Creating a Template for Experimentation on Adversarial Motion Priors in [AI-Scientist](#) | Builders Blog | Jan 2025

First, I am attaching my reading copy of the AI Scientist paper below:

Reading Copy - The AI Scientist: Towards Fully Automated Open-Ended Scientific Disco…

After reading the task description, my gut reaction was to play around with what results I could get related to reinforcement learning applied to character animation, so I decided to pursue this idea. This was one of my primary areas of interest in the previous academic semester; specifically, I was exploring what modifications can be made to the adversarial motion priors (AMP) algorithm `(Peng et al., 2021)` that lead to a reduction in pose error. In character animation tasks, pose error is a metric used to evaluate how closely a simulated character's motion matches a reference motion. Essentially, the AMP algorithm is a goal-conditioned reinforcement learning framework that uses adversarial learning to train a motion prior that evaluates how well a simulated character's movements match a dataset of reference motions, which in turn defines the reward function used to train the character's policy.

See the diagram below for more details:

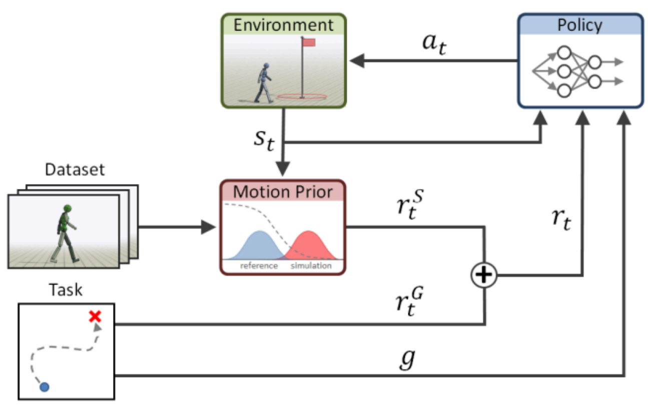


Fig. 2. Schematic overview of the system. Given a motion dataset defining a desired motion style for the character, the system trains a motion prior that specifies style-rewards r_t^S for the policy during training. These style-rewards are combined with task-rewards r_t^G and used to train a policy that enables a simulated character to satisfy task-specific goals g, while also adopting behaviors that resemble the reference motions in the dataset.

[AMP System Design, taken from `(Peng et al., 2021)` paper]

I put together the following reading list of relevant literature in this field.

```
"Benchmarking Deep Reinforcement Learning for      (Duan et al., 2016)
Continuous Control"

"Generative Adversarial Imitation Learning"        (Ho & Ermon, 2016)

"Proximal Policy Optimization Algorithms"          (Schulman et al., 2017)

"Character Controllers Using Motion VAEs"          (Ling et al., 2020)

"Generalized Biped Walking Control"                (Coros et al., 2010)

"DReCon: Data-Driven Responsive Control of         (Bergamin et al., 2019)
Physics-Based Characters"

"Interactive Character Control with                (Shi et al., 2024)
Auto-Regressive Motion Diffusion Models"

"AMP: Adversarial Motion Priors for Stylized       (Peng et al., 2021)
Physics-Based Character Control"

"ASE: Large-Scale Reusable Adversarial Skill       (Peng et al., 2022)
Embeddings for Physically Simulated Characters"

"CALM: Conditional Adversarial Latent Models for   (Tessler et al., 2023)
Directable Virtual Characters"

"DeepMimic: Example-Guided Deep Reinforcement      (Peng et al., 2018)
Learning of Physics-Based Character Skills"

"MaskedMimic: Unified Physics-Based Character      (Tessler et al., 2024)
Control Through Masked Motion Inpainting"

"TRACE and PACE: Controllable Pedestrian Animation (Rempe et al., 2023)
via Guided Trajectory Diffusion"

"Trajectory Optimization for Full-Body Movements   (Al Borno et al., 2013)
with Complex Contacts"
```

