
Pretraining Models via CLIP-Selected Motion Trajectories

Alex Lyons

Department of Machine Learning
Carnegie Mellon University
alyons@andrew.cmu.edu

Het Sekhalia

Department of Mechanical Engineering
Carnegie Mellon University
hsekhali@andrew.cmu.edu

Ziyu Li

Department of Computational Biology
Carnegie Mellon University
ziyuli@andrew.cmu.edu

Abstract

Training deep reinforcement learning agents for physics-based humanoid motion is computationally expensive, especially when on complex skills from motion capture. We study how pretraining on a set of similar motions, selected by a model leveraging human priors, affects the efficiency and quality of learning new complex humanoid motions. Building on the DeepMimic framework, we introduce a CLIP-based sampling method. It assigns higher pretraining weight to motions that are more semantically similar to the target behavior. These similarity scores are computed using only CLIP’s text encoder. Agents are first pretrained on motions drawn from CLIP derived distribution and are then finetuned on the target motion. We compare Proximal Policy Optimization (PPO) and Twin Delayed DDPG (TD3) across four target skills with ablations over sampling temperature and pretraining subset size. Our experiments show that this similarity guided pretraining can provide meaningful semantic priors for physics-based control, leading to faster learning and higher final performance compared to training from scratch or sampling uniformly in pretraining.

1 Introduction

One key problem in reinforcement learning is often training an agent can be very costly. For many cases such as robotics, there is also the practical concern of it being resource-heavy to train the agents, and concern for failure causing damage incentivizes wanting to minimize training while still maintaining performance.

To remedy this, pretraining has been adapted into many reinforcement learning frameworks, where the agent learns to perform similar tasks to the target task first, before fine-tuning to learn the target task. These tasks can be simpler and less expensive than the target task, meaning that less training is required on the target task while still achieving similar results.

One example task is learning human locomotion, where the goal is for a humanoid agent to learn to imitate human motions taken from motion capture data. Using the pretraining paradigm, one could pretrain a humanoid agent to learn several motions, and then fine-tune on a target motion.

However, pretraining is a non-trivial task, and curating what to pretrain on requires careful insight. Since the goal is to replicate human motions, one natural insight is that leveraging prior knowledge on motions from humans on how motions relate can be useful for curation. One source of priors

commonly used in machine learning is using pretrained vision-language models, which embed human information on various subjects, such as motions, that can be utilized for planning how to pretrain on different motions.

Therefore, we utilized vision-language models, specifically CLIP, for getting human priors on motion for curating pretraining. Specifically, we use CLIP to embed different pretraining motions, create a sampling distribution by comparing those embeddings to an embedding of the target motion, and then sample using that distribution during pretraining. Thus, the agent trains more on similar motions to the target motion, while still leveraging the more dissimilar motions for learning motion primitives and improving generalization before fine-tuning. We not only found that our CLIP-based pretraining technique results in better performance compared to learning the target motion from scratch, but also outperforms uniformly sampling from the pretraining motions. Therefore, we have shown that leveraging human priors in pretraining using CLIP can improve performance, and therefore result in having to spend less time training on the target motion.

2 Related Works

2.1 DeepMimic

One challenge for training humanoid agents in simulation is finding a balance between replicating the human motion and still adhering to physics. DeepMimic Peng et al. [2018] bridged this gap by combining motion-capture reference clips with physics-based RL in a unified framework. By directly rewarding a policy for imitating reference motions while also achieving task goals, DeepMimic produced physically realistic and accurate behaviors. The framework includes useful features such as phase-aware policies which incorporate at what stage of the motion the current agent is at, many hyperparameters available for initialization and termination conditions, and finally the implementation of sampling from many different motions during training. Our work builds on the last feature by using similarity-based motion selection during a pretraining stage, reducing manual curation and improving training efficiency. In doing so, we aim to make RL-based motion imitation more scalable, data-efficient, and adaptable across arbitrary motions.

2.2 Pretraining in RL

Previous work has shown that instead of training an agent to learn to perform a task from scratch, one can leverage pretraining on several similar tasks to speed up the process. For example, one work investigated pretraining an agent on multiple different ATARI games before learning a target ATARI game Taiga et al. [2023]. In this work, given a target ATARI game, the pretrain an actor-critic agent on various numbers of similar ATARI games for a given number of frames to the target game. They found that pretraining on more games improved the generalization of the agent when fine-tuning on the target game.

