IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 February 2025, accepted 2 May 2025, date of publication 12 May 2025, date of current version 19 May 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3568784

RESEARCH ARTICLE

Data-Efficient Approach to Humanoid Control by Fine-Tuning a Pre-Trained GPT on Action Data

SIDDHARTH PADMANABHAN[1], KAZUKI MIYAZAWA[1], TAKATO HORII[1], (Member, IEEE), AND TAKAYUKI NAGAI[1,2], (Member, IEEE)

[1]Graduate School of Engineering Science, Osaka University, Toyonaka-shi, Osaka 560-8531, Japan
[2]AIX, The University of Electro-Communications, Chofu-shi, Tokyo 182-8585, Japan

Corresponding author: Siddharth Padmanabhan (s.padmanabhan@rlg.sys.es.osaka-u.ac.jp)

This work was supported by JST Moonshot Research and Development under Grant JPMJMS2011.

ABSTRACT Recent advances in imitation learning have enabled robots to learn multiple tasks from large-scale datasets. However, developing a model for multi-tasking humanoid control faces significant challenges. Human kinematic data is available in open-source datasets for humanoid motion learning, but learning policies from this data requires simulation due to the lack of actions. While dynamic data can accelerate learning via supervision, datasets typically lack substantial amounts of such action labels, that are also difficult to be directly used for training due to the unique structure of each human/humanoid systems. In this study, we pre-trained a Generative Pre-trained Transformer (GPT) based model on expert policy-rollout observations only (without actions) from a humanoid motion dataset. Upon fine-tuning on a smaller dataset with both observations and action labels, we demonstrate that our GPT-based model can predict actions to achieve human-like movements faster in training than training a GPT on the entire dataset from scratch directly. Furthermore, performance evaluation based on motion generation across various behaviors showed that our approach achieves efficient learning comparable to baselines.

INDEX TERMS GPT, humanoid, imitation learning, motion prediction, whole-body control.

I. INTRODUCTION

Humanoid whole body control is becoming an increasingly interesting domain for enabling humanoids to be platforms for testing powerful deep learning models such as transformers for whole body control. However, for any character or robot that operates in continuous domain, reinforcement learning is time consuming and difficult for multiple reasons; designing reward statement is usually task-specific, character/robot specific and/or possibly simulator specific, and the online interaction between the training model and the simulator consumes time. Imitation learning has come a long way, and many papers have shown the efficacy of using motion capture data as reference data to train humanoids to learn natural human-like movements [1], [2], [3]. Including action data along with kinematic data in training via supervision can significantly speed up the learning by avoiding training against reward feedback and interacting online in a simulator. Although many datasets provide copious amounts of kinematic data, few provide action data. It is quite difficult to obtain action data for humanoids due to limited practicality. For instance, neither is it easy to obtain action data from motion capture nor from whole-body teleoperation of humanoids due to embodiment gap and system latency [4], [5], [6]. It is particularly difficult to generalize motion for humanoids on a variety of tasks, either due to the complex nature of the skeleton structure and/or the task. Expressing a single controller that can perform multiple movements is holy grail for whole-body control of humanoids. Till now, there is limited research focused on leveraging limited amounts of action data coupled with kinematic data to learn a multi-tasking controller for humanoids.

On the other hand, research is progressing on foundation models that can be adapted to multiple tasks [7]. For instance, in the field of natural language processing, a single model can perform multiple tasks by self-supervised pre-training on a large amount of textual data [8]. Similarly, in humanoid

The associate editor coordinating the review of this manuscript and approving it for publication was Yu-Huei Cheng.

control, if we can create a foundation model that can adapt to multiple tasks, we can improve humanoid adaptability.

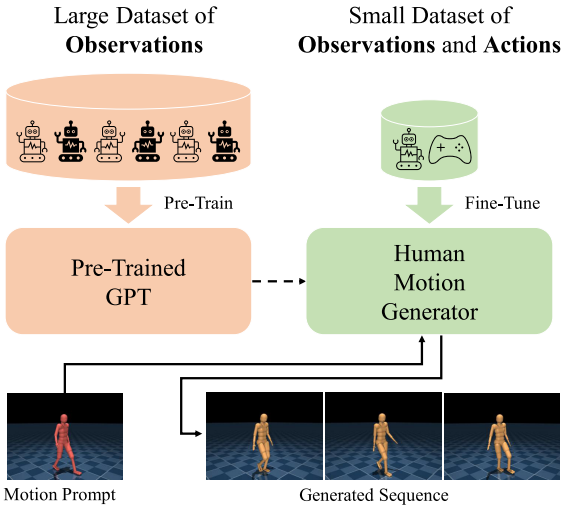


FIGURE 1. Proposed approach overview: A GPT-based policy is trained for motion generation. The policy generates physically plausible motions in simulation through autoregressive prediction. The training process has two stages: pre-training on a large dataset of observations only, followed by fine-tuning on a small dataset containing both observations and actions.

In this study, we explore the potential of using a pre-trained motion foundation model, originally trained on non-physics data, and fine-tuning it on a smaller physics-based dataset. This approach allows us to avoid the extensive training time typically required for large multi-task models in an online simulator environment. Instead, we utilize a substantial dataset comprised of observations and actions derived from policy rollouts across multiple tasks. This facilitates data-efficient imitation learning.

We hypothesize that by using a GPT-based motion pre-trained model, we would only have to fine tune this model on a smaller dataset, which we refer to as a Human Motion Generator (HMG), to plan physically plausible trajectories for humanoid motion, therefore significantly reducing the training time and dataset size (Fig. 1).

The application of this approach in humanoid control for multi-tasking is both novel and necessary, since existing methods often require large amounts of task-specific data to achieve generalization. By leveraging a foundation model approach, we aim to improve data efficiency by enabling knowledge transfer from a pre-trained model to a control downstream task, therefore reducing the need for extensive training with action labels. In this paper we:

1) propose a foundation model training approach for humanoid control,
2) we compare the data efficiency of our proposed method with baseline methods in our evaluations, and
3) perform a comprehensive evaluation between our proposed model HMG and models trained from scratch by comparing the performances based on motion prediction metrics, trajectory generation lengths, empirical analysis of humanoid motion.

