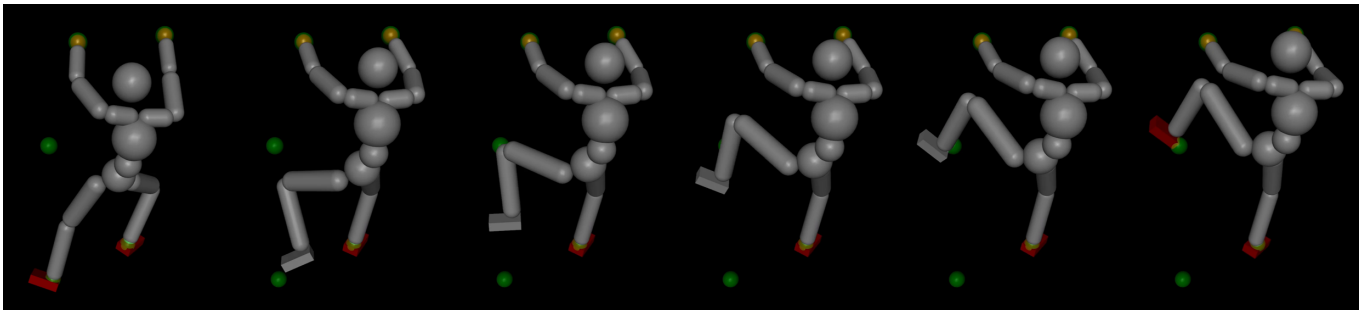
AI-Driven Rock Climbing

MATTHEW LEUNG, Simon Fraser University, Canada
YICHENG LIN, Simon Fraser University, Canada
RYAN ZRYMIAK, Simon Fraser University, Canada



We present a diffusion-based approach for generating realistic climbing motions from human demonstrations. Our system synthesizes full-body motion sequences conditioned on starting holds and target holds, producing coordinated, multi-limb behaviors that generalize to unseen wall and hold configurations. Generated motions are created auto-regressively in segments and concatenated to form complete sequences, and can be extended by redefining starting and target holds to produce new climbing paths. These high-quality motion sequences have applications for athlete training, offering data-driven insights for route planning and skill development, and provide valuable reference trajectories for robotics research, supporting the study of control strategies and motion planning in complex vertical environments.

1 Introduction

The challenge of a bouldering problem is universal, yet the strategies used to solve it are deeply personal. This project is motivated by the potential to leverage AI to decode and reproduce these strategies for two distinct audiences.

For athletes and coaches, this work offers the possibility of transforming training methodologies. An AI "climbing coach" could analyze a climber's strengths and weaknesses to generate customized training problems, propose efficient or non-intuitive movement sequences "beta" that a human might overlook, and provide a data-driven framework for route grading. Such a system moves beyond subjective advice, offering a quantitative and analytical tool for skill development.

At the same time, the generation of realistic climbing motions has important implications for robotics research. Bouldering is a demanding activity that requires full-body coordination, dynamic balance, and detailed contact planning. Producing high-quality climbing motions provides valuable reference trajectories for future robotic climbing systems, enabling the study of control strategies, contact scheduling, and motion planning in complex vertical environments. These datasets can help bridge the gap between motion generation and physically grounded robotic behaviors.

Authors' Contact Information: Matthew Leung, mcleung@sfu.ca, Simon Fraser University, Burnaby, British Columbia, Canada; Yicheng Lin, yla912@sfu.ca, Simon Fraser University, Burnaby, British Columbia, Canada; Ryan Zrymiak, rza80@sfu.ca, Simon Fraser University, Burnaby, British Columbia, Canada.

Modern AI motion systems remain limited in their ability to synthesize complex, full-body behaviors. Most existing methods focus on simple, repetitive actions such as walking, running, or jumping. These approaches are insufficient for tasks like climbing, which require multi-limb coordination, continuous interaction with the environment, and precise management of contact forces.

Climbing is inherently a vertical, full-body activity. Each movement involves coordinated use of the hands, feet, arms, and core to maintain balance and progress upward. A motion-generation system must therefore understand not only *how* to move, but also *how* to interact with surfaces: choosing appropriate holds, distributing body weight, and transitioning smoothly between contact points.

Conventional AI systems cannot yet reason at this level of complexity. If asked to "climb a wall," current models lack the ability to plan which hold to grasp, how to shift balance dynamically, or how to maintain stability across multiple contact points. This highlights the gap between existing motion-generation methods and the type of intelligent, adaptive behavior exhibited by human climbers.

