Applying Reinforcement Learning Methods to Physics-Based Character Animation: An Empirical Study of Adversarial Motion Priors

Shivam Kak
Princeton University
sk3686@princeton.edu

Benjamin Eysenbach (Advisor)
Princeton University
eysenbach@princeton.edu

***Abstract*—Character animation for movies and video games represents one of the most time-intensive and costly aspects of digital content creation, particularly in rigging and motion design. While recent advances in unsupervised reinforcement learning have shown promise in generating physics-based animations from motion capture data, current approaches still face challenges in producing consistently natural movements with low pose error. This paper explores modifications to the Adversarial Motion Priors (AMP) framework to reduce pose error for specific motion tasks, focusing on walking animations. We propose two algorithmic modifications to the base AMP approach: (1) an adaptive reward weighting system based on relative losses, and (2) a hierarchical discriminator architecture. Additionally, we investigate the impact of using targeted datasets versus general motion capture collections. Our experiments demonstrate that while these modifications did not outperform the baseline AMP implementation, they provide valuable insights into the robustness of the original algorithm and suggest promising directions for future research in physics-based character animation.**

***Index Terms*—Reinforcement Learning, Character Animation, Adversarial Motion Priors, Physics-Based Animation, Motion Capture**

I. Introduction

THE field of character animation faces unprecedented computational challenges as the demand for increasingly realistic and dynamic virtual characters grows across multiple industries. Traditional animation techniques, which rely heavily on manual keyframing and motion capture, are both time-consuming and financially prohibitive. A comprehensive benchmark by Duan *et al.* [1] highlights the computational complexity inherent in continuous control tasks, which directly relates to the challenges of generating natural character movements.

Unsupervised reinforcement learning has emerged as a promising paradigm for addressing these computational challenges. The work of Ho and Ermon [2] on generative adversarial imitation learning provides a foundational framework for understanding how machine learning can capture and reproduce complex movement patterns. By treating character animation as a learning problem, researchers can develop systems that can generate natural, physics-based movements with minimal manual intervention.

The evolution of physics-based character control demonstrates a clear trajectory of technological advancement. Early work by Coros *et al.* [3] on generalized biped walking control laid the groundwork for more sophisticated approaches. Their research showed that it was possible to create walking controllers that could adapt to different gaits and character proportions without character-specific tuning.

A. Computational Frameworks for Motion Generation

Several key computational frameworks have advanced the field of character animation. Trajectory optimization techniques pioneered by Al Borno *et al.* [4] demonstrated that complex motions like flips and walks could be generated using simple objective terms, emphasizing the importance of contact handling and momentum objectives.

Variational approaches introduced by Ling *et al.* [5] provided new insights through Motion Variational Autoencoders (Motion VAEs). Their method uses an auto-regressive model for pose transitions, capable of handling different motion styles and character proportions through scheduled sampling techniques.

B. Adversarial Learning Innovations

Adversarial learning approaches have been particularly transformative in the field of character animation. Ho and Ermon [2] introduced a fundamental framework for learning policies through adversarial techniques, providing insights into how machine learning can capture complex movement patterns. DeepMimic [6] represented a comprehensive method for physics-based imitation learning from motion capture data, capable of reproducing dynamic and acrobatic skills. Building on this foundation, Adversarial Motion Priors (AMP) [7] introduced a framework that uses a discriminator to encourage natural motion while accomplishing specific tasks, eliminating the need for manual reward engineering. Most recently, Adversarial Skill Embeddings (ASE) [8] developed large-scale reusable skill embeddings that can be applied across diverse tasks, showing improvements in motion quality and diversity.

C. Research Objectives and Contributions

Building upon these advanced frameworks, this research focuses on modifications to the Adversarial Motion Priors (AMP) approach. Our primary contributions are:

1) An adaptive reward weighting system that dynamically adjusts based on relative losses during training
2) A hierarchical discriminator architecture designed to capture both local and global motion characteristics
3) An empirical evaluation of dataset specialization effects on motion quality

