Implicit Bézier Motion Model for Precise Spatial and Temporal Control

Luca Vögeli*
DisneyResearch|Studios
Zürich, Switzerland
voegeli.luca@gmail.com

Dhruv Agrawal*
ETH Zürich
Zürich, Switzerland
DisneyResearch|Studios
Zürich, Switzerland
dhruv.agrawal@inf.ethz.ch

Martin Guay
DisneyResearch|Studios
Zürich, Switzerland
martin.guay@disneyresearch.com

Dominik Borer
DisneyResearch|Studios
Zürich, Switzerland
dominik.borer@disneyresearch.com

Robert W. Sumner
DisneyResearch|Studios
Zürich, Switzerland
ETH Zürich
Zürich, Switzerland
sumner@disneyresearch.com

Jakob Buhmann
DisneyResearch|Studios
Zürich, Switzerland
jakob.buhmann@disneyresearch.com

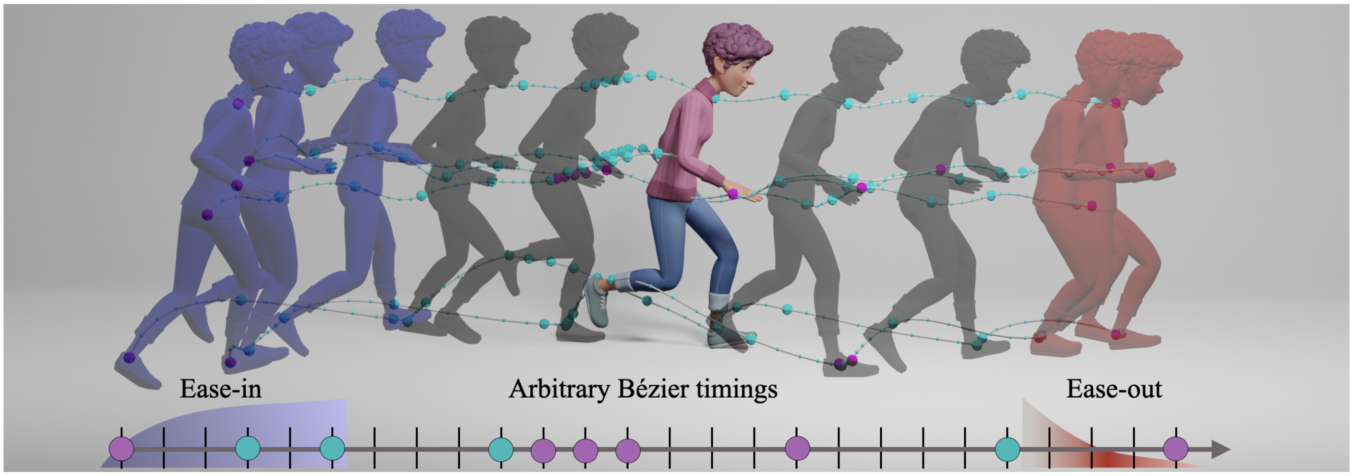


Figure 1: Example of a generated motion sample with our Implicit Bézier Motion Model. The model predicts Bézier control points (cyan) at arbitrary user defined frames and additionally allows for precise spatial control (purple). The model offers a novel input condition to shape the temporal profile of easing-in/out of the motion.

ABSTRACT

Creating high-quality character animation remains an intricate and cumbersome process that requires skill, training, and craftsmanship to master. Recently, diffusion models have unlocked the ability to generate diverse movements from high-level condition signals such as text. For artist-friendly control, motion diffusion leveraging Bézier curves have been shown to allow precise joint-level conditioning. Yet, these works have been limited to joints at a fixed temporal stride, while animators require more temporal flexibility when keyframing or manipulating tangents to achieve animation principles such as easing in & out. In this work, we introduce a new *Implicit* Bézier Motion Model (IBMM), which during training is exposed to all possible configurations of control points, enabling control at arbitrary timings. This allows both precise and sparse joint-level control, anywhere in time and for any joint. In addition, we introduce a new quantitative measure of ease-in and -out, which leads to a novel condition over the motion generation process to reflect this artistic principle.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; **Animation**; • **Human-centered computing** → *Systems and tools for interaction design.*

*Equal contribution

MIG '25, December 3–5, 2025, Zurich, Switzerland
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2236-3/2025/12
https://doi.org/10.1145/3769047.3769052

KEYWORDS

Motion Diffusion, Character Animation, Bézier Curves, Artistic Control

ACM Reference Format:

Luca Vögeli, Dhruv Agrawal, Martin Guay, Dominik Borer, Robert W. Sumner, and Jakob Buhmann. 2025. Implicit Bézier Motion Model for Precise Spatial and Temporal Control. In *The 18th ACM SIGGRAPH Conference on Motion, Interaction, and Games (MIG '25), December 3–5, 2025, Zurich, Switzerland.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3769047.3769052

1 INTRODUCTION

Animating 3D characters remains a challenging and time-consuming task. Traditional tools such as control rigs and inverse kinematics (IK), along with their modern learned counterparts [Agrawal et al. 2023; Oreshkin et al. 2022], can accelerate the posing of individual frames. However, crafting compelling and coherent motion remains a significant manual effort, requiring hundreds of keyframes for a few seconds of animation.

Recently, generative motion models have shown remarkable progress [Cohan et al. 2024; Guo et al. 2024] in facilitating animation tasks. These models can generate high-quality human movements from text [Meng et al. 2025] or music [Zhang et al. 2025], they can be leveraged to improve motion capture [Li et al. 2025; Mu et al. 2025], or help in traditional animation tasks such as inbetweenning [Cohan et al. 2024; Goel et al. 2025]. However, while they excel at producing plausible motion priors, they often lack the fine-grained and interactive control necessary for professional animation workflows.

Even though motion is a dense sequence of 3D transforms for every single joint, artists generally control a much smaller set of control handles. At a joint level, this is achieved by control rigs that control multiple joints, e.g. via IK. At a temporal level, artists rely on concepts of keyframes which are interpolated by parametric curves to craft a motion. These concepts of sparse representations have recently been introduced within a generative motion setting to improve the quality [Bae et al. 2025; Hwang et al. 2025], and the controllability [Studer et al. 2024] of the generated motion.

In particular, Bézier Motion Model (BMM) [Studer et al. 2024] offers precise joint-level control by working in a spatially and temporally sparse representation of artist-friendly Bézier curves. While offering precise spatial control, BMM is limited to predict a fixed set of control points at uniform time intervals. This prevents artists from fine-grained temporal control, such as moving control points in time or adding control points in regions that would require more detail.

In this paper, we introduce an *Implicit* Bézier Motion Model (IBMM), a novel framework designed specifically to overcome this critical limitation of a predefined temporal stride. IBMM implicitly learns a Bézier fit during training for arbitrary temporal control points, removing the need for a pre-fitting of the data. Therefore, IBMM allows artists to both precisely control any part of the curve, as well as sparse-level authoring. In addition to the new training methodology, we revised the transformer-based architecture to further increase the accuracy of constraints. Lastly, we introduce a novel stylistic parameter—inspired by existing artistic concepts—that captures at a high level, the global motion dynamics. Artists can then leverage this control to ease-in and ease-out of the generated motion.

By providing more flexible and precise control, IBMM offers a more powerful and intuitive tool for animators bridging the gap between generative power and artistic control. We evaluate our model for the use case of inbetweening to compare it to the work of [Studer et al. 2024].

2 RELATED WORKS

Synthesis of character motion in traditional motion authoring software such as Maya [Autodesk Inc. 2025] and Blender [Blender Online Community 2025] has been limited to keyframing rig parameters and traditional interpolation. [Ciccone et al. 2019] built an optimization-based keyframing system to accelerate this process. More recently, research has moved to more sophisticated data-driven generative models.

Early data-driven approaches such as [Harvey et al. 2020; Tang et al. 2022] employed auto-regressive methods to predict dense motion given current state and a singular target frame. [Qin et al. 2022] used a two-stage transformer approach in which the first stage predicts a noisy approximation of the motion that is refined by the second stage. More recently, [Akhoundi et al. 2025] show that a simple method with a single transformer predicting in a root-local space can match or outperform more elaborate methods. [Agrawal et al. 2024] use a skeletal transformer to model skeletal and temporal dependencies in motion. They require a single context frame and allow joint-level sparse constraints, which they use to build a neural motion rig enabling iterative motion authoring.

