InterLevel: Synthesizing Stair-Navigation Skills in Character-Scene Interactions

Jack Shilton[1,2], **Boeun Kim**[1, 3], **Hyung Jin Chang**[1]

[1]University of Birmingham
[2]Keio University
[3]Korea Electronics Technology Institute
shiltonjack@gmail.com, {b.e.kim, h.j.chang}@bham.ac.uk

Abstract

Synthesising realistic and responsive virtual characters capable of traversing complex 3D environments remains a challenging task. Existing approaches have generated convincing human motions on a two-dimensional plane; however, many neglect the necessity of traversal through three-dimensional settings. Consequently, these methods often result in unrealistic motions, with frequent clipping and floating feet. Moreover, some methods rely on auxiliary data, such as height maps, which are difficult to obtain. To address these limitations, we present **InterLevel**, a novel reinforcement learning approach for training physically simulated characters to navigate multi-level environments. Our system leverages a novel reward function that encourages the character's movement within 3D space, and we utilise a wide variety of stair gradients, dimensions, and orientations to ensure a generalised policy. **InterLevel** achieves an average progress of 46.8%, significantly exceeding the 30.1% of the current state-of-the-art method. Furthermore, the visualizations demonstrate a clear qualitative gap between our method and the existing method. While the existing method fails in most cases, our **InterLevel** consistently generates plausible motions, even on stairs with large gradients.

Introduction

Simulation of a 3D traversal policy can be a ground-breaking stride in allowing virtual characters to quickly navigate various floors of a given world space, opening the field to new possibilities for simulated avatars in computer interaction. The ability to realistically traverse staircases and multi-level environments holds many potential applications. In fields such as robotics and video games, it allows humanoid avatars to traverse real-world environments, including elevation changes. While previous works (Tessler et al. 2023; Lee and Joo 2023; Peng et al. 2022, 2018; Rudin et al. 2022) have achieved realistic human motion simulations, they still lack methods to adequately traverse multi-level environments, often resulting in characters feeling disconnected.

Some of these works (Peng et al. 2018; Rudin et al. 2022) simulate this process using auxiliary 3D environment information, such as height maps. However, these models remain constrained by their dependency on access to auxiliary information that is difficult or sometimes impossible to obtain.

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

To address this limitation, our proposed method focuses on developing a policy that can generate practical and reliable motions that can traverse various environments using realistic data without a need for auxiliary information. This will allow virtual characters to navigate the environment more realistically and believably and possibly enable humanoid robots to perform these actions in a real environment.

However, synthesising physically plausible navigational motion in a real environment is a complex challenge. Simply repeating prerecorded motion capture data of a human climbing up stairs quickly leads to unrealistic results, with frequent clipping and floating feet if not perfectly aligned with the environment (Hassan et al. 2021; Holden et al. 2020). Physics-based methods allow for more flexibility when performing these actions (Pan et al. 2023; Peng et al. 2018); however, consistently simulating these motions to allow a character to climb stairs of varying dimensions and shapes is an ongoing challenge. Stairs introduce a unique challenge for traversal, as the humanoid must precisely and consistently plant their feet on each step before shifting their centre of mass up the stairs.

This work presents InterLevel, a reinforcement learning approach that allows physically simulated characters to climb stairs realistically and consistently. Our system uses a novel reward function that encourages the avatar's movement within a 3-dimensional space. We use various generated staircases at multiple angles to allow for generalising unseen staircases. Compared to prior work, InterLevel shows more realistic and precise climbing motions and can handle numerous stair gradients, dimensions, and numbers at various orientations from the character. Our system provides the following contributions: (1) A reinforcement learning approach to train realistic physics-based humanoid characters to navigate varied terrain. (2) A novel reward function that encourages the character to follow a given trajectory. (3) Characters that can navigate various stairs with greater stability and efficiency compared to prior work.

Related Work

Data-Driven 3D Motion Generation

Data-driven motion generation techniques, such as motion matching, have shown impressive results in creating realistic