II. RELATED WORK

A. USING PRE-TRAINED KNOWLEDGE

Generalization of human motion prediction is a difficult problem due to several reasons, such as the varying skeletal structure and idiosyncrasies in the human motion data. To tackle this issue, in [9] a model is trained through a curriculum and continual learning manner, such that a model can be first trained on a diverse dataset to be robust and then fine tuned to predict motion of new subjects. In [10], a two stage training is implemented; a pre-training stage where 3-D motion is derived from noisy partial 2-D observation, then a fine tuning stage where the model is fine-tuned to solve downstream tasks such as 3-D pose estimation, action recognition, and mesh recovery. There is a similar implementation in [11], where initially the humanoid is trained to walk in a fully observed condition (having access to all sensory information from the humanoid and external environmental information that is difficult to obtain) and then this policy is distilled to another policy trained in a partially observed condition. However, in these publications, models were not trained on action data from physics simulation, rather just on kinematic data. Moreover, these publications have not investigated the change in data-efficiency in fine-tuning pre-trained models. Pre-training a model and then fine-tuning that model on downstream tasks is a common practice adopted in NLP [12], [13], and in human motion prediction for human-robot interaction. In this work, a GPT is pre-trained on observation data, resulting in a generalized representation of humanoid kinematics over a variety of of behaviors, and later fine tuned on action data that can control the humanoid in a physics simulator.

B. TRANSFORMER BASED MODELS FOR HUMANOID/ROBOT CONTROL

Transformers have been shown to be powerful to generate human motion [11], [14]. In [15], the researchers use the idea of dual attention to capture spatial and temporal dependencies of known data without relying on hidden states in RNNs or temporal encodings like Discrete Cosine Transformation. This model effectively generates poses that are temporally coherent. MotionGPT fuses language and motion to enhance performance of motion-related tasks such as text-to-motion, motion generation, and motion in-betweening [16]. In [17], the researchers also demonstrates GPT's capability to generate motion in a physics engine after taking in one second long motion prompts. In this paper, we train a GPT policy to control a humanoid, similar to what was done in [17], but by employing a more data-efficient training strategy.

C. PREDICTION MODELS FOR HUMANOID

Many papers have focused on optimizing human motion prediction by using various deep learning techniques [18], [19], [20]. Instead of using memory-based networks like RNN or Transformers, a simple feed-forward network with

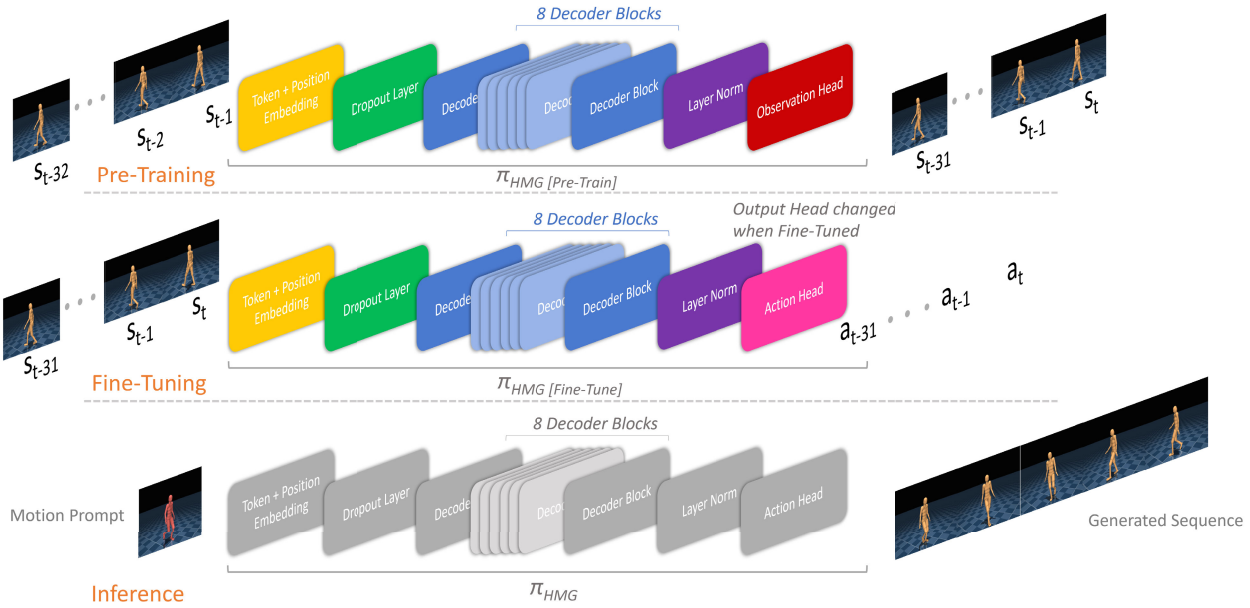


FIGURE 2. Detailed Proposed Approach of HMG: On top is the pre-training phase where a GPT is trained on a large observation dataset consisting of only observations; in the middle is the fine tuning phase, where the same GPT weights are used except the observation head which is replaced with an untrained action head to output actions, and the pre-trained model is fine-tuned on a small dataset consisting of both observations and actions; the bottom shows the inference of the resulting HMG. The GPT weights in gray depict that they are frozen, and the GPT weights in other colors denote that they are trainable.

fully connected layers, layer normalization, and transpose operations, can be used to generate human motion using spatial and temporal information [21]. Similarly considering the idea that human motion prediction depends on spatial and temporal information, by using a scene image, the past body poses of human, and the past 2-D locations as contexts, future 3-D poses and 3-D locations can be predicted [22]. FrankMocap [23] proposes a modular approach where regression is performed for face, hands, and body individually and then integrated later to produce a whole body pose output. In [24], researchers propose DLow, a sampling strategy to obtain diverse set of samples from a trained generative model. This sampling strategy serves to tackle two problems; lack of diversity and inability to cover minor nodes in the data distribution. Here, we use GPT for motion generation since it is powerful for predicting long sequences of structured data given a small context.

D. DATASETS

There are many popular datasets for human motion capture data available to the public [25], [26], [27], but not many have simulation data consisting of control outputs coupled with observational data. In this paper we use both the large and small versions of the MoCapAct dataset [17], to train and evaluate our proposed model HMG and other GPTs for comparative evaluation. MoCapAct dataset is a dataset available to the public that consists of several rollouts of states and actions of the MuJoCo ball-joint humanoid played in the MuJoCo simulator. There are two versions of this dataset, a large one that has around 580 hours of motion data and a small one that has around 49 hours of motion data (roughly one-tenth of the large dataset), and both contain the same clips. In [17], a GPT policy was trained from scratch whereas we show a more data-efficient approach to train a GPT via pre-training and fine-tuning, while also obtaining similar, if not, better motion generation capability.