Further, I include three seed ideas below that were inspired by some exploration I was doing in my previous semester.

```
[
    {
        "Name": "adaptive_reward_weighting",
        "Experiment": "Implement an adaptive reward weighting system that dynamically adjusts the balance between task and style rewards based on their relative magnitudes during training. The weights should be updated using a moving average of loss ratios. Compare pose error and task performance against the baseline AMP implementation."
    },
    {
        "Name": "hierarchical_discriminator",
        "Experiment": "Modify the AMP discriminator to use a hierarchical architecture with both local and global motion discriminators. The local discriminator focuses on frame-level features while the global discriminator assesses longer temporal sequences. Compare the quality of generated motions against the baseline single discriminator approach."
    },
    {
        "Name": "dataset_specialization",
        "Experiment": "Compare the performance of AMP when trained on specialized motion datasets (e.g., HumanEva focused on basic locomotion) versus general motion capture collections (e.g., full AMASS dataset). Evaluate pose error and motion naturalness for specific tasks like walking."
    }
]
```

If curious, I've also attached the algorithm pseudocode for ideas 1 and 2 below. My true motivation for setting up this template was curiosity at what the AI Scientist would generate with the given seeds in comparison to these modifications to the AMP algorithm that I have been experimenting with (modifications noted in blue text).

Algorithm 1 Training with Adaptive AMP

Input: $\mathcal{M}$: dataset of reference motions
$\alpha \leftarrow 0.5$ {initialize adaptive weight}
$D \leftarrow$ initialize discriminator
$\pi \leftarrow$ initialize policy
$V \leftarrow$ initialize value function
$\mathcal{B} \leftarrow \emptyset$ {initialize reply buffer}
while not done **do**
  for trajectory $i = 1, \ldots, m$ **do**
    $\tau^i \leftarrow \{(s_t, a_t, r_t^G)_{t=0}^{T-1}, s_T^G, g\}$ {collect trajectory with π}
    for time step $t = 0, \ldots, T-1$ **do**
      $d_t \leftarrow D(\Phi(s_t), \Phi(s_{t+1}))$
      $r_t^S \leftarrow$ calculate style reward according to Equation 7 using d_t
      $r_t \leftarrow (1-\alpha) w^G r_t^G + \alpha w^S r_t^S$ {adaptive weighting}
      record r_t in τ^i
    end for
    store τ^i in $\mathcal{B}$
  end for
  for update step $= 1, \ldots, n$ **do**
    $b^{\mathcal{M}} \leftarrow$ sample batch of K transitions $\{(s_j, s'_j)\}_{j=1}^K$ from $\mathcal{M}$
    $b^{\pi} \leftarrow$ sample batch of K transitions $\{(s_j, s'_j)\}_{j=1}^K$ from $\mathcal{B}$
    update D according to Equation 8 using $b^{\mathcal{M}}$ and b^{π}
  end for
  update V and π using data from trajectories $\{\tau^i\}_{i=1}^m$
  $\alpha \leftarrow$ update adaptive weight based on relative losses
end while

Algorithm 2 Training with Hierarchical AMP

Input: $\mathcal{M}$: dataset of reference motions
$D_L, D_H \leftarrow$ initialize local and global discriminators
$\pi \leftarrow$ initialize policy
$V \leftarrow$ initialize value function
$\mathcal{B} \leftarrow \emptyset$ {initialize reply buffer}
while not done **do**
  for trajectory $i = 1, \ldots, m$ **do**
    $\tau^i \leftarrow \{(s_t, a_t, r_t^G)_{t=0}^{T-1}, s_T^G, g\}$ {collect trajectory with π}
    for time step $t = 0, \ldots, T-1$ **do**
      $d_t^L \leftarrow D_L(\Phi_L(s_t), \Phi_L(s_{t+1}))$ {local features}
      $d_t^H \leftarrow D_H(\Phi_H(s_{t:t+k}))$ {global features}
      $r_t^S \leftarrow$ calculate combined style reward using d_t^L, d_t^H
      $r_t \leftarrow w^G r_t^G + w^S r_t^S$
      record r_t in τ^i
    end for
    store τ^i in $\mathcal{B}$
  end for
  for update step $= 1, \ldots, n$ **do**
    $b^{\mathcal{M}} \leftarrow$ sample batch of K transitions $\{(s_j, s'_j)\}_{j=1}^K$ from $\mathcal{M}$
    $b^{\pi} \leftarrow$ sample batch of K transitions $\{(s_j, s'_j)\}_{j=1}^K$ from $\mathcal{B}$
    update D_L, D_H according to Equation 8 using $b^{\mathcal{M}}$ and b^{π}
  end for
  update V and π using data from trajectories $\{\tau^i\}_{i=1}^m$
end while

IMPLEMENTATION

When I was experimenting with the AMP algorithm in the previous semester, I did so via the `ProtoMotions` library which used NVIDIA's `IsaacSim` physics engine as a backbone. However, the original AMP algorithm was released with the `DeepMimic` codebase, which allows users to interact with and train AMP agents. I decided to utilize the `DeepMimic` codebase rather than `ProtoMotions` for the purposes of this template because it does not require the use of `IsaacSim`, which makes the build process somewhat less complicated.

