

Special Section on MIG 2025

IBMM++: Improved and versatile motion generation with implicit Bézier curves[☆]

Dhruv Agrawal^{a,b,*,1}, Luca Vögeli^{b,1}, Dominik Borer^b, Robert W. Sumner^{b,a},
Martin Guay^b, Jakob Buhmann^b

^a ETH Zürich, Zürich, Switzerland^b DisneyResearch|Studios, Zürich, Switzerland

ARTICLE INFO

Keywords:

Animation
Machine learning
Interaction design

ABSTRACT

Creating high-quality character animation remains an intricate and cumbersome process. 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 Vögeli et al. we presented a new *Implicit* Bézier Motion Model (IBMM), which during training is exposed to all possible configurations of control points, enabling control at arbitrary timings. In addition, we introduced 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. In this extended version (IBMM++), we further improve constraint accuracy, reduce foot sliding, and enable motion extension, orientation constraints, longer sequences, and stable iterative editing through architectural updates, a revised tangent formulation, a randomized constraint sampling strategy, and dedicated inference-time methods, validated on a larger internal dataset with more varied motions.

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 [1,2], 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 [3,4] in facilitating animation tasks. These models can generate high-quality human movements from text [5] or music [6], they can be leveraged to improve motion capture [7,8], or help in traditional animation tasks such as inbetweening [3,9]. 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 [10,11], and the controllability [12] of the generated motion. In particular, Bézier Motion Model (BMM) [12] 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 Vögeli et al. [13], we presented 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

[☆] This article is part of a Special issue entitled: 'CAG_SS_MIG 2025' published in Computers & Graphics.

* Corresponding author at: ETH Zürich, Zürich, Switzerland.

E-mail address: dhruv.agrawal@disneyresearch.com (D. Agrawal).

¹ Equal Contribution.

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.

While the preliminary version of IBMM [13] established arbitrary temporal control, deploying the model inside iterative artist workflows revealed also some practical limitations. Constraint accuracy degraded when many handles were specified simultaneously — for instance when editing or stitching mocap clips — because attention to nearby conditions was diluted by distant ones. The fixed sinusoidal positional encoding prevented the model from generating sequences longer than those it had seen during training, and the requirement of always conditioning on the first and last frames precluded motion extension beyond the input boundaries. Finally, naively adding new control points to an already predicted curve could cause large, abrupt changes that disrupt iterative editing.

This extended version (IBMM++) targets each of these limitations. We update the transformer backbone to use a two-stage self- then cross-attention scheme combined with rotary positional encoding [14], which sharpens attention on the relevant local conditions and lets the model extrapolate beyond the trained sequence length. We revise the Bézier tangent formulation to use asymmetric in/out tangents scaled by the inter-control-point distance, which naturally reduces foot sliding when control points are close in time, such as during foot contacts. We replace the boundary-frame alignment with a randomized constraint sampling strategy that requires only two arbitrary anchor constraints, which enables motion extension. At inference, we leverage a top-k attention masking to focus the model on the most relevant nearby constraints when many handles are present, smooth noise modeling to make adding new control points a continuous editing operation, and cross-window blending to maintain continuity across stitched motion segments. We evaluate our model on inbetweening to compare against [12], and additionally ablate each of our architectural and inference-time extensions against the preliminary IBMM [13]. We validate these improvements both quantitatively, on various metrics, and qualitatively, on a larger internal dataset spanning more varied motions than LaFAN1 [15] alone.

Our contributions in this extended version are as follows:

- An improved transformer backbone combining cross-attention and rotary positional encoding with an asymmetric, distance-scaled Bézier tangent formulation, which together raise constraint accuracy, reduce foot sliding, and enable generation beyond the trained sequence length.
- A randomized constraint sampling strategy at training time that removes the requirement of fixed boundary frames and adds support for orientation constraints and motion extension.
- Inference-time mechanisms, such as top-k attention masking, smooth noise modeling, and cross-window blending, that scale to dense constraint sets, stable iterative editing, and stitched motion segments.

2. Related works

Synthesis of character motion in traditional motion authoring software such as Maya [16] and Blender [17] has been limited to keyframing rig parameters and traditional interpolation. [18] 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 [15,19] employed autoregressive methods to predict dense motion given current state and a singular target frame. [20] 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, [21] show that a simple method with a single transformer predicting in a root-local space can match or outperform more elaborate methods. [22] 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 [23] and VQ-based [4,24], have been widely adopted for motion generation at scale. Early works such as [25–27] trained denoising diffusion models to generate motion from a text or music prompt. [28] 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. [29] allow generating motion for different body shapes by conditioning their model with SMPL [30] shape parameters in addition to text. GenMO [8] is one of the first multi-modal diffusion models that can be conditioned concurrently with text, music, keyframes, and images. [5] 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 [31,32], noise optimization [33], and conditioning [3,34] has been extensively explored. Zhao et al. [35] extend this line of work to a real-time, autoregressive setting that ingests streaming text and waypoint constraints, while Kimodo [36] scales the model and training data to support a broad constraint vocabulary spanning full-body keyframes, sparse joint positions and rotations, 2D waypoints, and foot contacts. 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. [10] reduce temporal resolution by analytically finding the most important frames within a motion sequence, training a diffusion model to generate these frames, and interpolating the remaining frames before decoding. [11] 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. AutoKeyframe [37] sparsifies in time from a different angle by autoregressively predicting the keyframes themselves from a trajectory and action label, which can then be passed to a subsequent in-betweening model. Closer to the in-betweening setting, Yun et al. [38] leverage video diffusion priors to in-between arbitrary characters. All of these approaches, however, assume that keyframes are placed at fixed or pre-specified timings. Goel et al. [9] relax this by predicting a time-warping function that retimes mistimed keyframes.

Our work is most closely related to the Bézier Motion Model (BMM) [12], 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 [1,2,20,39–41]. 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.

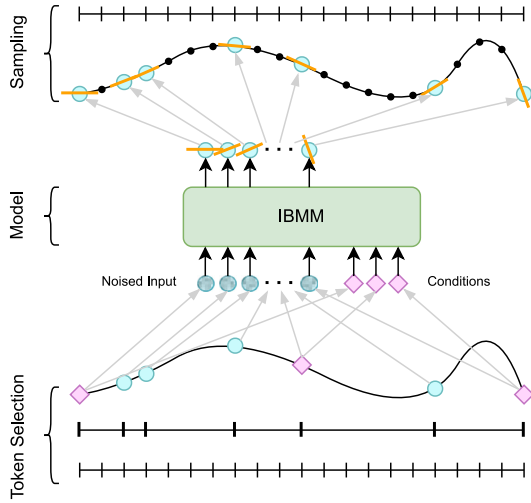


Fig. 1. Overview of the temporal token selection and sampling of IBMM. A random set of temporal control points (●) can be specified together with spatial constraints (◆). IBMM denotes the input and maps it to Bézier control points (positions and tangents) at the specified temporal indices. Finally, the dense curves are sampled to reconstruct the full motion.