Our goal is to develop an AI system capable of generating natural and realistic climbing motions by learning from human climbing demonstrations. Rather than replaying predefined animations, the model learns to synthesize new sequences that capture coordinated, multi-limb climbing behaviors and generalize to unseen wall or hold configurations. These generated motions offer significant potential not only for animation and human–computer interaction, but also as a foundation for future research in agile and adaptive climbing robots.

2 Related Works

Our project builds on a broader research trajectory that has progressed from traditional optimization-based pipelines to modern deep learning approaches for synthesizing climbing motions.

Optimization and Graph-Based Planning. Early methods relied heavily on combinatorial search and trajectory optimization. For instance, [NaderiKourosh et al. 2017] decomposed climbing into a high-level graph-based planner for hold sequencing and a low-level sampling-based optimizer for generating kinematically feasible motions. Although effective for structured problems, these

approaches lack the generalization ability and end-to-end learning benefits offered by modern data-driven models.

Deep Imitation and Reinforcement Learning. Reinforcement learning has also been explored for climbing-motion synthesis, such as in [Naderi et al. 2019]. A major advancement in this direction is the Adversarial Motion Priors (AMP) framework [Peng et al. 2021], which uses generative adversarial imitation learning to develop agile, naturalistic motor skills from motion-capture data. AMP introduces a discriminator-based style reward that encourages human-like movement. Building on this foundation, [Kang et al. 2024] adapted AMP specifically to rock climbing, demonstrating its ability to reproduce complex full-body behaviors in highly constrained environments.

Diffusion Models for Motion Synthesis. Recent work increasingly leverages diffusion models for high-quality motion generation. MDM [Tevet et al. 2023] introduced a diffusion-based framework for general human motion synthesis, and subsequent methods [Li et al. 2024; Yi et al. 2024] extended this idea to produce scene-aware motions. However, the absence of physics-based simulation in these approaches often leads to artifacts such as floating, ground penetration, and self-collisions. PARC [Xu et al. 2025] addresses these limitations by integrating a physics-based simulation module, enabling the generation of agile, parkour-style kinematic sequences conditioned on target terrain. Together, these methods highlight the growing potential of diffusion models to synthesize diverse, coherent, and contextually grounded motions in complex environments.

3 Method Overview

In this work, we introduce a diffusion-based approach for generating climbing motions. An overview of the method is shown in Figure 1. Our framework employs a motion generator that produces kinematic motions conditioned on the climbing holds. Given a target hold, the generator first synthesizes the initial N frames of the climber reaching towards it. These generated frames are then fed back into the model to produce the next N frames, and this process is repeated until the climber reaches the target hold. The resulting segments are concatenated to form the complete motion sequence, starting from the initial holds and progressing toward the target. After generating a sequence, we can redefine the starting holds using the final frame of the produced motion and specify a new target hold, allowing the system to generate an additional sequence that climbs toward a new configuration of holds.

4 Background

In this section, we review the core machine learning concepts underlying our framework, focusing on diffusion models, a class of generative models that have recently demonstrated high effectiveness for motion synthesis [Tevet et al. 2023]. Diffusion models generate samples from a target data distribution by learning to reverse a predefined forward noising process. This process starts with a clean data sample $x_0 \sim D$ and incrementally adds Gaussian noise over K steps, eventually transforming x_0 into pure noise $x_K \sim \mathcal{N}(0, I)$. The model is then trained to reconstruct the original sample from its noisy versions. While the original DDPM formulation predicts the noise added at each step [Ho and Salimans 2022], many recent

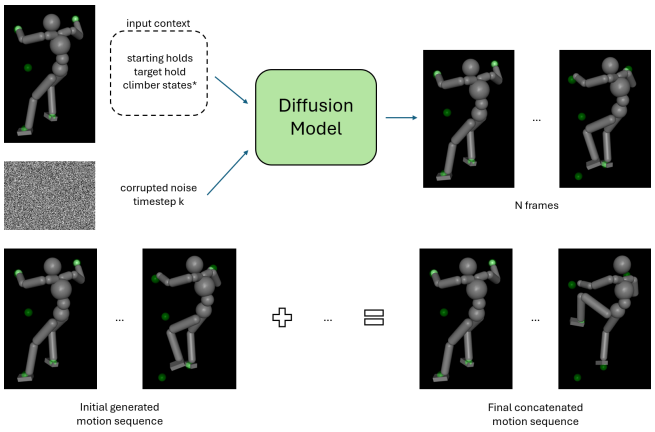


Fig. 1. Overview of our method. The diffusion model is conditioned on the starting holds, target hold, and optionally the climber states to generate N frames of a motion sequence. The motion is then generated auto-regressively, and the resulting segments are concatenated to produce the final motion sequence. See Section 5 for details.