Although these models made remarkable early progress, they struggle in training on larger datasets and generating high-frequency details because of averaging artifacts common in deterministic approaches. To combat this phenomenon, generative models, both diffusion-based [Ho et al. 2020] and VQ-based [Guo et al. 2024; Shi et al. 2025], have been widely adopted for motion generation at scale. Early works such as [Dabral et al. 2023; Tevet et al. 2023; Tseng et al. 2023] trained denoising diffusion models to generate motion from a text or music prompt. [Zhang et al. 2023a] use a hybrid retrieval mechanism that selects unusual motions from a large dataset and integrates them into the motion generation phase using a semantics-modulated transformer and a condition injection scheme. [Liao et al. 2025] allow generating motion for different body shapes by conditioning their model with SMPL [Loper et al. 2015] shape parameters in addition to text. GenMO [Li et al. 2025] is one of the first multi-modal diffusion models that can be conditioned concurrently with text, music, keyframes, and images. [Meng et al. 2025] train a diffusion model inspired by VQ-based models to generate motion from long and specific text sequences.

To better incorporate motion generation into animation software, controllable motion generation with keyframe and trajectory-based control through ControlNet [Xie et al. 2024; Zhang et al. 2023b], noise optimization [Karunratanakul et al. 2024] and inpainting [Cohan et al. 2024] has been extensively explored. However, these works struggle for very sparse control signals such as sparse joint-level constraints. Recent works have exploited the sparsification of the input space to solve this problem. [Bae et al. 2025] reduce temporal resolution by analytically finding the most important

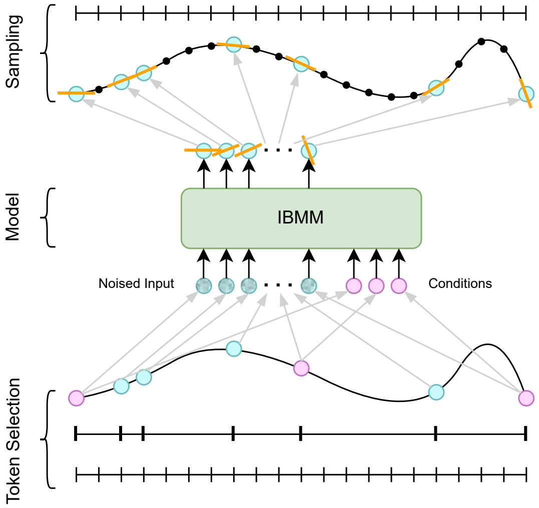


Figure 2: Overview of the temporal token selection and sampling of IBMM. A random set of temporal control points (cyan) can be specified together with spatial constraints (pink). IBMM denoises the input and maps it to Bézier control points (positions and tangents) at the specified temporal indices. Finally, the dense curves are sampled to reconstruct the full motion.

frames within a motion sequence, training a diffusion model to generate these frames, and interpolating the remaining frames before decoding. [Hwang et al. 2025] instead sparsify the spatial dimension using a two-stage approach that predicts only a subset of joints, called *keyjoints*, in the first stage. The second stage then predicts full-body motion from the first stage output.

Our work is most closely related to the Bézier Motion Model (BMM) [Studer et al. 2024], which sparsifies both temporal and spatial dimensions by predicting Bézier control points at fixed uniform intervals for the end-effector joints. A second stage recovers the full-body-motion that can be addressed by multiple inverse kinematic systems [Agrawal et al. 2023; Huang et al. 2017; Oreshkin et al. 2022; Ponton et al. 2025, 2023; Qin et al. 2022]. Although BMM closely follows sparse constraints, the fixed timing of the Bézier control points limits flexibility and disallows generating finer details between the Bézier points. Additionally, fitting Bézier curves during training is extremely slow and requires pre-processing.

In this work, we address both these limitations by allowing an arbitrary number of control points to be placed at arbitrary timings within the motion sequence and by implicitly learning the Bézier interpolation within the neural network.

3 METHODOLOGY

IBMM builds upon BMM [Studer et al. 2024], which factorizes motion into parametric (cubic) Bézier curves for a subset of joints that are then mapped to a full FK skeleton via a learned IK model [Oreshkin et al. 2022]. But instead of a fixed stride between the Bézier control points, our method allows any arbitrary temporal discretization as depicted in Figure 2. Through training, IBMM then effectively learns to generate controllable motion curves by predicting a variable number of arbitrarily timed Bézier control points.

In the following, we discuss our data representation, the model with an improved condition representation, and the novel control mechanism over ease-in/out.

3.1 Data Representation

We define a motion sequence as $[x_i]_1^N$, a series of N poses. Each pose, x_i, encapsulates the global positions $p_i \in \mathbb{R}^3$ and global orientations $r_i \in \mathbb{R}^6$ [Zhou et al. 2019] for the joints J at the i-th frame. For better comparison, we stick to the same subset of joints J as BMM, which includes the hip, head, wrists, and feet joints.

IBMM takes as input a sparse subset of B poses, whose indices $[\tau_i]_1^B$ are sampled from uniformly sized bins in the full sequence.[1] As we evaluate our model on an inbetweening task, we assume that the boundary frames are always available. Hence, τ_1 and τ_B are always set to 1 and N respectively. For more details on the index sampling, please refer to Appendix A.

The input poses $[x^t_{\tau_i}]_1^B$ are ground truth poses noised at these indices, according to the diffusion timestep t. Alongside these poses, the model receives condition tokens $c_{\tau_i,j}$. Each token includes the ground truth position p_{τ_i} and a one-hot encoded identifier ID_j for a specific joint j, with further details provided in Section 3.2.

Unlike BMM, the predicted tangents are not an input to our model, removing the necessity of a slow fit of the Bézier curves for training. Furthermore, in contrast to BMM's approach of shifting the start pose to the origin, we align motion sequences by centering them, orienting them forward, followed by a final normalization. This improved alignment simplifies the task and improves overall generation quality. For further details on pre-processing, see Section 4.1.

The model output $[\hat{y}_{\tau_i}]_1^B$ consists of predicted positions $\hat{p}_{\tau_i}$, orientations $\hat{r}_{\tau_i}$, and tangents $\hat{v}_{\tau_i}$ for each of the B input frames. Subsequently, the complete dense motion sequence $[\hat{x}_i]_1^N$ is reconstructed by interpolating the sparse outputs using cubic Bézier sampling.

The final mapping from the predicted subset skeleton J to the full FK skeleton is performed with a learned IK model [Oreshkin et al. 2022] as in BMM.

3.2 Model Architecture

An overview of the model architecture can be seen in Figure 3. IBMM leverages a transformer encoder as the backbone architecture for the diffusion model. The input sequence consists of one token encoding the noise level, the B Bézier pose tokens, k condition tokens, and two ease-in/out tokens (elaborated further in Section 3.3). The Bézier pose tokens are embedded with a linear layer, while the noise level token goes through an MLP. The temporal information is then added to the feature vector with sinusoidal positional encoding (PE).

[1] The binning helps to space out the control points to prevent sampling in just one region.

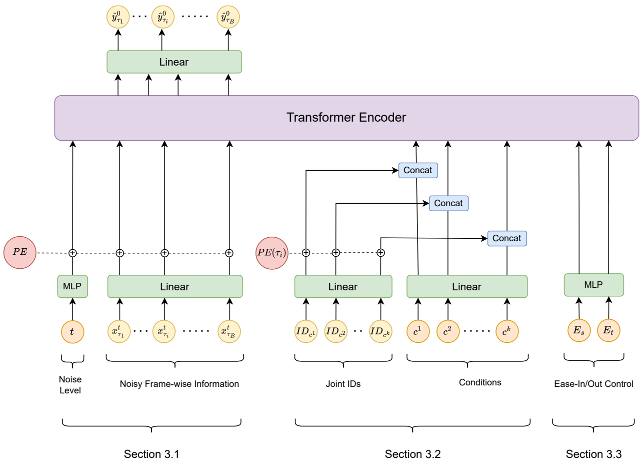
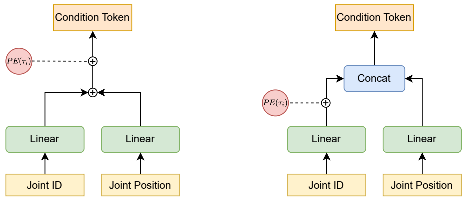


Figure 3: Overview of the IBMM model architecture. The transformer encoder gets noisy frame-wise information as input. In addition, positional condition tokens are provided to make the prediction controllable. Note that the notation c of a condition is an abbreviation for $p_{\tau_i,j}$. Lastly, the condition tokens for controlling the ease-in/out are added.