III. METHODOLOGY

The training methodology implemented and architecture are shown in Fig. 2. The first step is to train the humanoid motion foundation model. This serves as a pre-training phase. The motion foundation model is a minGPT [28] with model size 57M parameters. We used the same model architecture, size and the relevant training hyperparameters for motion completion from [17] since this model was trained on the MoCapAct dataset. The motion foundation model is trained only on the observations taken from the large version of the MoCapAct dataset, i.e, the input and output were observations. This dataset contains 100 rollouts of each motion behavior. After the pre-training phase, this foundation model is fine-tuned by loading the previously obtained weights, and replacing the final linear feed-forward layer with a new learnable layer suited to output actions. Using both observations and actions taken from the small version of the MoCapAct dataset the foundation model is fine-tuned to realize the physics of the behaviors. This smaller dataset contains 10 rollouts of each motion behavior. The observation data is normalized between -1 to 1 in both the training stages. The action range in fine-tuning stage is between -1 to 1. The fine-tuning method used

in this work is basic and commonly implemented. Later in the Experiment Section, we show in our evaluations of fine-tuning our pre-trained model on different dataset sizes and show that our proposed model surpasses models trained from scratch quicker with significantly smaller dataset size.

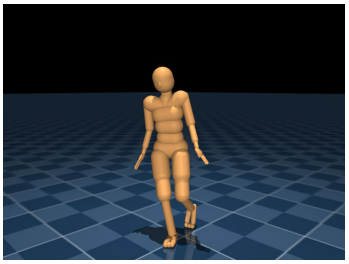


FIGURE 3. MuJoCo humanoid from dm_control package in MuJoCo simulator.

The weights of the pre-trained model were updated using cross-entropy loss during training, while Mean Square Error (MSE) was used as an evaluation metric during validation. The losses have a minor change during the fine-tuning process where the losses are calculated between actions but not observations. The cross entropy loss is defined below

$$L_{ce} = \sum_{i=1}^{N} \hat{\tau}_i \log(\tau_i), \tag{1}$$

where $\hat{\tau}_i$ can be either generated observation $\hat{s}_t$ or generated action $\hat{a}_t$ depending on whether the model is trained in pre-training stage or fine-tuning stage, and τ_i denotes either real observations s_t or actions a_t. N is the output size, therefore if τ_i is action then N is the number of actions and if τ_i is observation then N is the number of observations. The observable quantities used in training the pre-trained model and HMG is given in Table 1. The mode of control is positional control and the output actions are joint positions. Just like in [17], the simulator used in this work to test our motion prediction policy is MuJoCo [29] and the humanoid used in the experiments is a standard MuJoCo humanoid from the dm_control package that has 56 DoF (Fig. 3). The architecture, datasets, and hyperparameters used are the same as those used in [17]. The GPT policy consists of eight decoder layers and each layer has eight attention heads. The learning rate for pre-training is 3×10^{-6}. It is common practice in deep learning to fix a learning rate for fine-tuning 10-100 times lower than that of the pre-training learning rate to assure stability in training and avoid drastic changes in the network features, hence the learning rate used during fine-tuning was reduced to 3×10^{-7} and was empirically observed to work well. During inference, the weights of the resulting policy are frozen. A motion prompt of 32 steps of a particular expert behavior is provided to the policy at the initial step and then the policy auto-regressively generates the motion using the feedback from the simulator.

IV. EXPERIMENT

In our experiments, we compare and evaluate our proposed HMG with Scratch-Small and Scratch-Large, where *Small* and *Large* denotes the size of the dataset the models are trained on.

The datasets used are the publicly available large and small versions of MoCapAct dataset. These datasets contains noisy observation and action data collected from expert policies. The pre-trained model is trained on a large dataset of only observations and then it is fine tuned on a smaller dataset consisting of both observations and actions. The time taken for pre-training is around 52 hours (2M steps), and the time taken for fine-tuning is around 12 hours (400K steps), thus the fine tuning phase takes almost a quarter of the time taken to train the motion foundation model. Each model was assigned to four NVIDIA A100 GPUs for training.

We trained and tested three models: our proposed model HMG, Scratch-Large, and Scratch-Small, and performed a comprehensive comparative evaluation between them. The differences between these models in terms of training configuration is given in Table 2. HMG is our proposed foundation model fine-tuned on a motion prediction downstream task. As previously stated, we chose two GPTs, namely Scratch-Large and Scratch-Small, trained on the large and small versions of the MoCapAct dataset respectively from scratch, to compare their performances with the performance of our model trained by using our proposed method on motion completion. Motion completion can be defined as the generation of motion based on a motion prompt input of a given length. The length of motion prompt used in this work is 32 steps that amounts to one second. The weights for Scratch-Large are publicly available and Scratch-Small was trained at our facility. We could see that although our proposed model and Scratch-Large possess similar capabilities, our model still slightly outperforms the latter.

The following subsections delineates the models' evaluations, where we see how data-efficient the various training approaches are, and how well the models are capable of generating behaviors via prediction lengths, empirical differences in behaviors, and motion prediction metrics to evaluate motion quality, motion similarity to the ground truth, and motion diversity.

A. DATA EFFICIENCY

Fig. 4 compares the average episode lengths generated by each model for different dataset sizes. Our proposed model was fine-tuned on datasets of quarter, half, and full size of the *Small* dataset. The performances were compared based on episode lengths. We define an episode length as the total length of an episode for the humanoid to complete the task before episode termination (when it falls down) or the maximum episode length (set to about 15 seconds). From Fig. 4, we can see that HMG quickly generates higher average prediction lengths on the validation datasets after being trained on the *Small* dataset, compared to Scratch-Large

TABLE 1. List of observables and actions taken from MoCapAct dataset.