2.3 CLIP

One previous work that has been used for comparing both text and image similarity is CLIP Radford et al. [2021] ADD CITATION. CLIP consists of a text encoder and image encoder, which are trained to map inputs to a joint embedding space. The encoders are trained using a contrastive loss between around 400 million matching image-text pairs, such that the cosine similarity (i.e. dot product) between pairs of embeddings that represent similar semantic concepts is higher. For example, the embeddings for an image of a cat and the text "a cat" will have a high cosine similarity. However, the text encoder alone produces embeddings that measures the similarity between different text inputs. For example, given a text embedding of "a cat", it will have a higher similarity to "a dog" than "a car". CLIP has been shown to be robust, and is commonly used in many frameworks that require computing similarity between semantic concepts. Hence, we can use CLIP to measure the similarity between different concepts, using their text representations, such as motions.

2.4 RoboCLIP

As evidence that these pretrained vision-language models are useful for priors in robotic learning, RoboCLIP has been shown to use a CLIP-type model for training robotic agents to perform human

tasks Sontakke et al. [2023]. Specifically, instead of using CLIP, they use the S3D model Xie et al. [2018], which is a model consisting of a text and video encoder, which similar to CLIP is trained via contrastive learning. Their novel approach uses S3D as a reward model for performing arbitrary tasks via text description or demonstration. This is done by creating either a text embedding describing the target task or a video embedding of a demonstration of the task. Then while training, they render videos of the agent performing the task, embed it using the video encoder, and compute the cosine similarity between the target embedding and the current embedding as a sparse reward. Their results show that using this reward allows the robotic agent to learn tasks using the human priors encoded in the S3D model. Therefore, this shows that using CLIP-type models can be useful for evaluating motion similarity, which serves as a fundamental assumption for our pretraining method.

3 Methods

3.1 Overview

Based on the utility shown in previous work of pretraining and using human-priors from vision-language models, our method combines the two approaches by curating how pretraining is done using CLIP. Our pipeline involves sampling from different motions during pretraining based on a distribution defined using a CLIP model and the motions’ similarities to the target motion, and then fine-tuning on the target motion to achieve a faster convergence and better performance compared to training from scratch and uniformly sampling in pretraining.

3.2 CLIP-Based Motion Sampling

To get semantic representations of each of the motions used for pretraining, we utilized the CLIP text encoder by entering text descriptions of each motion. We only utilized the text encoder because when we used the vision encoder to create embeddings of the motions based on renders of the reference motions, they were too out of distribution, and created embeddings that didn’t accurately represent how similar different motions should be based on what we know about actions. After computing the text embeddings for the different motions, the embeddings are normalized, and then the dot product of the target embeddings is computed against the dot products of all the pretraining motion embeddings to get their cosine similarity scores. Given the cosine similarity scores of the pretraining motions, softmax is applied to create a distribution of how likely to sample each motion. Consequently, motions with a higher similarity score to the target motion will be sampled more.

However, we noticed that since all the motions are actions, they are inherently semantically related, and therefore the resulting softmax distributions were close to uniform. To remedy this, we also included a temperature parameter to the cosine similarities before inputting them into softmax, such that using a lower temperature makes the resulting softmax distribution less spread. Therefore, having a lower temperature resulted in similar motions to the target motion to be even more likely sampled during pretraining.

3.3 Pretraining on Similar Motions

Given a set number of pretraining motions, and a distribution generated using CLIP of how often to sample them, a single agent is trained to learn all the motions for a set number of samples. We utilize DeepMimic as the framework for pretraining the agent on the distribution of motions. DeepMimic uses a humanoid model with about 34-36 total degrees of freedom, including major limbs such as the hips, knees, ankles, spine, shoulders, elbows, and wrists. The state provided to the policy includes joint orientations and velocities, root position and orientation, and root linear and angular velocities. The action is a vector of target joint poses to generate stable joint torques in the physics simulator. Notably, the body pose (root, joints, etc) of the reference motion for the given timestep is also included in the observation space of the agent. This allows the agent to implicitly condition its action on the currently sampled motion from the CLIP-based distribution.

3.4 Fine-Tuning to the Target Motion

Once the agent is pretrained on the motions, it is then fine-tuned on only the target motion, again using the target motions body pose as part of the agent’s observation space. Our hypothesis is that by

pretraining on the similar motions, it will have learned motion primitives that are useful for learning the new target motion, compared to learning it from scratch. Additionally, by pretraining on many different motions, the agent will have learned to handle inputting many different body poses in its reference motion observation space, it will become more generalizable, and therefore more easily interpret the new reference body pose in its observation space.