To create the condition token $c_{\tau_i,j}$, both $p_{\tau_i,j}$, and ID_j are embedded using their own separate linear layers. The PE of τ_i is added only to the embedded ID_j and the two embeddings are concatenated. This ensures that spatial information $p_{\tau_i,j}$ is separated from the structural and relational information of the tokens. This enables the transformer to focus more on the relations between the pose and condition tokens, while still having the full unperturbed information of the 3D positions. The difference between our condition token creation and the one used in BMM can be seen in Figure 4.

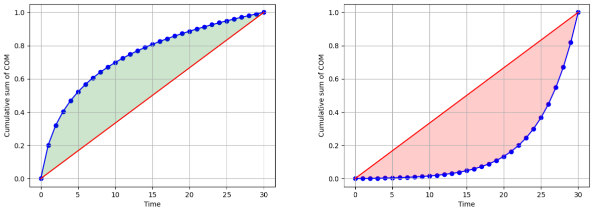


(a) BMM **(b) IBMM**

Figure 4: Comparison of the condition token creation between BMM and IBMM. (a) BMM condition token creation, and (b) our condition token creation. Note that in our condition tokens, the positional encoding is only added to the joint IDs.

3.3 Ease-In/Out Control

When crafting animations, artists also apply principles that go beyond precise spatial control. One such concept is the ease-in or ease-out of a motion. In an eased-in motion, the character movements starts slowly and then gradually accelerates towards the end of the sequence. Ease-out is the inverse, with the character slowing down towards the end of the motion.



(a) ease-out **(b) ease-in**

Figure 5: Illustration of an ease-in/out concept. (a) shows ease-out of a motion with the COM slowing at the end of the sequence. (b) shows ease-in to a motion with a slow initial velocity and then accelerating.

In this section, we introduce a new control handle that allows the animators to control the desired ease-in and ease-out of the generated motions. Compared to text-to-motion models [Meng et al. 2025] which can provide a rough control over timing with "slow" or "fast" labels, our control signal is inherently continuous, allowing for a more fine-grained control.

We start with the observation that the ease-in/out of a movement can be measured by studying the trajectory of the character's center of mass (COM). In Figure 5, we plot the ratio of the current distance traveled by the COM to the total distance over time. For Figure 5a, we see that the ground truth trajectory travels a large percentage of the path at the start and slows down towards the end. This is identical to the behavior of an ease-out animation. Inversely, in Figure 5b, the character starts out slow and accelerates towards the end of the motion–depicting an ease-in motion.

To introduce this concept into our model, we add two new tokens, E_i and E_o, to the input sequence of the transformer. They measure the total distance between the ground truth trajectory and a constant velocity motion for the start $[x_i]_{i<k}$ and the end $[x_i]_{i\geq N-k}$ of the motion. This can be formulated as follows.

$$E_i = \sum_{n=0}^{k-1} \bar{p}_n - \left(\bar{p}_0 + \frac{n}{k}\left(\bar{p}_{k-1} - \bar{p}_0\right)\right) \tag{1}$$

$$E_o = \sum_{n=k}^{N} \bar{p}_n - \left(\bar{p}_k + \frac{n-k}{N-k}\left(\bar{p}_N - \bar{p}_k\right)\right), \tag{2}$$

where $\bar{p}_n$ is the mean of all J joints at frame n. In this work, we use $k = \frac{N}{2}$ for simplicity. By calculating E_i and E_o separately, we can control both ease-in and ease-out for a single animation sequence.

These tokens are then embedded via an MLP and appended to the input sequence beside $[x^t_{\tau_i}]^B_1$ and $c_{\tau_i,j}$. Lastly, we add an attention mask so that the input tokens only attend to one of E_i and E_o, according to their frame index.

3.4 Loss Functions

For training our model, we use the same loss configurations as in BMM. This entails MSE losses for the predicted 3D trajectories $[\hat{p}_i]^N_1$ and orientations $[\hat{r}_i]^N_1$ against ground truth values $[p_i]^N_1$

and $[r_i]_1^N$, respectively. There are further MSE losses between the 3D conditions $[c_{\tau_i,j}]_1^B$ and the corresponding predicted positions $[\hat{p}_{\tau_i,j}]_1^B$, and dense 3D velocities calculated via finite differences. Additionally, we also adopt the same foot sliding and foot contact losses. For further details, we refer to [Studer et al. 2024].

However, as the data is not pre-fit to a specific configuration of Bézier control points, we do not require direct losses on the Bézier positions, $\mathcal{L}_b$, and tangents, $\mathcal{L}_t$, used in the original BMM work. Thus, in our case, the fitting to the Bézier control points is implicitly done via the dense losses on the trajectories during training.

4 EVALUATION

Our evaluation is conducted on the LaFAN1 dataset [Harvey et al. 2020], a widely recognized locomotion dataset. Given this dataset's nature and the focus on artist-controllable motion generation, we measure foot sliding and handle accuracy to measure motion quality and generalization to artist inputs, respectively. For more details on the metrics used, refer to Appendix E.

Building directly upon BMM, we quantitatively and qualitatively compare the benefits of our randomized token selection to their uniformly spaced tokens, and our motion generation performance against theirs. We evaluate the effectiveness of the ease-in/out handle in changing the overall timing of the motion. Finally, we show in the supplementary video instances of how IBMM is used to create and control motion.

4.1 Dataset and Implementation Details

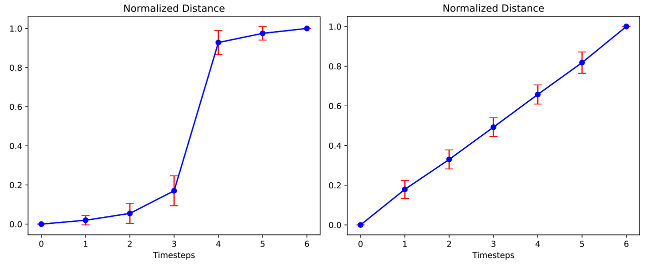
We use the LaFAN1 dataset [Harvey et al. 2020] to evaluate our model and compare it to BMM. It contains motion capture data recorded on five subjects and is approximately 4.6 hours long. The motions include walking, running, climbing obstacles, jumping, dancing, and many more.

Each clip is individually projected to the ground plane by subtracting the minimum height of the clip from all positions. The clips are split into overlapping segments of 30 or 90 frames, depending on the sequence length for which the model is to be trained. Additionally, the data is aligned based on the boundary frames in the following way. First, each segment is rotated around the vertical axis such that the ground projected direction from start to end pose aligns with the positive Y-axis. Second, the motion is now shifted along the Y-axis such that the middle is located at the origin. Finally, we use Z-score normalization to set mean to 0 and variance to 1 over the training data.

The training data includes all clips from subjects one to four, while the validation data consists of all clips from subject five. We trained two model versions for 230 epochs (~64 hours) on an Nvidia RTX3090.

- The first, a **30-frame model**, uniformly samples 4 to 12 Bézier poses and is conditioned on 0 to 6 intermediate joint-wise constraints per segment.
- The second, a **90-frame model**, uniformly samples 10 to 24 Bézier poses and is conditioned on 0 to 13 intermediate joint-wise constraints per segment.

The network hyper-parameters are kept the same as for BMM. The only exception is the latent size, which we increased from 512 to 1024.



(a) BMM **(b) IBMM**

Figure 6: Ratio of the distance between two control points for an additional control point that is temporally shifted between them. The results show the average and standard deviation over 1000 different samples. (a) BMM reflects a step-like function, indicating that time is mapped to discrete values. On the contrary, (b) IBMM has a continuous linear profile allowing for more detailed control over the time axis.

4.2 Quantitative Evaluation

4.2.1 Varying time indices. BMM is trained using Bézier curves where the control points are placed at a fixed stride of six frames. Consequently, during its training phase, the transformer-based architecture is only exposed to temporal information at these discrete six-frame intervals. However, a fundamental characteristic of transformer models is their token-based processing, which allows for variations in the number of inputs, and thereby, the number of control points. Additionally, the temporal information of the control points is represented by positional encoding, which is a continuous function. Thus, one could in principle also vary the time index in a continuous manner to shift the timing of a control point.