3. Methodology

IBMM builds upon BMM [12], 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 [1]. But instead of a fixed stride between the Bézier control points, our method allows any arbitrary temporal discretization, as depicted in Fig. 1. 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 $r_i \in \mathbb{R}^6$ [42] for the joints J at the i th frame. For better comparison, we use 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.² In order to generate the whole sequence we assume $B \geq 2$, and τ_1 and τ_B are always set to 1 and N respectively.

The input poses $[x_{\tau_i}^t]_1^B$ 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.

² The binning helps to space out the control points to prevent sampling in just one region.

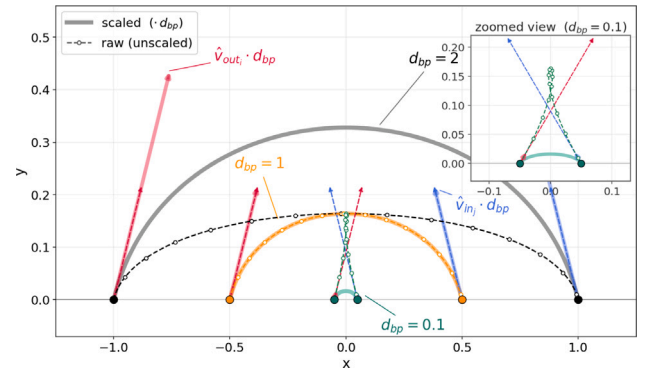


Fig. 2. Illustration of scaling tangents according to the distance between the control points.

Adaptive tangent scaling. Unlike BMM and Vögeli et al. [13], we update the Bézier representation to allow unsymmetrical tangents at the each control point. Therefore, $\hat{v}_{\tau_i}$ is a six dimensional vector that is split into $[\hat{v}_{in_{\tau_i}}, \hat{v}_{out_{\tau_i}}]$ where $\hat{v}_{in_{\tau_i}}$ and $\hat{v}_{out_{\tau_i}}$ are tangents entering and exiting each control point respectively. Note that $\hat{v}_{in_0}$ and $\hat{v}_{out_N}$ are invalid and dropped during interpolation.

Furthermore, to improve foot sliding in the generated motion, we scale the tangents between two consecutive control points, $\hat{v}_{out_i}$ and $\hat{v}_{in_j}$, using the euclidean distance between $\hat{p}_i$ and $\hat{p}_j$. The effect of this scaling is illustrated in Fig. 2 where we decrease the distance (d_{bp}) between the two control points but maintain the length of the tangents. When $d_{bp} = 1$ (orange) the tangents are unchanged, thus the two interpolations are identical. However, when two control points are very close $d_{bp} \approx 0$, such as during a foot step, larger tangents cause interpolation to move away from the control points, as shown by the dashed green curve. By automatically scaling the tangents, the green curve also shrinks with the distance, which for the trajectory of a foot would reduce sliding artifacts.

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 [1] as in BMM.

3.2. Model architecture

An overview of the model architecture can be seen in Fig. 3. The original IBMM [13] leveraged a transformer encoder as the backbone architecture for the diffusion model. The input sequence consisted 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 were embedded with a linear layer, while the noise level token went through an MLP. The temporal information was then added to the feature vector with sinusoidal positional encoding (PE).

In this work, we build upon the IBMM backbone by making two fundamental changes to improve constraints accuracy and generate longer motion sequences. Firstly, the original IBMM treated the conditions and the generated motion as a single sequence in the transformer encoder and thus the generated tokens simultaneously attended both to the other tokens and the conditions. This hurt its constraints accuracy as the attention to the conditions is reduced due very similar neighboring generated frames. We now instead split the attention into two stages, first, a self-attention block among the generated frames and, second, a cross-attention between the generated frames to the conditions. This second cross-attention step ensures that each condition is strongly attended to by its corresponding generated frame. Lastly, the global conditions, diffusion timestep and ease-in/out, that were originally

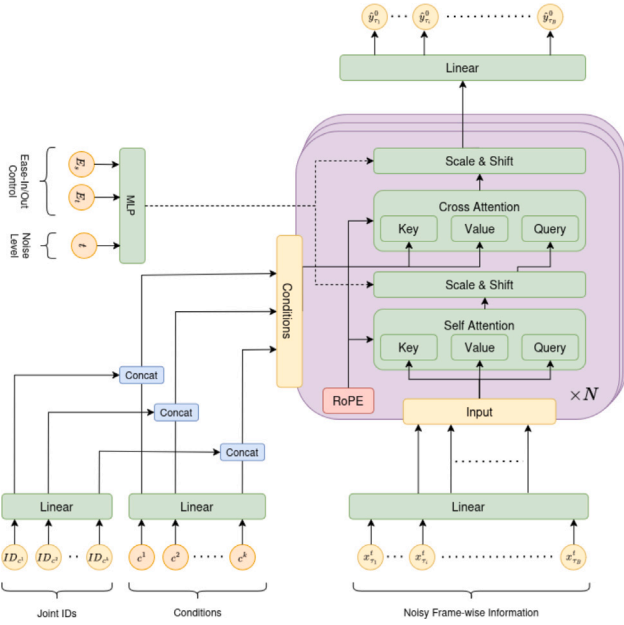


Fig. 3. Overview of the IBMM model architecture. The transformer encoder gets noisy frame-wise information as input. In addition, positional and ease-in/out conditions are provided for cross-attention to control the generation. Note that the notation c of a condition is an abbreviation for $p_{\tau_i,j}$. Note: The architecture presented here represents an optimized version of the framework originally proposed in [13]. We additionally skip illustrating Layer Norm and Feed Forward Network for simplicity.

added additional tokens in the sequence, are now used to modulate attention output in each transformer layer [43].

Secondly, using absolute sinusoidal positional encoding, as in the original version, limits generating motion beyond the sequence length for which the model was trained on as the new PE values are never seen by the model. Therefore, to increase the flexibility of our updated version, we adopt rotary positional encoding (RoPE) [14] that encodes each frame relative to its neighbors. This allows us to generate longer sequences than seen during training.

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.

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.

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 [5] 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 Fig. 4, we plot the ratio of the current distance traveled

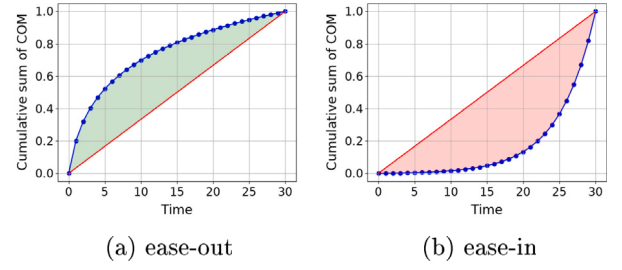


Fig. 4. Illustration of an ease-in/out concept. (a) shows ease-out of a motion with the COM slowing at the end of the sequence. (b) shows ease-in to a motion with a slow initial velocity and then accelerating.

by the COM to the total distance over time. For Fig. 4(a), 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 Fig. 4(b), 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.

