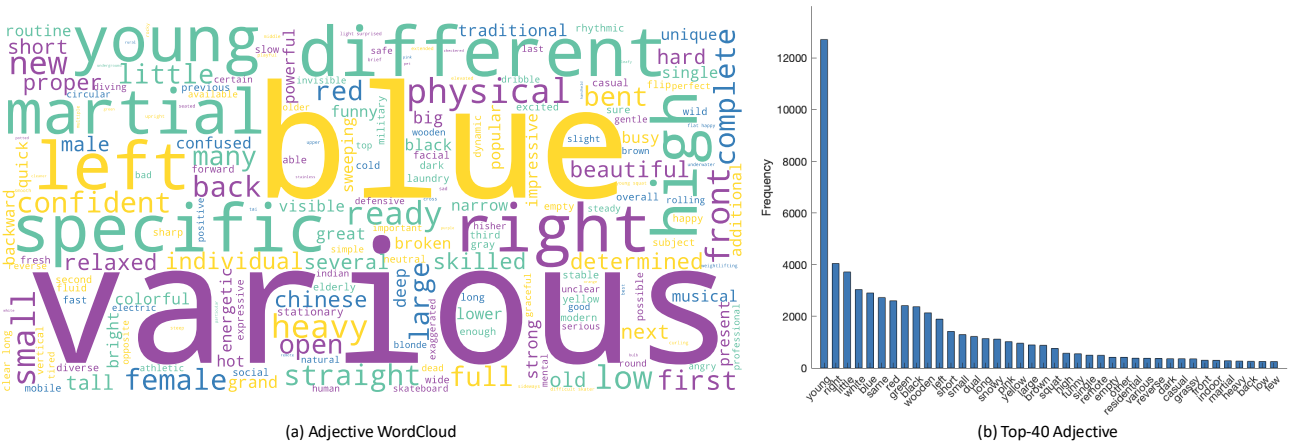
(a) Verbs WordCloud (b) Top-40 Verbs

Fig. 8: *Verbs* Word Cloud and Top-40 Frequent *Verbs*.



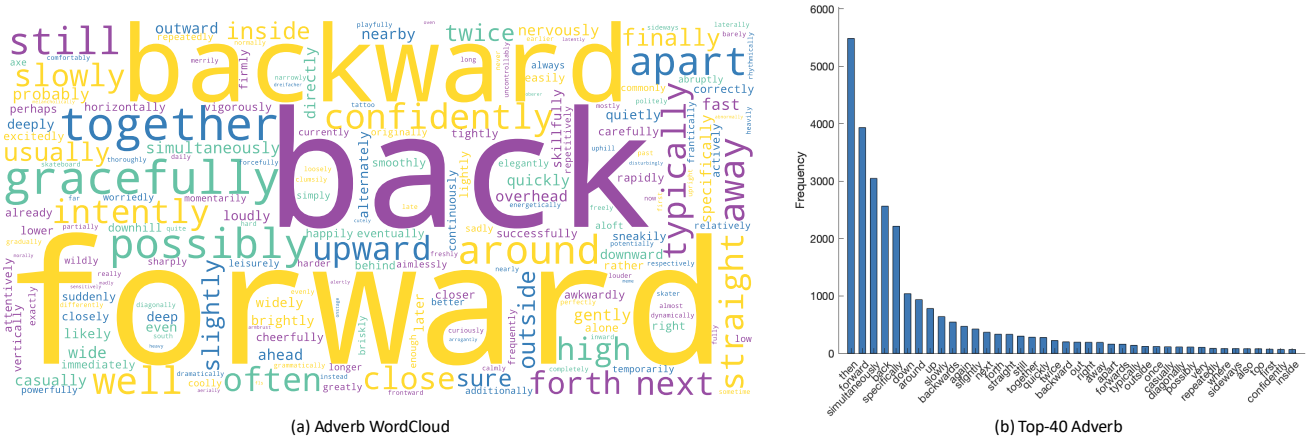
(a) Nouns WordCloud (b) Top-40 Nouns

Fig. 9: *Nouns* Word Cloud and Top-40 Frequent *Nouns*.



(a) Adjective WordCloud (b) Top-40 Adjective

Fig. 10: *Adjectives* Word Cloud and Top-40 Frequent *Adjectives*.



(a) Adverb WordCloud (b) Top-40 Adverb

Fig. 11: *Adverbs* Word Cloud and Top-40 Frequent *Adverbs*.

J. Data examples from Humanoid-X Dataset

We show visualized data examples from Humanoid-X in fig. 13, fig. 14, fig. 15, fig. 16, fig. 17, fig. 18, fig. 19, fig. 20, fig. 21, fig. 22, fig. 23, fig. 24. For motion video clips, we sample 5 frames from each video clip shown. For text, we directly present the text descriptions of the motions shown. For the human pose, we sample the SMPL visualization of the corresponding frames in the video clip. For the humanoid pose, we set the humanoid keypoints in MuJoCo and collected the MuJoCo-rendered images of the corresponding frames in the video clip. For humanoid actions, we render the humanoid control policy rollout in IsaacGym and collect the rendered images of the corresponding frames in the video clip.

K. Dataset Comparisons

We compared our Humanoid-X datasets with other human datasets in Table V. As a dataset on real-world humanoid robots, we believe Humanoid-X is unique and fundamentally different from datasets on human modeling and understanding: (1) Humanoid-X is the *first* large-scale humanoid robot datasets that contain diverse modalities not just human models. (2) Humanoid-X collects robot actions from Internet videos through an innovative pipeline, which has not been studied before. (3) Humanoid-X demonstrates the potential of large-scale humanoid robot learning, while prior works only focus on learning with small datasets.

L. UH-1 Action Tokenizer

For a given input sequence $X = [x_1, x_2, \ldots, x_T]$ with $x_t \in \mathbb{R}^{d_1}$ (representing either a humanoid keypoint or an action), UH-1 Action Tokenizer is designed to reconstruct this sequence using a learnable codebook $C = \{c_1, c_2, \ldots, c_N\}$ with $c_n \in \mathbb{R}^{d_2}$ and a learnable autoencoder with an encoder $\mathbb{E}$ and a decoder $\mathbb{D}$. In this context, T denotes the number of input frames, N the codebook size, d_1 the dimension of input tokens, and d_2 the dimension of the codes. For sequence reconstruction, the encoder $\mathbb{E}$ maps the input sequence into latent representations $Z = \mathbb{E}(X) = [z_1, z_2, \ldots, z_{T/k}]$ with $z_i \in \mathbb{R}^{d_2}$, where k represents the temporal downsampling rate of the encoder. Each z_i is subsequently quantized through the codebook into $\hat{z} \in C$ by selecting the nearest code $c_n \in C$, which can be formally expressed as:

$$\hat{z} = \underset{c_n \in C}{\arg\min} \|z_i - c_n\|_2. \tag{9}$$

Finally, the decoder $\mathbb{D}$ reconstructs the original input sequence as $X_{\text{re}} = \mathbb{D}(\hat{Z})$. The temporal downsampling enables each code in the codebook to represent k input tokens, encoding primitive humanoid motion skills and facilitating the generation of smooth actions.

In general, UH-1 Action Tokenizer is optimized by minimizing a standard objective function:

$$\mathcal{L}_{\text{vqvae}} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{embed}} + \alpha \mathcal{L}_{\text{commit}}, \tag{10}$$

$$\mathcal{L}_{\text{recon}} = \mathcal{L}_1(X, X_{\text{re}}), \tag{11}$$

$$\mathcal{L}_{\text{embed}} = \|\text{sg}[Z] - \hat{Z}\|_2, \mathcal{L}_{\text{commit}} = \|Z - \text{sg}[\hat{Z}]\|_2. \tag{12}$$

In this formulation, α is a hyperparameter that regulates the relative influence of each loss term, and sg$[\cdot]$ denotes the stop gradient operator. The embedding loss $\mathcal{L}_{\text{embed}}$ promotes the quantized codebook embeddings to move closer to the continuous output of the encoder, while $\mathcal{L}_{\text{commit}}$ encourages the encoder to commit to particular codebook entries.