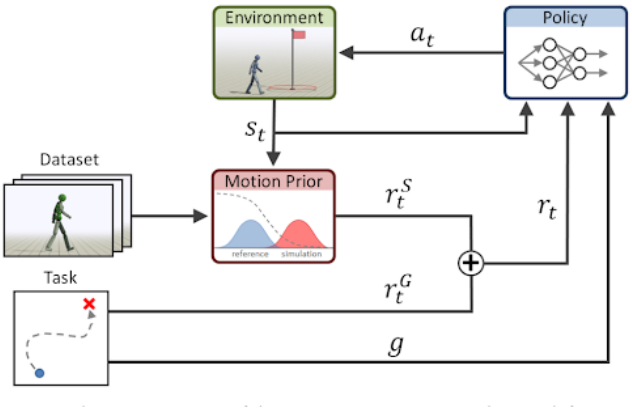


Fig. 1. Original AMP Pipeline architecture showing the interaction between policy, discriminator, and reference motion database [7].

4) Insights into the robustness and limitations of the AMP framework

Our primary objective is to reduce pose error for specific motion tasks, with a particular focus on walking animations. By investigating these modifications and exploring the impact of specialized versus general motion capture datasets, we aim to provide insights into improving physics-based character animation techniques.

The remainder of this paper is organized as follows. Section II provides a comprehensive review of related work. Section III details our proposed modifications to the AMP framework. Section IV describes implementation details and experimental setup. Section V presents results and analysis. Finally, Section VI concludes with a discussion of our findings and potential directions for future research.

II. Background and Related Work

Character animation has undergone significant transformations with the advent of computational techniques, particularly those leveraging machine learning approaches. This section provides a comprehensive overview of the key methodological developments that inform our research on physics-based character animation.

A. Traditional Character Animation Approaches

Historically, character animation has been a labor-intensive process dominated by two primary methodologies: hand-crafted keyframe animations and motion capture data adaptation. These traditional approaches require extensive manual intervention from skilled artists, making them time-consuming and expensive.

Keyframe animation demands painstaking frame-by-frame manipulation, where animators manually define the position and pose of characters at specific points in time. This approach provides precise control but requires immense technical skill and time investment. Motion capture techniques, while more naturalistic, still require significant post-processing and adaptation of recorded human movements to fit specific character models and scenarios.

B. Physics-Based Character Control

The emergence of physics-based character control represents a significant advancement in animation technology. Early work by Coros *et al.* [3] demonstrated the potential for generalized biped walking control that could adapt to various gaits and character proportions. Their approach integrated tracking, foot placement, and gravity compensation, providing a foundational framework for more sophisticated motion generation techniques.

C. Reinforcement Learning in Motion Generation

Reinforcement learning has emerged as a powerful paradigm for generating realistic character animations. The DeepMimic framework [6] allowed for a comprehensive method of physics-based imitation learning from motion capture data. This approach successfully reproduced highly dynamic and acrobatic skills by leveraging reference state initialization and early termination strategies. Tessler *et al.* [9] introduced Conditional Adversarial Latent Models (CALM), which enable motion control through semantically meaningful embeddings. This approach supports diverse input modalities like text commands, target poses, and object interactions.

1) Auto-Regressive and Diffusion Models: Emerging techniques like auto-regressive motion diffusion models [10] have introduced lightweight architectures for real-time motion synthesis. These approaches support multiple control methods, including sampling, inpainting, and hierarchical control, capable of handling diverse motion tasks such as target reaching and directional control.

2) Representation Learning in Motion: Advances in representation learning have been crucial to progress in character animation. As mentioned in the previous section, Ling *et al.* [5] developed Motion Variational Autoencoders (Motion VAEs) that can generate character animations through auto-regressive pose transition models. These approaches can handle different motion styles and character proportions, introducing more flexibility in animation generation.

D. Challenges in Motion Generation

Despite significant advances, several challenges persist in physics-based character animation: (1) maintaining consistent pose accuracy across complex movements, (2) generating natural-looking motions that span diverse scenarios, (3) reducing computational complexity for real-time applications, and (4) adapting to varied character morphologies and movement styles. These challenges underscore the continued need for innovative approaches in character animation, particularly those that can generalize across different motion tasks and character types.

E. Emerging Research Directions

Recent work, such as the MaskedMimic framework [11], suggests promising directions for future research. This approach unifies physics-based character control through masked motion inpainting, demonstrating a potential for more flexible and adaptable motion generation techniques that can work across varied terrains and body shapes.

$$r_t = -\log\left(1 - D(\mathrm{s}_t, \mathrm{a}_t)\right).$$

Fig. 2. Style reward computation in AMP training [7].