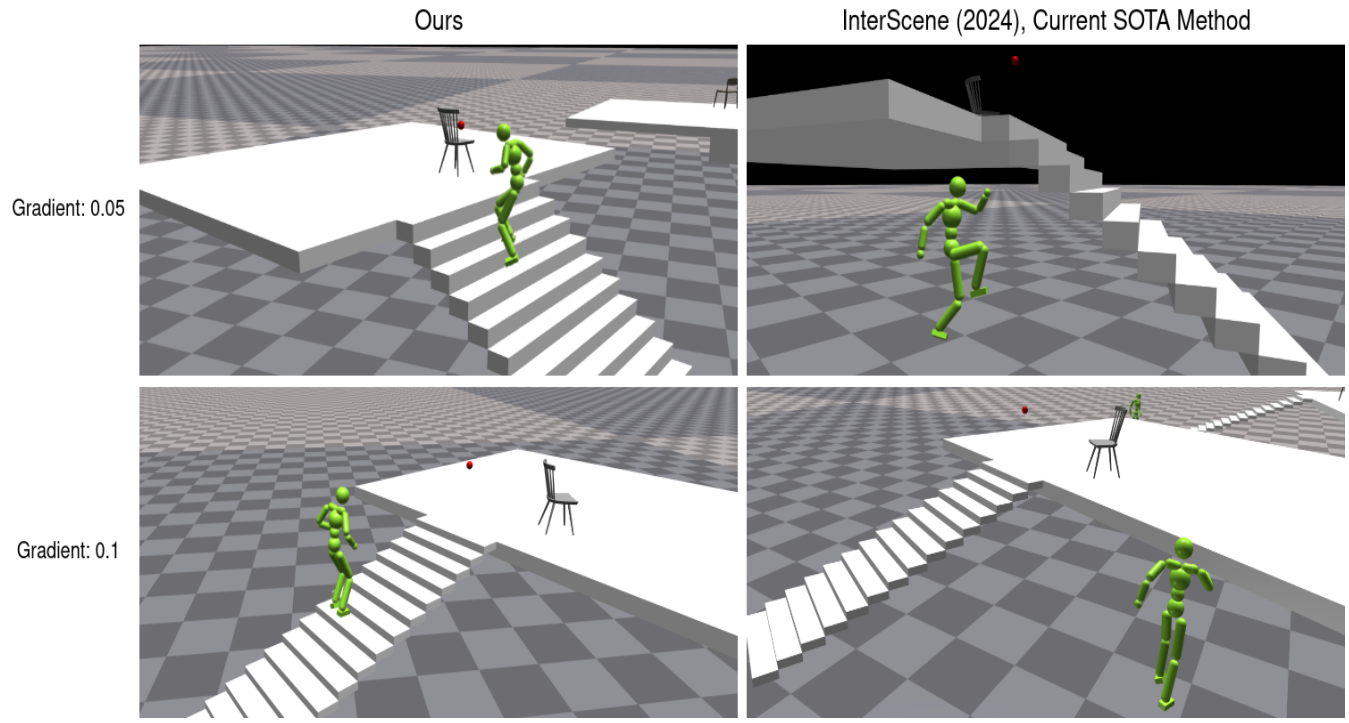

Figure 1: Our InterLevel policy uses a physics-based data-driven 3D motion synthesis approach to create effective motion for navigating 3D scenes. InterScene (Pan et al. 2023), a SOTA method, often fails on stairs with varying heights, whereas our approach ensures stable ascent.

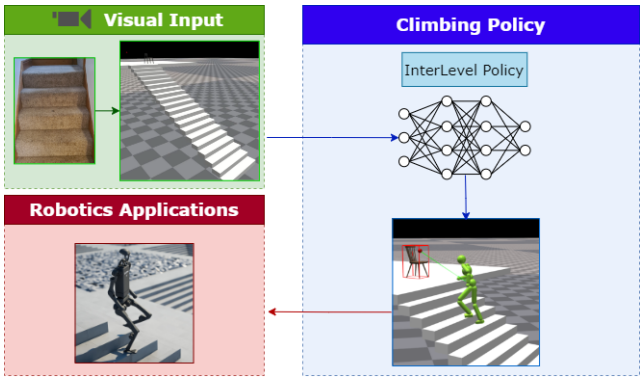

Figure 2: Example framework for real-world applications.

character animation. (Holden et al. 2020) proposed Learned Motion Matching, a technique that utilised multiple neural networks to emulate various parts of Motion Matching, enabling unique player-controlled animations based on a predefined database of animations. Their approach demonstrated the ability to generate responsive character animations in real time, having specific applications within video games. (Zhang et al. 2018) introduced Mode-Adaptive Neural Networks. This quadruped-based motion control architecture leverages unstructured motion capture data and a gating network to dynamically update and blend different animations within a dataset. This approach successfully facilitated smooth transitions between different locomotion models. (Starke et al. 2020) presented a Local Motion Phase framework for bipedal and quadruped characters using optimisation techniques to enable responsive character control from unstructured motion capture data. Their method automatically extracted local motion phases from the data and leveraged neural networks to produce responsive, natural-looking animations.

While these data-driven approaches create realistic-looking motions, they are susceptible to clipping and floating joints and are limited to the behaviours in their dataset. Therefore, generating motions for novel environments, such as staircases, is challenging for purely data-driven approaches. (Qing et al. 2023; Nguyen, Bao, and Nguyen 2022) attempt to solve this issue by blending the motions of two actions based on the current state of the environment. However, they are still restricted to their dataset. In contrast, physically driven approaches have a greater ability to generalise to novel environments that are not present within the training data.

Physics-Based Methods

Physics-based models synthesise motions by training the given network within a physically driven environment, solving for actions that accomplish the given task or behaviour. This allows the model to adapt to new environments and objectives, and transferring this trained model to a real-world application is possible.

Early works, like (Hodgins et al. 1995; Yin, Loken, and van de Panne 2007), proved that performing basic movement and sports tasks with physically driven characters was possible. Over time, these approaches have continued to improve, with recent advancements in deep reinforcement learning showing promising results in producing robust policies for complex tasks. (Peng et al. 2018) proposed DeepMimic, a deep reinforcement learning framework, utilising reference motion capture to create a reward system that can train a variety of control policies on different tasks in a physically realistic environment. Their approach created robust and dynamic motions but required considerable motion capture data for training. (Peng et al. 2022) introduced Adversarial Skill Embeddings (ASE), a project that is a critical foundation for our research. ASE is a training framework that learns various reusable skills that work with completing navigation and obstacle-based tasks. By learning a diverse set of skills in a latent space, ASE allows for generating adaptive and versatile character behaviours.

Physics-based approaches have shown promising results for character-environment interactions, such as walking and quadruped robots on stepping stones (Nguyen, Bao, and Nguyen 2022). However, many of these typically rely on reward function designs locked to a 2D plane, limiting their ability to handle complex 3D environments. Our work expands on this concept by introducing a novel reward function and learning scheme specifically designed for 3D navigation, enabling characters to traverse multi-level environments and areas of varying heights.

Character-Scene Interactions

The ability to synthesise realistic interactions between avatars and the environment is crucial to producing realistic animations and applying them to real-world objects. Recently, numerous works have focused on developing systems for full-body character control and scene interaction.