Given the unique properties of humanoid keypoints and actions, we propose an adjusted reconstruction loss, $\mathcal{L}_{\text{recon}}$, which integrates a forward difference loss and a root regularization term:

$$\mathcal{L}_1(X, X_{\text{re}}) + \beta \mathcal{L}_1(\Delta[X], \Delta[X_{\text{re}}]) + \gamma \mathcal{L}_1(X_{\text{re}}^{\text{root}}, \mathbf{0}), \tag{13}$$

where β and γ are hyperparameters for balancing the additional loss components, and $\Delta[\cdot]$ represents the forward difference operator.

M. UH-1 Transformer

We formulate the language-conditioned humanoid keypoint or action generation tasks as auto-regressive prediction of the next codebook index. Formally, let $s_i \in \{1, 2, \cdots, N\} \cup \{\text{End}\}$ denote the current index to predict, $s_{1:i-1}$ represent the preceding context of indices, and l the language instruction embedding encoded by CLIP [72].

The UH-1 Transformer is then trained to model the conditional probability distribution $P(s_i|s_{1:i-1}, l)$. A special [End] token is incorporated into the indices set to signal the termination of sequence generation. For an input sequence $X = [x_1, x_2, \cdots, x_T]$, the encoder $\mathbb{E}$ and codebook C of the UH-1 Action Tokenizer map this sequence into the codebook indices as $S = [s_1, s_2, \cdots, s_{T/k}, \text{End}]$; given this sequence of indices S, it can also be mapped back to $\hat{Z} = [c_{s_1}, c_{s_2}, \cdots, c_{s_{T/k}}]$, which is subsequently projected into the output space by the decoder $\mathbb{D}$ as $X_{\text{re}} = \mathbb{D}(\hat{Z})$.

To train this transformer model, we minimize the negative log-likelihood over the training dataset $\mathcal{D}$:

$$\mathcal{L}_{\text{trans}} = -\sum_{S \in \mathcal{D}} \log \prod_{i=1}^{|S|} p(s_i|s_{1:i-1}, l). \tag{14}$$

This objective encourages accurate predictions of the next index in the context of previous indices and language instructions.

N. Implementation Details

The implementation of our model architecture follows previous work [39]. For the UH-1 Action Tokenizer, we employ a straightforward convolutional architecture consisting of 1D convolutions, residual blocks, and ReLU activation functions. Temporal downsampling and upsampling are achieved using convolutions with a stride of 2 and nearest-neighbor interpolation, respectively. The codebook size is configured as 2048×512, with a downsampling rate $k = 4$. During training, action sequences are cropped to a temporal length of $T = 64$. For the UH-1 Transformer, it is based on an 18-layer transformer model featuring 16 attention heads and a dimensionality of $1,024$.

Training the UH-1 Action Tokenizer and the UH-1 Transformer on HumanoidML3D (a selected set of Humanoid-X) requires approximately 8 hours and 30 hours, respectively, on a single NVIDIA RTX™ 6000 Ada GPU, while training on the full set of Humanoid-X requires approximately 40 hours and 400 hours, respectively.

O. Real Robot Experiment

Success Rate. The success rate of real robot pose control is evaluated using two criteria: (1) Stability: the humanoid robot must maintain stability while performing actions; any instance of falling or failing to maintain balance results in an unsuccessful trial. (2) Accuracy: the humanoid robot must accurately perform the desired actions based on text instructions. This is assessed by five human evaluators, and if the majority agree that the robot does not perform the actions correctly, the trial is considered unsuccessful.

PD Controller. The output actions a of our model are the target DoF positions for controlling the humanoid robot. We use the PD control to transform actions a into motor torques τ, which can be represented as

$$\tau = K_p(a - q) - K_d dq, \tag{15}$$

where K_p and K_d are the proportional coefficients of the motor position and speed errors respectively, q is the current angle position of the motor rotor, and dq is the current rotor angular velocity of the motor rotor. We use the standard K_p and K_d provided in the official robot documents in our experiments.

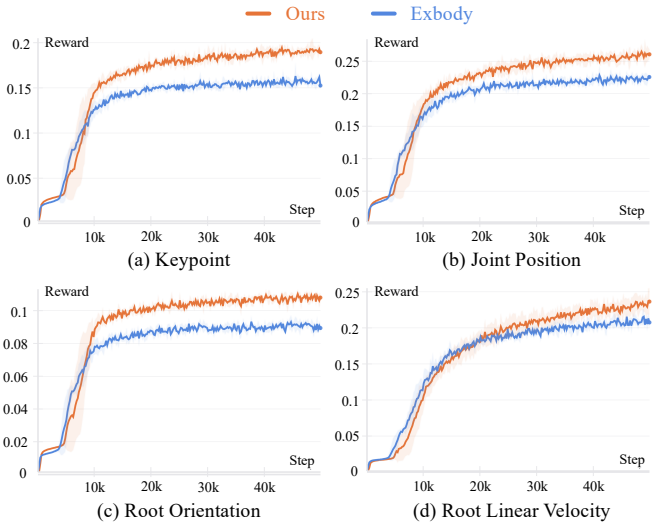


Fig. 12: **Ablation on different RL policies**, measured by task cumulative reward value. The solid line represents the mean return value, while the shaded regions correspond to the standard deviation, both calculated across five different random seeds. Our retargeted training data enhances the performance of the RL policy in tracking the imitation of body keypoints, joint positions, root orientation and root linear velocity.

Real Robot Experiments. We demonstrate the real humanoid robot pose control with text instructions in fig. 25, fig. 26, fig. 27, fig. 28, fig. 29, fig. 30, fig. 31. We also demonstrate human-humanoid interactions in fig. 32 and fig. 33. From these figures, we show that our method generates accurate and diverse poses to control the real humanoid robot with text instructions.

P. Ablation on Goal-conditioned Control Policy

To investigate the impact of humanoid keypoints on the goal-conditioned RL policy, we compare our motion retargeting approach, originated from [23], with another approach in [21]. We evaluate the quality of the humanoid keypoints generated by different motion retargeting methods by measuring the tracking rewards in the subsequent reinforcement learning step, maintaining other factors as the same. As illustrated in fig. 12, we launch experiments in five random seeds for both methods. We empirically found that our motion retargeting method improves the performance of the RL policy on the evaluation metrics in [21] tracking the imitation of body keypoints, joint positions, root orientation and root linear velocity in the form of training rewards. The results show that our retargeted data enhances the performance of the RL policy, thus suggesting that our retargeting method can generate humanoid pose data more executable for humanoid robots.