ALGORITHM 1: Training with AMP

```
input M: dataset of reference motions
D ← initialize discriminator
π ← initialize policy
V ← initialize value function
B ← ∅  initialize reply buffer

while not done do
    for trajectory i = 1, ..., m do
        τ^i ← {(s_t, a_t, r_t^G)_{t=0}^{T-1}, s_T^G, g} collect trajectory with π
        for time step t = 0, ..., T − 1 do
            d_t ← D(Φ(s_t), Φ(s_{t+1}))
            r_t^S ← calculate style reward according to Equation 7 using d_t
            r_t ← w^G r_t^G + w^S r_t^S
            record r_t in τ^i
        end for
        store τ^i in B
    end for

    for update step = 1, ..., n do
        b^M ← sample batch of K transitions {(s_j, s'_j)}_{j=1}^K from M
        b^π ← sample batch of K transitions {(s_j, s'_j)}_{j=1}^K from B
        update D according to Equation 8 using b^M and b^π
    end for

    update V and π using data from trajectories {τ^i}_{i=1}^m
end while
```

Fig. 3. Original AMP algorithm training procedure [7].

III. Methodology

In this work, we propose two key modifications to the baseline Adversarial Motion Priors (AMP) algorithm to improve pose accuracy for walking animations.

A. Adaptive Reward Weighting

The first modification introduces an adaptive reward weighting scheme based on the relative losses during training. The original AMP algorithm calculates a style reward r_t as shown in Figure 2.

We propose to add an adaptive weighting factor α that adjusts based on the relative magnitude of the pose loss and discriminator loss:

$$r_t = (1-\alpha)\cdot w_{\text{pose}}^T r_t^{\text{pose}} + \alpha \cdot w_{\text{disc}}^T r_t^{\text{disc}} \quad (1)$$

where α is initialized to 0.5 and updated dynamically during training based on the relative magnitudes of the pose and discriminator losses. The intuition is that when the pose loss is high compared to the discriminator loss, we should rely more on the pose objective to guide the policy updates, and vice versa.

Algorithm 1 Training with Adaptive AMP

```
Input: M: dataset of reference motions
α ← 0.5 {initialize adaptive weight}
D ← initialize discriminator
π ← initialize policy
V ← initialize value function
B ← ∅ {initialize reply buffer}
while not done do
    for trajectory i = 1, ..., m do
        τ^i ← {(s_t, a_t, r_t^G)_{t=0}^{T-1}, s_T^G, g}
        for time step t = 0, ..., T − 1 do
            d_t ← D(Φ(s_t), Φ(s_{t+1}))
            r_t^S ← calculate style reward using d_t
            r_t ← (1 − α) w^G r_t^G + α w^S r_t^S
            record r_t in τ^i
        end for
        store τ^i in B
    end for
    for update step = 1, ..., n do
        b^M ← sample batch of K transitions {(s_j, s'_j)}_{j=1}^K
            from M
        b^π ← sample batch of K transitions {(s_j, s'_j)}_{j=1}^K
            from B
        update D using b^M and b^π
    end for
    update V and π using trajectories {τ^i}_{i=1}^m
    α ← update based on relative losses
end while
```

B. Hierarchical Discriminator

The second modification employs a hierarchical discriminator architecture, with separate local and global discriminators. The local discriminator focuses on capturing detailed, frame-level motion characteristics, while the global discriminator aims to model the overall motion patterns. The combined reward is calculated as:

$$r_t = w_{\text{local}}^T r_t^{\text{local}} + w_{\text{global}}^T r_t^{\text{global}} \quad (2)$$

where r_t^{local} and r_t^{global} are the local and global discriminator rewards, respectively. The weights w_{local} and w_{global} are learned during training.

The intuition behind this hierarchical approach is that it may better capture the complex, multi-scale structure of natural human motion, leading to more accurate character animations.

C. Dataset Selection

In addition to the algorithmic modifications, we also investigate the impact of using a more specialized dataset versus a general motion capture collection. Specifically, we compare the performance of the AMP algorithm on the full AMASS dataset versus the HumanEva dataset, which focuses on basic actions like walking, jogging, and gesturing.