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

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

where $\bar{p}_n$ is the mean of all J joints at frame n and $\|\cdot\|$ denotes the Euclidean norm. 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 $[x_{\tau_i}^t]_1^B$ 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. Training

In the original IBMM, the data was aligned based on the boundary frames, x_0 and x_N , in the following way. First, each segment was rotated around the vertical axis such that the ground projected direction from start to end pose aligned with the positive Y -axis. Second, the motion was shifted along the Y -axis such that the middle was located at the origin. Finally, a Z-score normalization was used to set mean to 0 and variance to 1 over the training data.

In this work, we now remove the requirement of always conditioning on boundary frames x_0 and x_N . Instead, we assume at least two constraints are given to the model. Using these two constraints, we align the motion such that the direction from the first constraint to the last is aligned with the global Y -axis. By randomizing the time index of the first and last constraints, our model is able to be extrapolate motion beyond the boundaries of the input constraints. The remaining alignment and normalization follows identically to the original work.

Furthermore, as artists frequently use FK constraints to orient the joints, we now enable adding orientations as additional constraints. As we operate on a simplified skeleton, specifying the global orientation of a joint without its position is an unintuitive control. Therefore, we instead assume that setting an orientation constraint, additionally constrains the position for that joint.

Loss functions. For training our model, we use the same loss configurations as in BMM. This entails MSE losses for the predicted 3D trajectories $[\hat{p}_i]_1^N$ and orientations $[\hat{r}_i]_1^N$ against ground truth values $[p_i]_1^N$ and $[r_i]_1^N$, respectively. There are further MSE losses between the positional and orientational conditions $[c_{\tau,j}]_1^B$ and the corresponding predicted positions $[\hat{p}_{\tau,j}]_1^B$, 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 [12].

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.

3.5. Inference methods

In our editing examples with IBMM, we also observed a few issues that are harder to grasp from a training objective, since they mostly occur when using the model in an iterative and practical use case. These challenges include: editing when many constraints are set (e.g. when editing or stitching given mocap clips), iteratively adding new constraints, and sequence stitching. However, IBMM++ can mitigate those issues with additional inference-based methods.

Attention masking. In scenarios with a large number of full body constraints, we observe a lower constraints accuracy when manipulating sparse handles. We attribute this to the global cross-attention between the input pose token and all of the condition tokens diluting the attention weight on the local condition token. Basically every new condition will contribute to the attention weighting, no matter how far away they are and if more relevant constraints are close by. Especially, for motion that is much more Markovian, the attention should focus the most on the nearest constraints, and adding new far away constraints should not influence the motion too much anymore. While RoPE already helps to focus the attention to the closest constraints, it cannot fully cut off attention to far away constraints.

In order to improve constraints manipulation in this scenario, we use a *top-k* mask over the cross-attention at inference. Hence, an input pose token only attends to the *k* most relevant condition tokens. As demonstrated in Section 4.4.3, we find *k* = 10 to give the best results. While this *top-k* masking could also be done during training, we have not observed that this mitigates the problem better than simply using it as an inference-only method. Note, the *top-k* masking can also be applied at the self-attention stage, however, with a different *k*, since those tokens describe a full pose frame and not an individual joint. For those layers, we choose *k* = 5.

Smooth noise modeling. Since IBMM allows adding new control points at arbitrary timings, it is important to account for this also in the way the noise is created. By inserting a new point into the middle of the motion, the noise vector gets longer and naively mapping this new noise vector back to the ordered input would shift the noise for each point following the added one. Thus, we precompute and cache the noise for every single frame and slice the noise according to the control points' temporal location.

Additionally, we observed that adding new control points — even on the already predicted curves — can sometimes lead to bigger changes. While such changes are explainable, since a new input is added to the network, we observed that this chance can be especially strong, when the added noise sample on this point is more extreme. To ensure a smoother editing behavior when adding new control points, we cache a piecewise linear version of the noise. So similarly to the stride of the Bézier curves, the noise is sampled in a temporally reduced space with linear interpolation in between. For the stride length of the noise sampling, we used the mean stride length of the Bézier curves during training. Note that this noise cache is completely independent of the output motion and only depends on the frame index. Hence, it

remains constant over iterative edits to ensure predictable user edits but the user maintains the freedom to resample the noise for any part of the motion.

While the smoothed noise might suggest that due to lower frequent input the results would lead to less overall diversity, empirically we observe the opposite. Especially for the extreme case of *s* = 1, i.e. a Bézier control point at every timestep, IBMM with *iid* noise shows little diversity and collapses to a mean motion. We attribute this effect to a domain gap to training, as IBMM is trained for a much sparser domain. We suspect that too high frequent noise pushes the motion into multiple opposite directions which are then averaged out, leading to mode collapse. While the smoothed noise is not *iid* as for training it still mimics the sparsity and thus is closer to the training setup.

Cross window blending. While the model can deal with temporal variable input sizes and can also extrapolate beyond the maximal window length, there is a limit to the window size. Therefore, the motion needs to be broken up into smaller segments. From a practical perspective, this not only makes the evaluation faster, but also mimics the way artist work. They break up animation into smaller subsequences or “beats”, generally between important keyframes. To improve continuity, IBMM averages the tangent prediction of the boundary frames for adjacent windows. To further strengthen the continuity, we add a predicted frame from the adjacent window as an additional constraint. Thus the prediction between the two windows is no longer independent but has an additional context about the dynamics leading into the start boundary (or out of the end boundary). In practice, we observed that adding a single pose at a distance of ~6 frames from the boundary frames communicated the momentum of the motion without hurting constraint accuracy close to the boundaries.

4. Evaluation

Given the need to generate artifact-free motions and our focus on artist-controllable motion generation, we measure foot sliding and handle accuracy to measure motion quality and generalization to artist inputs, respectively.

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.

To validate the improvements made in this work, we refer to the original as IBMM and our improved version as IBMM++ throughout the following validation.

4.1. Dataset and implementation details

We use the LaFAN1 dataset [15] to evaluate our model and compare it to BMM. It contains motion capture data recorded on five subjects and is approximately 4.6 h long. The motions include walking, running, climbing obstacles, jumping, dancing, and many more. The training data includes all clips from subjects one to four, while the validation data consists of all clips from subject five. Additionally, our internal dataset is a more action-themed dataset and consists of ~40 h of walking, running, parkour, and fighting sequences at 24 fps. We use this dataset for all evaluations involving IBMM++. We split the dataset into 95% training and 5% validation datasets.

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.