motion-diffusion methods instead train a denoising model G to directly predict the clean motion x_0 from a noisy input [Cohan et al. 2024; Tevet et al. 2023; Xu et al. 2025]. A detailed comparison of these parameterizations is provided in [Karunratanakul et al. 2023]. The denoising model G is optimized using a reconstruction loss:

$$\mathcal{L}_{\text{rec}}(G) = \mathbb{E}_{x_0, C\sim D}\ \mathbb{E}_{k\sim p(k)}\ \mathbb{E}_{x_k\sim q(x_k|x_0)}\left[\|x_0 - G(x_k, k, C)\|^2\right] \quad (1)$$

where $p(k)$ is typically a uniform distribution over $[1, K]$, and C denotes optional conditioning context (e.g., text descriptions or control signals). Once trained, new motions are synthesized by using the denoising model G.

5 Motion Generator

Our motion generator is implemented as a diffusion model that synthesizes climbing motion sequences, following the approach of [Xu et al. 2025]. It is conditioned on the initial holds and a specified target hold. An overview of the architecture is shown in Figure 2. Given contextual information C, the generator produces a motion sequence $x = \{x^1, x^2, \ldots, x^N\}$ describing the character's movement along the terrain toward the target hold. Each frame x^i is represented by the following features:

- $p \in \mathbb{R}^{J\times 3}$, joint positions
- $q \in \mathbb{R}^{J\times 3}$, joint rotations
- $g \in [0, 1]^4$, grip indicators

where J denotes the number of joints in the character's body. The first joint corresponds to the root, representing the global position and orientation of the character. The rotational quantities are parameterized using exponential maps, and the grip indicators are for the left and right hands and feet.

The conditioning context C for our diffusion model comprises: (1) the starting holds of the left and right hands and feet $h_{\text{start}} \in \mathbb{R}^{4\times 3}$, along with the target hold $h_{\text{target}} \in \mathbb{R}^3$, all expressed in the character's local coordinate frame, and (2) the first two frames of the motion sequence. This frame condition is optional, allowing

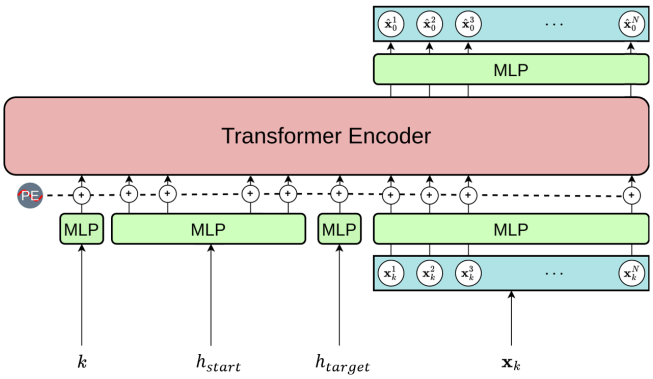


Fig. 2. The transformer encoder based architecture of our motion generator. The inputs are first embedded into tokens using individual MLPs. These tokens are then processed by the transformer encoder, and the tokens corresponding to motion frames are decoded by an MLP to produce the denoised motion sequence.

the model to operate in two modes: generating a new motion from no prior frames or extending a motion autoregressively. Using two frames, rather than one, gives the model information about the velocity.

Our diffusion model adopts a transformer encoder architecture, shown in Figure 2, follows the design of MDM [Tevet et al. 2023]. The generator G takes as input the diffusion timestep k, the noisy motion frames x_k, and the conditioning context $C = \{h_{\text{start}}, h_{\text{target}}, x^1, x^2\}$, and predicts the corresponding clean motion sequence $\hat{x}_0$:

$$G(k, x_k, C) = \hat{x}_0 = \{\hat{x}_0^1, \hat{x}_0^2, \ldots, \hat{x}_0^N\}.$$

The conditioning inputs are first converted into token sequences through dedicated embedding networks. The diffusion timestep k, the starting holds h_{start}, and the target hold h_{target} are each processed by separate MLP-based embeddings that map them into transformer tokens. Each frame of the noisy motion sequence x_k is likewise encoded into a token using a frame-wise MLP.

If the first two clean frames (x_0^1, x_0^2) are supplied as additional context, they replace the corresponding noisy frames (x_k^1, x_k^2) and are embedded using the same MLP. Positional encodings are then added to all tokens. The resulting sequence of tokens is fed through the transformer encoder, and the tokens associated with motion frames are finally decoded by an MLP to produce the denoised prediction $\hat{x}_0$.