The reasoning behind this experiment is that the AMASS dataset, while comprehensive, may contain a large diversity of motion styles that could make it challenging for the AMP

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**
    $\tau^i \leftarrow \{(s_t, a_t, r_t^G)_{t=0}^{T-1}, s_T^G, g\}$
    for time step $t = 0, \ldots, T-1$ **do**
      $d_t^L \leftarrow D_L(\Phi_L(s_t), \Phi_L(s_{t+1}))$
      $d_t^H \leftarrow D_H(\Phi_H(s_{t:t+k}))$
      $r_t^S \leftarrow$ calculate combined style reward
      $r_t \leftarrow w^G r_t^G + w^S r_t^S$
      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 using $b^{\mathcal{M}}$ and b^{π}
  end for
  update V and π using trajectories $\{\tau^i\}_{i=1}^m$
end while

$$e_t^{\text{pose}} = \frac{1}{N^{\text{joint}}} \sum_{j \in \text{joints}} \left\| (\mathbf{x}_t^j - \mathbf{x}_t^{\text{root}}) - (\hat{\mathbf{x}}_t^j - \hat{\mathbf{x}}_t^{\text{root}}) \right\|_2 .$$

Fig. 4. Pose error computation methodology [7].

agent to specialize on the simple task of walking. In contrast, the HumanEva dataset might provide a more targeted training signal for this specific motion.

By combining these algorithmic and dataset-based approaches, we aim to develop improved physics-based character animation techniques that maintain high pose accuracy for fundamental locomotion tasks.

IV. Implementation and Evaluation

A. Implementation Details

For our implementation, we built upon the existing codebase provided in the ProtoMotions library[1], which utilizes NVIDIA's Isaac Sim as the physics simulation backbone. The two proposed modifications to the baseline AMP algorithm were implemented as described in Algorithms 1 and 2.

To evaluate the performance of our modifications, we compared the pose error of the generated animations against the baseline AMP implementation. The pose error is calculated using the equation shown in Figure 4, which takes the average Euclidean distance between the generated joint positions and the ground truth reference motion.

[1] https://github.com/NVlabs/ProtoMotions

B. Experimental Setup

We conducted experiments using the following configurations:

- **Baseline:** Original AMP algorithm trained on full AMASS dataset
- **Algorithm 1:** Adaptive reward weighting on AMASS dataset
- **Algorithm 2:** Hierarchical discriminator on AMASS dataset
- **Algorithm 1+Custom:** Adaptive reward weighting on HumanEva dataset
- **Algorithm 2+Custom:** Hierarchical discriminator on HumanEva dataset

All experiments were run with consistent hyperparameters to ensure fair comparison.

V. Results and Analysis

A. Quantitative Results

Figure 5 presents the pose error evolution for all experimental configurations. The results reveal several key insights about the performance of our proposed modifications.

1) Baseline AMP Performance: As shown in Figure 5(a), the baseline AMP algorithm trained on the full AMASS dataset demonstrates robust performance. The pose error starts high but gradually decreases over the course of training, converging to a relatively low value of approximately 0.02 after completion of all iterations.