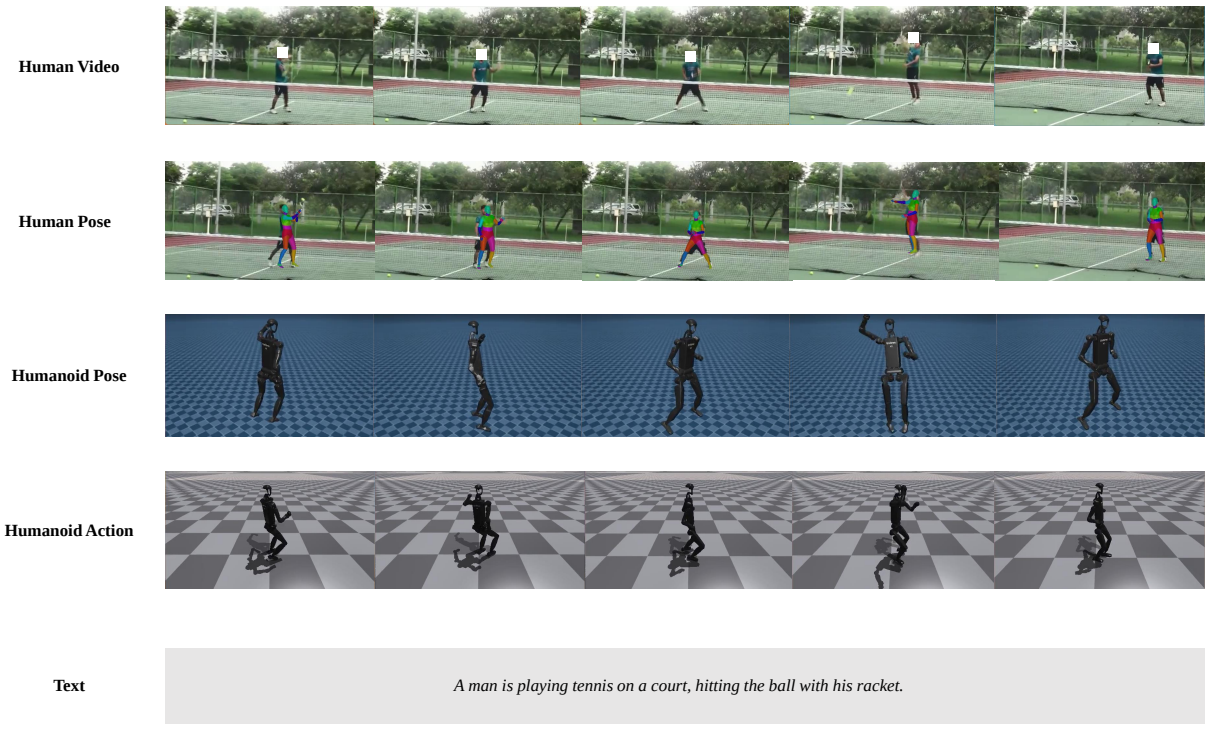


Fig. 13: **Data samples in Humanoid-X**.

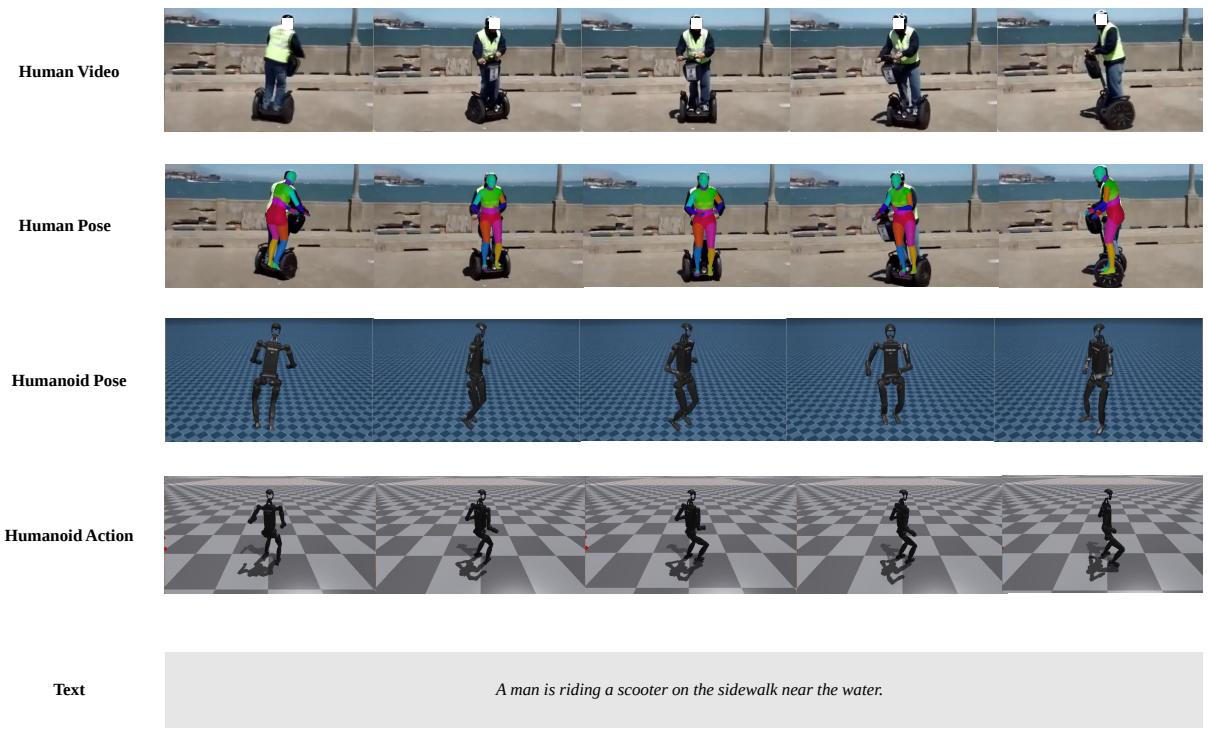


Fig. 14: **Data examples in Humanoid-X**.

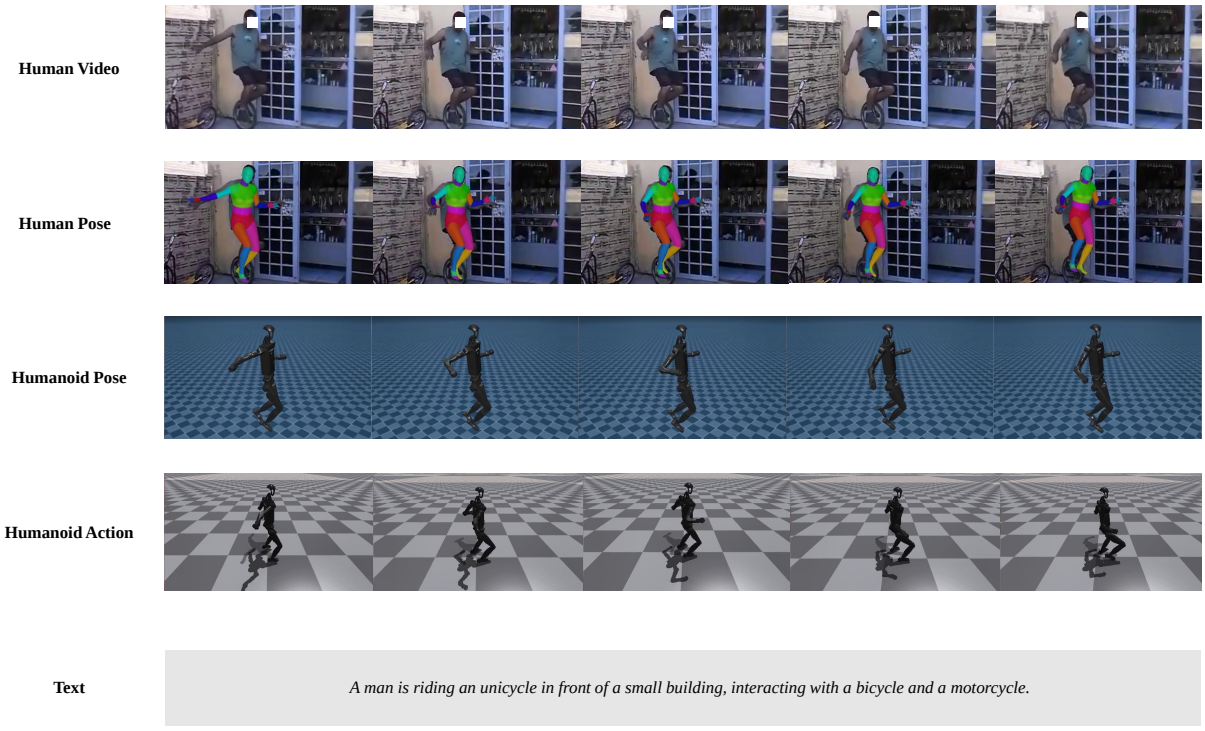


Fig. 15: **Data examples in Humanoid-X**.