3.5 Experiments

To evaluate the effectiveness of our pretraining method, we compared agents trained using our method against the same type of agent trained to learn the motion from scratch as a baseline. The baseline agents and our agents trained on the target motion for the same number of samples.

To fully understand how using a CLIP-based distribution for sampling in pretraining, we experimented with several ablations: The agent type, the temperature used for scaling in softmax, and the number of motions sampled from in pretraining.

To evaluate how different types of agents would learn using the CLIP-based pretraining, we trained both PPO Schulman et al. [2017] and TD3 Fujimoto et al. [2018] agents in our setup. PPO was selected since it is well known for being good in continuous action spaces, such as humanoid locomotion, and for being stable. The stability is interesting in our case, since we wanted to see how a PPO agent would interact in multi-task pretraining environment, which is inherently more unstable compared to the typical environment PPO is used for. TD3 was selected since it is also good in continuous action spaces, and it is known for its good sample efficiency due to its extensive use of a replay buffer, and its multiple networks for stability. It is useful for our method, since mixing samples across the different motions via replay buffer naturally fits well with our CLIP-distribution based learning.

Second, we experimented with different temperatures for scaling the softmax distributions. When manually inspecting the distributions formed from different sets of motions, we found that using temperatures of 0.01, 0.05, and ∞ (i.e. uniform) would provide a good range of different behaviors, where again the higher temperatures lead to more evenly spread distributions. We initially planned to also perform experiments where only the most-similar motion is sampled from (i.e. equivalent to a temperature of 0). However, this led to too much instability when switching to fine-tuning due to lack of generalization ability in understanding the new reference body poses in the observation space, causing too many body part collisions, leading the simulation to crash.

Finally, we experimented with using different number of motions during pretraining. The subset of the motion capture dataset we used consisted of 21 motions, meaning there is a total of 20 possible motions used for pretraining for 1 given target motion. We experimented with using 4, 10, and all 20 motions for pretraining. For each target motion we fine-tuned on, we selected the subsets of pretraining motions manually to include both similar and dissimilar motions.

3.6 Experimental Setup

For running our experiments, we aimed to learn four different target motions: Zombie walk, dance, spin kick, and getup-facedown (a humanoid standing up after lying facedown on the ground). For both pretraining and fine-tuning, the agents were trained using a set 160000000 samples before concluding. Comparisons were drawn between the baselines and our various methods per motion by comparing the mean test return behavior exhibited across fine-tuning on the target motions.

The motions we learn to replicate come from the CMU Motion Capture Database Carnegie Mellon University [2003], which consists of many motions performed by humans, which were captured by markers attached to body suits using infrared. The resulting captures produced files consisting of the body motions, which were converted to skeleton motions including joints and their angles. The reference skeleton motions is what is used for the DeepMimic agent to learn from.

We utilized the MimicKit implementation of the DeepMimic framework, due to its popularity and ease of use for the motion capture dataset Peng [2025]. For PPO, we used a learning rate of $5e-5$, a discount factor of 0.99, and a clip ratio of 0.2. For TD3, we used a learning rate of $1e-4$ with the same discount factor, and a buffer size of 1000000. These hyperparameters were chosen through experimentation and by referring to example agent parameters given by the MimicKit framework.

4 Results

4.1 Comparison of RL Agents

Across all tasks, PPO consistently outperforms TD3 when initialized from the same pretrained motion distributions. As shown in Fig. 1, PPO not only reaches higher final returns, but also learns much faster in the early stages. TD3, on the other hand, improves more slowly and generally plateaus at a lower level.

We attribute these differences to PPO’s policy clipping mechanism, which prevents overly large updates and helps keep training stable. This stabilization is especially helpful in motion-imitation settings, where the initial policy may be far from the target motion. TD3, while theoretically more sample-efficient, tends to be more sensitive to noise and struggles to fully leverage the pretrained prior.

4.2 Performance Evaluation Across Sampling Temperature

We next examine how sampling temperature during pretraining influences downstream learning for both TD3 and PPO. Across all tasks, we found that pretraining provides a meaningful initialization that accelerates learning and improves overall performance. The temperature parameter during pretraining plays an important role, but its impact differs between the two agents.