To test the model's ability to interpret unseen, continuous time values, we take a sequence of Bézier control points generated by BMM and introduce an additional control point p_{new}. The timing of this new point is moved linearly from the time of the third original control point (p_3) to the fourth (p_4). For each incremental step, corresponding to a single frame's duration, we calculate the normalized distance $d(p_{new})$ on its way from p_3 to p_4. We define this distance as the ratio of the distance between p_3 and p_{new} to the total distance from p_3 through p_{new} to p_4:

$$d(p_{new}) = \frac{\|p_3 - p_{new}\|}{\|p_3 - p_{new}\| + \|p_4 - p_{new}\|} \tag{3}$$

An ideal model, capable of continuous-time understanding, would move this point smoothly and linearly along the already predicted trajectory.

As illustrated in Figure 6a, BMM does not generalize to intermediate timings. The plot of the normalized distance does not increase linearly, but instead has the characteristics of a step-wise function. This result indicates that BMM has not learned a continuous representation of time but rather the fixed discrete time to which it was exposed during training. Thus, any intermediate control point is effectively mapped to its closest timing at the fixed stride, preventing the introduction of additional control points without causing severe artifacts in the motion. While changing the timing of the

Table 1: Handle accuracy evaluated on BMM and different versions of IBMM. The handle accuracy is measured over the full validation dataset with 10% intermediate handles.

Model	Handle. Acc. (cm)↓
BMM	2.563
BMM$^{+1024\text{ dim}}$	1.409
IBMM	**0.389**
IBMM	0.418 (Random)
IBMM$^{\text{-token, -align}}$	1.326
IBMM$^{\text{-align}}$	0.639
IBMM$^{\text{-token}}$	0.690

control points is still possible, the effects are, due to the non-linear behavior, hard to control and not artist friendly.

In contrast, the results for IBMM, shown in Figure 6b, demonstrate that the model generalizes to continuous timings. When subjected to the identical experimental setup, the normalized distance of the interpolated control point increases linearly. This smooth, linear progression confirms that IBMM has successfully learned a continuous representation of time. For more qualitative comparison of this behavior we refer to the supplementary video.

4.2.2 Handle accuracy. Accurately satisfying conditions is essential for artistic control. To evaluate our model's performance at meeting these conditions, we measure handle accuracy and compare it to BMM.

Since BMM only allows for sparse conditions to be set at a fixed stride, we evaluate IBMM on the same fixed stride, even though it was trained with sparse conditions on random frames.

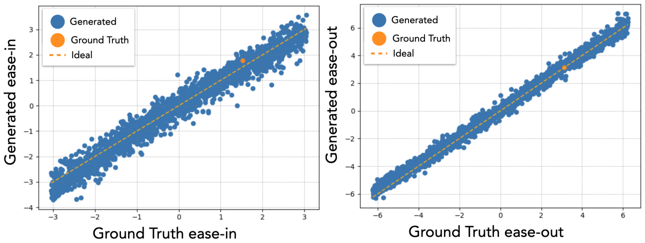
The results in Table 1 confirm that IBMM outperforms BMM significantly in accurately hitting sparse conditions. Also, when sampling sparse constraints at random frames (Random), IBMM shows a substantial improvement, demonstrating that our model also generalizes to this setting. Since IBMM uses an increased latent dimension compared to BMM, we also provide handle accuracy for BMM with the same increased latent dimension BMM$^{+1024\text{ dim}}$. IBMM$^{\text{-token, -align}}$, which uses BMM's condition tokens and alignment, achieves handle accuracy similar to that of BMM$^{+1024\text{ dim}}$, indicating that fitting the Bézier curves implicitly has no negative effect on performance. Evaluating IBMM$^{\text{-align}}$ and IBMM$^{\text{-token}}$ that leave out one of the two added components respectively shows that the new condition tokens and alignment both have a positive effect on the handle accuracy, with their combination leading to the best IBMM version.

4.2.3 Foot sliding. Minimizing foot sliding is crucial for generating physically plausible and realistic character animations. We evaluate our model's performance on foot sliding using our own metric designed for locomotion in particular that we call Foot Sliding Ratio Locomotion (FSRL). The metric is explained in more detail in Appendix E.

We conduct evaluations at strides of $s = 6$ and $s = 2$ on walking and running clips. A stride of 6 was proposed in the original BMM, while a smaller stride of 2 is expected to yield better performance, as a denser set of control points allows the Bézier curve to more accurately approximate the ground truth foot trajectory.

Table 2: Foot Sliding Ratio Locomotion measured on BMM and IBMM for different strides s on walking and running data. BMM is trained on different strides s, while for IBMM, s indicates the stride only for the evaluation. While the height threshold is fixed to 1.5 cm for walking and running, the lower and upper velocity threshold is provided in the table as $v_{lower} < v < v_{upper}$, where v is the foot velocity in cm per frame.

FSRL ↓	Stride 6		Stride 2		GT
	BMM	IBMM	BMM	IBMM	
Walk ($0.75 < v < 1.4$)	0.074	0.038	0.047	**0.037**	0.040
Run ($2.2 < v < 4$)	0.052	0.044	0.043	**0.023**	0.043



(a) ease-in **(b) ease-out**

Figure 7: Influence of the ease-in/out control for the generated motion samples. The samples indicate a linear relationship showing that IBMM has faithfully learned to follow the condition.

The results in Table 2 confirm that IBMM consistently outperforms BMM, with IBMM with stride 2 achieving the best FSRL for walking and running. Both BMM and IBMM improve when going from stride 6 to stride 2. This further corroborates that a higher temporal resolution is beneficial for ground contacts. However, the arbitrary timing nature of IBMM allows users to increase the temporal resolution at frames near ground contact and maintain sparse resolution for further away frames, benefiting both from reduced foot sliding and controllable generation.

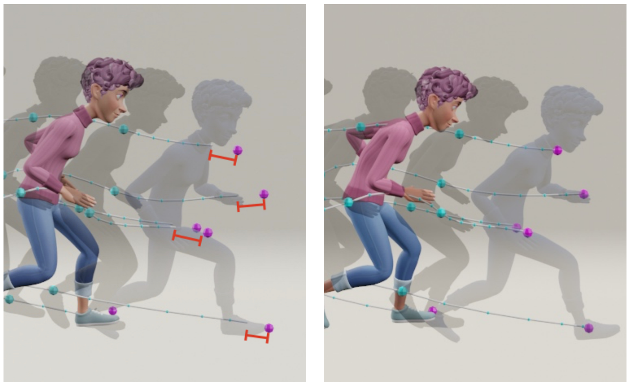
For both strides, IBMM has fewer frames classified as sliding compared to the ground truth motion. We expect that this is due to the foot sliding losses used during training that incentivize the model to learn better ground contact than simple reconstruction.

4.2.4 Ease-in/out control. In Figure 7, we vary E_i and E_o by a multiplier in the range $[-2,\ 2]$ from the ground truth. We plot the input ease-in/out value against the generated value. The orange lines in each plot represent perfectly generated motion with the specified amount of ease-in/out. We see that the generated ease-in/out follows this line closely for the entire range.

4.3 Qualitative Evaluation

We present evaluating our model on test clips from LaFAN1 and AMASS [Mahmood et al. 2019] datasets in our supporting video. In the following sections, we discuss the effects of our improvements on usability over [Studer et al. 2024].

4.3.1 Better constraints hitting. Figure 8 visually shows the improvement in handle accuracy. While BMM does not hit the constraints accurately, it is visible that IBMM manages to hit them much better. Note that no overwriting is used in this case, which would lead to perfectly hitting the conditions. However, simple overwriting does not take joint correlations into account and can lead to artifacts that are further elaborated in Appendix F.

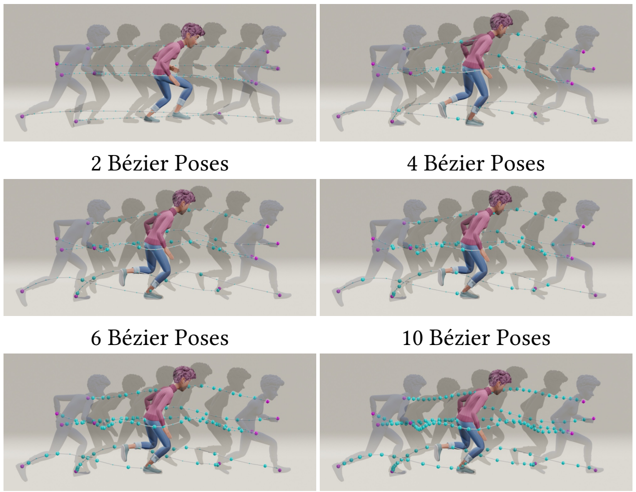


(a) BMM **(b) IBMM**

Figure 8: Qualitative comparison of the handle accuracy between BMM and IBMM. (a) shows the distance between the predictions and conditions (purple spheres) for BMM, while (b) shows that the predictions of IBMM are much closer to the conditions.