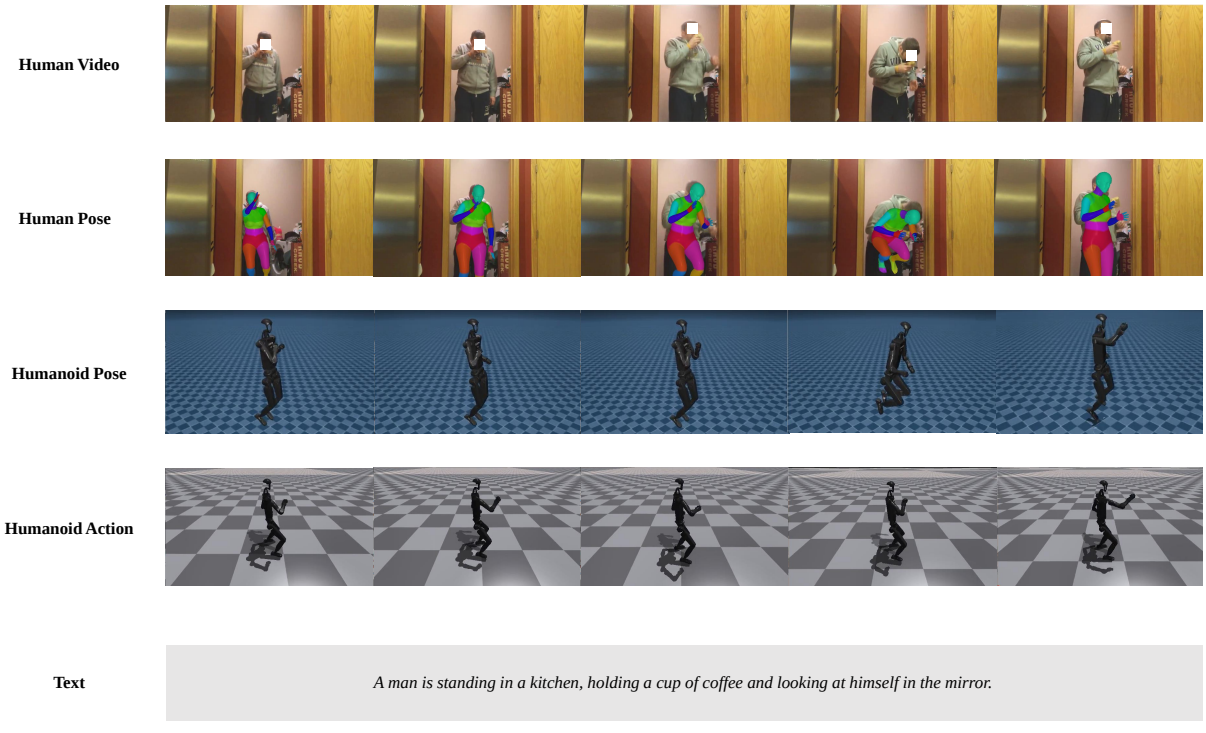


Fig. 16: **Data examples in Humanoid-X**.

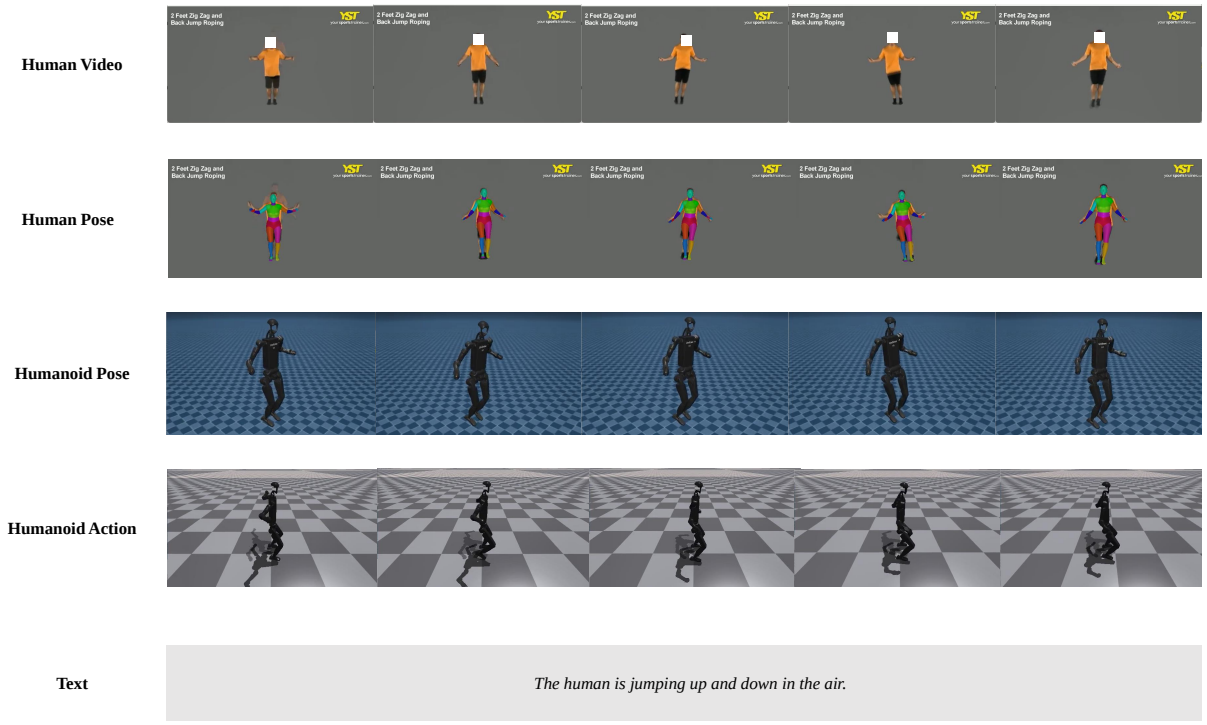


Fig. 17: **Data examples in Humanoid-X**.

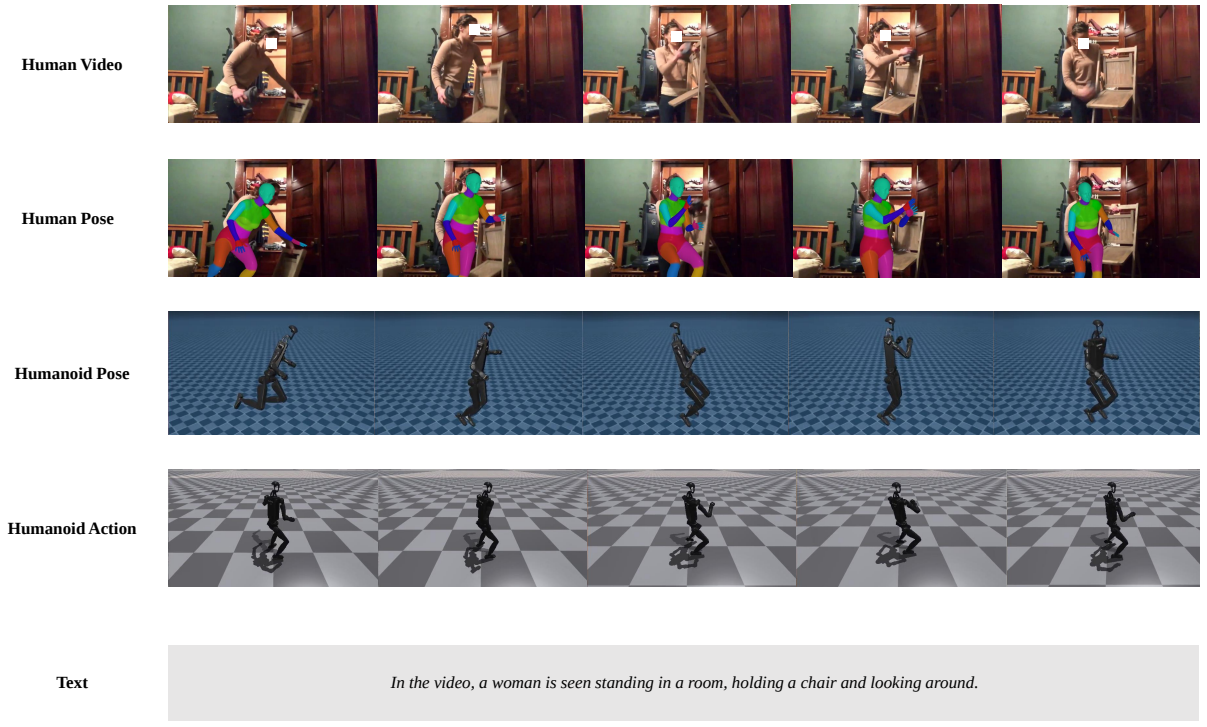


Fig. 18: **Data examples in Humanoid-X**.