5.1 Training

Following the approach of [Xu et al. 2025], our motion generator is trained using a standard diffusion-model objective, with additional geometric loss terms that promote spatial coherence and physical plausibility. Training samples a noisy motion x_k for $k \sim [1, K]$, predicts a denoised estimate $\hat{x}_0$, and computes the corresponding diffusion loss. The overall training objective is

$$\mathcal{L}(G) = \mathcal{L}_{\text{rec}}(G) + \mathcal{L}_{\text{velocity}}(G) + \mathcal{L}_{\text{joint}}(G) + \mathcal{L}_{\text{hold}}(G) \tag{2}$$

where $\mathcal{L}_{\text{rec}}$ is the reconstruction loss, $\mathcal{L}_{\text{velocity}}$ enforces consistency of velocities, $\mathcal{L}_{\text{joint}}$ enforces joint-space consistency, and $\mathcal{L}_{\text{hold}}$ penalizes deviations from the assigned holds.

Reconstruction Loss. The reconstruction loss follows Equation 1, with one modification: because direct subtraction is not a valid metric for rotations, we decompose x_0 into positional p_0, rotational q_0, and gripping-label g_0 components. Let $\hat{p}_0$, $\hat{q}_0$, and $\hat{g}_0$ denote the corresponding components extracted from the prediction $\hat{x}_0 = G(k, x_k, C)$. The modified reconstruction loss is

$$\mathcal{L}_{\text{rec}}(G) = \mathbb{E}_{x_0, C\sim\mathcal{D}}\, \mathbb{E}_{k\sim p(k)}\, \mathbb{E}_{x_k\sim q(x_k|x_0)} \Big[\|p_0 - \hat{p}_0\|^2 + \|q_0 \ominus \hat{q}_0\|^2 + \|g_0 - \hat{g}_0\|^2 \Big] \tag{3}$$

where $\ominus$ denotes the geodesic distance on the rotation manifold.

Velocity Loss. The velocity loss enforces consistency of velocities between the original and predicted motion. Positional velocities $\dot{p}_0$ are computed via finite differences, while angular velocities $\dot{q}_0$ are computed by taking quaternion differences and converting them to their exponential-map representation. The loss is

$$\mathcal{L}_{\text{velocity}}(G) = \mathbb{E}_{x_0, C\sim\mathcal{D}}\, \mathbb{E}_{k\sim p(k)}\, \mathbb{E}_{x_k\sim q(x_k|x_0)} \Big[\|\dot{p}_0 - \hat{\dot{p}}_0\|^2 + \|\dot{q}_0 - \hat{\dot{q}}_0\|^2 \Big] \tag{4}$$

Joint-Consistency Loss. To ensure that the predicted joint positions are kinematically consistent with the predicted rotations, we penalize discrepancies between the predicted joint positions $\hat{p}_0$ and the positions obtained from forward kinematics applied with $\hat{q}_0$:

$$\mathcal{L}_{\text{joint}}(G) = \mathbb{E}_{x_0, C\sim\mathcal{D}}\, \mathbb{E}_{k\sim p(k)}\, \mathbb{E}_{x_k\sim q(x_k|x_0)} \Big[\|\hat{p}_0 - \text{FK}(\hat{q}_0)\|^2 \Big] \tag{5}$$

Hold-Deviation Loss. The hold-deviation loss penalizes predicted end-effector positions that deviate from the designated holds during gripping ($g_0 = 1$). For each gripping frame, the error is measured relative to the nearer of the start hold h_{start} or the target hold h_{target}:

$$\mathcal{L}_{\text{hold}}(G) = \mathbb{E}_{x_0, C\sim\mathcal{D}}\, \mathbb{E}_{k\sim p(k)}\, \mathbb{E}_{x_k\sim q(x_k|x_0)} \Big[g_0 \cdot \min\Big(\|\hat{p}_0 - h_{\text{start}}\|^2, \|\hat{p}_0 - h_{\text{target}}\|^2 \Big) \Big] \tag{6}$$

Hold-Augmented Training. To improve robustness to variations in hold placement, we randomly offset the target hold with probability 0.1 during training. For these augmented samples, we apply only the joint-consistency and hold-deviation losses:

$$\mathcal{L}(G) = \mathcal{L}_{\text{joint}}(G) + \mathcal{L}_{\text{hold}}(G) \tag{7}$$

5.2 Motion generation

Given the initial and target holds, our trained model can generate long motion sequences by conditioning autoregressively on its own previously generated frames, as illustrated in Figure 1.

