Factorized Motion Diffusion for Precise and Character-Agnostic Motion Inbetweening

Justin Studer*
DisneyResearch|Studios
Switzerland
ETH Zürich
Switzerland
jstuder.research@gmail.com

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

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

Seyedmorteza Sadat
DisneyResearch|Studios
Switzerland
ETH Zürich
Switzerland
seyedmorteza.sadat@inf.ethz.ch

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

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

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

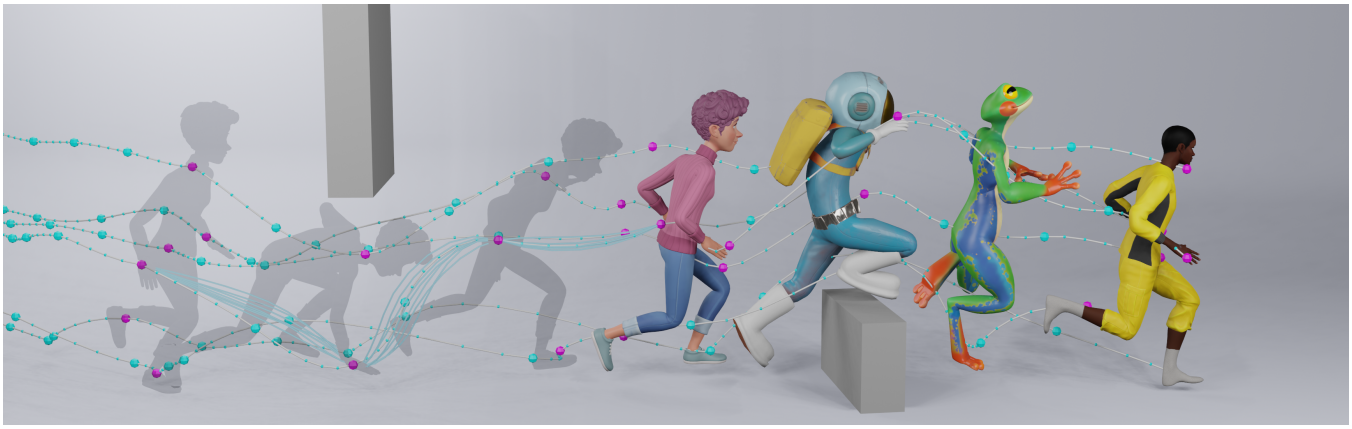


Figure 1: Demonstration of BMM, a factorized approach for motion generation that can be accurately controlled with intermediate joint-level constraints (purple). A generative model creates character-agnostic Bézier motion curves that can be applied to various humanoid characters via a separate IK-module. The generative process of BMM accurately satisfies constraints while generating realistic and varied motions, as shown by the cyan curves samples of the right hand.

Abstract

Animation is a challenging and time-consuming process where animators must manipulate hundreds of controls over space and time to create compelling motions. Recent advances in motion diffusion models have shown impressive results for general motion generation and hold the potential to reduce the number of controls manipulated by animators to achieve high quality results. However, these models are limited by their inability to match sparse constraints precisely, preventing frame-level joint control required by artists. Additionally, recent models are trained for specific characters, preventing reuse, and are incompatible for characters with only

*Work done during an internship at DisneyResearch|Studios.

MIG '24, November 21–23, 2024, Arlington, VA, USA

ACM ISBN 979-8-4007-1090-2/24/11
https://doi.org/10.1145/3677388.3696338

a small datasets available. To tackle these shortcomings, we propose a novel factorization of motion between a character-agnostic Bézier Motion Model (BMM), which can be trained on a large motion dataset, followed by a character-specific posing model, trainable on a much smaller pose dataset, that enables reuse across many characters. BMM provides accuracy for meeting sparse joint-level constraints by working in a reduced space of Bézier curves that better aligns the condition signal with the prediction space of our model. Additionally, the Bézier curves offer animators an intuitive interface compatible with existing authoring software. Through quantitative and qualitative comparisons, we show the effectiveness of our factorization and parametric subspace, enabling user control with higher fidelity.

CCS Concepts

• **Computing methodologies** → **Machine learning**; **Motion processing**.