Observables	Description
Joint Pose	joint angles of each DoF in radians
Velocimeter	root velocity in Cartesian XYZ directions
Gyrometer	root orientation in Cartesian XYZ directions
End Effector Pose	end effector orientation
World Z Axis	direction of Cartesian Z axis
Actuator Activation	boolean values on whether actuators are active or not
Touch Sensors	contact forces between humanoid and ground
Torque Sensors	joint torque of each DoF
Body Height	root height from ground

Actions	Description
Joint Pose	joint angles of each DoF in radians
Control Mode	position control mode

TABLE 2. Models used for evaluation.

	Dataset		Training		
Model	Large Dataset	Small Dataset	Pre-Trained	Fine-Tuned	Trained from scratch
HMG	✓(Pre-Training Only)	✓(Fine-Tuning Only)	✓(2M steps)	✓(400K steps)	✗
Scratch-Large	✓	✗	✗	✗	✓(2M steps)
Scratch-Small	✗	✓	✗	✗	✓(2M steps)

(which is trained on the *Large* dataset that is 10 times the size of the *Small* dataset) which requires to be trained on the *Large* dataset to generate an average prediction length of 5.75 seconds. Additionally, the average generated episode lengths of the HMG trained on quarter and half the size of the *Small* dataset are lower than that of Scratch-Large and higher than that of Scratch-Small. This shows that our proposed model is significantly data-efficient in learning motion prediction downstream task.

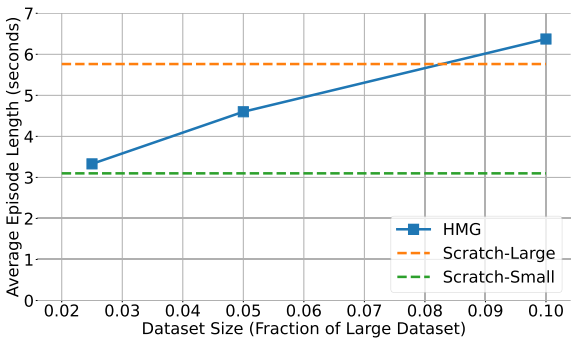


FIGURE 4. Performance based on Dataset Sizes: The x-axis denotes the dataset size given in fraction of the *Large* dataset and the y-axis denotes the average episode length. The average episode length for Scratch-Large (orange) is taken when it is trained on the *Large* dataset and for Scratch-Small (green) it is the *Small* dataset.

B. BEHAVIORAL DIFFERENCES

1) QUANTITATIVE ANALYSIS

Using the differences in episode lengths of motion predictions, we were able to find the exact behaviors generated by the models that outperformed the other models in terms of episode lengths. We considered the minimum difference in episode lengths of generated behaviors from the validation dataset (which consists of 63 behaviors), between models to be 6 seconds which we considered significant.

We compared the episode lengths generated by HMG, Scratch-Large, and Scratch-Small. Our analysis showed that HMG produced more episodes with longer durations than Scratch-Large (Table 3) and significantly longer than Scratch-Small (Table 4). Additionally, the difference in average episode lengths when HMG either outlasted Scratch-Large or Scratch-Small was higher than when either Scratch-Large or Scratch-Small outlasted HMG (Scratch-Small did not outlast HMG in any motion as shown in Table 4). These results demonstrate that our proposed model has superior capability in predicting longer trajectories, allowing the humanoid to survive longer in the simulation environment.

2) QUALITATIVE ANALYSIS

We further looked into the exact behaviors that led to these differences in prediction lengths, to observe empirically how different the motion predictions were. Figs. 5 to 7 show the comparison between the frames of the humanoid behaviors generated by HMG, Scratch-Large, and Scratch-Small. When the humanoid is red it denotes that the humanoid is given the motion prompt, and when in bronze denotes that the humanoid is under motion prediction. Fig. 5 shows a simple locomotion behavior. Fig. 6 shows a humanoid behavior that involves immediate change in direction and running. Fig. 7 shows a humanoid behavior that involves arm gestures while standing.

There were differences between the models' generations of moderate speed cyclic movements like walking, fast cyclic movements such as running, and non-cyclic movements as well. Walking behaviors patterns were observed to be similar, with the only difference being the generation length, with HMG coming out on top. All the models suffer when predicting running behaviors and the humanoid falls quickly to the ground. We attribute this issue to the flight phase in running which might make it more difficult to predict the proper footstep planning. For non-cyclic behaviors like arm gestures or side-stepping, HMG generates movements that are not necessarily close to the ground

TABLE 3. Motion Generation Durability (Between HMG and Scratch-Large): We observed that HMG outperformed Scratch-Large in 9 out of 63 behaviors. The average of the episode lengths for these 9 behaviors were taken for both the models and the difference was computed as shown in the table. The same was done for behaviors where Scratch-Large outperformed HMG.

Model	No. of generated motions that outlasted the other model (from validation dataset)	Δ Avg. Episode Length (sec)
HMG	**9 out of 63**	**8.630**
Scratch-Large	5 out of 63	8.082

TABLE 4. Motion generation durability (between HMG and scratch-small): evaluation done in Table 3 was applied here as well.

Model	No. of generated motions that outlasted the other model (from validation dataset)	Δ Avg. Episode Length (sec)
HMG	**15 out of 63**	**11.025**
Scratch-Small	0 out of 63	N/A

truth, but helps it survive longer in the episode compared to Scratch-Large and Scratch-Small. This indicates that our proposed model has learned different representation that abstract a better understanding of the relationship between observations and actions, that in turn helps in understanding the dynamics of the humanoid in the simulation environment to preserve the imitation and survival as much as possible.

C. GENERATED TRAJECTORY LENGTH

In this subsection, we are going to cover the models' capability for generating long sequences given a motion prompt. The resulting episode lengths of generated trajectories of behavior clips from the training and validation datasets were recorded. The length of the motion prompts was 32 steps long, which constitutes one second of motion from the expert policy.

From Figs. 8 and 9, we can observe that the generated episode lengths of HMG are similar or slightly longer to those lengths generated by the Scratch-Large model, and significantly longer than Scratch-Small before episode termination. Scratch-Small fails to compete with the other models as it is unable to generate longer episode lengths. This shows that Scratch-Small lacks the ability to predict sequences greater than 5 seconds in length, indicating that training a GPT model on a small dataset for motion generation without pre-training performs poorly. From these box plots, it is clear that HMG slightly outperforms Scratch-Large in terms of generating longer predictions, whereas Scratch-Small significantly lacks similar capabilities.