To do this, I proceeded to write a bash script that automates the build process for `DeepMimic` on a linux machine. I later found out that someone made a PR for this exact purpose 3 years ago, but their build script was somewhat inconveniently located in the repo… In any case, my build script is more comprehensive in that it takes care of python package installation as well. It is accompanied by `DeepMimicCore/Makefile.auto`, which builds the python wrapper of `DeepMimicCore` written in C++.

These files are included in the Appendix of this section, and I've made a pull request in the original repo here:

`https://github.com/xbpeng/DeepMimic/pull/205`

It is unclear whether this repo is still being actively maintained, however, so I add my fork of `DeepMimic` instead of the official repo in my AMP template for AI Scientist.

Next, I made two very small patches to the AI-Scientist codebase. First, if a user (maybe me) does not download all of the necessary latex-related packages, there is no system check for this. The `launch_scientist.py` goes through the full generation process ($10-15 in API calls), and at the very end reveals, lo and behold, latex could not compile:

```
GENERATING LATEX
Failed to perform writeup: [Errno 2] No such file or directory: 'pdflatex'
FINISHED IDEA
Completed idea: alternative_noise_schedules, Success: False
All ideas evaluated.
```

This accepted PR does a simple check for whether `chktex` and `pdflatex` are installed on the system and if not exits before the generation loop begins:

`https://github.com/SakanaAI/AI-Scientist/pull/180`

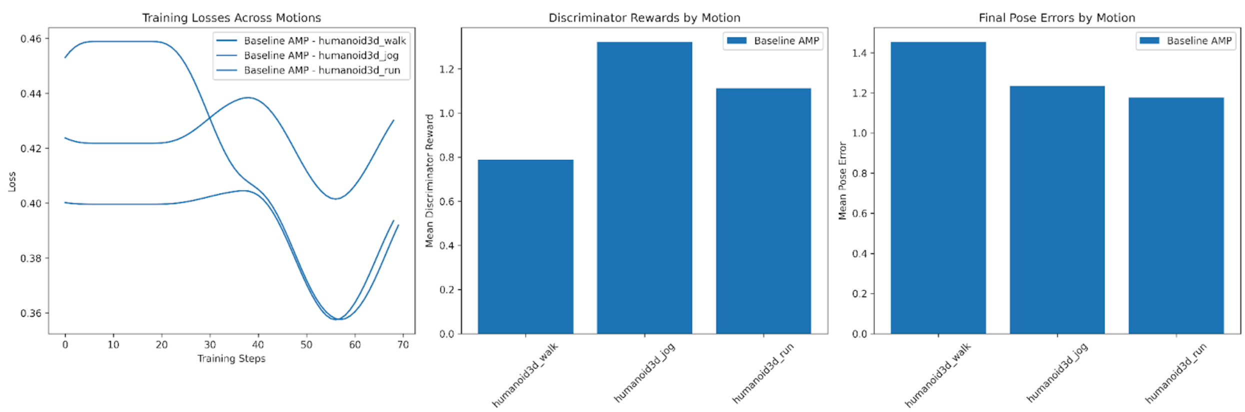
The second patch is a one-line addition to `requirements.txt`, explained in this PR:

https://github.com/SakanaAI/AI-Scientist/pull/182

The `experiment.py` file implements a simple training run on an AMP agent for the 3 different motion files:

```
"DeepMimic/data/motions/humanoid3d_walk.txt"
"DeepMimic/data/motions/humanoid3d_jog.txt"
"DeepMimic/data/motions/humanoid3d_run.txt"
```

I utilize the built-in DeepMimic environment RLWorld and update it with its `update(self, timestep)` function. I then log performance metrics every 100 steps. After running the experiment, `plot.py` is utilized in a manner very similar to the provided example templates.