On the LaFAN1 dataset, we trained two model versions for 230 epochs (~64 h) 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.

For efficiency, on our internal dataset we only train the longer **90-frame model** with identical Bézier and constraint sampling setup as on LaFAN1.

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.

4.2. Quantitative evaluation

In the following section, we first compare IBMM to BMM [12] on varying time indices, handle accuracy, and foot sliding. Next, we validate our improvements from this work on handle accuracy and foot sliding.

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 :

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

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

As illustrated in Fig. 5(a), 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 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 Fig. 5(b), 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.

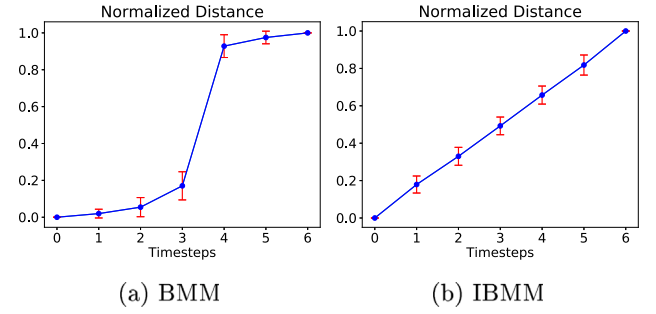


Fig. 5. Ratio of the distance between two control points for an additional control point that is temporally shifted between them. The results show the average and standard deviation over 1000 different samples. (a) BMM reflects a step-like function, indicating that time is mapped to discrete values. On the contrary, (b) IBMM has a continuous linear profile allowing for more detailed control over the time axis.

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 dim}	1.409
IBMM	0.389
IBMM	0.418 (Random)
IBMM ^{-token, -align}	1.326
IBMM ^{-align}	0.639
IBMM ^{-token}	0.690

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. (GT=Ground Truth).

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

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 dim}. IBMM^{-token, -align}, which uses BMM’s condition tokens and alignment, achieves handle accuracy similar to that of BMM^{+1024 dim}, indicating that fitting the Bézier curves implicitly has no negative effect on performance. Evaluating IBMM^{-align} and IBMM^{-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.

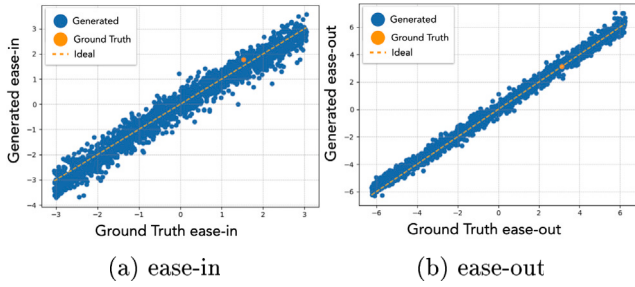


Fig. 6. Influence of the ease-in/out control for the generated motion samples. The samples indicate a linear relationship showing that IBMM has faithfully learned to follow the condition.

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). For more details on the metric, we refer to Appendix D of Vögeli et al. [13].

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. 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 Fig. 6, 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.

4.2.5. Comparison to Vögeli et al. [13]

In Table 3, we evaluate our architectural improvements to IBMM [13] on our internal dataset. Since we now evaluate on a more diverse dataset, we use Foot Sliding (FS) as a metric instead of FSRL, which was more specific to locomotion. FS is defined as sum of planar velocity of the feet joint when the foot is in contact ($p_z \leq 5$ cm) with the ground. For the positional and orientational accuracy, we use the same metric as in Table 1, using 6D representation for the rotations. Note that the ablated version $\text{IBMM}^{++}_{\text{CA}, -\text{RoPE}}$ represents the original IBMM but also trained to predict orientation constraints. First, we see that introducing cross-attention gives the large improvement over the base model for positional and orientational constraints accuracy, while introducing RoPE provides further gains. Another consequence of using cross-attention is that the sequence is now split into 2 shorter sequences of generated frames and conditions. As the computation time for attention is quadratic in number of tokens, this improves our execution time by $\sim 40\%$.

Table 3

Ablating importance of adding Cross-Attention and RoPE to IBMM when adding 10% random intermediate handles.

Model	Pos Acc. (cm)↓	Orient Acc. ↓	FS	Time
IBMM++	0.389	0.317	0.001	5:22
IBMM++-RoPE	0.399	0.323	0.087	5:10
IBMM++-CA	0.399	0.318	0.001	8:27
IBMM++-CA, -RoPE	0.410	0.333	0.003	8:39

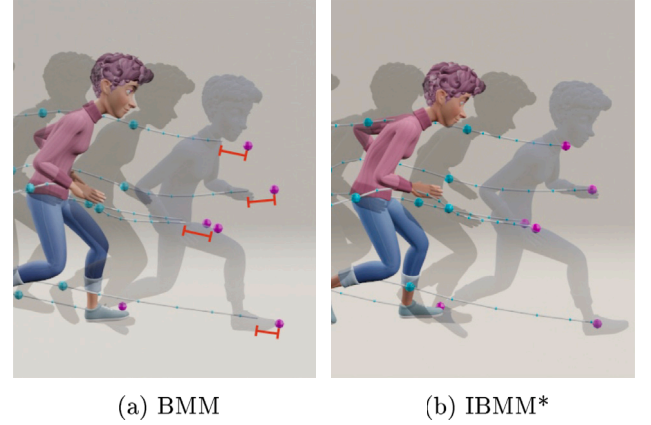


Fig. 7. Qualitative comparison of the handle accuracy between BMM and IBMM*. (a) shows the distance between the predictions and conditions (purple spheres) for BMM, while (b) shows that the predictions of IBMM* are much closer to the conditions.

4.3. Qualitative evaluation

We present evaluating our model, IBMM, on test clips from LaFAN1 and AMASS [44] datasets in our supporting video. In the following sections, we discuss the effects of our improvements on usability over [12]. In extension to Vögeli et al. [13], we further demonstrate various improvements made in IBMM++ on our internal dataset. We use IBMM* to denote results that are representative of both IBMM and IBMM++.

4.3.1. Better constraints hitting

Fig. 7 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 of Vögeli et al. [13].

4.3.2. Different amounts of evenly spaced Bézier poses

Fig. 8 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. Fig. 9 shows an example where dense conditions on

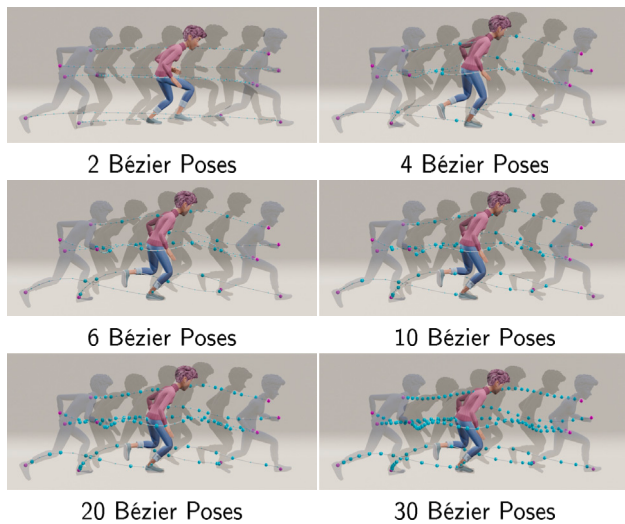


Fig. 8. Illustration of sampling different numbers of evenly spaced Bézier poses with IBMM*.