Keywords

Motion Generation, Motion Diffusion, Bézier Curves, Character Animation

ACM Reference Format:

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 *The 16th ACM SIGGRAPH Conference on Motion, Interaction, and Games (MIG '24), November 21–23, 2024, Arlington, VA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3677388.3696338

1 Introduction

Creating animations using traditional keyframing is widely known to be a labor-intensive and challenging task. While both optimization- and learning-based inverse kinematics (*IK*) methods [Agrawal et al. 2023; Aristidou and Lasenby 2011; Oreshkin et al. 2022; Voleti et al. 2022] have established themselves as facilitators in the authoring toolkit, they do not offer control over the entire motion. On the other hand, diffusion models [Ho et al. 2020; Sohl-Dickstein et al. 2015] show promise for high-quality motion generation. However, they suffer from poor controllability when provided with sparse spatial constraints as conditions. Recent methods [Cohan et al. 2024; Karunratanakul et al. 2024, 2023; Xie et al. 2023] offer limited sparse constraints or employ expensive optimization schemes to address inaccuracy. However, to apply motion generation methods in practice, an artist would require even sparser control, effectively influencing a joint positioning at a specific frame in real-time. Additionally, diffusion models require significant amounts of data and are trained for one particular topology. Consequently, characters with different topologies—or which hold less data—cannot be directly animated by large motion models.

When leveraging recent diffusion models for devising an animation workflow, we observe how these models struggle to match sparse spatial constraints accurately. This use case is particularly challenging for diffusion models due to the sparseness of the condition and the discrepancy between the condition and prediction space. While conditions are typically provided in 3D world space, the individual poses forming the motion are represented with local orientations. The hierarchical nature of forward kinematics (FK) causes errors to accumulate along the kinematic chain; a problem that is further amplified by the presence of noise during inference of the diffusion models, resulting in poor conformity to conditions.

In this paper, we propose a diffusion-based motion generation method in which the generation process is factorized into two distinct steps. In the first step, our Bézier Motion Model (BMM) is trained to generate Bézier curves for a characteristic subset of joints applicable to most humanoid hierarchies. In the second step, an inverse kinematics (IK) system turns the Bézier curves into character-specific poses. Our motion parameterization uses 3D Bézier curves for a subset of the joints, removing the forward kinematics in the generative process, and thus directly aligns the condition with the prediction space. Additionally, this compresses the representation along the spatial (joints) and temporal axis, which addresses the sparseness of the condition signal and thus increases its impact on the diffusion process. This enables BMM to consistently satisfy joint-level constraints better than existing methods. Moreover, our factorized configuration allows the generated trajectories to be used on different skeletons, and with many different IK methods, such as traditional IK, learned IK [Agrawal et al. 2023; Oreshkin et al. 2022], as well as learned motion models [Cohan et al. 2024; Qin et al. 2022]—as long as such a system exists for the respective topology. Our compressed representation allows us to employ a smaller and faster diffusion model, and we demonstrate a workflow where an artist can use familiar Bézier controls to manipulate full body motion with precise joint-level control at interactive rates. Besides the algorithmic advantages of Bézier curves, they also facilitate integration into existing authoring software such as Maya [Autodesk Inc. 2023] and Blender [Blender Online Community 2023] thanks to the widespread availability of their data structure. And lastly, artists can directly refine motions locally post-inference, without requiring any conversion across rigs.

To evaluate our approach, we show with qualitative and quantitative comparisons to baseline methods that BMM can hit constraints more accurately for an animation task of inbetweening with sparse joint-level constraints. Additionally, we demonstrate the compatibility of BMM with multiple IK-modules that can drive characters with different skeleton hierarchies and proportions. Lastly, to showcase the interactive usage of our model, we provide in our supplementary video examples of how movements can be interactively authored with BMM and its animator-friendly Bézier curves.

2 Related Work

In its current state, manual motion authoring using software such as Maya [Autodesk Inc. 2023] and Blender [Blender Online Community 2023] involve manipulating rig parameters for setting full pose keyframes, followed by some basic interpolation. This repetitive process typically results in having to specify a number of poses for a high-quality animation. Learned Inverse Kinematics models such as ProtoRes [Oreshkin et al. 2022] can recover a complete pose from a sparse set of rig-like set of constraints. While such models can accelerate posing, authoring an animation includes additional complexity due to the coordination of poses over time.

2.1 Neural Motion Completion

[Harvey et al. 2020] were amongst the early works that explored using a recurrent network to generate motion from a starting context and a target frame. [Oreshkin et al. 2023] extended their work to use a transformer and work in a delta space of linear interpolation to generate higher-quality motion. The method proposed by [Qin et al. 2022] generates motion in two stages. First, a transformer creates a coarse, low-frequency motion that can reach the target. This is then refined by a second transformer that adds details. While these works can generate realistic motions, their need for dense context and fully specified intermediate or target keyframes make them unsuitable for an interactive motion authoring system. Furthermore, when trained on larger datasets, these deterministic models suffer from averaging artifacts and struggle to utilize the additional data available.