(Hassan et al. 2023) proposed a method for synthesising physical character-scene interactions using scene-conditioning and a visual discriminator. Their approach is conditioned on the characters' movements and the objects in the environment, allowing for the generation of realistic interactions. By leveraging randomised placements, sizes, and properties of objects, their method can generalise to a broader range of scenarios. (Rempe et al. 2023) introduced Trace and Pace, a diffusion model that uses a trajectory-driven, physics-based humanoid to emulate pedestrian interactions in various environments. In their approach, each agent navigates the environment while avoiding collisions with other agents, resulting in realistic crowd behaviour. (Pan et al. 2023) presented InterScene, a system for synthesising character motions using a finite state machine and various pre-trained policies to explore and interact with a provided 3D scene. However, InterScene struggles to produce policies that are not localised to a 2D plane in the scene, due to the given reward function.

Previous works therefore struggle in handling multi-level environments, with no methods that look specifically to handle various slopes or staircases.

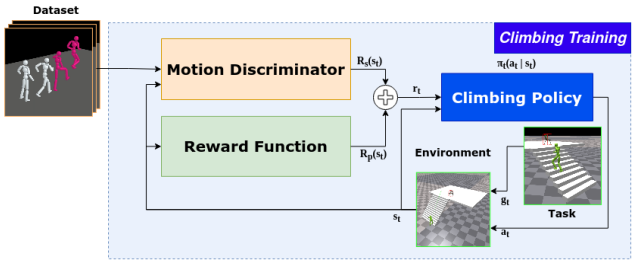

Figure 3: InterLevel Training Framework.Our system consists of a Motion Discriminator that distinguishes between reference climbing motions and generated motions, providing a style-based reward signal $R_s(s_t)$ for the climbing policy. The policy $\pi_t(a_t|s_t)$ controls a simulated character in a physics-based environment to climb the generated staircase. The Reward Function encodes the key features of the task, such as location and velocity targets $R_p(s_t)$, and overall promotes human-like behaviour through the total reward r_t. The character and environment state s_t are passed as observations to the policy at each time step t, which outputs actions a_t to control the character.

We formulate our training environment with various staircase configurations at different slopes and orientations relative to the character's starting state. This allows for greater flexibility and generalisation compared to methods that rely on a fixed environment or predefined object layouts. By training on a diverse set of staircases, our policy learns to adapt to different inclines and orientations, enabling more robust multi-level navigation.

Methodology

Overview

Our InterLevel system consists of a reinforcement learning framework for training physically simulated characters to climb objects, specifically stairs, realistically. We formulate the problem as a Markov Decision Process (MDP), where the character acts as the agent and our 3D environment s_t as the state. At each time step t, the character observes the task state g_t, current state s_t, and its joint positions, velocities, etc. and passes it to a policy $\pi_t(a_t|s_t)$ that outputs a given action a_t. The goal is to climb the stairs in a realistic way that resembles the motion capture data provided.

Observations

At each step, the environment and character state s_t consists of the following 118-dimensional (118D) information:

- Root Height (1D)
- Root Rotation (6D)
- Root Linear and Angular Velocity (6D)
- Local Joint Rotations (72D)
- Local Joint Velocities (28D)
- Distance in XYZ to Target (3D)
- Distance Between M_B and M_T (1D)

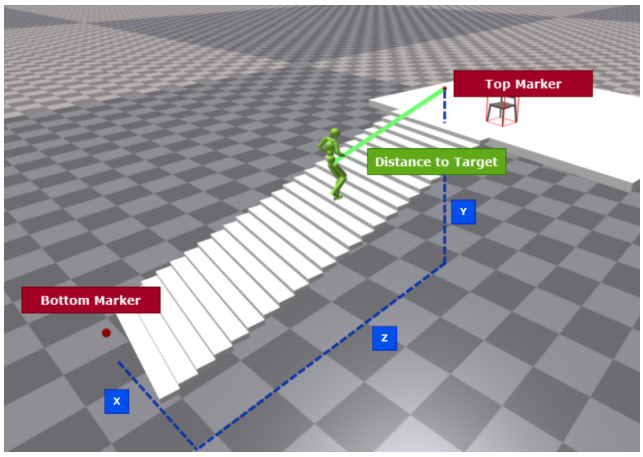

Figure 4: Labelled InterLevel Environment.The distance to the target is visualised using a solid green line between the hips of the humanoid and the current target marker. M_B and M_T are displayed by a red sphere in the scene.

- Angle Difference Between Root and M_T, Relative to the Line Connecting M_B and M_T (1D)

Where M_B and M_T, represent the position of the bottom and top markers respectively. Defined as $M_B = (M_B^x, M_B^y, M_B^z)$ and $M_T = (M_T^x, M_T^y, M_T^z)$.

Rotations are represented using a continuous 6D rotation representation to ensure cohesive data representation (Zhou et al. 2020). The XML for defining the skeleton of the agent is the same as described in (Peng et al. 2021, 2022; Hassan et al. 2023; Pan et al. 2023), having 12 movable internal joints with 28 degrees of freedom.

Actions

The agent's action space is represented by a 28D vector corresponding to the target orientations of each joint. The vector is generated by the trained policy and is subsequently passed to the Issac Gym environment to drive the movements of the humanoid character in the scene.