4.3.2 Different amounts of evenly spaced Bézier poses. Figure 9 shows the possibility of using various numbers of Bézier poses evenly sampled along the 30 frames of the predicted motion. As can be seen, having only two Bézier poses at the start and end frames results in near-linear interpolation. When adding more Bézier poses, IBMM starts to predict reasonable walking motions. It can be seen that the influence of additional Bézier poses decreases as their total number increases, effectively converging to a dense curve. Interestingly, this behavior resembles how a curve is fitted by Bézier curves at different levels of granularity.

4.3.3 Dense and sparse conditions. In addition to the possibility of different amounts of evenly spaced Bézier poses, the model also allows for dense and sparse conditioning at the same time. Figure 10 shows an example where dense conditions on the hip trajectory are used at the beginning of the motion, whereas towards the end, sparse conditions are defined on the trajectory of the right hand. As shown in the supplementary video, this flexibility is in particular helpful to better add detail in some parts of the motion, like creating steady steps over objects.



2 Bézier Poses

4 Bézier Poses

6 Bézier Poses

10 Bézier Poses

20 Bézier Poses

30 Bézier Poses

Figure 9: Illustration of sampling different numbers of evenly spaced Bézier poses with IBMM.

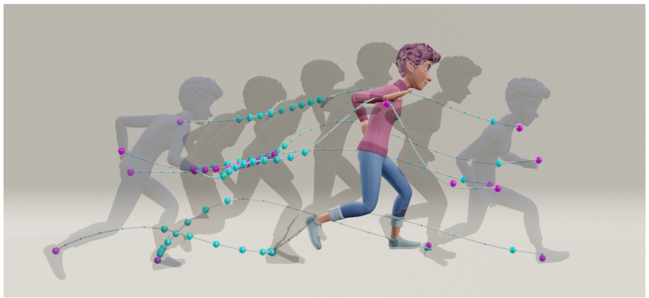


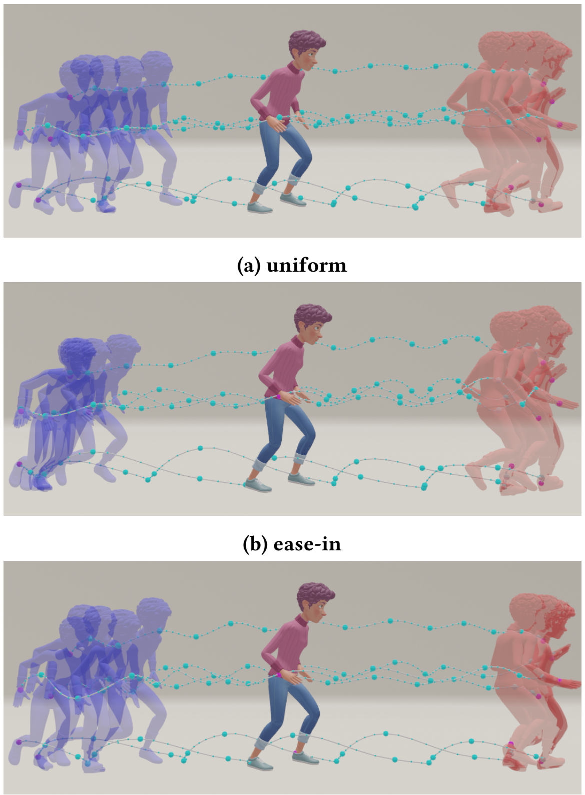
Figure 10: Illustration of dense and sparse conditioning of the Bézier poses. IBMM allows to create dense conditions on the hip in the beginning of the motion, while having sparse conditions on the right hand towards the end of the motion.

4.3.4 Ease-in/out control. We evaluate ease control for IBMM in Figure 11. The blue and red silhouettes show every fourth frame at the start and end of the motion sequence, respectively. In Figure 11a, one can see the motion generated by the model when both E_i and E_o are set to 0. Since the silhouettes are evenly spaced out, the character is moving at constant speed throughout the motion.

In contrast, in Figure 11b, one can observe that the starting frames, in blue, are closer together, showing that the character starts slowly and then accelerates. Lastly, the character slows down into the target keyframe in Figure 11c—depicting ease-out motion. Due to the continuous nature of this condition, the user can have a wide spectrum of possible motion profiles. For more examples, we refer to the supplementary video.

4.4 Animation Example with IBMM

In the supplementary video, we present an animation example that shows our model's capabilities. The entire animation was created within a custom user interface in Blender, which enables the user


(a) uniform

(b) ease-in

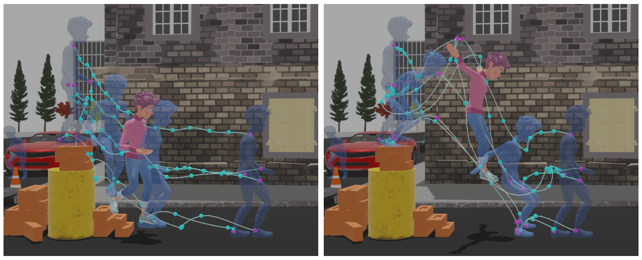
(c) ease-out

Figure 11: Comparison of different values for E_i and E_o: (a) Uniform motion with $E_i = 0$ and $E_o = 0$, (b) Ease-in motion with negative E_i, (c) Ease-out motion with positive E_o.

to control the generation process by adding constraints, choosing the number of Bézier controls, moving control points, sampling different motions, and adjusting the ease-in/out of a predicted motion.

The creation of the animation involved three main steps. First, we define a sparse set of keyposes to outline the desired motion in space and in time (90 frames apart). Next, these keyposes are automatically interpolated using IBMM to generate a smooth, continuous motion sequence. Finally, we refine the animation by adding and editing constraints to specific body parts to achieve the intended behavior with the objects in the scene.

This process allows for intuitive editing. When jumping or running over obstacles, like the car or the bricks, it is sufficient to adapt a few generated steps to the correct height and location of the feet placements. Changing the direction of the motion can be best manipulated with moving the hip trajectory and adjusting the foot placement if needed. For jumping off the stack of bricks, the model initially predicted a stair-walking motion as shown in Figure 12a. To create the jump, we added positional constraints on the hip and feet and tweaked their positions and timings such that the model knew when and where the takeoff and landing should be. In addition, we added one positional constraint on the hand to make the motion more dynamic in mid air.


(a) without edits

(b) with edits

Figure 12: Example of editing an animation with IBMM. (a) shows the model's initial prediction and (b) the final edited animation.

To create the animation of sitting down on the bench, we added multiple foot constraints and locked them to the same position to eliminate foot sliding that occurred during the initial prediction. A constraint on the right hand was used to steer the timing of the character's turn before sitting down. Note that the training dataset did not contain any sitting motions.

For the transition between keyposes, the model automatically creates dynamics that match the keypose. For example, when transitioning from a static standing pose to a running pose, the model generates a natural acceleration, with the character's speed increasing as it approaches the final running pose, while transitioning from a walking to a running motion. To further push the dynamics between the transitions, we relied on the ease-in/out mechanism of our model.

For details on how the stochastic nature of the diffusion model can be used, we refer to Appendix G.

5 DISCUSSION

Our work builds on a new insight in motion diffusion [Bae et al. 2025; Hwang et al. 2025; Studer et al. 2024], that sparsifying the input signal improves motion generation, and in the case of BMM [Studer et al. 2024] enables more precise spatial control.

However, BMM constrained artists to control poses only at fixed Bézier timing intervals–thereby limiting the type of animations that could be created. In this work, we removed this constraint by introducing an implicit BMM (i.e. IBMM) that works on an arbitrary number of control points at arbitrary timings. This not only increases the space of animations artists can create, but also allows additional control via the timing of the Bézier control points. Since control points can be at arbitrary timings, an artist can move one in the past or future and receive visual feedback from the generated motion, as shown in our supporting video. This further allows the artist to refine their animations and facilitates iterative workflows.

We further showed that an improved architecture and alignment of the data has large effects on motion fidelity and improves spatial control. Additionally, inspired by traditional animation principles, we also introduced a novel control mechanism over the ease-in and ease-out of motion. We showed that we could mathematically capture this principle and train a diffusion model to learn it directly from data, without the need for manual labeling. The trained model then follows this condition and even extrapolates to ease-in/out levels not present in the original dataset.