To generate new motions with the diffusion model, we aim to recover the clean sample x_0 from the predicted sample $\hat{x}_0$ using a modified DDIM update rule [Song et al. 2022; Xu et al. 2025]. Given a DDIM stride d and initial noise $x_K \sim \mathcal{N}(0, I)$, the intermediate

samples are computed iteratively as

$$x_{k-d} = \left(\sqrt{\bar{\alpha}_{k-d}} - \frac{\sqrt{\bar{\alpha}_k}\sqrt{1-\bar{\alpha}_{k-d}}}{\sqrt{1-\bar{\alpha}_k}} \right) \hat{x}_0 + \frac{\sqrt{1-\bar{\alpha}_{k-d}}}{\sqrt{1-\bar{\alpha}_k}} x_k \qquad (8)$$

where $\bar{\alpha}_k$ denotes the cumulative product of the diffusion schedule at timestep k. This update is applied repeatedly until the final clean sample x_0 is obtained.

6 Data Acquisition

For our data, we used climbing animations from the CIMI4D dataset [Yan et al. 2023] and the AscendMotion dataset [Yan et al. 2025] as reference trajectories. We then applied the Rokoko add-on in Blender to retarget the full motion sequences to our humanoid model, ensuring kinematic compatibility. After retargeting, we extracted short clips in which exactly one hand or foot moved from one hold to another, and exported these clips as training samples.

Because the extracted motion clips contained only joint positions and rotations, additional processing was required to recover hold and grip information. We defined the initial positions of each hand and foot as the starting holds, and the final position of the moving hand or foot as the target hold. We also collected grip-state information for each limb to identify which hand or foot was moving and over which frames. This was performed by playing the climbing animation visually and manually marking the frames where the corresponding limbs are moving.

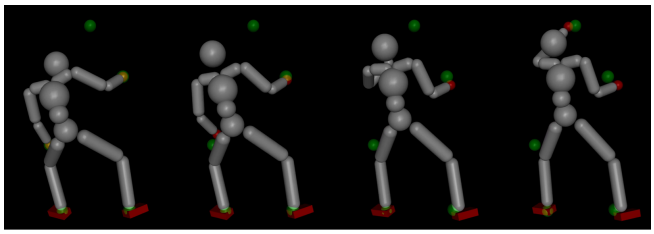
7 Experiments and Evaluations

The generator is trained on 20 motion clips, each approximately 2 seconds long, yielding roughly 40 seconds of total training data. Training was performed on a machine equipped with an RTX 3090 GPU and a Ryzen 9 9990X CPU, requiring approximately 4 hours of wall-clock time. Given the small size of the dataset, the following experiments examine how well the generator can learn and generalize meaningful climbing behaviors under these constrained conditions. All experiments use a diffusion timestep of $K = 1000$, a DDIM stride of $d = 10$, and a generator that produces motion sequences of $N = 15$ frames at 30 FPS. To assess the effectiveness of the proposed method, we conduct both qualitative evaluations of motion quality and quantitative analyses of limb involvement.

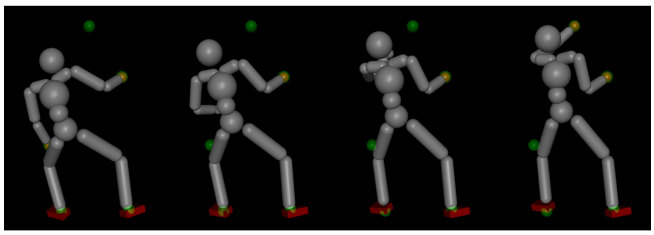
7.1 Qualitative Evaluation

We qualitatively evaluate the motion generator on three tasks: (1) reconstructing motions from the training data, (2) generating motions under small random perturbations of the hold positions, and (3) synthesizing motions from fully randomized hold configurations. While the generator exhibits some promising behaviors, it ultimately struggles to generalize beyond the training distribution.

First, we assess whether the generator has learned a meaningful representation of climbing motions by evaluating its ability to reconstruct trajectories from the training set. Results are shown in Figure 3. When conditioned on motions from the training data, the generator accurately reconstructs the sequences, often closely matching the original inputs. Remarkably, the generator is also able to correct minor drifting artifacts present in the raw dataset. The reconstructed sequences preserve the overall structure and timing of the ground-truth motions, indicating that the generator has effectively overfit the training examples. Although this does not demonstrate generalization, it confirms that the generator successfully maps corrupted motions back onto the motion manifold observed during training, providing a useful baseline for evaluation in more challenging scenarios.



(a)



(b)

Fig. 3. (a) Input motion from the training set. (b) Reconstructed motion produced by the generator. The motion generator faithfully recovers the original trajectory and removes drifting artifacts.