D. MOTION PREDICTION METRICS

To further evaluate the motion prediction in terms of motion quality, motion imitation, and generation diversity, the standard metrics for motion prediction were chosen, namely; Frechet Inception Distance (FID), Average Displacement Error (ADE), Final Displacement Error (FDE), and Diversity (DIV). We chose these metrics as the standard based on previous works that used these metrics to evaluate kinematic motion prediction [16], [30], [31], [32]. We believe that these metrics are appropriate here even for dynamics motion prediction since the mode of control is positional (units are homogeneous between kinematic and dynamic motion prediction). First the metrics will be defined as below.

FID is one of the most important metrics to determine how well the generated motion quality is. The idea behind FID is to compare the distributions between the features from the motion feature extractor based on the real and generated trajectories. The common practice is to train another RNN as a motion feature extractor on the relevant data and then extract the features. Training another RNN in this case is redundant, since the GPT is trained on reconstructing and predicting the motion given the motion prompt, plus, the representation of the features taken from the middle of the GPT network is rich [33]. Hence, the pre-trained model which was trained on observation data was used as the motion feature extractor to calculate FID and DIV. Equation (2) shows how to calculate FID

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}), \quad (2)$$

where, μ_r, μ_g, Σ_r, Σ_g are the means and co-variances of the features from the real and generated data respectively. Tr is the trace of the resulting matrix.

Understanding ADE is intuitively quite straightforward. It is essentially the average of the differences in joint poses across entire trajectories between the generated trajectories and the ground truth

$$\text{ADE} = \frac{1}{N \times T} \sum_{i=1}^{N} \sum_{t=1}^{T} \|\hat{\mathbf{j}}_{i,t} - \mathbf{j}_{i,t}\|, \quad (3)$$

where $\hat{\mathbf{j}}_{i,t}$ and $\mathbf{j}_{i,t}$ are the generated and real joint poses respectively at time step t and trajectory i.

FDE is the difference in joint poses at the last step of the trajectory between the generated motion and ground truth

$$\text{FDE} = \frac{1}{N} \sum_{i=1}^{N} \|\hat{\mathbf{j}}_{i,T} - \mathbf{j}_{i,T}\|, \quad (4)$$

where $\hat{\mathbf{j}}_{i,T}$ and $\mathbf{j}_{i,T}$ are the generated and real joint poses respectively at the final time step of the trajectory i. ADE and FDE scores are in radians.

FID, ADE, and FDE help us understand the quality of the motion produced and how close the generated motion is to the ground truth.

Another important metric is the generation diversity (DIV). This metric shows the variance in the generated motions across all behaviors the model is trained on. It is preferred if the variance is close to that of the real dataset. DIV is calculated by creating two sets of n randomly sampled motions from each generated data and calculating the diversity between the two. Equation (5) calculates the

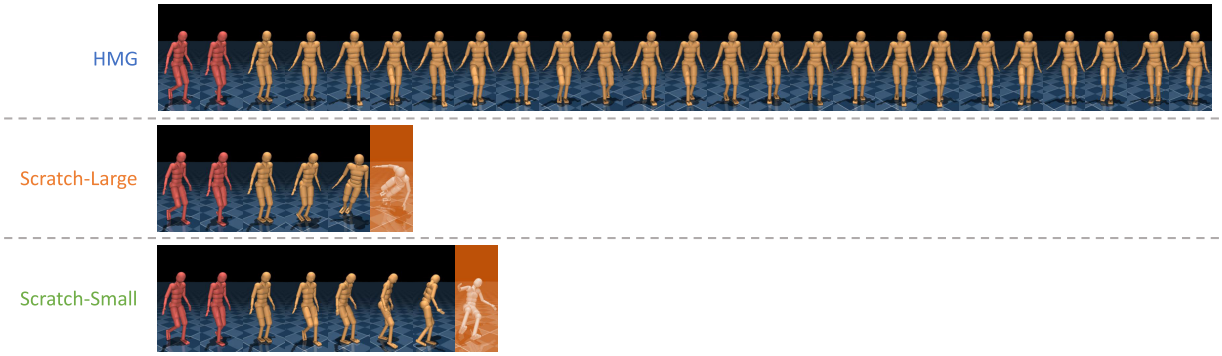


FIGURE 5. Walking Randomly: Top - HMG again maintains balance throughout the prediction while also succeeds in walking in different directions, Middle - Scratch-Large fails to change direction and falls early, Bottom - Scratch-Small also fails to change direction and falls early but lasts slightly longer than Scratch-Large in this generation. Every tenth frame was recorded and the frame in orange indicates episode termination.

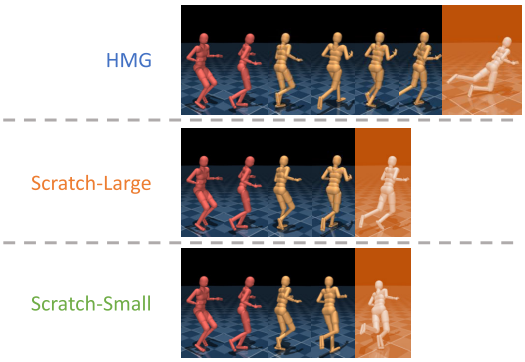


FIGURE 6. Running Towards the Left: Top - HMG survives a little longer than Scratch-Large and Scratch-Small but falls possibly due to the flight phase or fast changes in foot placement, Middle - Scratch-Large falls almost immediately from the start of the motion prediction, Bottom - Scratch-Small similar to Scratch-Large falls almost immediately from the start of the motion prediction. Every tenth frame was recorded and the frame in orange indicates episode termination.

diversity score of a model

$$\text{DIV} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{E_d}(\mathbf{v_i}, \mathbf{v_i'}) \tag{5}$$

where $\mathbf{v}, \mathbf{v}'$ are feature vectors from two sampled sets from the same generated models (expert policy, HMG, Scratch-Large, Scratch-Small) respectively across all behaviors, n is the number of samples in the set, and $\mathbf{E_d}$ is the Euclidean distance. We set n to be 200. We require ADE and FDE to be low so that the motion similarity is high, we require FID to be low to ensure that the motion quality is as close to that of ground truth, and we require DIV to be close to that of the real data. From Table 5, we can see that the HMG scores are in the desired range. However, DIV score for HMG seems to deviate from the real data slightly more than the DIV score for Scratch-Large. Scratch-Small deviates from real data significantly more than the other two models (FID and DIV is the highest among the three). Scratch-Small's imitation capability is also lower since the ADE and FDE scores are higher than the other models' scores.