For TD3, the effect is strong and consistent (Fig. 2). Lower temperatures ($\tau = 0.01$) produce the best performance, leading to much faster improvement and higher final returns. This is because a more concentrated distribution focuses pretraining on the most informative motions, giving TD3 a clearer and more structured started point. Moderate temperature ($\tau = 0.05$) offer partial benefits, while uniform sampling tends to underperform across tasks. Training from scratch performs the worst, showing that even simple pretraining provides a valuable inductive bias for TD3.

In contrast, PPO is far less sensitive to temperature (Fig. 3). While lower temperatures still help, especially early in training, the differences between $\tau = 0.01$, $\tau = 0.05$, and uniform sampling are relatively small. PPO consistently converges to similar final performance regardless of temperature settings. This stability aligns with PPO’s policy-clipping design, helping the algorithm make steady progress even when the pretraining distribution is broad or noisy.

4.3 Performance Evaluation Across Pretraining Subset Sizes

We vary the subset of motions used for pretraining (4, 10, or all 20 available trajectories) to measure how the breadth of motion diversity affects downstream RL. Larger subsets yield strong improvements for TD3 (Fig. 4), whereas PPO benefits only slightly (Fig. 5).

For TD3, using all 20 motions yields the strongest results across tasks, while a 4-motion subset often causes underfitting and slower learning, and 10 motions offer an intermediate improvement. This pattern indicates that TD3 depends heavily on having a diverse set of motions to sample from, which helps the model generalize rather than overfit to a narrow motion set. In contrast, PPO shows a much weaker dependence on subset size. Although larger subsets slightly accelerate early learning, PPO reaches similar final performance regardless of whether it is pretrained on 4, 10, or 20 motions. PPO’s stable optimization allow it to overcome the lack of diversity during fine-tuning, whereas TD3 gains from a broader and more varied pretraining distribution.

5 Discussion

Our experiments demonstrate that CLIP-guided motion pretraining is an effective mechanism for improving physics-based humanoid control within the DeepMimic framework. Across all four target skills (Dance, Getup-Facedown, SpinKick, Zombie Walk), pretraining on similar motions before fine-tuning and yields higher final returns compared to training from scratch, and results in faster convergence rates compared to uniformly sampling. This supports our hypothesis that pretraining on a motions selected using human-based priors encoded in CLIP is more effective than simply pretraining uniformly.

A key finding is the strong difference between PPO and TD3 in how they benefit from this pretraining. PPO reliably outperforms TD3 in both convergence speed and final performance under all configurations considered. Its clipped, on-policy updates appear well suited for humanoid learning and allow it to benefit from pretraining even when the CLIP-based sampling distribution is broad or noisy. TD3, in contrast, is more sensitive to the pretraining configuration: low-temperature CLIP sampling and larger pretraining subsets provide clear gains, while uniform sampling or very small subsets often lead to slower learning and returns. Therefore, CLIP-guided sampling plays a critical role in making TD3 competitive, whereas PPO is robust enough to succeed under a wide range of pretraining methods.

One failure case for our method is when there are too few motions during pretraining. As briefly mentioned before, when we experimented with only using one motion during pretraining, the transition from pretraining to fine-tuning was too severe, causing the simulation to crash. Specifically, since the reference body pose is in the observation space, since the reference body poses for two different motions is very different, switching from one set of poses as a model input to another set caused the agent to initially contort, leading to too many body collisions, causing the simulation to crash. We avoided this issue by using enough pretraining motions where the agent didn’t overfit and was able to handle multiple different reference body poses as input. However, this doesn’t fix the fundamental issue that if the tuning pose is too out of distribution, our method might cause the simulation to crash.

6 Conclusion

In order to reduce the training needed for learning potentially difficult and costly tasks, recent work has leveraged pretraining on similar tasks in order to learn priors before learning the target task. We aimed to further this work in the humanoid locomotion setting, by curating a distribution of motions to sample from during pretraining based on their similarities to the target motion. We utilized CLIP, a model pretrained on copious amounts of data, to curate the distribution based on human priors we have about motions and their similarities. We found that under many conditions, using our CLIP-based sampling method in pretraining not only outperforms learning the motion from scratch, but also uniformly sampling during pretraining, thus showing that leveraging these humanoid priors can in fact be more efficient.