Reward Function

Our approach's reward function consists of three main components: Location Reward (R_{location}), Facing Reward (R_{facing}), and Velocity Reward (R_{velocity}). Figure 5 visually represents the reward policy. The overall reward function produces a weighted combination of these terms to assess performance R_p, as

$$R_p = w_{\text{loc}} \cdot R_{\text{location}} + w_{\text{face}} \cdot R_{\text{facing}} + w_{\text{vel}} \cdot R_{\text{velocity}}, \quad (1)$$

where w_{loc}, w_{face}, and w_{vel} are weighted values for each reward. The function encourages the agent to follow the desired trajectory as they progress up the incline, providing a more significant reward as the agent successfully navigates the slope. Specifically, the Location Reward guides the character to M_T. The Facing Reward encourages the character to orient towards M_T, while the Velocity Reward encourages the policy to maintain a predefined velocity. We explain each reward in detail below.

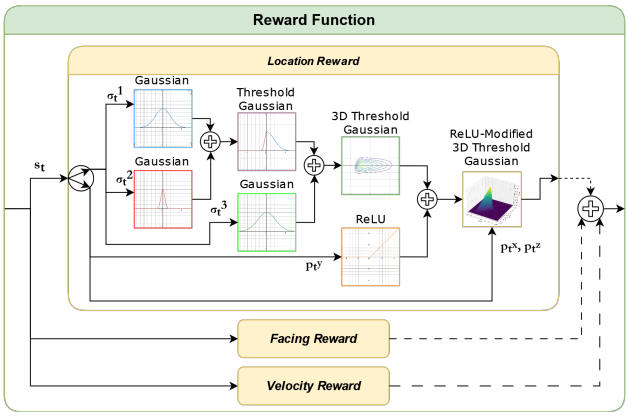

Figure 5: 3D Training Reward Policy. The total reward r_t comprises three main components: Location Reward, Facing Reward, and Velocity Reward. The Location Reward is calculated using a combination of Gaussian functions (Defined by σ_t^1, σ_t^2, and σ_t^3). The ReLU-Modified 3D Threshold Gaussian term creates a reward gradient guiding the character to M_T.

Location Reward The location reward is calculated using a combination of Gaussian and ReLU functions. The root position at time t, p_t, of the avatar is passed into the function, where the x (p_t^x), y (p_t^y), and z (p_t^z) positions are extracted. p_t^y is passed as a parameter for defining the weight and σ for each Gaussian distribution. Two initial distributions are specified with σ_t^1 and σ_t^2, representing the direction of the incline from the highest point, and the opposite direction past the peak, where a threshold across 0 is then used to define a Threshold Gaussian function that ensures that if the policy overshoots, it is not punished. A scaling factor is also applied to $G_x(p_t; \sigma_t^2)$ to ensure the peaks of the distributions are equal, ensuring the highest reward lies at the target position, otherwise, the smaller σ value past the target position will create a higher reward at the incorrect position.

$$G_x(p_t; \sigma_t) = \frac{1}{\sigma_t \sqrt{2\pi}} e^{-\frac{(p_t - \mu)^2}{2\sigma_t^2}} \quad (2)$$

$$TG_x = \begin{cases} G_x(p_t^x; \sigma_t^1) & \text{if } distance \geq 0 \\ w_{\text{scale}} \cdot G_x(p_t^x; \sigma_t^2) & \text{otherwise} \end{cases} \quad (3)$$

This is then combined with a third Gaussian function, weighted by σ_t^3, across the 3rd dimension to create a 3D reward function that will ensure that the agent is rewarded based on being more central on the incline, creating a 3D Threshold Gaussian.

Finally, the root p_t^y position is used to calculate where the agent is, relative to between M_B and M_T, which is passed to a ReLU function to be then used to scale the 3D Threshold Gaussian to create the final ReLU-Modified 3D Threshold Gaussian.

$$R_{\text{location}} = TG_x \cdot G_z(p_t^z; \sigma_t^3) \cdot ReLU((p_t^y - m_b^y)/(m_t^y - m_b^y)) \quad (4)$$

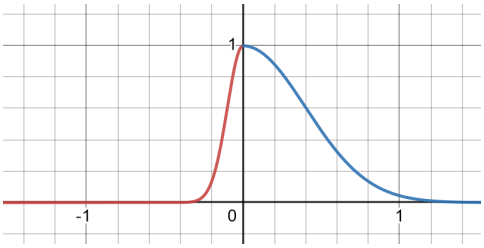

Figure 6: Graphical Example of Threshold Gaussian. Described in Equation 3. The reward is plotted on the y-axis, and the distance to the target is plotted along the x-axis.

Facing Reward The facing reward is formulated as a weighted reward based on the agent's current orientation relative to the direction of the desired trajectory, calculated as the dot product between the target direction and the agent's current direction, where 1 is perfectly aligned, 0 is perpendicular, and -1 is the opposite direction. In this case, it encourages the character to face in the direction of M_T to ensure that it will continue to move towards the target. It is then passed to a ReLU function to prevent a negative penalty from being applied.

$$R_{\text{facing}} = max(0, E_{facing}) = ReLU(E_{facing}), \quad (5)$$

$$E_{facing} = F_{tar} \cdot F_{cur}, \quad (6)$$

where F_{tar} and F_{cur} represent the target and current direction of the character.

Velocity Reward For the velocity reward, the previous p_t and p_{t-1} are used to calculate the agent's current velocity; a reward is then given proportional to how close the calculated velocity is to a predefined velocity value, defined by the user. This allows the policy to be trained to act quicker or slower. The Velocity Reward is written as

$$R_{\text{velocity}} = e^{-2 \cdot {E_{vel}}^2}, \quad (7)$$

$$E_{vel} = ReLU(S_{tar} - S_{cur}), \quad (8)$$

where S_{tar} and S_{cur} denote target speed and current speed, respectively.

Throughout the development process, we conducted extensive testing with these reward functions, enabling and disabling individual components to evaluate their impact on the agent's performance. This iterative approach allowed us to fine-tune the contribution of each reward and identify their influence on the training outcomes.