When training BMM, previous work [Studer et al. 2024] stated that while losses on the sparse Bézier points and tangents were important, they needed dense trajectory losses to reach the best motion fidelity. In our work, we demonstrate that these dense losses are not only auxiliary losses but sufficient to learn a Bézier fit of the data implicitly during training. Thereby, our simpler training setup eliminates the need for (online) Bézier fitting for the sampled control points, which would otherwise make training substantially slower.

As we build on BMM, we inherit many of its advantages. This includes being skeleton-agnostic and being able to work with smaller datasets. However, we also inherit some of its limitations. In particular, we do not address the IK module stage. Hence, artists can still only control the joints in J, and the frame-based IK module might not always be temporally smooth. Additionally, since the input to the model consists of pose tokens, we are currently constrained to having the same Bézier timings for all of the J joints. In comparison, in some artists' workflow keying happens at the joint level or for individual rig-controls and not for the full pose. An extension enabling this would be to switch to tokens at the joint level with independently sampled Bézier timings. However, in our experience, this goes against the sparsity motivation and reduces the controllability of the model.

Further possible extensions to our work include the introduction of other conditioning signals, such as text or audio descriptions, for more high-level control. As we focused on motion inbetweening, we assumed that the boundary frames are always provided. For a more generalized motion model that can be used for other tasks than inbetweening, this assumption should be relaxed.

6 CONCLUSION

In this work, we introduced the Implicit Bézier Motion Model (IBMM) that provides fine-grained spatial and temporal control over generated movements. We overcame a core limitation of the previous Bézier Motion Model (BMM) which is the fixed temporal stride of control points. IBMM eliminates the concept of stride altogether and enables artists to constrain any end-effector joint at any frame in time.

We also introduced a new global control to users: a direct handle for the global ease-in and ease-out of the motion. To our knowledge, this is the first global control over timing–while generating natural motion–without recourse to human labeling.

By improving controllability of generative motion models and bringing them closer to artists' workflows, we believe that this type of neural motion authoring can more easily be adopted for content creation in the professional creative industry and also enables more inexperienced users to engage with animation. We also hope that our work inspires future research in defining and measuring animation principles, such as *anticipation* or *squash/stretch*, that can be used as high-level controls that go beyond ease-in and ease-out.

ACKNOWLEDGMENTS

We would like to thank Nadine Arendt for insightful discussions and help in this work. Additionally, we want to thank our artists Violaine Fayolle and Dorianno VanEssen for their support with 3D assets and animation scenes.

REFERENCES

Dhruv Agrawal, Jakob Buhmann, Dominik Borer, Robert W Sumner, and Martin Guay. 2024. SKEL-Betweener: a Neural Motion Rig for Interactive Motion Authoring. *ACM Transactions on Graphics (TOG)* 43, 6 (2024), 1–11.

Dhruv Agrawal, Martin Guay, Jakob Buhmann, Dominik Borer, and Robert W. Sumner. 2023. Pose and Skeleton-aware Neural IK for Pose and Motion Editing. In *SIGGRAPH Asia 2023 Conference Papers*. 1–10.

Elly Akhoundi, Hung Yu Ling, Anup Anand Deshmukh, and Judith Bütepage. 2025. SILK: Smooth InterpoLation frameworK for motion in-betweening. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2900–2909.

Autodesk Inc. 2025. Autodesk Maya - a 3D modeling, animation, and rendering software. http://www.autodesk.com/maya Version 2025, Autodesk, Inc., San Rafael, CA.

Jinseok Bae, Inwoo Hwang, Young Yoon Lee, Ziyu Guo, Joseph Liu, Yizhak Ben-Shabat, Young Min Kim, and Mubbasir Kapadia. 2025. Less is More: Improving Motion Diffusion Models with Sparse Keyframes. *arXiv preprint arXiv:2503.13859* (2025).

Blender Online Community. 2025. Blender - a 3D modelling and rendering package. http://www.blender.org Blender Foundation, Stichting Blender Foundation, Amsterdam.

Loïc Ciccone, Cengiz Öztireli, and Robert W Sumner. 2019. Tangent-space optimization for interactive animation control. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–10.

Setareh Cohan, Guy Tevet, Daniele Reda, Xue Bin Peng, and Michiel van de Panne. 2024. Flexible motion in-betweening with diffusion models. In *ACM SIGGRAPH 2024 Conference Papers*. 1–9.

Rishabh Dabral, Muhammad Hamza Mughal, Vladislav Golyanik, and Christian Theobalt. 2023. Mofusion: A framework for denoising-diffusion-based motion synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9760–9770.

Purvi Goel, Haotian Zhang, C Karen Liu, and Kayvon Fatahalian. 2025. Generative Motion Infilling from Imprecisely Timed Keyframes. In *Computer Graphics Forum*. Wiley Online Library, e70060.

Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. 2024. Momask: Generative masked modeling of 3d human motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1900–1910.

Félix G Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. 2020. Robust motion in-betweening. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 60–1.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

Jing Huang, Qi Wang, Marco Fratarcangeli, Ke Yan, and Catherine Pelachaud. 2017. Multi-variate gaussian-based inverse kinematics. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 418–428.

Inwoo Hwang, Jinseok Bae, Donggeun Lim, and Young Min Kim. 2025. Motion Synthesis with Sparse and Flexible Keyjoint Control. *arXiv preprint arXiv:2503.15557* (2025).

Korrawe Karunratanakul, Konpat Preechakul, Emre Aksan, Thabo Beeler, Supasorn Suwajanakorn, and Siyu Tang. 2024. Optimizing diffusion noise can serve as universal motion priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1334–1345.

Jiefeng Li, Jinkun Cao, Haotian Zhang, Davis Rempe, Jan Kautz, Umar Iqbal, and Ye Yuan. 2025. GENMO: A GENeralist Model for Human MOtion. *arXiv preprint arXiv:2505.01425* (2025).

Ting-Hsuan Liao, Yi Zhou, Yu Shen, Chun-Hao Paul Huang, Saayan Mitra, Jia-Bin Huang, and Uttaran Bhattacharya. 2025. Shape my moves: Text-driven shape-aware synthesis of human motions. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 1917–1928.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34, 6 (Oct. 2015), 248:1–248:16.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. 2019. AMASS: Archive of Motion Capture as Surface Shapes. In *International Conference on Computer Vision*. 5442–5451.

Zichong Meng, Yiming Xie, Xiaogang Peng, Zeyu Han, and Huaizu Jiang. 2025. Rethinking Diffusion for Text-Driven Human Motion Generation: Redundant Representations, Evaluation, and Masked Autoregression. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 27859–27871.

Yuxuan Mu, Hung Yu Ling, Yi Shi, Ismael Baira Ojeda, Pengcheng Xi, Chang Shu, Fabio Zinno, and Xue Bin Peng. 2025. StableMotion: Training Motion Cleanup Models with Unpaired Corrupted Data. *arXiv preprint arXiv:2505.03154* (2025).

Boris N. Oreshkin, Florent Bocquelet, Felix G. Harvey, Bay Raitt, and Dominic Laflamme. 2022. ProtoRes: Proto-Residual Network for Pose Authoring via Learned Inverse Kinematics. In *International Conference on Learning Representations*. https://openreview.net/forum?id=s03AQxehtd_

Jose Luis Ponton, Eduard Pujol, Andreas Aristidou, Carlos Andujar, and Nuria Pelechano. 2025. Dragposer: Motion reconstruction from variable sparse tracking signals via latent space optimization. In *Computer Graphics Forum*, Vol. 44. Wiley Online Library, e70026.

Jose Luis Ponton, Haoran Yun, Andreas Aristidou, Carlos Andujar, and Nuria Pelechano. 2023. Sparseposer: Real-time full-body motion reconstruction from sparse data. *ACM Transactions on Graphics* 43, 1 (2023), 1–14.

Jia Qin, Youyi Zheng, and Kun Zhou. 2022. Motion In-Betweening via Two-Stage Transformers. *ACM Trans. Graph.* 41, 6 (2022), 184–1.

Junyu Shi, Lijiang Liu, Yong Sun, Zhiyuan Zhang, Jinni Zhou, and Qiang Nie. 2025. GenM^3: Generative Pretrained Multi-path Motion Model for Text Conditional Human Motion Generation. *arXiv preprint arXiv:2503.14919* (2025).