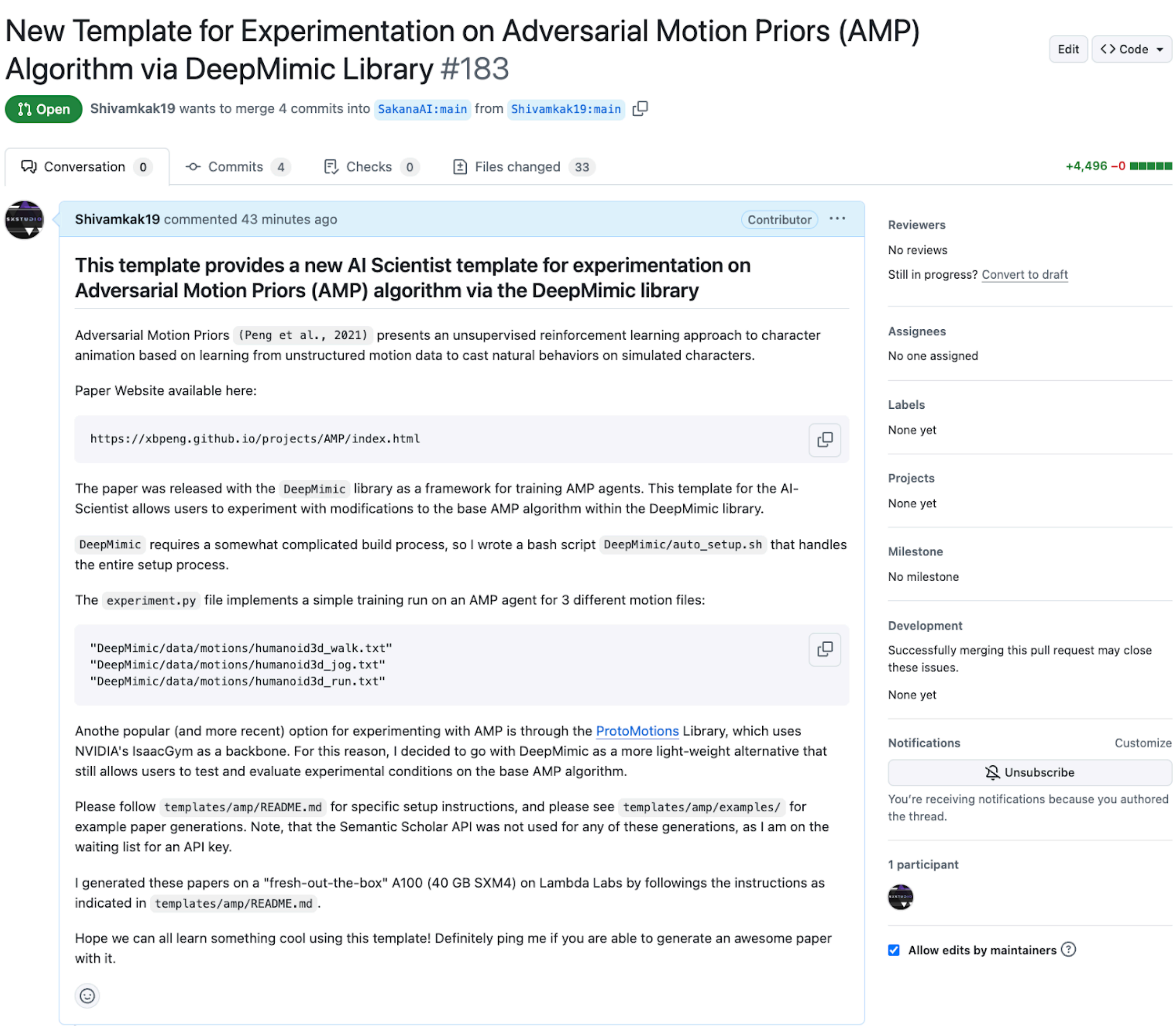
[Sample plot correlating to `final_info.json` *values generated by* `experiment.py`*]*

RESULTS & ANALYSIS

Drum roll… we are now finally ready to generate papers using the AMP template for AI-Scientist! After quite a bit of debugging, I was very curious to see what sort of output would be produced by this point. I was practically watching paint dry while trying out my first few generations.