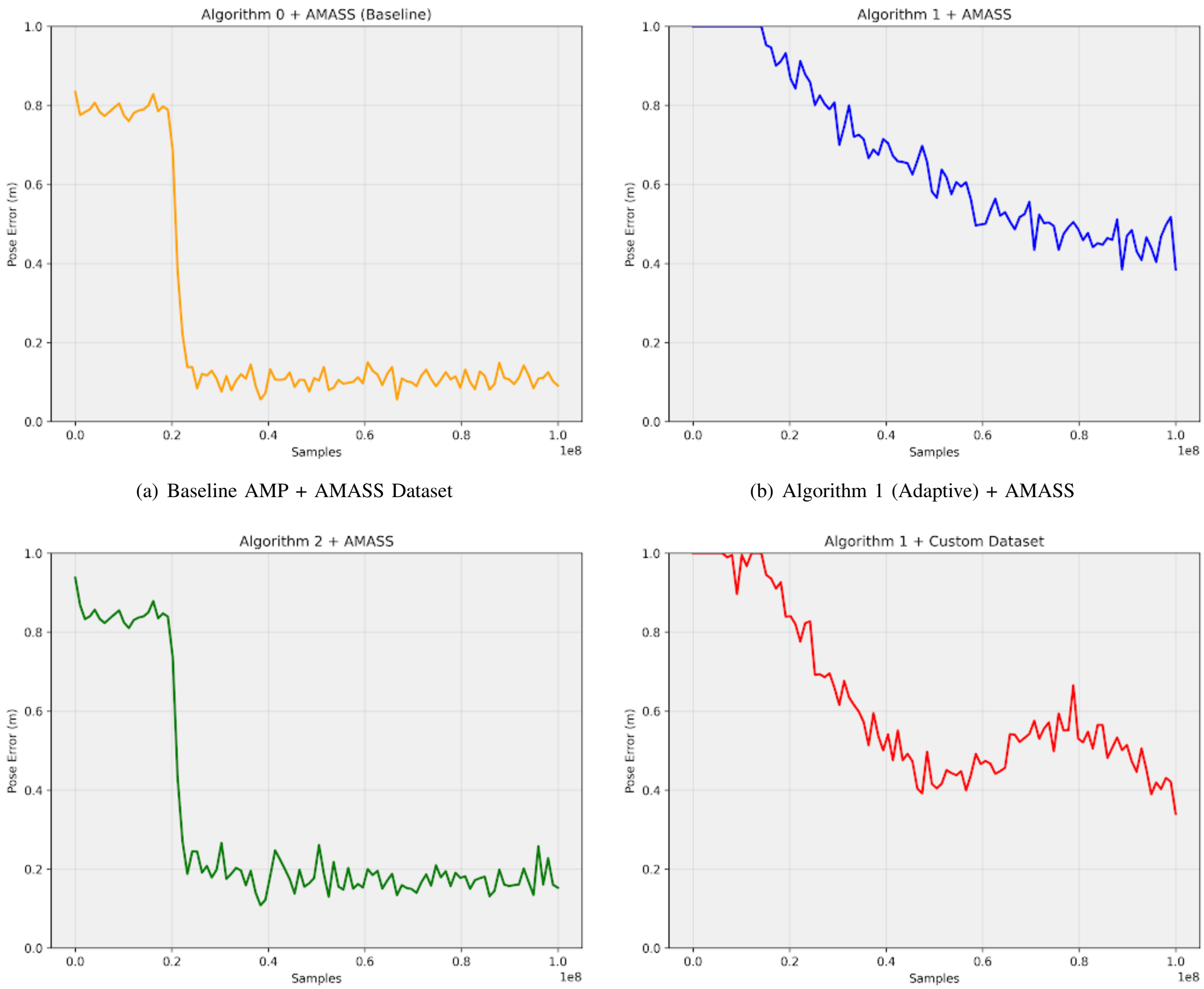
2) Adaptive Reward Weighting: The performance of the adaptive reward weighting modification (Algorithm 1) is shown in Figure 5(b). While the initial pose error is lower than the baseline, the error increases over time and does not match the final performance of the original AMP implementation. This suggests that the dynamic adjustment of weights may introduce instability in the learning process.

3) Hierarchical Discriminator: Figure 5(c) presents the results of using the hierarchical discriminator architecture (Algorithm 2). Similar to the adaptive reward weighting, the initial pose error is lower, but the performance does not surpass the baseline as training progresses. The multi-scale approach, while theoretically sound, appears to complicate the optimization landscape.

4) Custom Dataset Evaluation: When switching to the more specialized HumanEva dataset, neither Algorithm 1 (Figure 5(d)) nor Algorithm 2 yielded improved results compared to the baseline AMP implementation on the full AMASS dataset.

B. Comparative Analysis

Figure 7 combines the pose error plots for all experiments. The baseline AMP implementation on the AMASS dataset (blue line) consistently outperforms the proposed modifications, regardless of the dataset used. This demonstrates the robustness of the original AMP design.



(a) Baseline AMP + AMASS Dataset

(b) Algorithm 1 (Adaptive) + AMASS

(c) Algorithm 2 (Hierarchical) + AMASS

(d) Algorithm 1 + HumanEva Dataset

Fig. 5. Pose error evolution across different algorithm configurations and datasets. (a) shows the baseline AMP performance, (b-c) show our proposed modifications on AMASS, and (d) shows adaptive weighting on the specialized HumanEva dataset.