Fig. 19: **Data examples in Humanoid-X**.

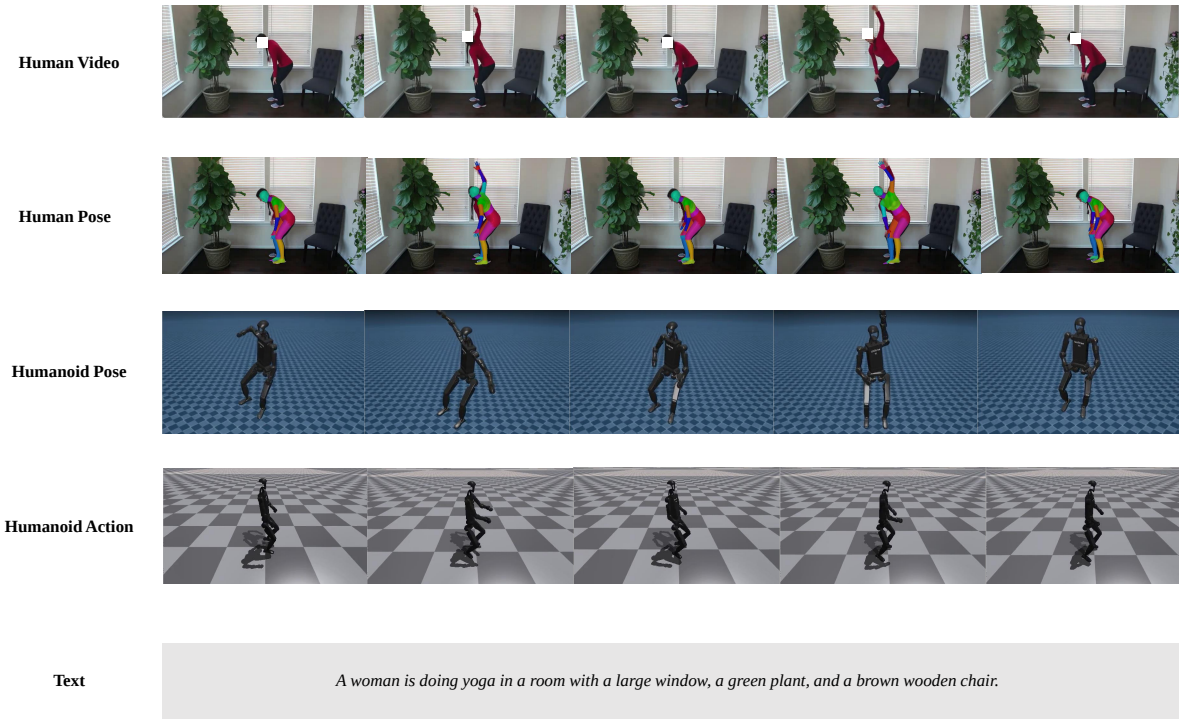


Fig. 20: **Data examples in Humanoid-X**.

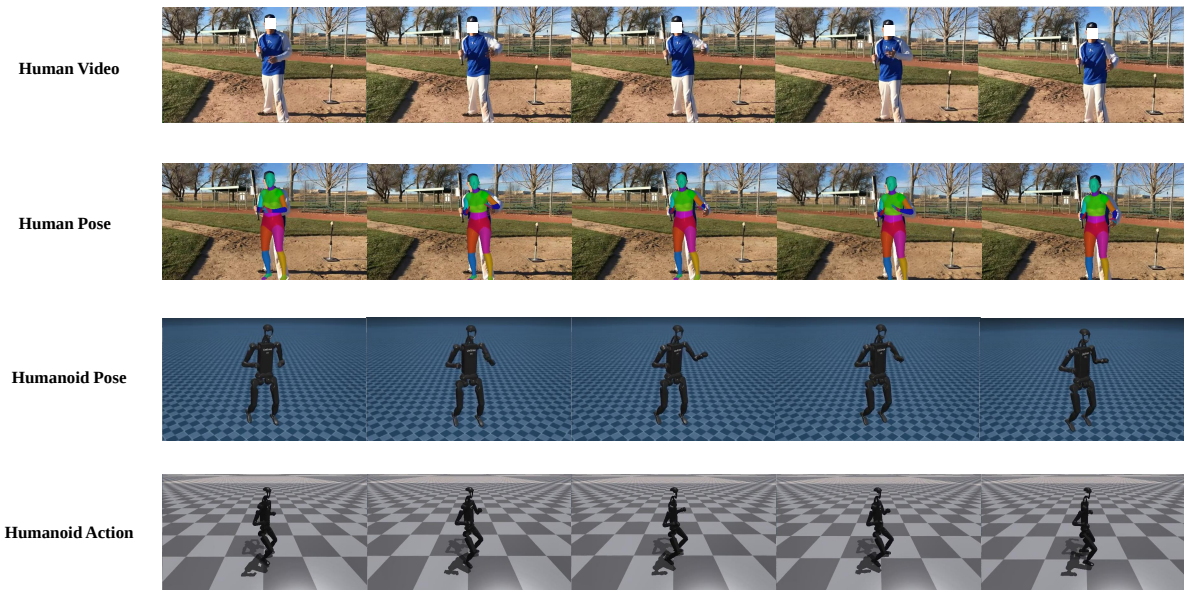


Fig. 21: **Data examples in Humanoid-X**.

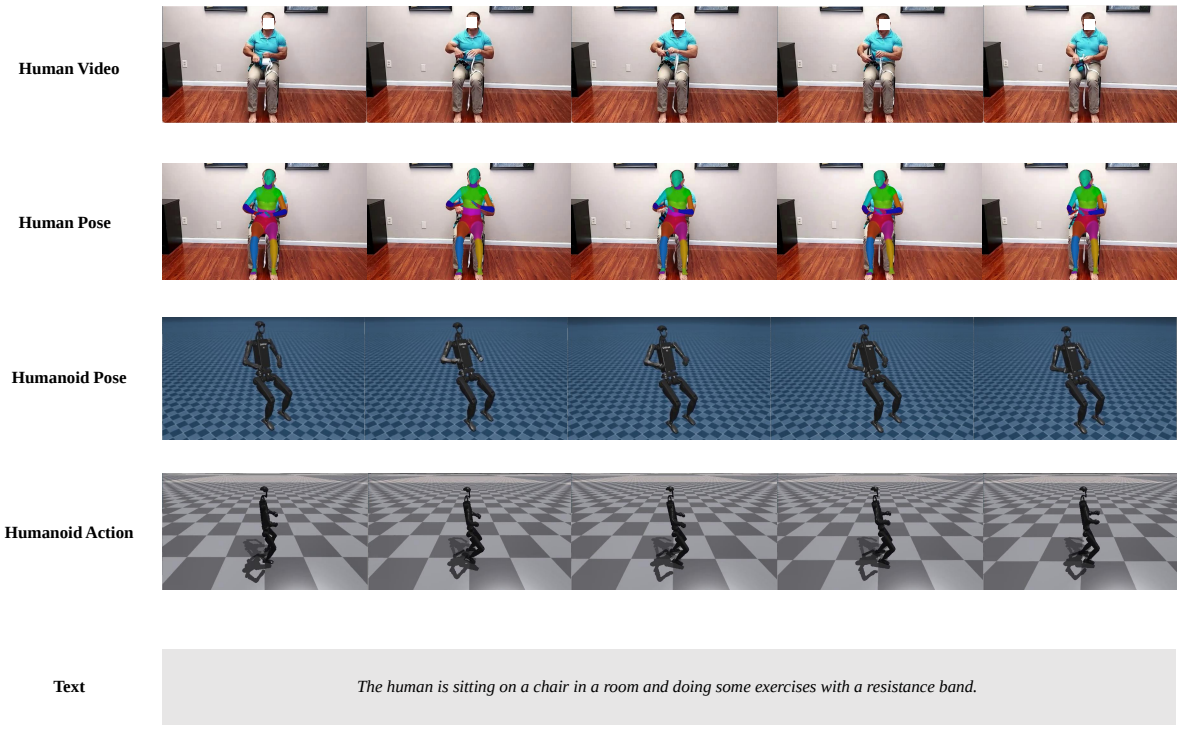


Fig. 22: **Data examples in Humanoid-X**.