Motion Discriminator

The style reward R_s, described in (Peng et al. 2021), is designed to increase the reward when the trained policy can create motions that appear to be similar to the given dataset, which is calculated using the discriminator, as seen in Figure 3. This can be described as a given motion $M = (s_i, a_i)$, where s_i are example states from the dataset, and a_i are actions from the policy, where the objective is for a_i to imitate the orientations seen in s_i to maximise a given goal function g and attempt to deceive the discriminator $D(s, a)$ into believing that the action a is sampled from the dataset.

Training

Dataset

For this system, we used clips from the SFU Motion Capture Database (Ying et al. 2018), specifically *0017_Running OnBench001* and *0017_RunningOnBench002*. Each clip was segmented into climbing and descending, focusing on the upward and downward movements. Among these segments, we utilized ascending motion clips to train the proposed model, however the descending clips are still available for potential training of a descending traversal model.

Simulation Setup

NVIDIA's Isaac Gym (Makoviychuk et al. 2021) is used for simulation, a powerful GPU-accelerated simulator designed for training policies for a large variety of robotics tasks.

To ensure that the policy remains generalised and robust, even in unseen conditions, we employ a level of randomness in the initial state of each episode, for the starting position of the character and the rotation of the staircase, exposing the agent to various trajectories and scenarios. As well as this, depth, width, and height are varied within the dataset.

The initial position of the character is a random value within a predefined range from the center of the environment. The staircase's rotation is sampled from a uniform distribution of values from 0 to 360 degrees around the vertical axis. This is then translated to ensure the staircase faces the center of the environment.

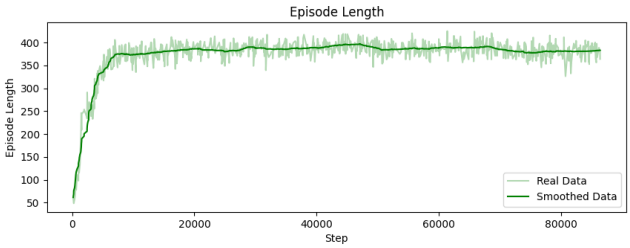

Figure 7: Episode length throughout training. The duration of each episode generally increases as the agent learns to climb the stairs longer before terminating. The smoothed curve (green) shows the upward trend against the raw data (light green).

Implementation

For training, we used the PPO algorithm(Schulman et al. 2017) with a constant learning rate of 2e-5, the same algorithm that showed promising results in (Pan et al. 2023; Peng et al. 2021). The batch size was set to 256, and the minibatch size was 64. The training process was configured to run for a maximum of 1,000,000 epochs, with intermediate results saved every 500 epochs and the best model saved after 50 epochs, the model was never trained to the full 1,000,000 epochs. The model utilized a multi-layer perceptron (MLP) with units [1024, 512] and ReLU activation. We employed gradient clipping with a norm of 1.0 and set the entropy coefficient to 0.0. The reward shaping scale value was set to 1, and the discount factor (gamma) was 0.99. The training

also included normalization of input and value, and the use of mixed precision was disabled.

Early Termination

We can confidently assert that if the agent falls to the ground or onto the staircase, then it is unlikely that we want to encourage this behaviour. As such, inspired by (Pan et al. 2023), we introduce an early termination criterion during the training process. (Pan et al. 2023) uses a pre-defined height, with the condition that if the hips of the character fall below that height, the episode is reset, instead we propose a specified distance from the hips in the vertical axis, where if the head or feet enter this boundary, it is considered horizontal, and the episode is terminated early. Due to this, we were able to effectively detect failures during training above ground level.

The episode will continue until one of the reset criteria is met; either the agent successfully ascends the staircase, the maximum number of time steps is reached, or an early termination condition is met. If the agent is unsuccessful in ascending the stairs, either by time or a fall, they receive a low reward and are reset.

Figure 7 illustrates the length of each episode across each step. This is used to monitor how long the agent can navigate the given environment before the episode terminates, either by completion of the task or by an early termination method being met. Initially, the policy appears to struggle with basic movement and maintaining balance, leading to lower episode length; however, after learning to balance and climb the stairs, the episode length can gradually increase.

Experiments

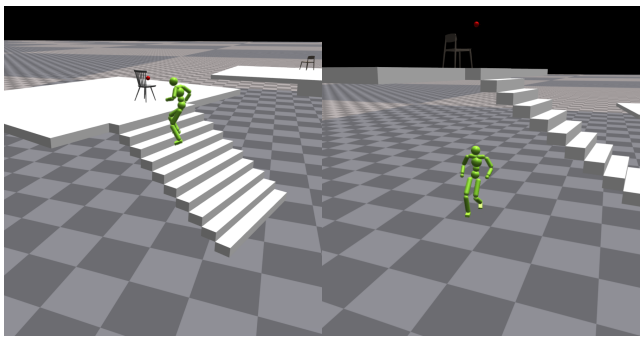

Figure 8: Qualitative comparison of InterLevel (Ours) (Left) against our ablated model, InterLevel-2D (Right).

Ablation Study

To demonstrate the benefits of incorporating a reward function that considers three dimensions, we conduct an ablation study regarding $R_{location}$. In the ablated version, the model is trained with a reward function where the ReLU scaling component of the $R_{location}$ is locked to 1. That is, Eq. (4) is changed into:

$$R_{\text{location}} = TG_x \cdot G_z(p_t^z; \sigma_t^3) \cdot 1 \tag{9}$$

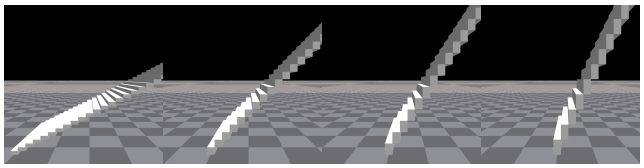

Figure 9: Example Staircase configurations used for training and evaluation. The staircases have step heights of 0.05 (Left), 0.1 (Middle Left), 0.15 (Middle Right), and 0.2 (Right) units, respectively, to allow the policy to generalise to varying inclines.