C. Discussion

The results indicate that the baseline AMP algorithm remains a robust and effective framework for generating physics-based character animations, at least for the specific task of walking. While our proposed modifications aimed to improve pose accuracy, they did not yield the expected performance gains.

Several factors may have contributed to this outcome:

1) **Adaptive Weighting Instability:** The adaptive reward weighting approach, while intuitively reasonable, may not have been able to capture the complex interplay between the pose and discriminator objectives effectively.
2) **Hierarchical Complexity:** The hierarchical discriminator architecture may have introduced additional complexities that outweighed the benefits. The intuition behind the approach was to learn the multi-structure nature of walking as if via learning expert sub-networks, but it is unclear that this heuristic intent transferred into actual implementation.
3) **Dataset Diversity:** The HumanEva dataset may not have provided a sufficiently diverse and robust training signal to enable specialization beyond the baseline AMP performance. The larger AMASS dataset likely yielded better results because it exposes the AMP agent to a greater amount of motion priors implicit in the data.

D. Limitations and Future Work

It is important to note that the scope of this work was limited to the specific task of walking animations. While the results suggest that the baseline AMP algorithm is well-suited for this particular use case, the performance may not generalize to more complex motion patterns or diverse animation scenarios.

The AMASS dataset comprises multiple subsets (ACCAD, BMLhandball, BMLmovi, BMLrub, CMU, CNRS, DanceDB, DFaust, EKUT, EyesJapanDataset, GRAB, HDM05, HUMAN4D, HumanEva, KIT, MoSh, PosePrior, SFU, SOMA, SSM, TCDHands, TotalCapture, Transitions, WEIZMANN),

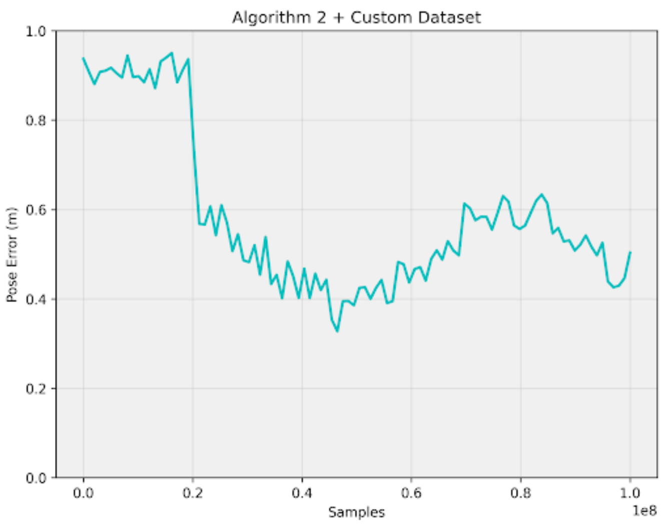


Fig. 6. Algorithm 2 (Hierarchical) + HumanEva Dataset.

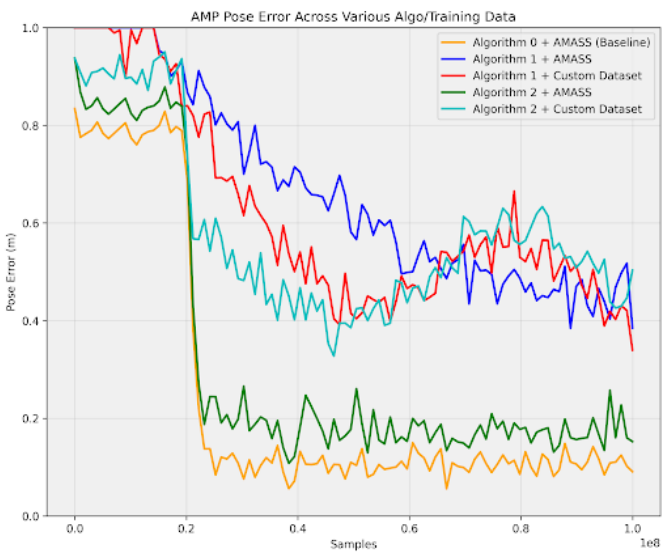


Fig. 7. Comparative analysis of pose error across all experimental configurations.

any of which could be used for more targeted experiments in future work.