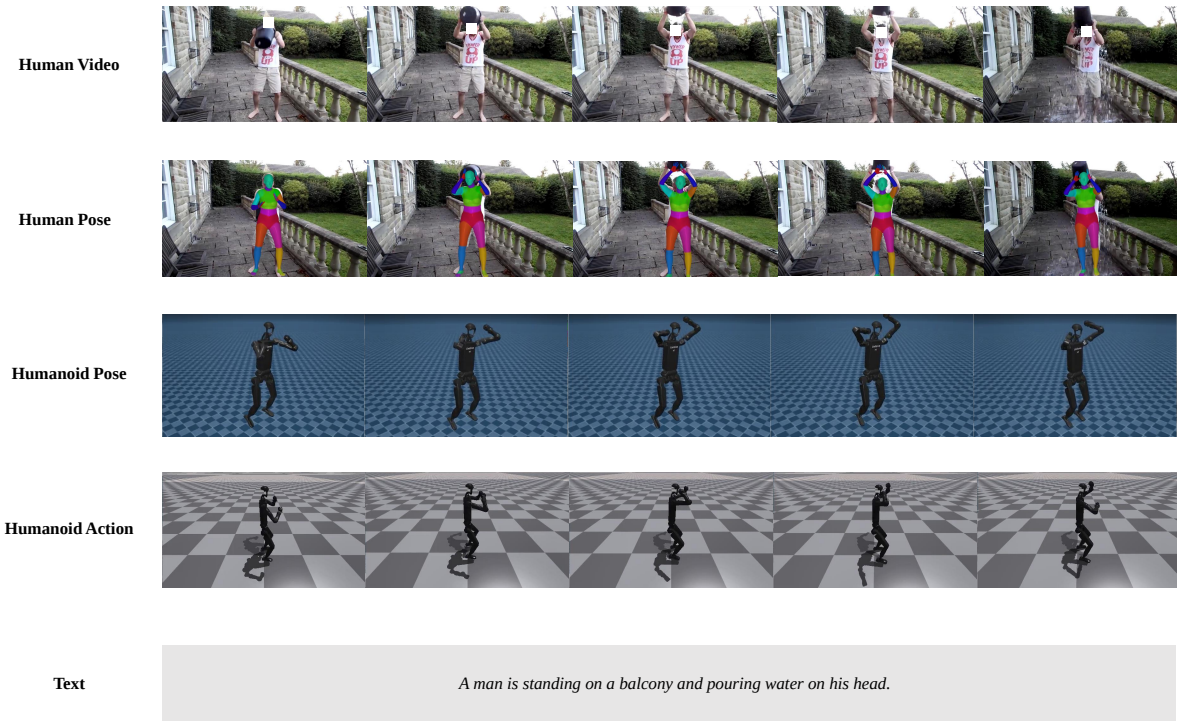


Fig. 23: **Data examples in Humanoid-X**.

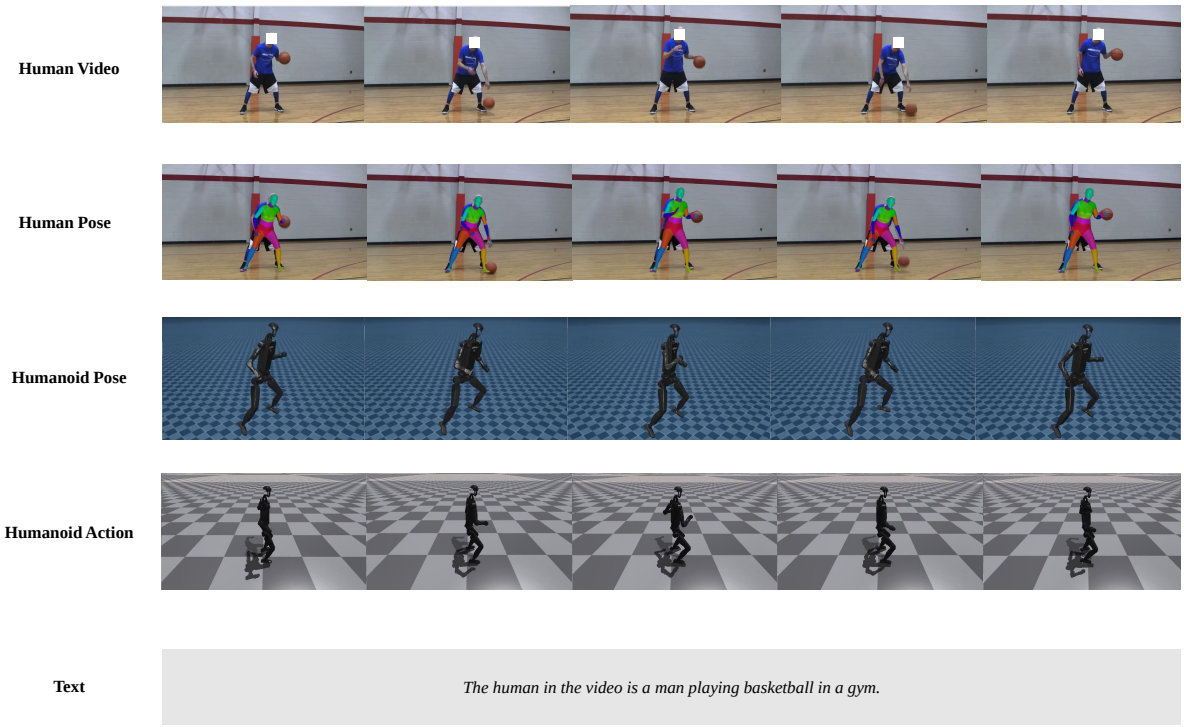


Fig. 24: **Data examples in Humanoid-X**.