Fig. 9. 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.

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.

4.3.4. Ease-in/out control

We evaluate ease control for IBMM in Fig. 10. The blue and red silhouettes show every fourth frame at the start and end of the motion sequence, respectively. In Fig. 10(a), 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 Fig. 10(b), 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 Fig. 10(c)—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.

4.3.5. Orientation control

By adding orientation handles to IBMM++, one can achieve new control mechanism that go beyond positional constraint hitting. Fig. 11 demonstrates this for rotating the hip joint from an originally sitting pose along the yaw, pitch, and roll axes. Notice how rotating a single

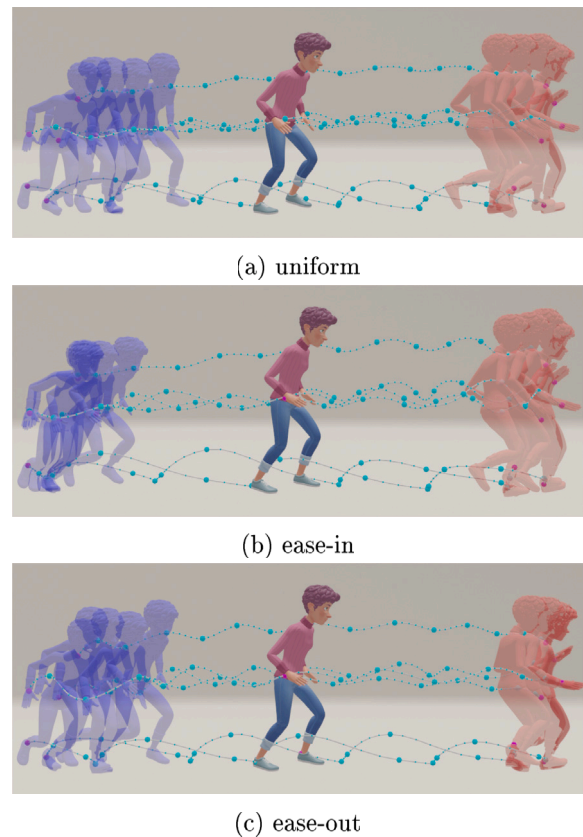


Fig. 10. Comparison of different values for E_i and E_o : (a) Uniform motion with $E_i = 0$ and $E_o = 0$, (b) Ease-in motion with negative E_i , (c) Ease-out motion with positive E_o .

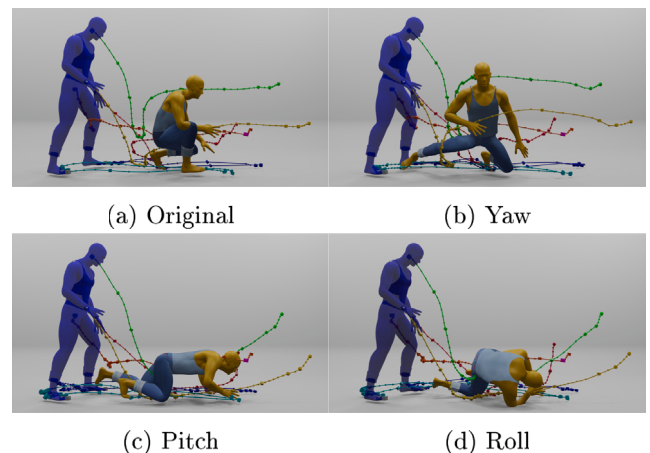


Fig. 11. Example of rotating a sparse hip joint along yaw, pitch, and roll axes.

constraint effects the full pose and changes the motion from sitting to crawling when changing the pitch, and to falling for roll.

Lastly, while IBMM++ works on a reduced skeleton, control over missing joints is very limited. However, the orientation control can help to improve this issue. For instance, by rotating the hand's orientation, the elbow can be affected much more directly, for overall better pose control.

4.3.6. Motion extension

In Fig. 12, we show results of generating motion beyond the last keyframes with different noise seeds but fixed full body constraints. The

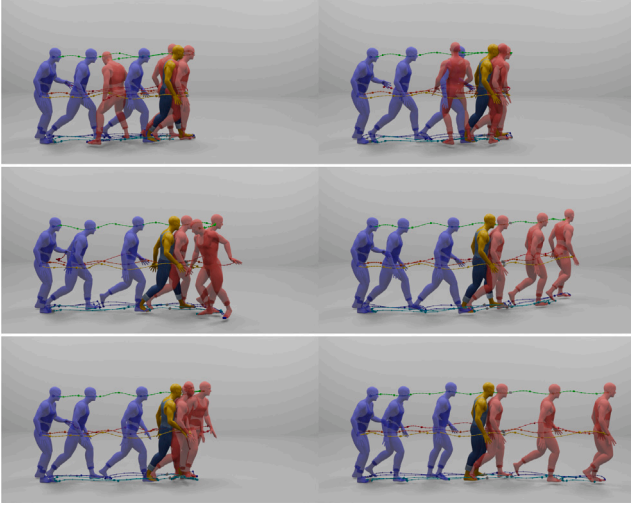


Fig. 12. Different extensions from the same full body constraints. The blue silhouettes are inbetween while the red frames extend beyond the final full body keyframe.

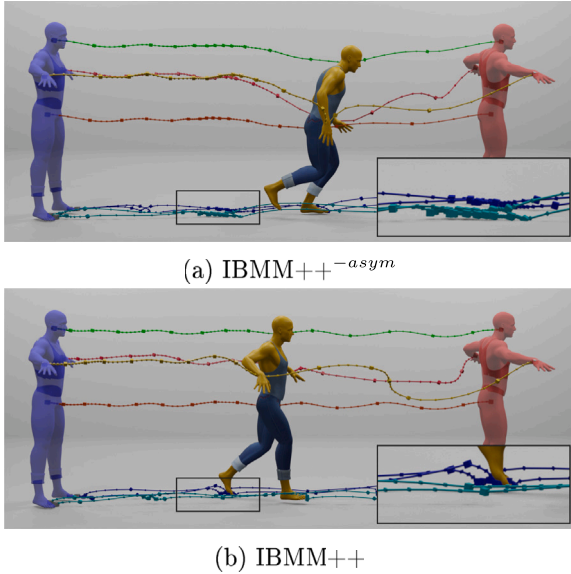


Fig. 13. Effect of adding adaptive tangent scaling when inbetweening between two T-pose full body constraints. Without the scaling, the generated motion shows foot sliding in (a), which is fixed with scaling (b). The insets magnify the first step: without scaling, the foot control points are strung out in a line along the ground, whereas with scaling they lift into an arc.

blue silhouettes temporally lie between the two full body constraints, while the red ones are after the last given keyframe. Note that while blue frames are similar across the different variants, the red frames show a large variability with some staying in place, continuing forward, or turning back.