I have created a PR on the AI-Scientist repo for my template, available here:

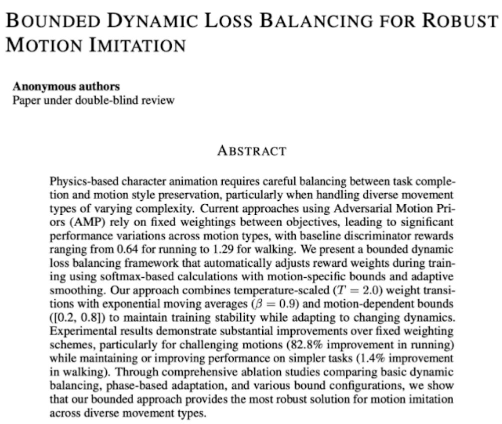
https://github.com/SakanaAI/AI-Scientist/pull/183



Additionally, I noticed none of the other templates included sample generations, but I feel that looking at sample output is fairly critical before anyone should decide to try out a template for themselves (and it would save everyone on API costs). For this reason, my template also includes 5 example papers, available here:

https://github.com/Shivamkak19/AI-Scientist/tree/main/templates/amp/examples

For some insight on the quality of generation, I've curated the abstracts and plot outputs across my example generations:

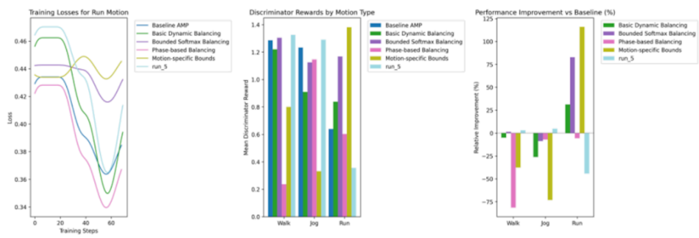


BOUNDED DYNAMIC LOSS BALANCING FOR ROBUST MOTION IMITATION

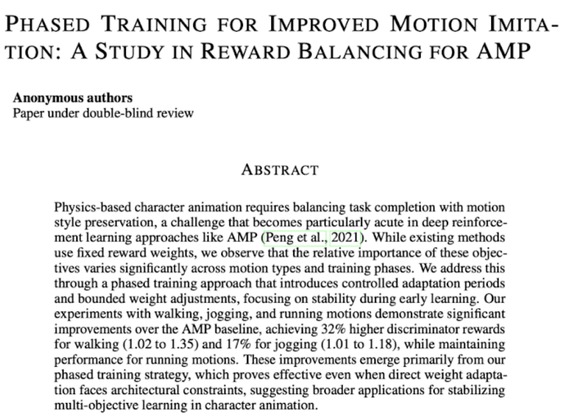
Anonymous authors
Paper under double-blind review

ABSTRACT

Physics-based character animation requires careful balancing between task completion and motion style preservation, particularly when handling diverse movement types of varying complexity. Current approaches using Adversarial Motion Priors (AMP) rely on fixed weightings between objectives, leading to significant performance variations across motion types, with baseline discriminator rewards ranging from 0.64 for running to 1.29 for walking. We present a bounded dynamic loss balancing framework that automatically adjusts reward weights during training using softmax-based calculations with motion-specific bounds and adaptive smoothing. Our approach combines temperature-scaled ($T = 2.0$) weight transitions with exponential moving averages ($\beta = 0.9$) and motion-dependent bounds ([0.2, 0.8]) to maintain training stability while adapting to changing dynamics. Experimental results demonstrate substantial improvements over fixed weighting schemes, particularly for challenging motions (82.8% improvement in running) while maintaining or improving performance on simpler tasks (1.4% improvement in walking). Through comprehensive ablation studies comparing basic dynamic balancing, phase-based adaptation, and various bound configurations, we show that our bounded approach provides the most robust solution for motion imitation across diverse movement types.