TABLE 5. Motion prediction scores: ↓ indicates that lower score is desired and → indicates that score closer to Real (ground truth data) is desired.

Model	FID ↓	ADE ↓	FDE ↓	DIV →
Real	0.000	N/A	N/A	7.530
HMG	**7.741**	**6.824**	**6.616**	8.222
Scratch-Large	8.415	6.866	6.910	**8.151**
Scratch-Small	290.726	7.271	7.250	17.935

V. DISCUSSION

In this study, we attempt to tackle a situation where we have access to copious amounts of kinematic data and limited access to dynamic data. This is to simulate the difficulty in obtaining large amounts of action data from humanoids. We hypothesize that by implementing the foundation model training approach we can bring about the similar or slightly better performances compared to the baselines. From Fig. 4 we concluded that by using a pre-trained model we can bring about higher data and training efficiency. From Tables 3 and 4, we can see that the HMG can generate higher average episode lengths than those generated by the baselines. Figs. 5 to 7 are some of the motion predictions were HMG outperforms the baseline models except for fast paced movements like running where all the models fails to keep the humanoid alive and keep the episode from terminating. Figs. 8 and 9 show that HMG overall generates higher episode lengths than the baselines. Overall our proposed approach displays higher performance in control compared to the baselines with a smaller set of action data.

A. KNOWLEDGE TRANSFER

We were interested to know how much of the knowledge gained during pre-training phase was passed down during fine-tuning. Each models' weights except the input embeddings and action heads were grouped together and cosine similarities were calculated between one another and themselves. A heat map was applied to this correlation and it can be observed that there is a high correlation between the pre-trained model and HMG, indicating that pre-training serves as a useful prior in fine-tuning (Fig. 10). For fine-tuning, there are other methods that have come up

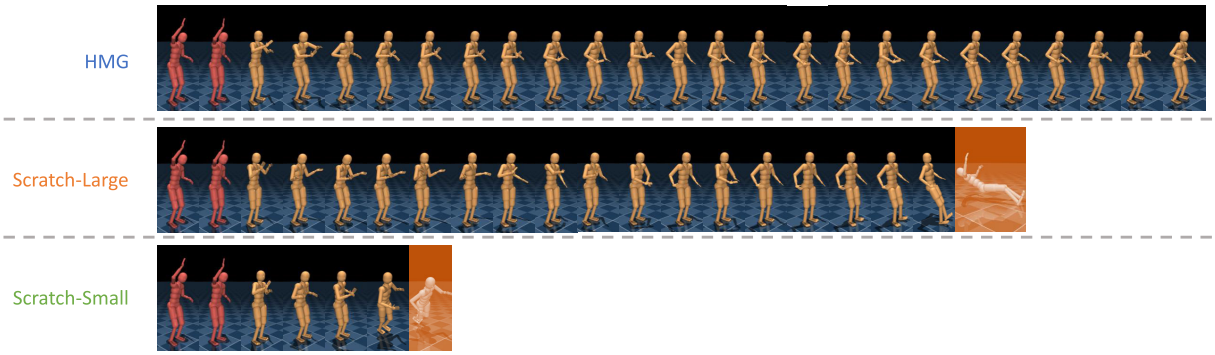


FIGURE 7. Gestures: Top - HMG's prediction does not match the ground truth but survives the entire duration of the episode, Middle - Scratch-Large's motion generation also does not accurately predict the gestures, loses balance after a while and falls, Bottom - Scratch-Small loses balance quite quickly and falls. Every tenth frame was recorded and the frame in orange indicates episode termination.

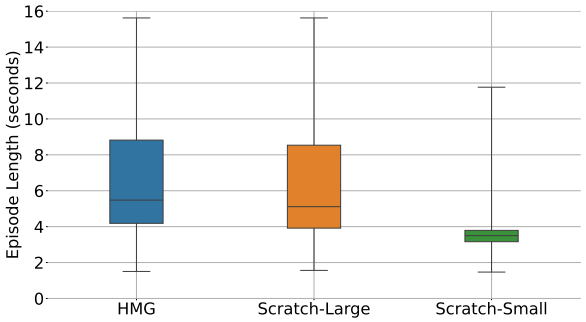


FIGURE 8. Comparison of generated episode lengths on training dataset.

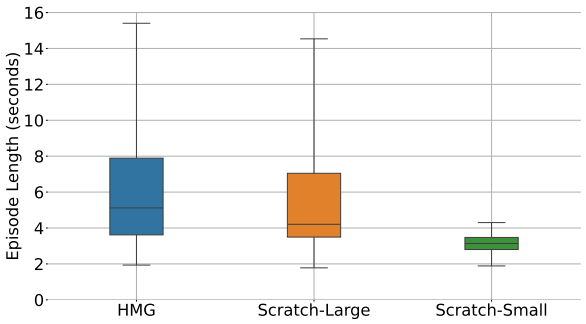


FIGURE 9. Comparison of generated episode lengths on validation dataset.

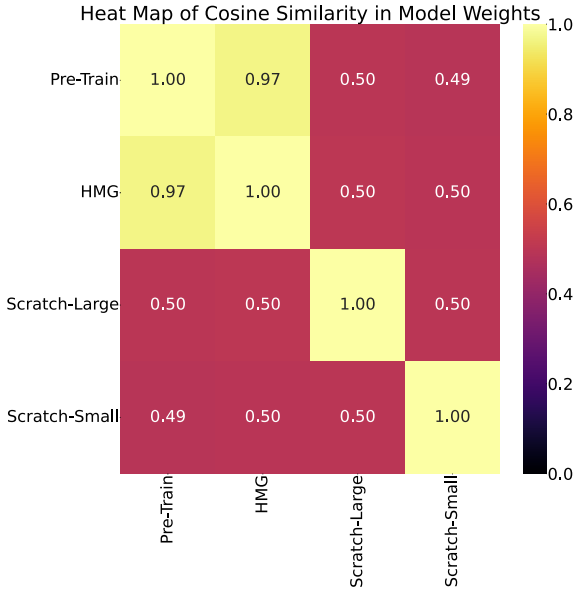


FIGURE 10. Model Weight Similarity: All of models' weights except the input embeddings and action heads were taken and cosine similarities were calculated between one another and themselves. A heat map was applied over this correlation and we can see that there is a high correlation between the pre-trained model and HMG, indicating that majority of the knowledge is preserved from pre-training.

such as LoRA [34] and Neural Adapters [35], however the current method itself yields results that are similar to that of Scratch-Large. It is possible that by implementing a better fine-tuning technique would not necessarily yield better results.