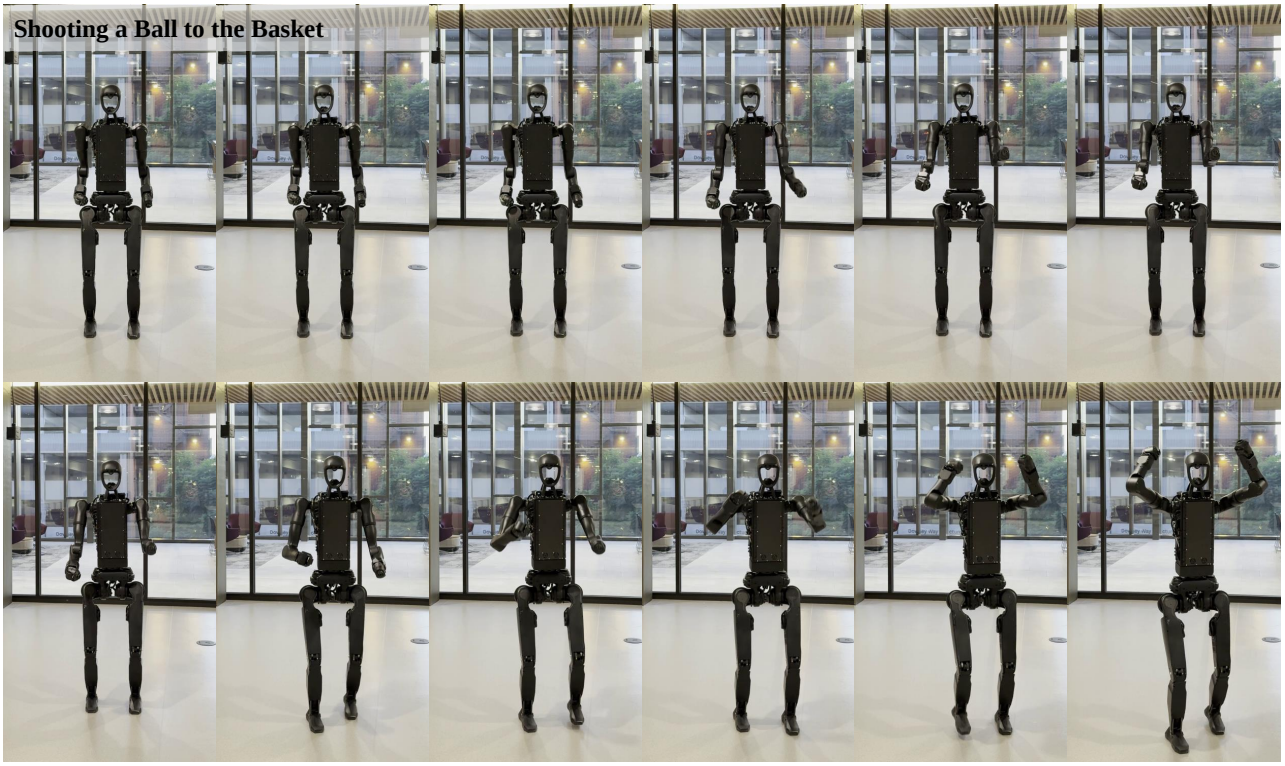


Fig. 25: **Real robot demonstrations. Text instruction:** ***Shooting a Ball to the Basket***.

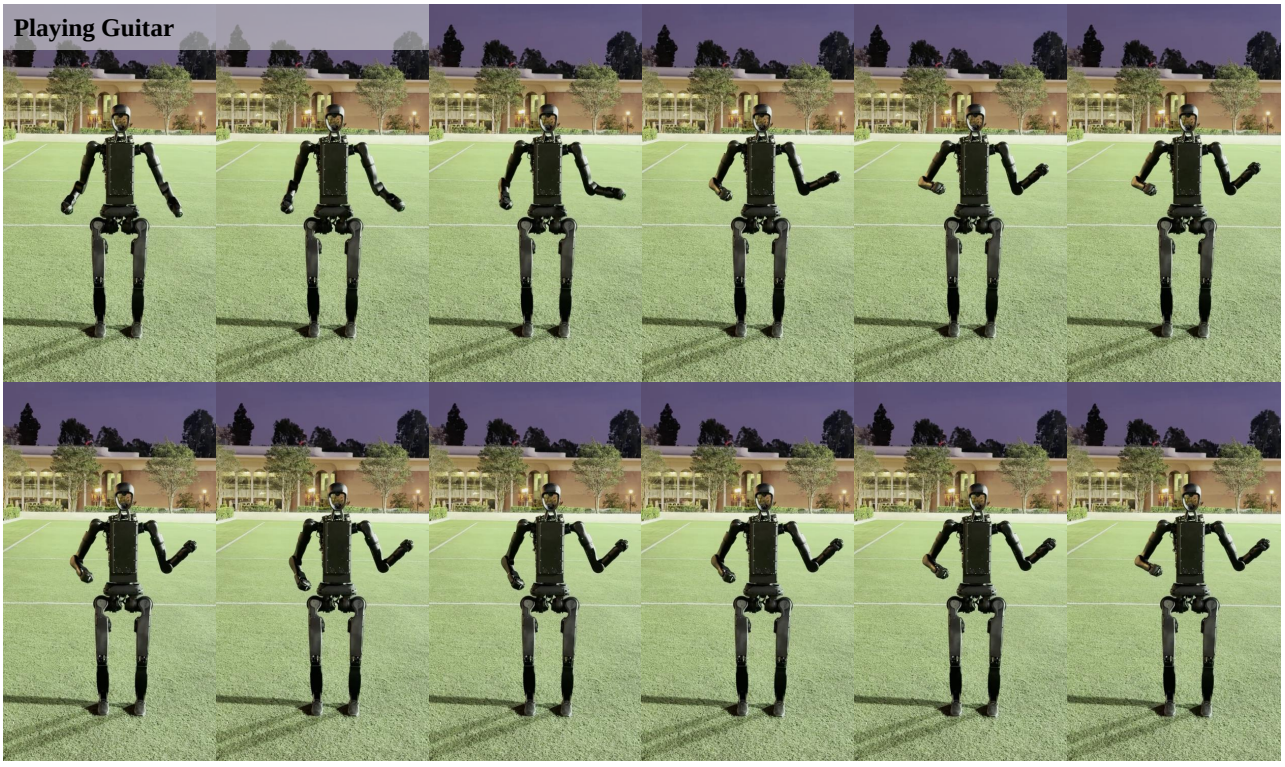


Fig. 26: **Real robot demonstrations. Text instruction:** ***Playing Guitar***.

Fig. 27: **Real robot demonstrations. Text instruction:** ***Putting in a Golf Tournament***.

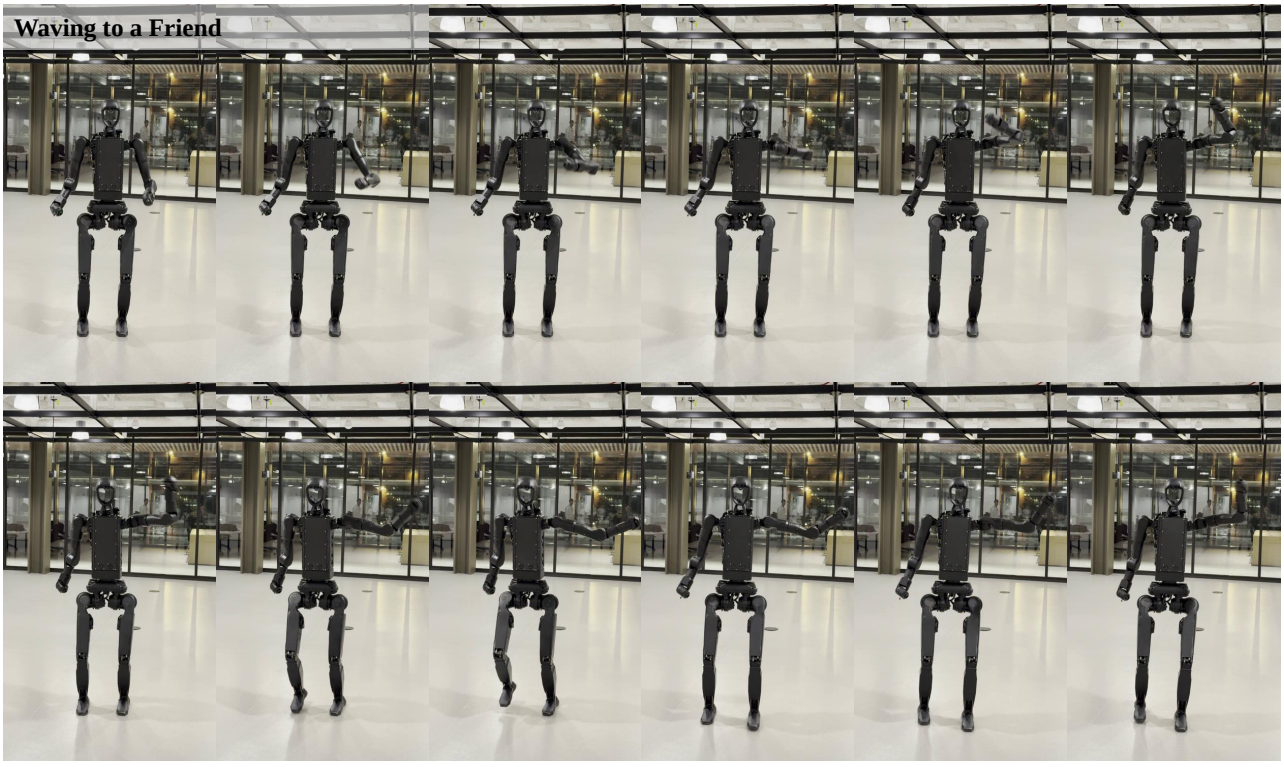


Fig. 28: **Real robot demonstrations. Text instruction:** ***Waving to a Friend***.