We call the ablated model without the scaling component InterLevel-2D. InterLevel-2D struggles to produce motions that begin to navigate the staircase, converging to instead navigate below the target. In contrast, the fully implemented policy can successfully climb the staircase and complete the task (Figure 8).

This demonstrates that our novel G_y component is an essential element in creating effective policies for navigating in 3D environments.

Quantitative Comparison

To assess the effectiveness of our trained policy, we conducted experiments on unseen staircases of various heights. We tested the model over 4096 trails, each with a randomised staircase from the dataset with heights ranging from 0.5m to 0.2m, recording the average progress up the stairs and the percentage of episodes terminated due to one of the early termination conditions being met, and the average time taken to complete the staircase in seconds.

Table 1 shows the results found, demonstrating the effectiveness of the InterLevel training algorithm against current state-of-the-art method, InterScene (Pan et al. 2023). Our approach can achieve consistently higher performance when navigating these staircases, achieving a significant increase in progress over InterScene and our ablated model, InterLevel-2D. As well as this, our method is the only method that was able to produce motions that completed the staircase.

These quantitative metrics (Table 1, 2, 3) show the effectiveness of this reinforcement learning approach and the proposed novel reward function. In Table 1, the progress value of our method is 16.7% higher than InterScene, this shows our model's ability to apply these movements to the environ-

Method	Progress (%)	ET (%)	Time (s)
InterScene	30.1%	**6.8%**	-
InterLevel-2D	34.8%	19.4%	-
InterLevel	**46.8%**	24.9%	**8.49**

Table 1: Quantitative results of our InterLevel against InterScene (Pan et al. 2023) and InterLevel-2D on staircases of varying heights over 4096 trials, example staircases in Figure 9. ET denotes early termination. – denotes no successful cases of the agent climbing all the stairs to the end.

Method	Progress (%)	ET (%)	Time (s)
InterScene	42.5%	**10.0%**	-
InterLevel-2D	**59.7%**	23.2%	-
InterLevel	48.8%	17.2%	**12.8**

Table 2: Quantitative results of our InterLevel against InterScene (Pan et al. 2023) and InterLevel-2D on staircases of 0.05m over 4096 trials. − denotes no successful cases of the agent climbing all the stairs to the end.

Method	Progress (%)	ET (%)	Time (s)
InterScene	30.4%	**4.6%**	-
InterLevel-2D	33.9%	7.6%	-
InterLevel	**55.2**%	16.4%	**12.3**

Table 3: Quantitative results of our InterLevel against InterScene (Pan et al. 2023) and InterLevel-2D on staircases of 0.1m over 4096 trials. − denotes no successful cases of the agent climbing all the stairs to the end.

ment effectively with a large variety of staircases. The Early Termination (ET) value of our method is 24.9%, this means that our model is attempting to traverse more of the staircase than InterScene, as the low ET value of the InterScene is due to avoidance of the obstacle, as opposed to traversal and interaction.

In Table 2, we present an experiment that only contains cases of 0.05 gradient staircases, different from all staircases seen in Table 1. We can obtain a consistently higher average with both InterLevel-2D and Interlevel in this case. In Table 3, we present cases of 0.1 gradient staircases. In this case, compared to Table 1, there was a larger gap between InterScene and Ours. This is due to the fact that with the higher gradient, InterScene is unable to reach a higher progress percentage without genuine traversal of the staircase. Furthermore, in these high-gradient cases, our final model obtains a much higher progress percentage than InterLevel-2D, which demonstrates the significance of the 3D information. By training with our novel reward function on a diverse set of staircases at varying angles, InterLevel learns a more generalised policy for being able to traverse these unseen staircases, simulating realistic behaviour.

When considering completed episodes, we analyse the time taken for completion. Showing that the only completed episodes are from the InterLevel policy. Over all staircases, InterLevel can achieve completion in 8.49 seconds, showing that it can not only complete the staircase effectively, but within a reasonable time frame. Subsequently, with 0.05 gradient staircases, it became 12.8 seconds, and 12.3 seconds with 0.1 gradient staircases.

Qualitative Comparison

Our proposed InterLevel generates high-quality motion in a wide range of scenarios. The agent can consistently take continuous steps up the staircase while maintaining balance and adapting to the varied heights of the staircases. As visualized in Table 4, our method significantly outperforms InterScene (Pan et al. 2023). InterScene (Pan et al. 2023) can capture the motion data appropriately, however, it struggles to apply this to the given environment, resulting in the policy circling underneath the target location. In Table 4 (a), the agent in the result of InterScene frequently loops underneath the target floor and falls on the staircase, without being able to recover. In Table 4 (b), as the agent approaches the underside of the floor, it is not blocked, due to the lower height, and remains at the position for the duration of the episode. In Table 4 (c), even at a gradient of 0.05, InterScene fails to traverse the staircase, falling over the low floor, and is unable to recover. In comparison, InterLevel can be seen in Table 4 (a, b, c) successfully navigating the same staircases, reaching the target marker at the correct height.

Limitation

While the results are promising, some artefacts are still present in the motion. One such artefact is that the avatar keeps its left foot at a 20-degree angle from the floor throughout all movements, caused by a misplaced marker during the dataset's recording. As a result, the policy learned to mimic this error, creating the effect we see from the avatar.