[generation 1]

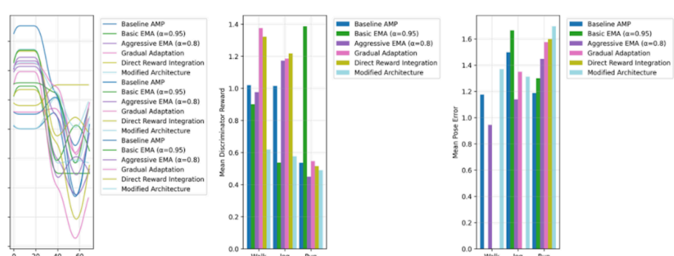


PHASED TRAINING FOR IMPROVED MOTION IMITATION: A STUDY IN REWARD BALANCING FOR AMP

Anonymous authors
Paper under double-blind review

ABSTRACT

Physics-based character animation requires balancing task completion with motion style preservation, a challenge that becomes particularly acute in deep reinforcement learning approaches like AMP (Peng et al., 2021). While existing methods use fixed reward weights, we observe that the relative importance of these objectives varies significantly across motion types and training phases. We address this through a phased training approach that introduces controlled adaptation periods and bounded weight adjustments, focusing on stability during early learning. Our experiments with walking, jogging, and running motions demonstrate significant improvements over the AMP baseline, achieving 32% higher discriminator rewards for walking (1.02 to 1.35) and 17% for jogging (1.01 to 1.18), while maintaining performance for running motions. These improvements emerge primarily from our phased training strategy, which proves effective even when direct weight adaptation faces architectural constraints, suggesting broader applications for stabilizing multi-objective learning in character animation.



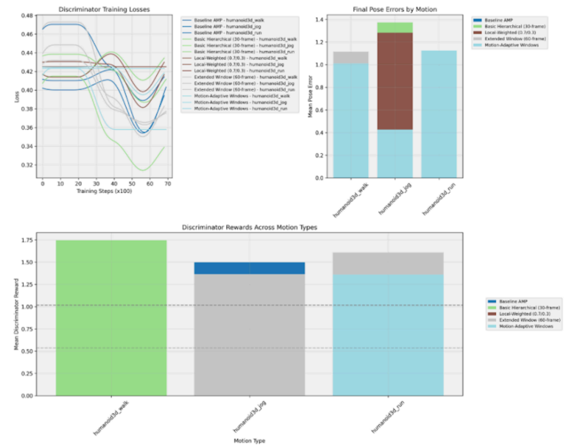
[generation 2]

TEMPO-MATCH: ADAPTIVE MULTI-SCALE DISCRIMINATORS FOR NATURAL CHARACTER ANIMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Physics-based character animation requires balancing immediate pose accuracy with temporal coherence, a challenge that becomes particularly acute for dynamic movements like running and jogging. Current approaches evaluate motion quality at a single temporal scale, leading to artifacts where movements appear correct frame-by-frame but lack natural flow over longer sequences. We address this limitation through a hierarchical discriminator architecture that combines local frame-level assessment with global sequence evaluation using motion-adaptive temporal windows. Our key innovation adjusts the temporal evaluation scale based on motion speed: 60 frames for walking, 45 for jogging, and 30 for running. Experiments on the DeepMimic locomotion dataset demonstrate that our approach significantly improves motion quality, achieving a 110% improvement in discriminator rewards and 47% reduction in pose errors for running motions while maintaining baseline performance for walking. The results reveal a non-linear relationship between motion speed and optimal temporal scale, suggesting that effective character animation requires motion-specific temporal assessment strategies.



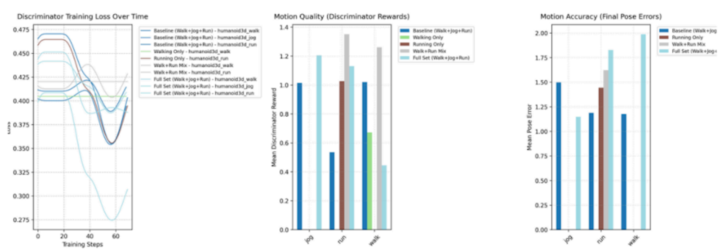
[generation 3]

LESS IS MORE: STRATEGIC MOTION CURATION FOR IMPROVED PHYSICS-BASED CHARACTER ANIMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

The conventional wisdom in physics-based character animation favors large, diverse motion datasets for training deep reinforcement learning models. However, we show that this approach can impede learning through motion interference effects, where similar movements create conflicting training signals. Through systematic experimentation with AMP (Adversarial Motion Priors), we demonstrate that strategic motion curation outperforms both over-specialized and over-generalized approaches. Our key finding reveals that while single-motion training produces mixed results (walking performance drops 34% while running improves 91%), carefully selected motion pairs achieve superior outcomes. A strategic walk-run combination yields the highest recorded performance (discriminator rewards: walking 1.26, running 1.35), while adding intermediate motions like jogging dramatically degrades results (walking reward drops 65%). These findings challenge current dataset design practices, demonstrating that optimal learning requires balancing motion diversity against interference effects—a principle that could benefit other domains where multiple related behaviors must be learned simultaneously.