Diffusion Models [Ho et al. 2020] have recently received significant attention due to the quality of the results when trained on very large datasets in several domains [Brooks et al. 2024; Esser et al. 2024; OpenAI 2023]. Human Motion Diffusion Model (MDM) [Tevet et al. 2022] demonstrated the ability to generate realistic movements from natural language. By inpainting desired values, their method can further be used for inbetweening of dense keyframes. However, their method does not produce satisfying results when conditioned on sparse individual joints. Guided Motion Diffusion [Karunratanakul et al. 2023] introduced a method that first generates a dense spatial trajectory on the root joint from sparse constraints, which is used as input to a secondary motion generation system.

OmniControl [Xie et al. 2023] enables control over arbitrary joints by training a ControlNet-like [Zhang et al. 2023a] control network on a pre-trained motion generation system and using optimization to enforce constraint accuracy. Diffusion Noise Optimization [Karunratanakul et al. 2024] backpropagates desired conditions through the entire diffusion chain at inference time to optimize the initial noise vector such that the output obeys the conditions without retraining. While this can generate plausible motion, backpropagating the entire diffusion chain is prohibitively slow for an interactive application. [Raab et al. 2024] allow motion transfer for tasks such as spatial editing and style transfer via mixed attention between a lead and a follower motion clips.

CondMDI [Cohan et al. 2024] overcomes the need for optimization by conditioning the model on dense spatial constraints such as keyframes or joint trajectories at training time. In our paper, we introduce a new factorization of the motion space that, compared to these recent diffusion models, demonstrates better accuracy to sparse user controls, such as moving a single joint at a specific frame.

2.2 Motion Transfer and Retargeting

The motion generation models mentioned above cannot be easily applied to different characters due to differences in bone proportions and skeleton hierarchies. An additional motion retargeting step is required to transfer the motion onto another character. Previous works such as [Lee and Shin 1999; Seol et al. 2013] have explored transferring motion via optimization. [Shin et al. 2001] modified end-effector trajectories based on the difference in character sizes and solved for full pose using inverse kinematics.

Neural retargeting has also been explored in recent times. [Aberman et al. 2020] used skeletal convolutions to reduce a source skeleton to a primal skeleton and deconvolve to the target skeleton. [Lee et al. 2023] also use a skeletal encoder-decoder network to learn skeleton agnostic embeddings to transfer motion to the target skeleton. However, both these models require paired motion data between the different skeletal hierarchies. Generating such data is very expensive, requiring manual clean-up of the retargeted motion.

In contrast, our method directly generates motions for the specified character using a general purpose Bézier motion model, trained on a large motion dataset, followed by a character-specified Inverse Kinematics model, trained on a smaller character-specific pose dataset. Thus, eliminating the need for post hoc retargeting.

3 Background

In this section, we first review the diffusion framework and subsequently outline Bézier curves.

3.1 Diffusion Models

Given a data point x_0 sampled from the data distribution $q(x_0)$, diffusion models define a forward process q that gradually adds noise to x_0 until it becomes indistinguishable from pure noise. Specifically, the forward diffusion process is described by

$$q(x_1, \ldots, x_T \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1}), \tag{1}$$

where $q(x_t \mid x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \boldsymbol{I})$ is the transition kernel from x_{t-1} to x_t. Here, $\beta_t \in (0, 1)$ is a noise schedule that determines how much information is destroyed at each time step t. If T is large enough, X_T converges to a sample from the standard Gaussian distribution $\mathcal{N}(0, \boldsymbol{I})$. Additionally, if β_t is small enough, the reverse or denoising distribution $q(x_{t-1} \mid x_t)$ can be approximated with a neural network $p_\theta(x_{t-1} \mid x_t)$ given by

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \tag{2}$$

When $p_\theta(x_{t-1} \mid x_t)$ approximates $q(x_{t-1} \mid x_t)$ sufficiently well, we can sample a new data point $x_0 \sim q(x_0)$ by starting from pure noise $x_T \sim \mathcal{N}(0, \boldsymbol{I})$ and iteratively applying p_θ until a clean sample is produced. By conditioning the denoising process $p_\theta(x_{t-1} \mid x_t)$ on additional inputs c, such as text prompts or spatial constraints, we can also perform conditional generation, i.e. sampling from $q(x_0 \mid c)$.