4.4. Ablations

In this section we ablate our representation (Section 3.1) and inference time (Section 3.5) contributions compared to the original IBMM [13].

4.4.1. Adaptive tangent scaling ablation

We introduced unsymmetrical tangents and adaptive tangent scaling to improve the foot sliding in the generated motion. To demonstrate

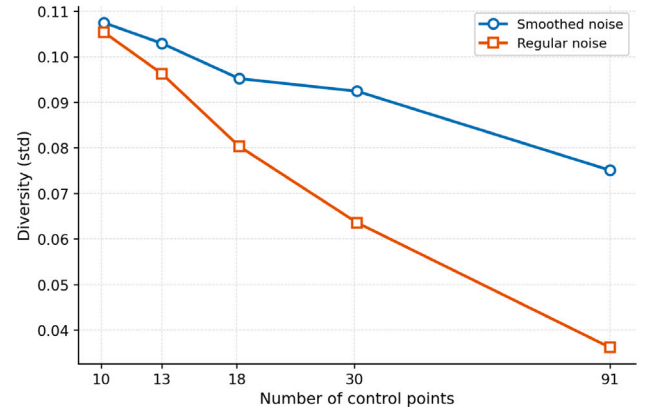


Fig. 14. Plotting diversity (std) in end-effector positions for different number of control points (B) with and without using smooth noise modeling.

the effect of these additions, in Fig. 13 we show an example of the generated motion between two T-pose constraints. Note that T-pose is an out of distribution pose and giving the identical start and end poses further push the sample away from the training data.

Without the tangent scaling, we see a large amount of foot sliding with many control points predicted right beside each other. In contrast, adding tangent scaling results in clean foot steps being generated for the same noise seed.

4.4.2. Smoothed noise modeling

In Fig. 14, we ablate smooth noise modeling by plotting the diversity in the generated samples for different number of Bézier control points (B). We measure diversity as the average standard deviation in the end-effector positions when sampling 128 variants for 320 sequences each.

At smaller B , both noise strategies perform similarly. However, as the model was not trained with very high B , we see that without smooth noise modeling, the diversity drops drastically. In comparison, with interpolated noise (with 9 piecewise sections), we see a much smaller decrease.

We further demonstrate the higher diversity of smooth noise modeling in our supporting video. When generating variants for a fixed inbetweening constraints set with $B = N$, we see that without smooth noise modeling all variants are very similar and exhibit substantial jitter artifacts. In comparison, using smooth noise modeling gives large variability and generates smooth motion.

4.4.3. Top-k attention

In Fig. 15, we edit a motion capture by deleting a few frames and regenerating while the remaining frame are used as full-body constraints. When using global attention, as in Vögeli et al. [13], the unconstrained input tokens more strongly attend to the many full-body conditions instead of the sparse hip constraint (in pink in Fig. 15(a)). Consequently, we get a large constraint satisfaction and the neighboring control points do not adapt appropriately.

In contrast, as shown in Fig. 15(b), attending only to top-k condition tokens fixes this issue as the unconstrained tokens can now more strongly attend to the sparse constraint.

Additionally, in Fig. 16, we measure the performance of different values of k at satisfying sparse positional constraints across different keyframe densities. We use positional constraint accuracy across the validation set of our internal dataset for the evaluation. Note that since the model was originally trained to receive only two full keyframes, evaluating with 60 – 80 keyframes is very far from its training distribution. Therefore, to measure the comparative performance at each keyframe density, we subtract the mean error across different values of k .

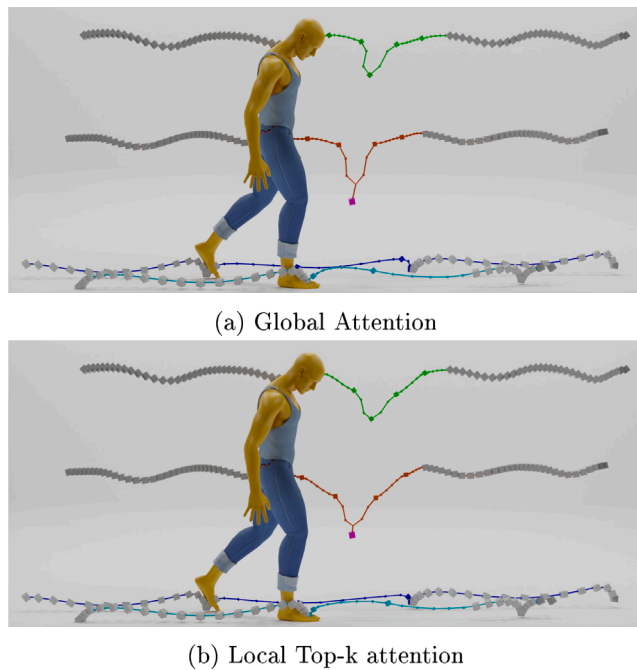


Fig. 15. Editing motion capture by deleting frames with and without localized top-k attention.

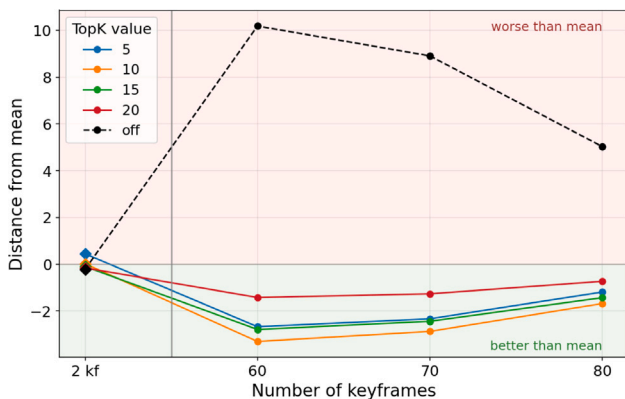
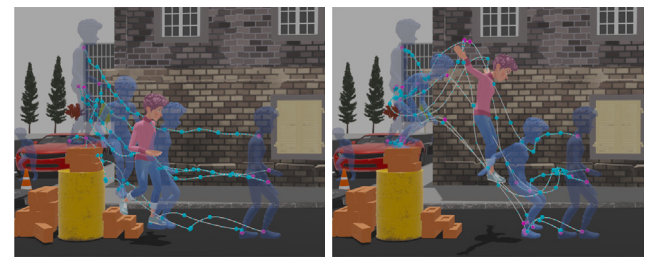


Fig. 16. Effect of varying the top-k value on positional constraint accuracy across different numbers of full-body keyframes.