Justin Studer, Dhruv Agrawal, Dominik Borer, Seyedmorteza Sadat, Robert W Sumner, Martin Guay, and Jakob Buhmann. 2024. Factorized Motion Diffusion for Precise and Character-Agnostic Motion Inbetweening. In *Proceedings of the 17th ACM SIGGRAPH Conference on Motion, Interaction, and Games*. 1–10.

Xiangjun Tang, He Wang, Bo Hu, Xu Gong, Ruifan Yi, Qilong Kou, and Xiaogang Jin. 2022. Real-time controllable motion transition for characters. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–10.

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

Jonathan Tseng, Rodrigo Castellon, and Karen Liu. 2023. Edge: Editable dance generation from music. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 448–458.

Yan Wu, Jiahao Wang, Yan Zhang, Siwei Zhang, Otmar Hilliges, Fisher Yu, and Siyu Tang. 2022. Saga: Stochastic whole-body grasping with contact. In *European Conference on Computer Vision*. Springer, 257–274.

Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. 2024. OmniControl: Control Any Joint at Any Time for Human Motion Generation. In *The Twelfth International Conference on Learning Representations*.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023b. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3836–3847.

Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. 2023a. Remodiffuse: Retrieval-augmented motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 364–373.

Zeyu Zhang, Yiran Wang, Wei Mao, Danning Li, Rui Zhao, Biao Wu, Zirui Song, Bohan Zhuang, Ian Reid, and Richard Hartley. 2025. Motion Anything: Any to Motion Generation. *arXiv preprint arXiv:2503.06955* (2025).

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. 2019. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5745–5753.

A SAMPLING STRATEGY

To sample the indices during training so that the model can learn all the different combinations of Bézier poses in different frames, we use a sampling strategy that we call random bins. This sampling strategy chooses a random number of bins for each batch. The segments are then divided into those bins, and in each bin, exactly one index is randomly sampled. A visual example can be seen in Figure 13. This sampling strategy ensures a fairly even distribution of indices along the segments, while still including enough randomness to allow the model to generalize to all input combinations. The interval that decides the number of bins should be chosen reasonably, since too few or too many bins can cause wasted training time or make it more difficult for the model to converge.

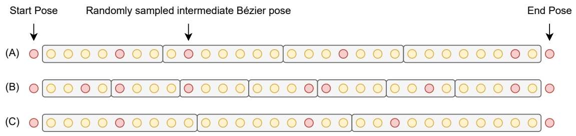
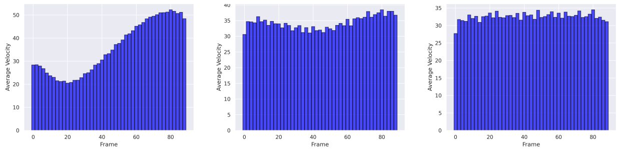


Figure 13: Explanation of the sampling strategy for control points. A random number of bins is chosen and in every bin exactly one random frame is selected to be an input pose. (A), (B), and (C) are examples of possible samples. Red points indicate that a pose is predicted in this frame. Note that in the start and end frame the pose is always predicted.

B IMPORTANCE OF ALIGNING AND NORMALIZING THE DATA

When training our model for 90 frames, with the alignment strategy as for BMM, we observed a strong temporal bias in the generated motion. The predictions are densely clustered at the beginning of the motion sequence and become progressively sparser towards the end. This behavior can be seen in Figure 14a showing the average velocity of each frame measured on approximately 7000 running samples. The predicted motions start with a low initial velocity that accelerates to a high final velocity. The reason for this is that in the data preprocessing of BMM the start pose of each motion gets shifted to the origin. Since the predicted motion is in a global space, we hypothesize that it is easier for the denoiser to map the zero-centered gaussian noise to poses around the origin since it does not need to account for the additional shift in global position. Additionally, poses close to the origin are also more frequent compared to poses further away. We believe these imbalances lead to the bias towards predicting poses close to the origin.

Our proposed alignment simplifies the problem of mapping the original noise vector to the final motion. Rotating every clip to align on the same axis and shifting the middle of the clip to the origin yields a significant improvement as can be seen in Figure 14b. The bias decreases and is shifted towards the middle of the predicted motion, since it now acts as the origin. Adding normalization on top, then fully removes the bias as can be seen in Figure 14c.



(a) (b) (c)

Figure 14: Comparison of average velocities of each frame over ~7k running samples. (a) IBMM without alignment and without normalization, (b) IBMM with alignment, and (c) IBMM with alignment and normalization.

C DIFFERENT ARCHITECTURE DESIGNS IBMM

Dense version of IBMM. One drawback of the current design choice of IBMM is that during inference, the predicted tangents are never reintroduced in the diffusion process and only used from the final inference step. Thus, the tangents predictions are not really produced via a diffusion denoising process but rather a direct regression that is conditioned on the denoised poses.

In order to reintroduce the tangents into the denoising process, the tangents would need to be represented as an input to the network. For BMM, this was possible due to the pre-fitting of the data, which does not exist in our case. However, there is also a way to implicitly add the information from the tangents back to the input via sampling the Bézier curves. In this case, the input to the network would be the dense motion that can be easily noised from ground truth data for training, and sampled from the predictions during inference. Hence, we refer to this design as the dense IBMM. Lastly, we add an additional flag to the input to indicate at which timestamps the model should predict the Bézier poses.

While this version of the network also learns to create motion, we observed that the training is overall less stable and the final model shows more artifacts in its generated results. Thus, we could not justify the added complexity for the purpose of adding the tangents back into the diffusion process. However, we imagine that a diffusion process that includes the tangents could potentially offer advantages over IBMM. For instance, one could directly control the tangents via inpainting throughout the whole diffusion process.

Per-joint version of IBMM. While keyframing is a long standing animation workflow, when creating and tweaking animations artists might set keys for individual rig controls and not for all controls at that frame. While IBMM allows setting keyframes at arbitrary timings, the notion of a keyframe is defined at a pose level and not at the joint level. Hence, adding a control point for time t for one joint would also add a prediction for all other joints at t. To be closer to a joint-level design, we would need to represent the inputs to the transformer also at a joint level.

We tested this design of a per-joint version of IBMM, where the input is not a full pose, but each joint individually, i.e. the position and the joint ID. While the model can learn to faithfully create motion from the training set and hit constraints very accurately, this model suffers when users move constraints. The individual joint trajectories become very independent and do not capture the correlations between the joints that are needed for a faithful pose

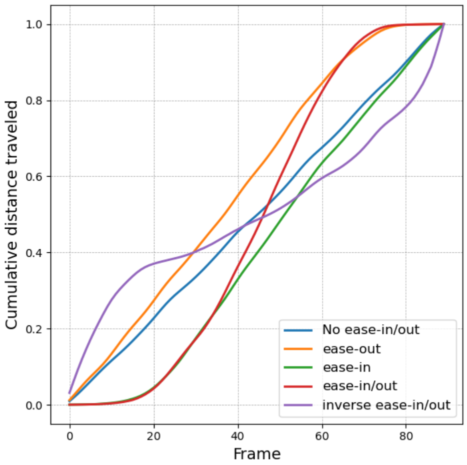


Figure 15: Per-frame cumulative distance traveled by COM for different combinations of ease-in and ease-out controls for a single motion clip.

within a frame. Additionally, the per-joint encoding leads to a much longer input sequence length and, thereby, a slower model.

D EASE-IN/OUT

To illustrate the effect of ease-in and ease-out controls on the timing of the motion, we plot the cumulative distance traveled as a function of the current frame for different combinations of ease-in/out controls in Figure 15. When no ease-in/out is applied to the clip (blue), one can observe a constant velocity throughout the motion clip. In contrast, applying only ease-in (green), the motion starts with a lower velocity, as the COM travels a smaller ratio of the total distance at the start, and then accelerates to a peak velocity for the rest of the clip. Applying only ease-out (orange) conversely results in a constant velocity at the start but then decelerates towards the end of the clip. The curves in red and purple show that both controls can be applied together and can even be inverted.

E EVALUATION METRICS

E.1 Handle Accuracy

The handle accuracy measures how accurately the conditions are hit by the predicted Bézier control points. Hitting the constraints closely is essential for controllability. It is calculated on all joints j of a single sample as,

$$\text{HandleAcc.}\left([\hat{p}_i]_1^N,\ C\right) = \frac{1}{|C|} \sum_{c_{\tau_i,j} \in C} \|(\hat{p}_{\tau_i,j} - c_{\tau_i,j})\|_2, \qquad (4)$$

where C is the set of all conditions with $c_{\tau_i,j}$ being the 3D position for joint j at frame τ_i and $\hat{p}_{\tau_i,j}$ being the corresponding prediction.