Further on, a significant issue occurs as the agent progresses up the staircase. The policy has learned to lock out its legs and lean forward as it approaches 80-90% up the incline, likely in an attempt to maximise the reward function without the danger of continuing the climb and potentially failing. These artefacts can be seen in Figure 10.

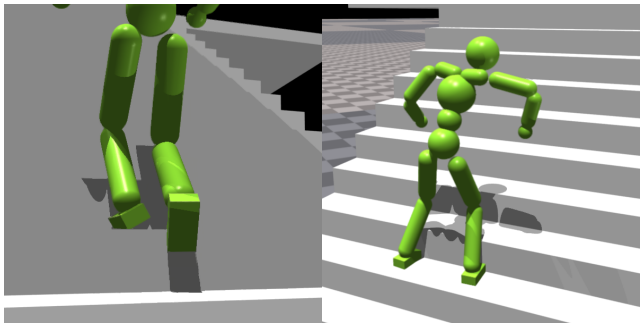

Figure 10: Artefacts observed in the generated climbing motions. The character's left foot remains at a 20-degree angle throughout the movement (left), and the character tends to lock out the legs and lean forward when approaching M_T (right).

Additionally, further hyper-parameter tuning can be used to potentially address these artefacts and improve the overall performance of the model. The current hyper-parameters, such as the learning rate, regularization coefficients, and reward function weighting, may not fully capture the nuances of the desired motions or penalize undesirable behaviors effectively. Adjusting these parameters could help the policy better generalize across a range of movements and environments, reducing the tendency to overfit to specific errors, such as the locked legs and foot misalignment.

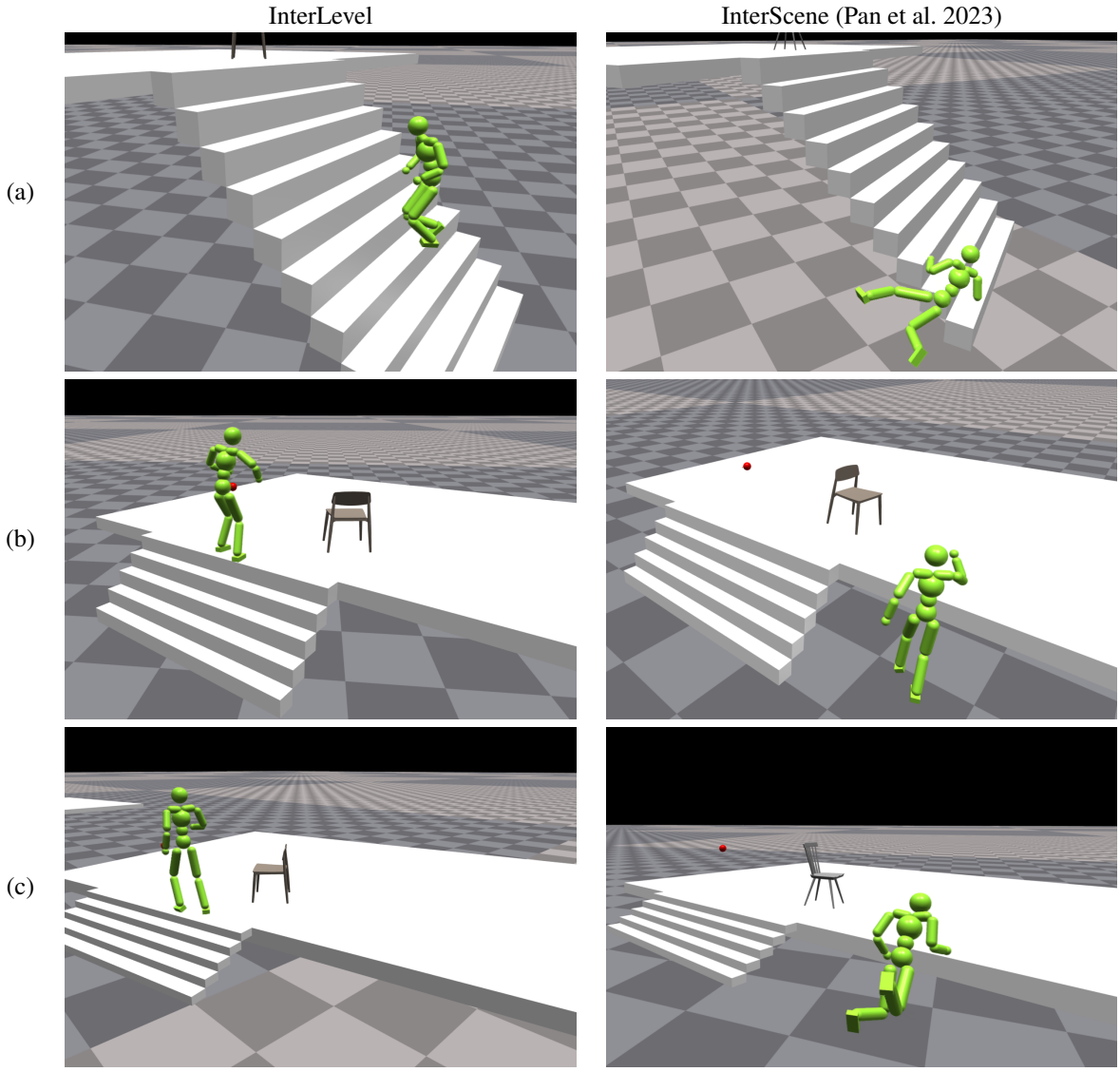

Table 4: Qualitative comparison between InterLevel (Ours) (Left) and InterScene (Pan et al. 2023) (Right), on staircases of varied gradients, both are trained on the same motion data.