First, we see that when attending to all input keyframes (“off” in Fig. 16), the model performs the best with only two keyframe constraints but drastically drops off in performance at higher keyframe densities. This corroborates the empirical evidence from Fig. 15(a). At higher keyframe densities, $k = 10$ consistently performs the best. Hence, we recommend it as a safe default value when working with dense data such as motion capture. Lastly, it is worth noting that varying k is an inference-time technique. Therefore, it can be easily exposed to the artist to select the best value for the current sequence.

4.5. 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 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.



(a) without edits (b) with edits

Fig. 17. Example of editing an animation with IBMM. (a) shows the model’s initial prediction and (b) the final edited animation.

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 Fig. 17(a). 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.

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.

5. Discussion

Our work builds on a new insight in motion diffusion [10–12], that sparsifying the input signal improves motion generation, and in the case of BMM [12] 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.

With IBMM, we removed this constraint by allowing 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.

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.

With IBMM, and now IBMM++ in this work, we further showed that an improved architecture and alignment of the data has large effects on motion fidelity and improves spatial control. Specifically, IBMM improved the condition token formulation that we inherit in this work, and we introduced adaptive tangent scaling for better foot-ground contact. However, while scaling the tangents helps reduce foot sliding, it does not completely eliminate it unless the two Bézier points coincide. Therefore, further research incorporating foot-contact modeling into the architecture is required to fully remove this artifact.

Using a relative positional encoding in RoPE [14] allowed us to generate sequences up to 25% longer than the ones the model was originally trained on. However, we must note that when generating sequences much longer ($\geq 1.5\times$) than the training sequence length, the model exhibits ringing artifacts where the additional frames oscillate between the two full body constraints. We attribute this to the generated position values growing much larger than seen during training. Using a redundant motion representation, such as additionally generating velocities, can potentially help improve this limitation, which we leave as future work.

When training BMM, Studer et al. [12] 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.

By relaxing the requirement of always providing both boundary full body keyframes, we unlocked a new workflow where artists can quickly auto-regressively layout motion sequences. By feeding generated frames as starting full body constraints at the next evaluated window, an artist can generate the future motion. This generated future motion can further be controlled via sparse control points.

As both IBMM and IBMM++ 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.

6. Conclusion

In Vögeli et al. [13], we introduced the Implicit Bézier Motion Model (IBMM). 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.

In this extended version (IBMM++), we further closed the gap to professional editing workflows. An improved transformer backbone with cross-attention and rotary positional encoding raises constraint accuracy and lets the model generate motion beyond the sequence length seen during training. An asymmetric, distance-scaled Bézier tangent formulation reduces foot sliding and enables better foot manipulation. A randomized constraint sampling strategy at training time removes the requirement of fixed boundary frames and adds support for orientation constraints and motion extension. Additionally, we introduce several inference-time mechanisms that enable IBMM++ to scale to dense constraint sets, offer stable iterative editing, and ensures smoothness across motion segments.

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.

CRedit authorship contribution statement

Dhruv Agrawal: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology. **Luca Vögeli:** Visualization, Validation, Methodology. **Dominik Borer:** Validation. **Robert W. Sumner:** Supervision, Funding acquisition. **Martin Guay:** Supervision. **Jakob Buhmann:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.cag.2026.104751>.

Data availability

The authors do not have permission to share data.

References

- [1] Oreshkin BN, Bocquet F, Harvey FG, Raitt B, Laflamme D. ProtoRes: Proto-residual network for pose authoring via learned inverse kinematics. In: International conference on learning representations. 2022, URL <https://openreview.net/forum?id=s03AQxehtd>.
- [2] Agrawal D, Guay M, Buhmann J, Borer D, W. Sumner R. Pose and skeleton-aware neural IK for pose and motion editing. In: SIGGRAPH Asia 2023 conference papers. 2023, p. 1–10.
- [3] Cohan S, Tevet G, Reda D, Peng XB, van de Panne M. Flexible motion in-betweening with diffusion models. In: ACM SIGGRAPH 2024 conference papers. 2024, p. 1–9.
- [4] Guo C, Mu Y, Javed MG, Wang S, Cheng L. Momask: Generative masked modeling of 3d human motions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2024, p. 1900–10.
- [5] Meng Z, Xie Y, Peng X, Han Z, Jiang H. Rethinking diffusion for text-driven human motion generation: Redundant representations, evaluation, and masked autoregression. In: Proceedings of the computer vision and pattern recognition conference. 2025, p. 27859–71.
- [6] Zhang Z, Wang Y, Mao W, Li D, Zhao R, Wu B, Song Z, Zhuang B, Reid I, Hartley R. Motion anything: Any to motion generation. 2025, arXiv preprint [arXiv:2503.06955](https://arxiv.org/abs/2503.06955).