B. EVALUATION ANALYSIS

In [17], box plots and histograms were used to evaluate their model's capability to generate motion. However, since it is hard to tell how well the humanoid can actually imitate and complete the ground truth motion prompts accurately, the standard metrics such as FID was chosen to evaluate motion quality, DIV for motion diversity, ADE and FDE for motion imitation [16], [36], [37]. However, there are some limitations to using ADE and FDE as evaluation metrics. Since we cannot control the motion generation beyond the motion prompt, it is not necessary that the generated motion correctly imitates the ground truth. For instance, when considering the gesture type movements in Fig.7 or other related behaviors, the difference in ADE and FDE scores are low between the models. This is because it is not just about the humanoid surviving in the simulation, but also the accuracy of the imitation. Nevertheless, since ADE and FDE are standard metrics used for imitation in motion prediction, they were still included in the evaluation. Despite this limitation, we still used this metric to evaluate and observed that HMG manages to stay closer to the ground truth than the other baselines. Therefore, to understand the performance of our model on a behavior level, from the motion prediction metric evaluation

we can see that our model performs as well as the baseline model and in some cases slightly better, thus there is no loss in performance when implementing foundation model approach.

From our evaluations, we could confirm that our proposed method of fine-tuning a pre-trained model is more data-efficient than training from scratch, and the imitation performance between both the methods are quite similar with our model slightly outperforming the other.

C. LIMITATIONS

There are a couple of limitations to this evaluation. Firstly, there is no ablation study done on different pre-training strategies to see what could enhance the knowledge transfer. Furthermore, although Fig. 10 shows how much knowledge was transferred from the pre-trained model to HMG, it would be better to analyze the weights further on how exactly the pre-training stage contributes to the downstream task performance through attention patterns and other feature representations. Finally, even though episode lengths and motion prediction metrics such as FID, ADE, FDE, and DIV were included in the evaluation, it is hard to understand whether the proposed training approach preserves or improves the physical realism. In the future, we would like to consider including other metrics like joint torque smoothness and center-of-mass stability in our evaluation.

We cannot control the generation of the motion by conditioning, for instance, if the model generates a motion given a prompt to stand, it may try to either continue to stand or walk. Another limitation is that it uses fully observable state space, whereas realistically we would prefer to only use partially observable variables. Further, to claim that our approach provides a good humanoid foundation model, it is better to show how well the pre-trained model can adapt to other downstream tasks. Since our focus is on making the training the model in a data-efficient manner, we chose not to expand in this direction in our experiments and have designated it as future work. In terms of real world application, if there are changes in the physics parameters in the environment, it may be necessary to re-collect a fine-tuning dataset. Perhaps by training the model using RL via feedback and using domain randomization techniques, it may help to mitigate this issue. We are curious to see if our foundation model can be trained and fine-tuned on simulation data obtained from other robots such as [38].

VI. CONCLUSION

We proposed a data-efficient approach for motion prediction by pre-training a humanoid motion foundation model on observation data and fine-tuning it on both observation and action data for a motion prediction downstream task. The proposed method's performance was evaluated after training from scratch based on several aspects, including training efficiency, motion prediction metrics, generation lengths, and empirical observations of various cyclic, non-cyclic, and fast-paced behaviors. We observed that HMG quickly generates higher average prediction lengths on the validation datasets after being trained on the *Small* dataset, compared to Scratch-Large which is trained on the *Large* dataset, proving it to be more data-efficient. We further show from the motion prediction metric evaluation, FID, ADE, FDE, and DIV, that our proposed model can generate more accurate and longer motion trajectories of higher quality than the state of the art. The main limitation of this approach is that we cannot control the generation of motion by conditioning. To extend this work, we are interested in modifying the input to include conditioning to control the motion generation and further implement our proposed approach to multiple humanoid data.

REFERENCES

[1] Z. Xie, X. Da, M. van de Panne, B. Babich, and A. Garg, "Dynamics randomization revisited: A case study for quadrupedal locomotion," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 4955–4961.

[2] X. Bin Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," 2020, *arXiv:2004.00784*.

[3] 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," 2023, *arXiv:2309.14225*.

[4] T. He, Z. Luo, W. Xiao, C. Zhang, K. Kitani, C. Liu, and G. Shi, "Learning human-to-humanoid real-time whole-body teleoperation," 2024, *arXiv:2403.04436*.

[5] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn, "HumanPlus: Humanoid shadowing and imitation from humans," in *Proc. Conf. Robot Learn. (CoRL)*, Jun. 2024.

[6] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile ALOHA: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," in *Proc. Conf. Robot Learn. (CoRL)*, Jan. 2024.

[7] R. Bommasani et al., "On the opportunities and risks of foundation models," 2021, *arXiv:2108.07258*.

[8] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019. [Online]. Available: https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

[9] M. Samin Yasar and T. Iqbal, "Improving human motion prediction through continual learning," 2021, *arXiv:2107.00544*.

[10] W. Zhu, X. Ma, Z. Liu, L. Liu, W. Wu, and Y. Wang, "MotionBERT: A unified perspective on learning human motion representations," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 15039–15053.

[11] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Real-world humanoid locomotion with reinforcement learning," 2023, *arXiv:2303.03381*.

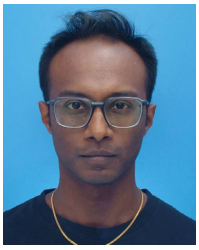
[12] T. B. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, J. Burstein, C. Doran, and T. Solorio, Eds., 2019, pp. 4171–4186, doi: 10.18653/v1/n19-1423.

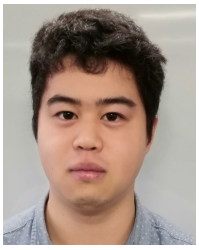
[14] M. Jänner, Q. Li, and S. Levine, "Offline reinforcement learning as one big sequence modeling problem," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2021, pp. 1273–1286. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/099fe6b0b444c23836c4a5d07346082b-Paper.pdf

[15] E. Aksan, M. Kaufmann, P. Cao, and O. Hilliges, "Attention, please: A spatio-temporal transformer for 3D human motion prediction," 2020, *arXiv:2004.08692*.