Next, we examine the generator's response to small perturbations applied to the climbing holds. The results are presented in Figure 4. For each evaluation sequence, we apply a slight random offset to the hold positions and query the generator for a motion conditioned on these modified inputs. The generator, however, fails to adapt its output to the perturbed holds and instead produces motions nearly identical to those generated using the unperturbed holds. This suggests that the generator relies heavily on memorized training-set patterns and has not learned a functional relationship between hold configurations and the resulting climbing motion. In other words, the generator "remembers" motions rather than reasoning about the spatial arrangement of the holds. This failure mode indicates insufficient motion diversity in the dataset as well as potential architectural limitations in how conditioning information is incorporated.

Finally, we evaluate the generator on fully novel hold configurations not seen during training. The results are shown in Figure 5. In this setting, the generator produces highly unstable and incoherent motions. Frequently, the generator snaps to a motion resembling an unrelated training example or generates movements with no plausible climbing intent. These behaviors further demonstrate that the generator has significantly overfit the small training set and has not learned a generalizable mapping from hold positions to motion trajectories.

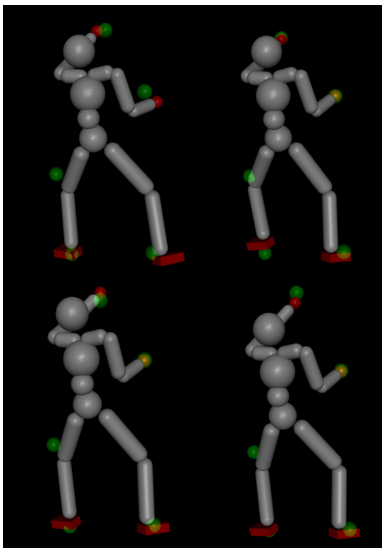


Fig. 4. Motion generation results when the climbing holds are subject to small random perturbations. Despite the modified hold positions, the generator produces motions nearly identical to those generated with the original holds, showing little adaptation.

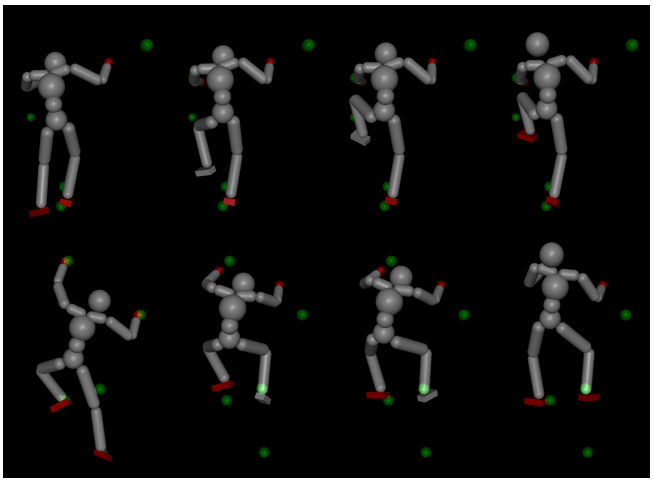


Fig. 5. Motion generation results for fully novel hold configurations not seen during training. The generator produces unstable and incoherent motions, often snapping to trajectories resembling unrelated training examples or exhibiting no plausible climbing intent. These failures highlight the generator's overfitting to the limited dataset and its inability to generalize to new hold arrangements.

7.2 Quantitative Evaluation

To complement the qualitative results, we evaluate the generator using several quantitative metrics designed to measure (1) how accurately the model reaches the target holds and (2) how consistently the limbs remain close to their designated holds throughout the full trajectory. These metrics allow us to assess whether the generated climbing motions are spatially coherent and consistent with the conditioning inputs.

We report results across four evaluation settings: (1) motions from the training set, (2) reconstructions of training-set motions, (3) motions generated from small perturbations of the original hold locations, and (4) motions generated from entirely novel hold configurations.

Final-Frame Target-Hold Distance. For each generated motion, we compute the Euclidean distance between each limb's end effector and its designated target hold at the *final frame*. This metric evaluates whether the synthesized motion succeeds at ultimately positioning the climber's hands and feet at the intended holds. Lower values indicate more successful target reaching.

Table 1 summarizes the results. As expected, the generator achieves low error on the training sequences, their reconstructions, and the slightly perturbed configurations, confirming strong memorization of the training trajectories. However, the error increases substantially for the novel hold configurations, consistent with the qualitative observation that the generator fails to adapt to unseen spatial layouts.

Table 1. Distance to target holds at the final frame (in cm). Lower values indicate better target-reaching accuracy.