However, as mentioned previously, a big limitation of our work is that it only works on motions that can be accurately described via text. Without the use of a vision-encoder, if someone wanted to train the humanoid to use motions taken from arbitrary videos or motion captures, they wouldn’t be able to do so accurately. Therefore, future work to remedy this could be involving a vision-encoder pipeline for computing the embeddings instead of the CLIP text encoder. As mentioned, using off-the-shelf image and video encoders produced unrealistic distributions, so to use a vision-encoder one would likely have to fine-tune or create one from scratch on the format of the motions they want to use.

References

- Carnegie Mellon University. The CMU Motion Capture Database. Website, Carnegie Mellon University, 2003. URL <https://mocap.cs.cmu.edu/>. Accessed: 2025-12-10.
- Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning*, 2018. URL <https://arxiv.org/abs/1802.09477>.
- Xue Bin Peng. Mimickit: A reinforcement learning framework for motion imitation and control. 2025. URL <https://arxiv.org/abs/2510.13794>.
- Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics*, 2018. doi: 10.1145/3197517.3201311.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th*

International Conference on Machine Learning, 2021. URL <https://arxiv.org/abs/2103.00020>.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

Sumedh A Sontakke, Jesse Zhang, Sébastien M. R. Arnold, Karl Pertsch, Erdem Bıyık, Dorsa Sadigh, Chelsea Finn, and Laurent Itti. Roboclip: One demonstration is enough to learn robot policies, 2023. URL <https://arxiv.org/abs/2310.07899>.

Adrien Ali Taiga, Rishabh Agarwal, Jesse Farebrother, Aaron Courville, and Marc G. Bellemare. Investigating multi-task pretraining and generalization in reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=sSt9fR0SZR0>.

Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification, 2018. URL <https://arxiv.org/abs/1712.04851>.

7 Appendix

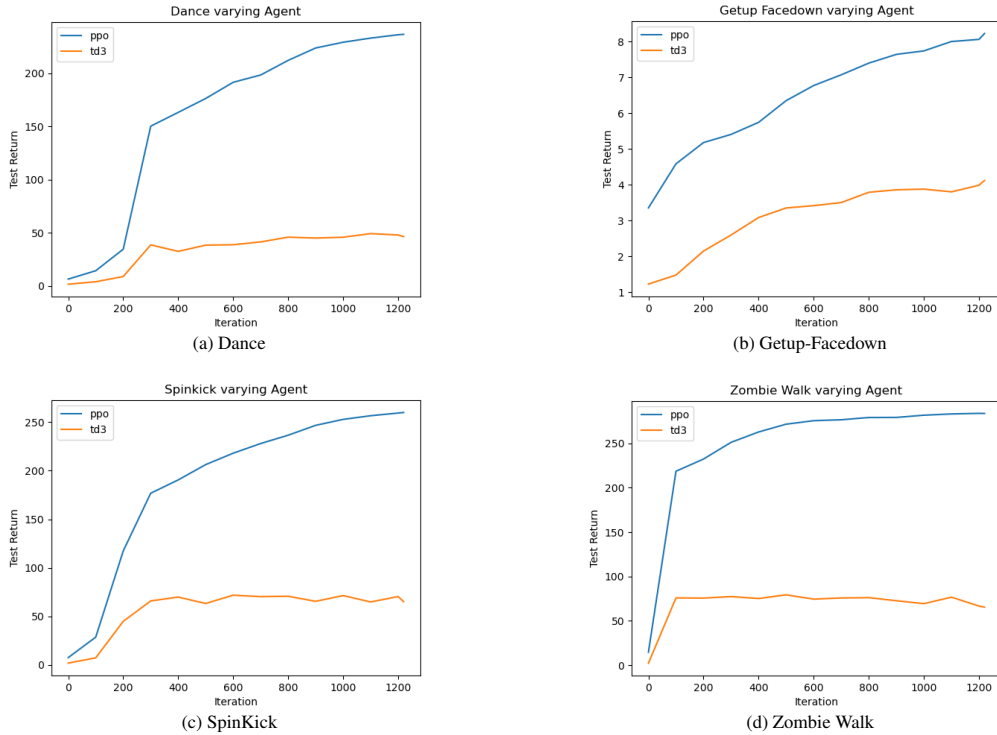
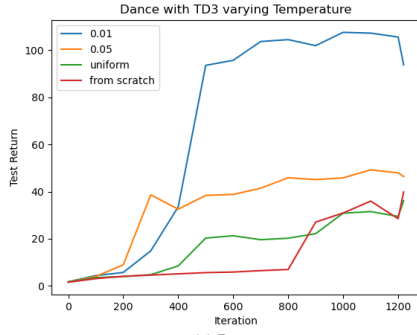
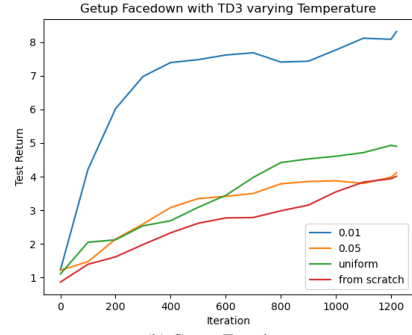