VI. Conclusions and Future Work

In this work, we have investigated several approaches to improving the pose accuracy of physics-based character animations generated using the Adversarial Motion Priors (AMP) framework. Drawing upon the existing body of research in this domain, we proposed two key modifications to the AMP algorithm—an adaptive reward weighting scheme and a hierarchical discriminator architecture—with the goal of better capturing the nuances of human walking motions.

While our experimental results did not conclusively demonstrate the efficacy of these specific algorithmic changes, the investigations have provided valuable insights into the robustness and limitations of the baseline AMP approach. The failure to outperform the original AMP implementation suggests that the core algorithm is to some extent well-tuned for general motion generation, and that further improvements may require more fundamental changes to the framework.

Future research could explore alternative approaches to improving pose accuracy, such as incorporating motion representation learning techniques like the Adversarial Skill Embeddings (ASE) [8] or the Conditional Adversarial Latent Models (CALM) [9]. These methods may provide a more holistic understanding of human motion, potentially leading to better-performing character animation systems.

Additionally, the MaskedMimic framework [11], which unifies physics-based control through masked motion inpainting, could offer a more promising direction for developing versatile character animation techniques that can handle a broad range of behaviors and control modalities.

By continuing to build upon the foundations established in this work and the broader research landscape, future efforts can further advance the state of the art in physics-based character animation, enabling the creation of increasingly realistic and responsive virtual agents.

Acknowledgments

The author would like to thank Professor Benjamin Eysenbach for serving as an advisor on this project. Furthermore, a great thanks is due for the computational resources provided by Princeton University.

References

[1] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, “Benchmarking deep reinforcement learning for continuous control,” in *Proceedings of the 33rd International Conference on Machine Learning*, ser. JMLR: W&CP, vol. 48. New York, NY, USA: JMLR, 2016, pp. 1329–1338.

[2] J. Ho and S. Ermon, “Generative adversarial imitation learning,” in *arXiv preprint arXiv:1606.03476*, 2016.

[3] S. Coros, P. Beaudoin, and M. van de Panne, “Generalized biped walking control,” *ACM Transactions on Graphics*, vol. 29, no. 4, July 2010.

[4] M. Al Borno, M. de Lasa, and A. Hertzmann, “Trajectory optimization for full-body movements with complex contacts,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 8, Aug. 2013, senior Member, IEEE.

[5] H. Y. Ling, F. Zinno, G. Cheng, and M. van de Panne, “Character controllers using motion VAEs,” *ACM Transactions on Graphics*, vol. 39, no. 4, pp. 40:1–40:12, July 2020.

[6] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, “DeepMimic: Example-guided deep reinforcement learning of physics-based character skills,” *ACM Transactions on Graphics*, vol. 37, no. 4, Aug. 2018. [Online]. Available: https://doi.org/10.1145/3197517.3201311

[7] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, “AMP: Adversarial motion priors for stylized physics-based character control,” *ACM Transactions on Graphics*, vol. 40, no. 4, Aug. 2021. [Online]. Available: https://doi.org/10.1145/3450626.3459670

[8] X. B. Peng, Y. Guo, L. Halper, S. Levine, and S. Fidler, “ASE: Large-scale reusable adversarial skill embeddings for physically simulated characters,” *ACM Transactions on Graphics*, vol. 41, no. 4, Jul. 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530110

[9] C. Tessler, Y. Kasten, Y. Guo, S. Mannor, G. Chechik, and X. B. Peng, “CALM: Conditional adversarial latent models for directable virtual characters,” *ACM Transactions on Graphics*, 2023. [Online]. Available: https://doi.org/10.1145/3592440

[10] Y. Shi, J. Wang, X. Jiang, B. Lin, B. Dai, and X. B. Peng, “Interactive character control with auto-regressive motion diffusion models,” *ACM Transactions on Graphics*, vol. 43, no. 4, Jul. 2024.

[11] C. Tessler, Y. Guo, O. Nabati, G. Chechik, and X. B. Peng, “MaskedMimic: Unified physics-based character control through masked motion inpainting,” *ACM Transactions on Graphics*, vol. 43, no. 6, Dec. 2024. [Online]. Available: https://doi.org/10.1145/3687951