Motion Type	**Mean**	**Max**
Training data	5.964	7.684
Reconstructed training data	3.227	4.979
Randomly perturbed holds	6.691	7.522
Completely random start holds	63.28	127.3

Per-Frame Limb-to-Hold Distance. To evaluate motion quality across the *entire* trajectory, we compute, for each sequence, the mean and maximum distance between each limb and its assigned hold across all frames. This metric measures how consistently the generator maintains contact or close proximity to the relevant holds—a key requirement for producing stable and physically plausible climbing motions.

Table 2 reports these values. On the training sequences, their reconstructions, and the slightly perturbed configurations, the generator achieves low mean and maximum distances, indicating reasonable reproduction of limb-placement patterns from the training manifold. In contrast, distances increase notably for the novel hold configurations. The larger maximum distances further suggest instability, with limbs drifting or snapping to implausible locations when the generator is pushed outside its memorized regime.

These quantitative results support the qualitative findings: the generator overfits the training data, reconstructing seen motions without the drifting artifacts present in some of the original sequences. Small perturbations to the hold positions yield similarly low errors, indicating that the model remains tightly confined to the training manifold. In contrast, novel hold configurations lead to large errors and unstable limb behavior. Overall, the metrics suggest that the generator memorizes the training sequences rather

Table 2. Mean and maximum distance between limbs and their designated holds across all frames (in cm). Lower values indicate closer and more consistent limb engagement.

Motion Type	Mean Distance	Max Distance
Training data	3.048	22.04
Reconstructed training data	2.633	8.547
Randomly perturbed holds	2.633	10.50
Completely random start holds	8.425	81.08

than learning a generalizable mapping from hold configurations to motion trajectories.

8 Limitations and Future Works

One limitation we faced stems from the quantity and quality of the training data. Although the datasets we used contain numerous climbing motions, many sequences exhibit a stationary body root, preventing meaningful extraction of motion direction. Additionally, we observed noise and artifacts in several sequences, such as hands and feet drifting away from holds or sudden, unrealistic drops between frames. Consequently, only a fraction of the data was suitable for training our motion generator, which limited its ability to generalize to unseen scenarios.

Future work can address these issues through improved data pre-processing and post-processing. Noise and artifacts could be mitigated using techniques such as linear least squares or minor optimizations to correct drifting limbs. Classical least-squares–based filters, such as the Savitzky–Golay method [Savitzky and Golay 1964], provide effective local polynomial fitting that can smooth noisy joint trajectories while preserving important motion structure. More advanced approaches, including penalized B-spline smoothing [Eilers 2003], offer greater control over continuity and are particularly well suited for cleaning motion-capture data. Recent studies also demonstrate that B-spline–based filtering can reliably remove high-frequency jitter, reconstruct missing data, and suppress artifacts introduced during capture [Memar Ardestani and Yan 2022]. Generated motions could also undergo a post-optimization step to enforce constraints [Xu et al. 2025], for example, ensuring limbs remain close to their intended holds during grips and removing unrealistic movements.

To increase the amount of high-quality data, one could leverage pose estimation methods [Luo et al. 2022; Xie et al. 2021] on climbing videos to extract motions directly. Given the wider availability of climbing videos, this approach has significant potential to expand the dataset.

Works such as [Xu et al. 2025; Yuan et al. 2023] demonstrate that a motion tracker operating within a physics-based simulation can be used to filter generated motions, ensuring physical plausibility. The validated motions can then be fed back into the training pipeline, effectively augmenting the dataset and improving the robustness and fidelity of the generative model.

Another limitation of our current method is that it accepts only a fixed number of holds as input, and motions are generated under the assumption that only one limb moves at a time. Future work could explore generating motions with arbitrary hold configurations and more dynamic behaviors, such as simultaneous multi-limb movements or jumping between holds, to capture a broader range of realistic climbing strategies.

9 Afterthoughts

The process of retargeting our climbing motions took us longer than expected. In Blender, we created a standard armature that we could use for all our retargeting and designed to match the humanoid model for the motion generation. However, we made repeated changes to this armature in order for it to match the constraints of the humanoid model. We eventually created a matching armature which would allow us to correctly retarget the climbing motions from our dataset in Blender, but had we been able to figure this out early on, we would have allowed ourselves more time to improve our results.

Our original plan was to build upon the PARC framework [Xu et al. 2025]. A substantial portion of our early work was spent exploring climbing simulation in IsaacSim. Although we eventually succeeded in implementing gripping behaviors, we encountered persistent difficulties in training a stable motion tracker, even on basic walking sequences. This led us to shift our focus entirely to the motion generation component. Had we prioritized the generator from the outset, we would have had significantly more time to iterate and improve the quality of our outcomes.