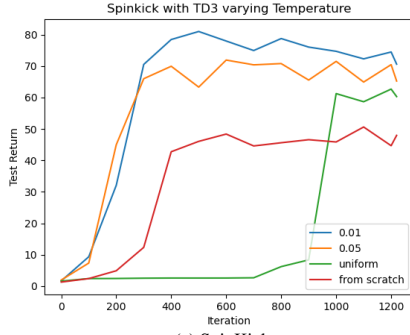
Figure 1: Learning curves for four skills for varying agent configurations.



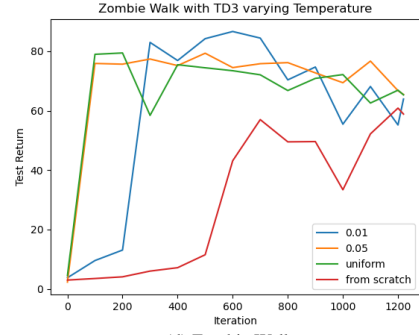
(a) Dance



(b) Getup-Facedown

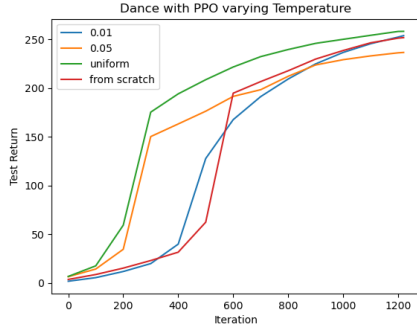


(c) SpinKick

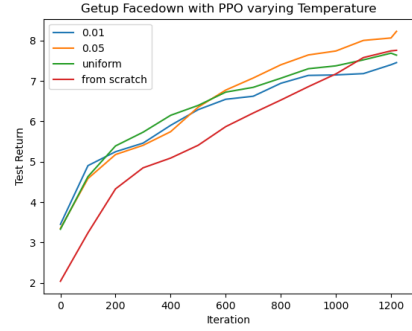


(d) Zombie Walk

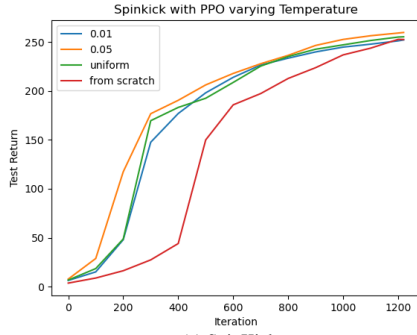
Figure 2: Learning curves for four skills for varying temperature on TD3.



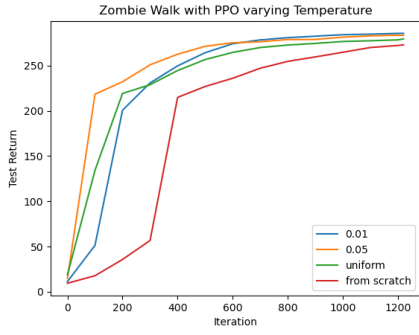
(a) Dance



(b) Getup-Facedown



(c) SpinKick



(d) Zombie Walk

Figure 3: Learning curves for four skills for varying temperature on PPO.