[16] B. Jiang, X. Chen, W. Liu, J. Yu, G. Yu, and T. Chen, "MotionGPT: Human motion as a foreign language," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2023.
[17] N. Wagener, A. Kolobov, F. V. Frujeri, R. Loynd, C.-A. Cheng, and M. Hausknecht, "MoCapAct: A multi-task dataset for simulated humanoid control," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2022, pp. 35418–35431. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/e5dd4fbb6fb4cb805b982bfb41c20aad-Paper-Datasets_and_Benchmarks.pdf
[18] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," in *Proc. Adv. Neural Inf. Process. Syst. 31*, Jan. 2018, pp. 2451–2463. [Online]. Available: https://worldmodels.github.io
[19] D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba, "Mastering Atari with discrete world models," 2020, *arXiv:2010.02193*.
[20] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse domains through world models," 2023, *arXiv:2301.04104*.
[21] W. Guo, Y. Du, X. Shen, V. Lepetit, X. Alameda-Pineda, and F. Moreno-Noguer, "Back to MLP: A simple baseline for human motion prediction," 2022, *arXiv:2207.01567*.
[22] Z. Cao, H. Gao, K. Mangalam, Q. Cai, M. Vo, and J. Malik, "Long-term human motion prediction with scene context," in *Proc. ECCV*, 2020, pp. 387–404.
[23] Y. Rong, T. Shiratori, and H. Joo, "FrankMocap: A monocular 3D whole-body pose estimation system via regression and integration," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 1749–1759.
[24] Y. Yuan and K. Kitani, "DLow: Diversifying latent flows for diverse human motion prediction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Jan. 2020, pp. 346–364.
[25] Y. Zhu, N. Samet, and D. Picard, "H3WB: Human3.6M 3D wholebody dataset and benchmark," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 20109–20120.
[26] A. R. Punnakkal, A. Chandrasekaran, N. Athanasiou, A. Quirós-Ramírez, and M. J. Black, "BABEL: Bodies, action and behavior with english labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 722–731.
[27] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. Black, "AMASS: Archive of motion capture as surface shapes," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5441–5450.
[28] A. Karpathy. (2020). *MinGPT*. [Online]. Available: https://github.com/karpathy/minGPT
[29] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 5026–5033.
[30] C. Guo, X. Zuo, S. Wang, S. Zou, Q. Sun, A. Deng, M. Gong, and L. Cheng, "Action2Motion: Conditioned generation of 3D human motions," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, doi: 10.1145/3394171.3413635.
[31] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, p. 6629–6640.
[32] T. Salzmann, M. Pavone, and M. Ryll, "Motron: Multimodal probabilistic human motion forecasting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 6447–6456, doi: 10.1109/CVPR52688.2022.00635.
[33] A. Toppo and M. Kumar, "A review of generative pretraining from pixels," in *Proc. 3rd Int. Conf. Adv. Comput., Commun. Control Netw. (ICAC3N)*, Dec. 2021, pp. 495–500.
[34] J. E. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *Proc. ICLR*, Jan. 2021, pp. 1–14. [Online]. Available: https://www.microsoft.com/en-us/research/publication/lora-low-rank-adaptation-of-large-language-models/
[35] N. S. Moosavi, Q. Delfosse, K. Kersting, and I. Gurevych, "Adaptable adapters," 2022, *arXiv:2205.01549*.
[36] Y. Zhang, M. J. Black, and S. Tang, "We are more than our joints: Predicting how 3D bodies move," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3371–3381.
[37] H. Ma, J. Li, R. Hosseini, M. Tomizuka, and C. Choi, "Multi-objective diverse human motion prediction with knowledge distillation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8151–8161.
[38] F. Al-Hafez, G. Zhao, J. Peters, and D. Tateo, "LocoMuJoCo: A comprehensive imitation learning benchmark for locomotion," in *Proc. 6th Robot Learn. Workshop, NeurIPS*, Jan. 2023, pp. 1–14.



SIDDHARTH PADMANABHAN received the Bachelor of Technology degree in mechanical engineering from Vellore Institute of Technology, India, in 2019, and the Master of Engineering degree from Osaka University, Japan, in 2021, where he is currently pursuing the Ph.D. degree, working on robot learning with focus in reinforcement learning, multi-task learning, supervised learning, imitation learning, and foundation models in humanoid whole-body control.



KAZUKI MIYAZAWA received the Ph.D. degree from Osaka University, Osaka, Japan, in 2022. He was a JSPS Research Fellowship for Young Scientists from 2019 to 2022. Since 2022, he has been an Assistant Professor with the Graduate School of Engineering Science, Osaka University. His current research interests include multimodal data integration, robot learning, concept formation, natural language processing, and reinforcement learning.

TAKATO HORII (Member, IEEE) received the M.E. and Ph.D. degrees in engineering from Osaka University, Osaka, Japan, in 2013 and 2018, respectively. He was a Project Researcher with The University of Electro-Communications, from 2017 to 2019. He then an Assistant Professor with Osaka University, from 2019 to 2020, and an Associate Professor with Osaka University, in 2020. His current research interests include computational modeling of human cognitive functions, such as emotion and creativity and machine learning algorithms.

TAKAYUKI NAGAI (Member, IEEE) received the B.E., M.E., and Ph.D. degrees from the Department of Electrical Engineering, Keio University, in 1993, 1995, and 1997, respectively. Since 1998, he has been with The University of Electro-Communications, and since 2018, he has been a Professor with the Graduate School of Engineering Science, Osaka University. From 2002 to 2003, he was a Visiting Scholar with the Department of Electrical Computer Engineering, University of California, San Diego. He is currently a specially-appointed Professor at AIX, UEC, a Visiting Researcher with Tamagawa University Brain Science Institute, and a Visiting Researcher with AIST/AIRC. His research interests include intelligent robotics, cognitive developmental robotics, and robot learning. He aims at realizing flexible and general intelligence like human by combining AI and robot technologies. He received the IROS Best Paper Award Finalist, the Advanced Robotics Best Paper Award, and the JSAI Best Paper Award.