References

Setareh Cohan, Guy Tevet, Daniele Reda, Xue Bin Peng, and Michiel van de Panne. 2024. Flexible Motion In-betweening with Diffusion Models. arXiv:2405.11126 [cs.CV] https://arxiv.org/abs/2405.11126

Paul H. C. Eilers. 2003. A Perfect Smoother. *Analytical Chemistry* 75, 14 (July 2003), 3631–3636. doi:10.1021/ac034173t

Jonathan Ho and Tim Salimans. 2022. Classifier-Free Diffusion Guidance. arXiv:2207.12598 [cs.LG] https://arxiv.org/abs/2207.12598

Kyungwon Kang, Taehong Gu, and Taesoo Kwon. 2024. *Learning Climbing Controllers for Physics-Based Characters.* The Eurographics Association. https://doi.org/10.2312/sca.20241165

Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. 2023. Guided Motion Diffusion for Controllable Human Motion Synthesis. arXiv:2305.12577 [cs.CV] https://arxiv.org/abs/2305.12577

Jiaman Li, Alexander Clegg, Roozbeh Mottaghi, Jiajun Wu, Xavier Puig, and C. Karen Liu. 2024. Controllable Human-Object Interaction Synthesis. arXiv:2312.03913 [cs.CV] https://arxiv.org/abs/2312.03913

Zhengyi Luo, Ryo Hachiuma, Ye Yuan, and Kris Kitani. 2022. Dynamics-Regulated Kinematic Policy for Egocentric Pose Estimation. arXiv:2106.05969 [cs.CV] https://arxiv.org/abs/2106.05969

Mehdi Memar Ardestani and Hong Yan. 2022. Noise Reduction in Human Motion-Captured Signals for Computer Animation based on B-Spline Filtering. *Sensors* 22, 12 (2022), 4629. doi:10.3390/s22124629

Kourosh Naderi, Amin Babadi, Shaghayegh Roohi, and Perttu Hämäläinen. 2019. A Reinforcement Learning Approach To Synthesizing Climbing Movements. In *2019 IEEE Conference on Games (CoG).* 1–7. doi:10.1109/CIG.2019.8848127

NaderiKourosh, RajamäkiJoose, and HämäläinenPerttu. 2017. Discovering and synthesizing humanoid climbing movements. *ACM Transactions on Graphics (TOG)* (July 2017). doi:10.1145/3072959.3073707 Publisher: ACMPUB27New York, NY, USA.

Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. 2021. AMP: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics* 40, 4 (July 2021). doi:10.1145/3450626.3459670 arXiv:2104.02180 Publisher: Association for Computing Machinery.

Abraham Savitzky and M. J. E. Golay. 1964. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry* 36, 8 (1964), 1627–1639. doi:10.1021/ac60214a047

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2022. Denoising Diffusion Implicit Models. arXiv:2010.02502 [cs.LG] https://arxiv.org/abs/2010.02502

Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. 2023. Human Motion Diffusion Model. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=SJ1kSyO2jwu

Kevin Xie, Tingwu Wang, Umar Iqbal, Yunrong Guo, Sanja Fidler, and Florian Shkurti. 2021. Physics-Based Human Motion Estimation and Synthesis From Videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 11532–11541.

Michael Xu, Yi Shi, KangKang Yin, and Xue Bin Peng. 2025. PARC: Physics-based Augmentation with Reinforcement Learning for Character Controllers. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*. 1–11. doi:10.1145/3721238.3730616 arXiv:2505.04002 [cs].

Ming Yan, Xincheng Lin, Yuhua Luo, Shuqi Fan, Yudi Dai, Qixin Zhong, Lincai Zhong, Yuexin Ma, Lan Xu, Chenglu Wen, et al. 2025. ClimbingCap: Multi-Modal Dataset and Method for Rock Climbing in World Coordinate. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 12312–12323.

Ming Yan, Xin Wang, Yudi Dai, Siqi Shen, Chenglu Wen, Lan Xu, Yuexin Ma, and Cheng Wang. 2023. CIMI4D: A Large Multimodal Climbing Motion Dataset under Human-scene Interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 12977–12988.

Hongwei Yi, Justus Thies, Michael J. Black, Xue Bin Peng, and Davis Rempe. 2024. Generating Human Interaction Motions in Scenes with Text Control. arXiv:2404.10685 [cs.CV] https://arxiv.org/abs/2404.10685

Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. 2023. PhysDiff: Physics-Guided Human Motion Diffusion Model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 16010–16021.