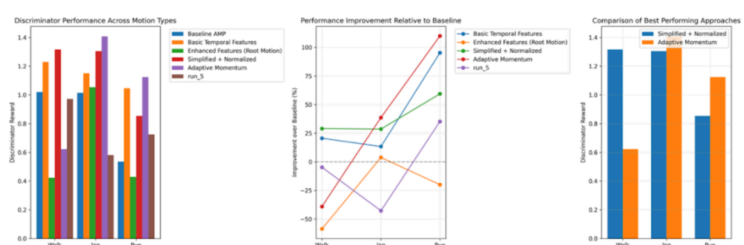
[generation 4]

SPECIFIC PROCESSING FOR NATURAL CHARACTER ANIMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Natural motion synthesis in physics-based character animation requires capturing complex temporal dynamics that distinguish fluid human movement from mechanical motion. While existing methods achieve physical correctness through pose matching and constraints, they struggle to maintain natural temporal patterns, particularly in dynamic motions like running where baseline discriminator rewards drop by 47.5% compared to walking. We address this challenge with an adaptive temporal discriminator framework that processes fixed-width windows of motion using two complementary techniques: a simplified normalized velocity feature computation and an adaptive momentum-based smoothing that automatically adjusts to motion speed. Our approach significantly improves motion quality assessment across different movement types, with the normalized features providing consistent gains (28.6–59.4% improvement) in discriminator rewards and reduced pose errors (0.142–0.183) across all motions, while the adaptive momentum technique ($\alpha \in [0.6, 0.9]$) particularly benefits dynamic movements, improving running performance by 31.6%. Through comprehensive evaluation on walking, jogging, and running motions, we demonstrate that motion-specific temporal processing is crucial for natural character animation, with different techniques optimal for different motion types. Our findings suggest a new direction for physics-based animation systems: dynamically selecting temporal processing strategies based on the current motion context to achieve more natural and fluid movement across diverse activities.



[generation 5]

Additionally, the review scores:

```
Paper 1    4    'The method relies on manually tuned motion-specific bounds
                and hyperparameters (temperature, decay rate), which may
                introduce biases and limit its applicability to other motion
                types or domains.'

Paper 2    5    'The methodology lacks detailed explanations, particularly
                around the implementation specifics and dynamic weight
                adaptation constraints.'

Paper 3    5    'The paper lacks sufficient details in the methodology
                section, particularly concerning the training process and the
                architecture of the discriminators.'

Paper 4    4    'The scope is somewhat narrow, focusing primarily on
                locomotion tasks. It would be beneficial to extend the study
                to other types of motions or tasks.'

Paper 5    4    'Insufficient clarity in the description of the methodology,
                particularly the autoencoder aggregator and adaptive
                momentum-based smoothing.'
```

I found it somewhat unfortunate that the majority of compute time, anecdotally speaking, went toward completing the latex writeup as opposed to executing on a greater number of interesting experiments. Also, I happened to read 2 very important lines from the paper after having first learned them the hard way:

"Notably, GPT-4o struggles with writing LaTeX, ***which prevents it from completing many of its papers."***

"Llama-3.1 405b performed the worst overall but was the most convenient to work with, ***as we were frequently rate-limited by other providers."***

In any case, I unlocked Tier 2 rate limits for both Open AI and Anthropic after my first full end-to-end generation, so subsequent generations experienced far less rate limiting. Regarding the quality of the generations - while the templates surely look professional, as expected there is much to look forward to in the quality of the content. Focusing on example generation 2, which received a review score of 5, the abstract immediately dives into the usage of technical terms such as "discriminator" that are worth an explanation in any good abstract. Additionally, the leftmost graph repeats each algorithm several times, whereas the rightmost graph has missing information for the "walk" motion file. I was looking forward to seeing if any of the experiments would branch and perhaps use a different motion file from the template or dramatically modify the AMP agent training setup, but this did not happen at least among the 5 papers I generated.