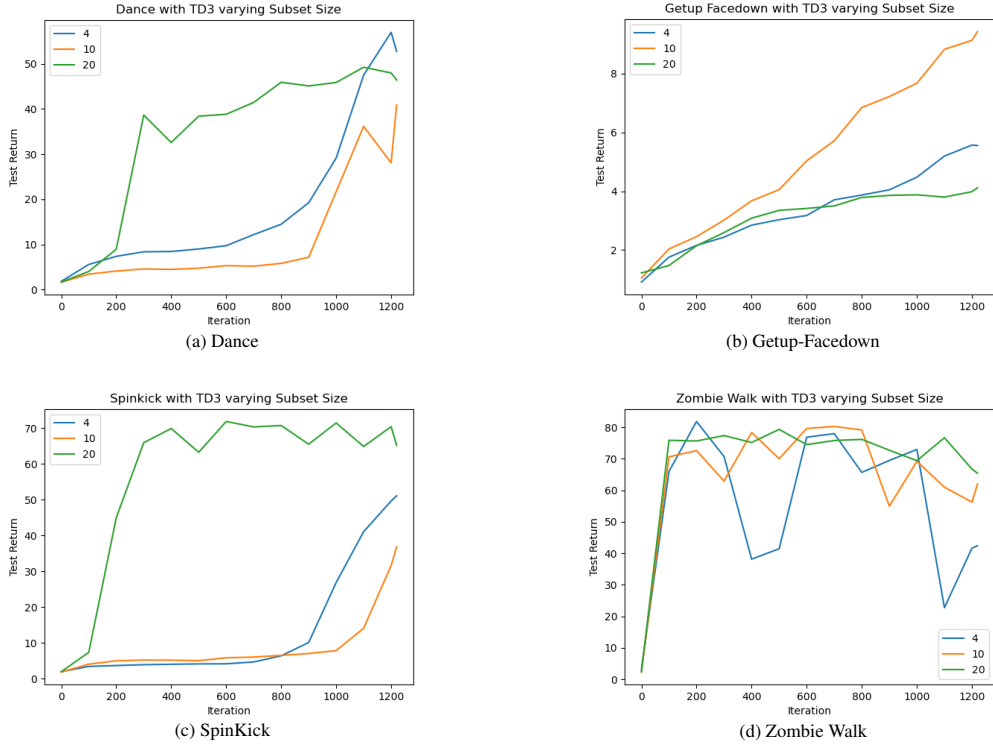


Figure 4: TD3 learning curves for four skills for varying subset size.

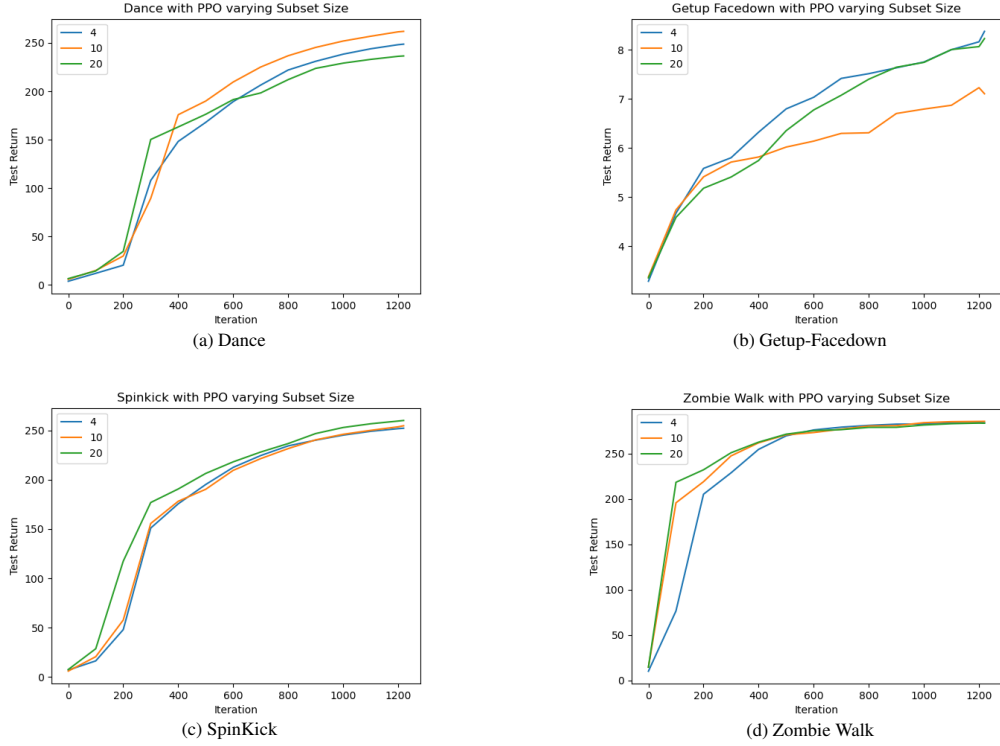


Figure 5: PPO learning curves for four skills for varying subset size.