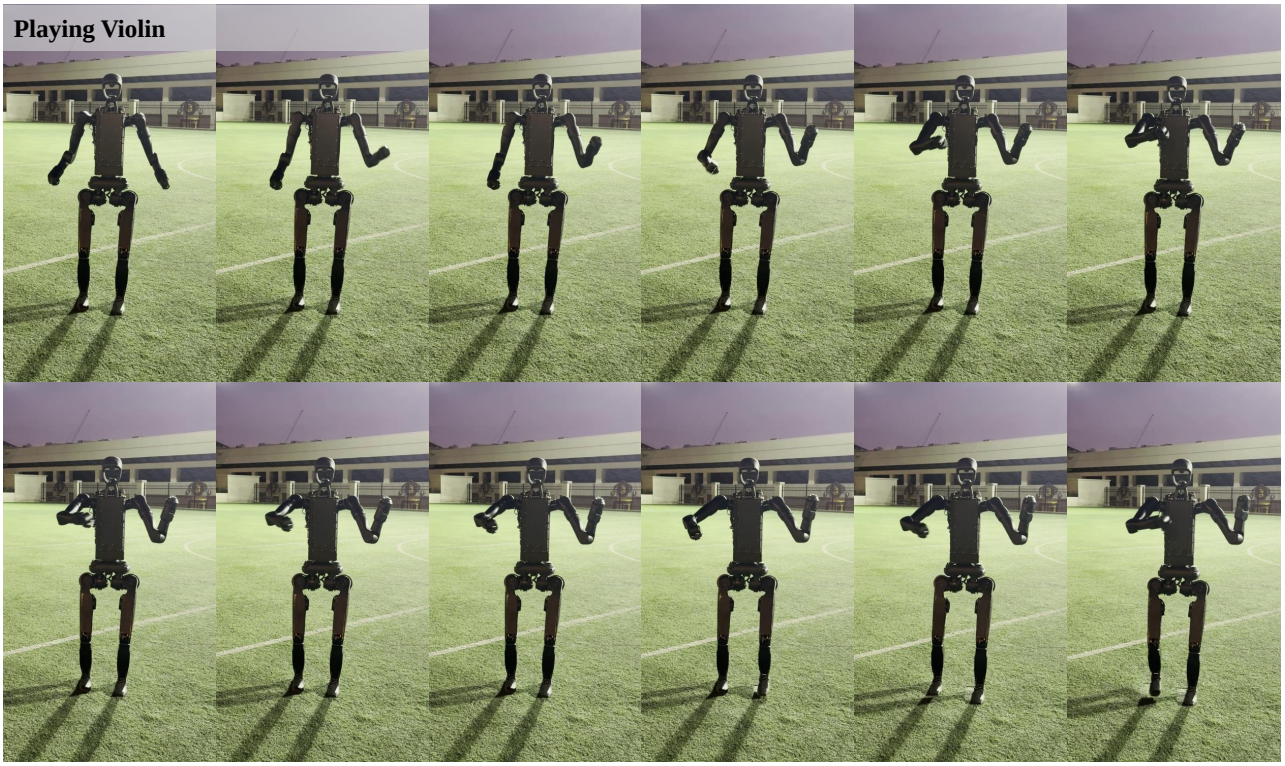


Fig. 29: **Real robot demonstrations. Text instruction:** ***Playing Violin***.

Fig. 30: **Real robot demonstrations. Text instruction:** ***Punching with the Right Hand***.

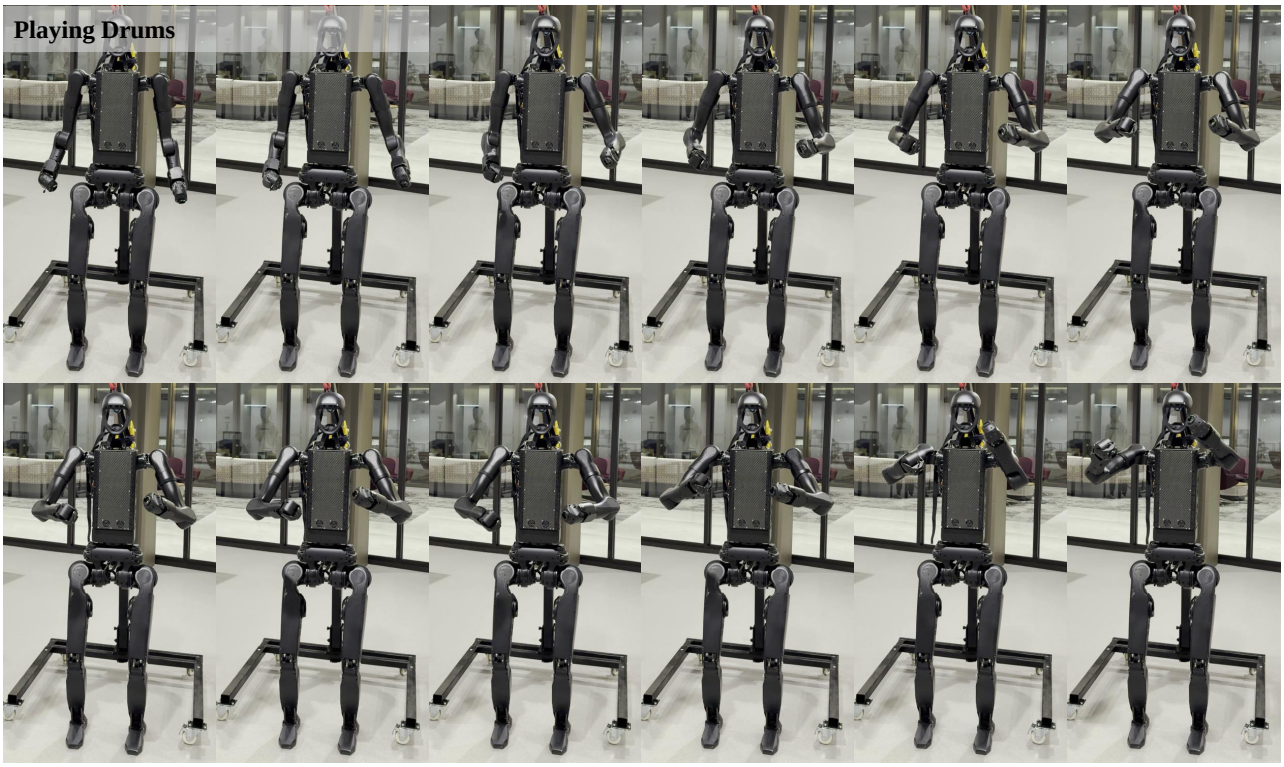


Fig. 31: **Real robot demonstrations. Text instruction: *Playing Drums*.**

Fig. 32: **Demonstration of human-humanoid interactions. Text instruction: *High-Five*.**

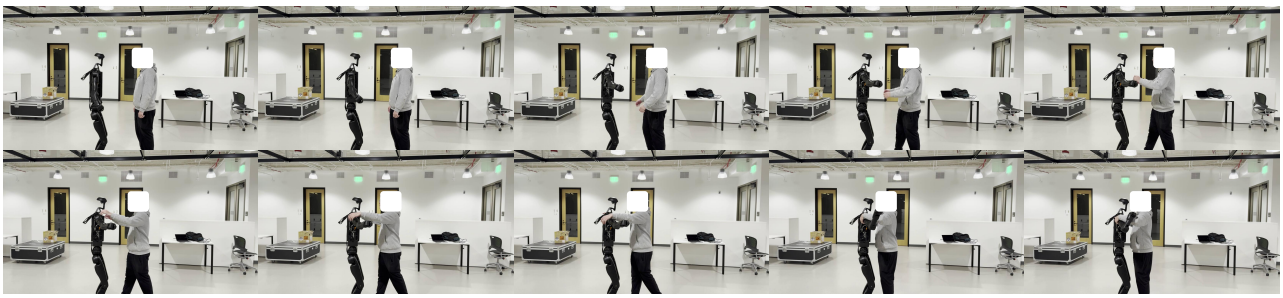


Fig. 33: **Demonstration of human-humanoid interactions. Text instruction: *Embrace*.**