- [7] Mu Y, Ling HY, Shi Y, Ojeda IB, Xi P, Shu C, Zinno F, Peng XB. StableMotion: Training motion cleanup models with unpaired corrupted data. In: Proceedings of the SIGGRAPH Asia 2025 conference papers. SA conference papers '25, New York, NY, USA: Association for Computing Machinery; 2025, <http://dx.doi.org/10.1145/3757377.3763817>.
- [8] Li J, Cao J, Zhang H, Rempe D, Kautz J, Iqbal U, Yuan Y. GENMO: A generalist model for human motion. In: 2025 IEEE/CVF international conference on computer vision. 2025, p. 11766–76. <http://dx.doi.org/10.1109/ICCV51701.2025.01094>.
- [9] Goel P, Zhang H, Liu CK, Fatahalian K. Generative motion infilling from imprecisely timed keyframes. In: Computer graphics forum. Wiley Online Library; 2025, e70060.
- [10] Bae J, Hwang I, Lee YY, Guo Z, Liu J, Ben-Shabat Y, Kim YM, Kapadia M. Less is more: Improving motion diffusion models with sparse keyframes. 2025, arXiv preprint [arXiv:2503.13859](https://arxiv.org/abs/2503.13859).
- [11] Hwang I, Bae J, Lim D, Kim YM. Motion synthesis with sparse and flexible keyjoint control. 2025, arXiv preprint [arXiv:2503.15557](https://arxiv.org/abs/2503.15557).
- [12] Studer J, Agrawal D, Borer D, Sadat S, Sumner RW, Guay M, Buhmann J. Factorized motion diffusion for precise and character-agnostic motion inbetweening. In: Proceedings of the 17th ACM SIGGRAPH conference on motion, interaction, and games. 2024, p. 1–10.
- [13] Vögeli L, Agrawal D, Guay M, Borer D, Sumner RW, Buhmann J. Implicit Bézier motion model for precise spatial and temporal control. In: Proceedings of the 2025 18th ACM SIGGRAPH conference on motion, interaction, and games. Zurich Switzerland: ACM; 2025, p. 1–10. <http://dx.doi.org/10.1145/3769047.3769052>, [Accessed 2026-01-08].
- [14] Su J, Ahmed M, Lu Y, Pan S, Bo W, Liu Y. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing* 2024;568:127063. <http://dx.doi.org/10.1016/j.neucom.2023.127063>.
- [15] Harvey FG, Yurick M, Nowrouzezahrai D, Pal C. Robust motion in-betweening. *ACM Trans Graph* 2020;39(4). 60–1.
- [16] Autodesk Inc. Autodesk maya - a 3D modeling, animation, and rendering software. 2025, URL <http://www.autodesk.com/maya>, Version 2025, Autodesk, Inc., San Rafael, CA.
- [17] Blender Online Community. Blender - a 3D modelling and rendering package. 2025, URL <http://www.blender.org>, Blender Foundation, Stichting Blender Foundation, Amsterdam.
- [18] Ciccone L, Öztireli C, Sumner RW. Tangent-space optimization for interactive animation control. *ACM Trans Graph* 2019;38(4):1–10.
- [19] Tang X, Wang H, Hu B, Gong X, Yi R, Kou Q, Jin X. Real-time controllable motion transition for characters. *ACM Trans Graph* 2022;41(4):1–10.
- [20] Qin J, Zheng Y, Zhou K. Motion in-betweening via two-stage transformers. *ACM Trans Graph* 2022;41(6). 184–1.
- [21] Akhouni E, Ling HY, Deshmukh AA, Bütepage J. SILK: Smooth Interpolation framework for motion in-betweening. In: Proceedings of the computer vision and pattern recognition conference. 2025, p. 2900–9.
- [22] Agrawal D, Buhmann J, Borer D, Sumner RW, Guay M. SKEL-betweener: a neural motion rig for interactive motion authoring. *ACM Trans Graph* 2024;43(6):1–11.
- [23] Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models. *Adv Neural Inf Process Syst* 2020;33:6840–51.
- [24] Shi J, Liu L, Sun Y, Zhang Z, Zhou J, Nie Q. GenM³: Generative pretrained multi-path motion model for text conditional human motion generation. 2025, arXiv preprint [arXiv:2503.14919](https://arxiv.org/abs/2503.14919).
- [25] Tevet G, Raab S, Gordon B, Shafir Y, Cohen-or D, Bermano AH. Human motion diffusion model. In: The eleventh international conference on learning representations. 2023, URL <https://openreview.net/forum?id=SJ1kSyO2jwu>.
- [26] Tseng J, Castellon R, Liu K. Edge: Editable dance generation from music. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023, p. 448–58.
- [27] Dabral R, Mughal MH, Golyanik V, Theobalt C. Mofusion: A framework for denoising-diffusion-based motion synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023, p. 9760–70.
- [28] Zhang M, Guo X, Pan L, Cai Z, Hong F, Li H, Yang L, Liu Z. Remodiffuse: Retrieval-augmented motion diffusion model. In: Proceedings of the IEEE/CVF international conference on computer vision. 2023, p. 364–73.
- [29] Liao TH, Zhou Y, Shen Y, Huang CHP, Mitra S, Huang JB, Bhattacharya U. Shape my moves: Text-driven shape-aware synthesis of human motions. In: Proceedings of the computer vision and pattern recognition conference. 2025, p. 1917–28.
- [30] Loper M, Mahmood N, Romero J, Pons-Moll G, Black MJ. SMPL: A skinned multi-person linear model. *ACM Trans Graph (Proc. SIGGRAPH Asia)* 2015;34(6):248:1–248:16.
- [31] Zhang L, Rao A, Agrawala M. Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF international conference on computer vision. 2023, p. 3836–47.
- [32] Xie Y, Jampani V, Zhong L, Sun D, Jiang H. OmniControl: Control any joint at any time for human motion generation. In: The twelfth international conference on learning representations. 2024.
- [33] Karunratanakul K, Preechakul K, Aksan E, Beeler T, Suwajanakorn S, Tang S. Optimizing diffusion noise can serve as universal motion priors. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2024, p. 1334–45.
- [34] Wu L, Yu J, Jin L, Wang H, Zheng B, Yang X, Jiang H, Xia F, Ling F, Deng J, et al. Unifying precise keyframes and semantic control via multi-level diffusion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2026, p. 23473–83.
- [35] Zhao K, Li G, Tang S. DartControl: A diffusion-based autoregressive motion model for real-time text-driven motion control. In: International conference on learning representations. 2025.
- [36] Rempe D, Petrovich M, Yuan Y, Zhang H, Peng XB, Jiang Y, Wang T, Iqbal U, Minor D, de Ruyter M, Li J, Tessler C, Lim E, Jeong E, Wu S, Hassani E, Huang M, Yu J-B, Chung C, Song L, Dionne O, Kautz J, Yuen S, Fidler S. Kimodo: Scaling controllable human motion generation. 2026, arXiv preprint [arXiv:2603.15546](https://arxiv.org/abs/2603.15546).
- [37] Zheng B, Chen K, Yao Y, Zeng Z, Jiang X, Wang H, Lasenby J, Jin X. AutoKeyframe: Autoregressive keyframe generation for human motion synthesis and editing. In: ACM SIGGRAPH 2025 conference papers. 2025, <http://dx.doi.org/10.1145/3721238.3730664>.
- [38] Yun K, Hong S, Kim C, Noh J. AnyMoLe: Any character motion in-betweening leveraging video diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2025.
- [39] Huang J, Wang Q, Fratarcangeli M, Yan K, Pelachaud C. Multi-variate gaussian-based inverse kinematics. In: Computer graphics forum, vol. 36, (8):Wiley Online Library; 2017, p. 418–28.
- [40] Ponton JL, Yun H, Aristidou A, Andujar C, Pelechano N. Sparseposer: Real-time full-body motion reconstruction from sparse data. *ACM Trans Graph* 2023;43(1):1–14.
- [41] Ponton JL, Pujol E, Aristidou A, Andujar C, Pelechano N. Dragposer: Motion reconstruction from variable sparse tracking signals via latent space optimization. In: Computer graphics forum, vol. 44, (2):Wiley Online Library; 2025, e70026.
- [42] Zhou Y, Barnes C, Lu J, Yang J, Li H. On the continuity of rotation representations in neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019, p. 5745–53.
- [43] Peebles W, Xie S. Scalable diffusion models with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. IEEE; 2023, p. 4172–82. <http://dx.doi.org/10.1109/iccv51070.2023.00387>.
- [44] Mahmood N, Ghorbani N, Troje NF, Pons-Moll G, Black MJ. AMASS: Archive of motion capture as surface shapes. In: International conference on computer vision. 2019, p. 5442–51.