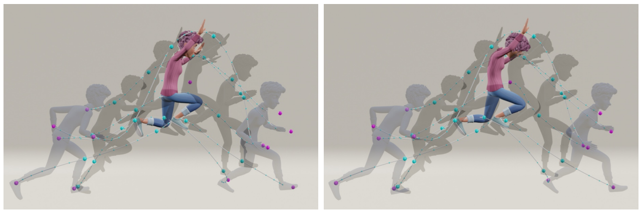
E.2 Foot Sliding Ratio Locomotion

A widely used metric to measure foot sliding is the Foot Sliding Ratio (FSR) [Wu et al. 2022]. FSR measures foot sliding when both feet are below a height threshold and above a velocity threshold. For a locomotion dataset such as LaFAN1 [Harvey et al. 2020], this ignores the majority of frames where only one foot is in contact with the ground at a time. Hence, we report our own more suitable foot sliding metric, which we call FSR Locomotion (FSRL). FSRL is the ratio of frames where an individual foot's height is below a height threshold, while its planar velocity is within a specified range, to the total number of frames where the foot's height is below the threshold. Compared to FSR, we introduce an upper bound for velocity to filter out false positives in the ground truth motion. During Locomotion, the feet can have sudden accelerations and move fast before the onset of a step.

We empirically find that a height threshold of 1.5 cm to detect ground contact gives a good trade-off between accounting for shoe or joint location variation across subjects and counting too many false positives. Using this height threshold, we measure that the foot is on the ground for roughly 14 frames per step for walking and 5 frames per step for running.

To define the velocity range, we statistically measure all velocities for which the foot height is below the height threshold. We then visually determined that using the 90th and 94th percentile as the lower and upper bound, respectively, most consistently detects foot sliding. This resulted in the velocity window $[0.75, 1.4]$cm/frame for walking and $[2.2, 4]$cm/frame for running. In each case, this equates to cumulative foot sliding of $[10.5, 20]$cm/step.

With FSRL, we attempt to build a standardized foot sliding ratio for easy comparisons across works. Foot sliding is one of the most noticeable artifacts in an animation and FSRL can provide an important measure of the quality of the generated motion, particularly for datasets where pre-trained feature extractors are not available to measure FID.



(a) no overwrite **(b) overwrite**

Figure 16: Limitation of the overwriting mechanism. (a) Original prediction of BMM without overwriting. (b) Overwriting prediction of BMM with the ground truth conditions leads to an unnatural pose at the peak of the jump.

F ARTIFACTS AFTER OVERWRITING

To perfectly hit the constraints, [Studer et al. 2024] propose an overwriting mechanism that replaces the positions of the control point with its corresponding constraint position before the dense trajectories are sampled. However, this overwriting only influences the specific Bézier control point to which the condition belongs. Hence, it has no influence on other Bézier control points belonging to the same joint and also does not adjust the other trajectories. This can lead to unreasonable poses if the overwritten Bézier control point is moved too far, as shown in Figure 16. There, the corrected hip

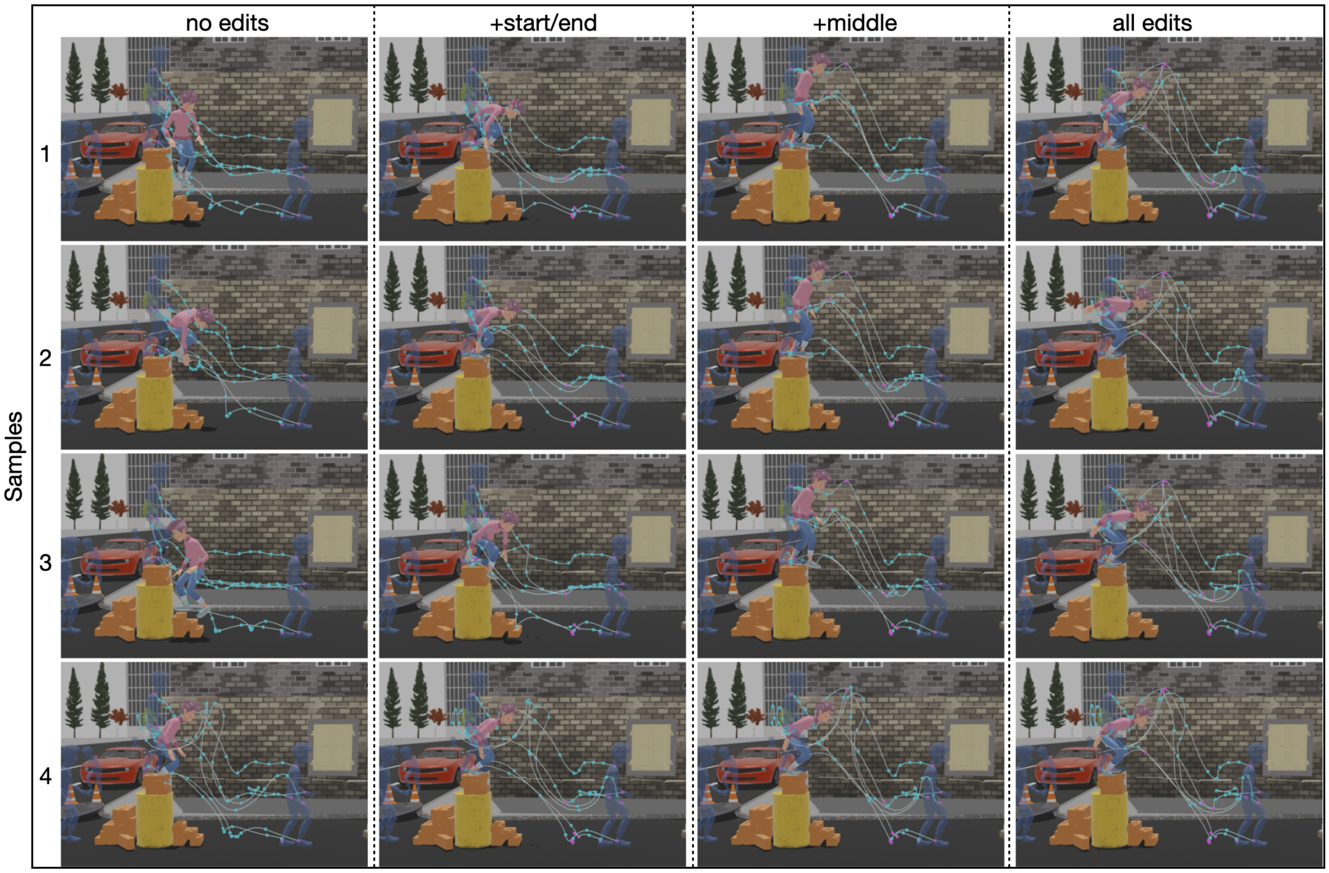


Figure 17: The vertical axis shows different samples for the task of animating the jump from the stack of bricks. Along the horizontal axis more manual edits are added to the input. Note, the original edits have been created based on the first sample.

position is too close to the head position resulting in an unnatural pose.

G INFLUENCE OF NOISE SAMPLE FOR ANIMATION TASK

Since the training data of LaFAN1 [Harvey et al. 2020] includes not only locomotion but also interactions with obstacles, we naturally get interesting motion when the start or end pose are not at ground level. For instance, when animating the stepping onto the bricks or the jump off them, for the animation example in the supplementary video, we observed motions such as walking up stairs, jumping, or other motions that seem to interact with an obstacle.

In Figure 17, we show this for four distinct samples. In the unconditional case (first row), the model can create different motions, such as 1) walking down stairs, 2) using the hand to safely hopping down, 3) walking down backwards, and 4) jumping down. While the last sample already creates a jump from the start, and is easier to shape into the desired motion, other samples require more constraints. However, in the end all samples lead to a satisfactory result. This shows that direct spatial control is sufficient for such an animation task, however, finding a “good” noise vector can also significantly help to speed up the animation process.

In our demo in Blender, the user is able to set the seed (as an integer) of the noise sampler, thus allowing to quickly explore multiple starting points for the animation and key them to the specific animation segment. While this search can be done interactively, a more direct way of finding a “good” noise vector could improve the overall user experience. Hence high level conditions such as text or sketch could be a useful complimentary input modality to the precise spatial control of IBMM. Exploring a multi-module system, ways to find optimal noise vectors, and a better user interface are interesting directions for future research.