Conclusion

InterLevel presents a novel reinforcement learning system for synthesising stair-climbing motions for physically simulated characters. Our core contributions include a novel reward function that encourages human-like movements while successful navigation towards a target. Effective generalisation is achieved through training with various staircases of different heights and orientations, leading to high performance, even in unseen environments. Quantitative and qualitative analysis demonstrates the approach's effectiveness while being able to achieve stable and realistic motions in new environments.

Future work should focus on extending the system for more complex environments, such as varied step heights and downward slopes. It should also investigate real-world applications of the policy. Additionally, alternative tracking measures should be explored, such as distance sensors from the front of the feet, which could prove more applicable to real-world implementations. Addressing these issues could enhance the overall adaptability of the system, making significant contributions to character-scene interactions.

Acknowledgments

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2024-RS-2024-00437102) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation), and also supported by IITP, MSIT (No. RS-2023-00222106).

References

Hassan, M.; Ceylan, D.; Villegas, R.; Saito, J.; Yang, J.; Zhou, Y.; and Black, M. 2021. Stochastic Scene-Aware Motion Prediction. In *Proceedings of the International Conference on Computer Vision 2021 (ICCV)*, 11354–11364. Piscataway, NJ: Institute of Electrical and Electronics Engineers (IEEE).

Hassan, M.; Guo, Y.; Wang, T.; Black, M.; Fidler, S.; and Peng, X. B. 2023. Synthesizing Physical Character-Scene Interactions. In *SIGGRAPH '23: ACM SIGGRAPH 2023 Conference Proceedings*, 63. New York, NY, USA: Association for Computing Machinery (ACM). ISBN 9798400701597.

Hodgins, J. K.; Wooten, W. L.; Brogan, D. C.; and O'Brien, J. F. 1995. Animating human athletics. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, 71–78. New York, NY, USA: Association for Computing Machinery (ACM). ISBN 0897917014.

Holden, D.; Kanoun, O.; Perepichka, M.; and Popa, T. 2020. Learned motion matching. *ACM Transactions on Graphics*, 39(4).

Lee, J.; and Joo, H. 2023. Locomotion-Action-Manipulation: Synthesizing Human-Scene Interactions in Complex 3D Environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 9663–9674. Paris, France: Institute of Electrical and Electronics Engineers (IEEE). Available: The Computer Vision Foundation open access.

Makoviychuk, V.; Wawrzyniak, L.; Guo, Y.; Lu, M.; Storey, K.; Macklin, M.; Hoeller, D.; Rudin, N.; Allshire, A.; Handa, A.; and State, G. 2021. Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning.

Nguyen, C.; Bao, L.; and Nguyen, Q. 2022. Continuous Jumping for Legged Robots on Stepping Stones via Trajectory Optimization and Model Predictive Control. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, 93–99.

Pan, L.; Wang, J.; Huang, B.; Zhang, J.; Wang, H.; Tang, X.; and Wang, Y. 2023. Synthesizing Physically Plausible Human Motions in 3D Scenes. arXiv:2308.09036.

Peng, X. B.; Abbeel, P.; Levine, S.; and van de Panne, M. 2018. DeepMimic: Example-guided Deep Reinforcement Learning of Physics-based Character Skills. *ACM Transactions on Graphics*, 37(4): 143:1–143:14.

Peng, X. B.; Guo, Y.; Halper, L.; Levine, S.; and Fidler, S. 2022. ASE: Large-Scale Reusable Adversarial Skill Embeddings for Physically Simulated Characters. *ACM Transactions on Graphics*, 41(4): 1–17.

Peng, X. B.; Ma, Z.; Abbeel, P.; Levine, S.; and Kanazawa, A. 2021. AMP: adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics*, 40(4): 1–20.

Qing, Z.; Cai, Z.; Yang, Z.; and Yang, L. 2023. Story-to-Motion: Synthesizing Infinite and Controllable Character Animation from Long Text.

Rempe, D.; Luo, Z.; Peng, X. B.; Yuan, Y.; Kitani, K.; Kreis, K.; Fidler, S.; and Litany, O. 2023. Trace and Pace: Controllable Pedestrian Animation via Guided Trajectory Diffusion.

Rudin, N.; Hoeller, D.; Reist, P.; and Hutter, M. 2022. Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347.

Starke, S.; Zhao, Y.; Komura, T.; and Zaman, K. A. 2020. Local motion phases for learning multi-contact character movements. *ACM Transactions on Graphics*, 39: 54:1 – 54:13.

Tessler, C.; Kasten, Y.; Guo, Y.; Mannor, S.; Chechik, G.; and Peng, X. B. 2023. CALM: Conditional Adversarial Latent Models for Directable Virtual Characters. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings*, 37. Los Angeles, CA, USA: Association for Computing Machinery (ACM).

Yin, K.; Loken, K.; and van de Panne, M. 2007. SIMBICON: simple biped locomotion control. *ACM Transactions on Graphics*, 26(3).

Ying, G. J.; Yin, K.; Kumar, K. D.; Geng, H.; and Mahadevan, C. 2018. SFU Motion Capture Database. Available at: http://mocap.cs.sfu.ca. The database was created with funding from NUS AcRF R-252-000-429-133 and SFU President's Research Start-up Grant.

Zhang, H.; Starke, S.; Komura, T.; and Saito, J. 2018. Mode-adaptive neural networks for quadruped motion control. *ACM Transactions on Graphics*, 37(4).

Zhou, Y.; Barnes, C.; Lu, J.; Yang, J.; and Li, H. 2020. On the Continuity of Rotation Representations in Neural Networks.