3.2 Bézier Curves

Bézier curves are parametric functions that can be used for smooth interpolation between two points, or for approximation of continuous functions. A Bézier curve of degree n is defined by

$$\boldsymbol{B}(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i \boldsymbol{P}_i, \qquad 0 \le t \le 1, \tag{3}$$

where $\{\boldsymbol{P_i}\}_{i=0}^{n}$ is a set of control points. In particular, a Bézier curve of degree n is defined by $n+1$ control points. For our purposes, we make use of cubic Bézier curves ($n = 3$), which means that each curve segment is defined by four points, or equivalently, a start and an endpoint, and a tangent at each one of them.

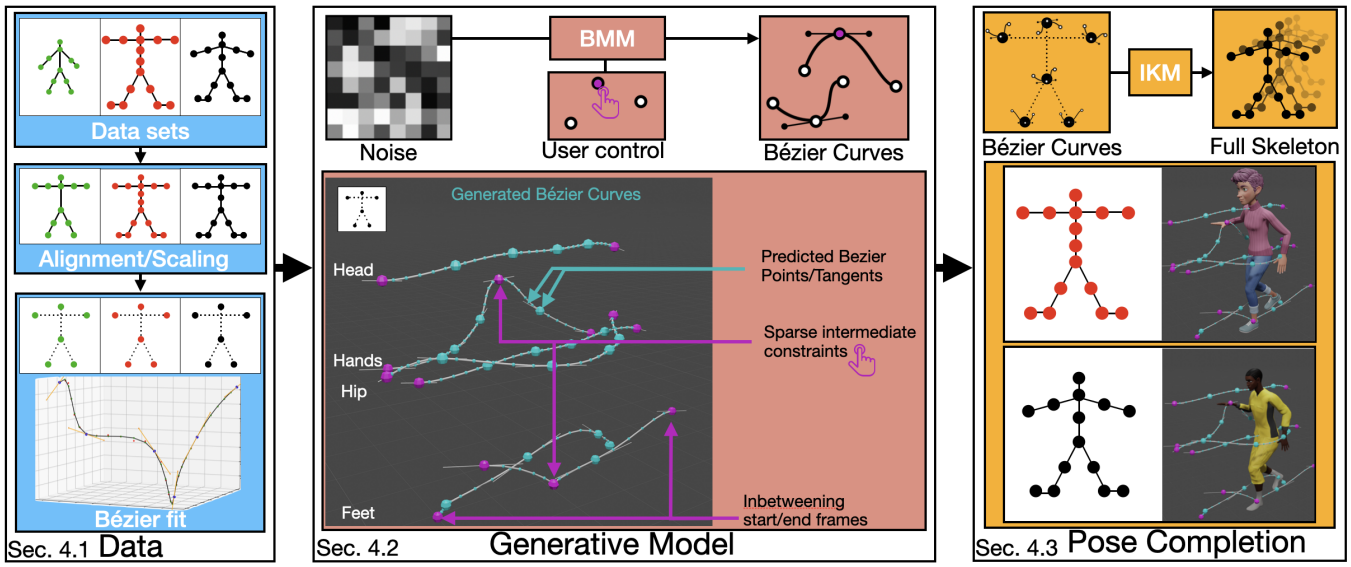


Figure 2: Schematic overview of our approach that can be divided into three steps: data preparation, a generative model, and a pose completion model. In the data preparation step, different humanoid skeletons are broken down to their main characteristic joints and represented by Bézier curves in 3D. Next, our Bézier Motion Model (BMM) is trained to generate these Bézier curves for the use case of inbetweening with sparse joint-wise constraints (purple). Finally, a skeleton-specific IK-module is used to map the generated Bézier curves to full skeletal motion.

4 Method

Measuring the positions and orientations of all of the joints at each frame is a highly redundant representation of human motion. We hypothesize that one can find a more compressed representation of human motion akin to using parabolic equations for a free-falling object instead of recording dense positions.

In this section, we detail our proposed trajectory-based motion generation method that compresses motion representation in time using Bézier curves and in space by leveraging Inverse Kinematics on a characteristic subset of joints. In the first stage, a diffusion model is used to create dense trajectories, parameterized as Bézier curves that can be conditioned via sparse joint-level position targets. In the second step, an independent Inverse Kinematics system is used to retrieve the poses of the final motion for a specific character.

4.1 Data Representation

Recent learned Inverse Kinematics models such as [Oreshkin et al. 2022] and [Agrawal et al. 2023] have shown that a pose can be reconstructed from only a small subset of known joints. Hence, we can leverage such models to compress our motion representation. To make our approach applicable to a wide set of humanoid skeletons, we restrict ourselves to the trajectories of only six joints J_{sub}, namely the hips, the hands, the feet, and the head, see Figure 2. As shown in the supplementary video and Appendix A.1, J_{sub} is sufficient to reconstruct motions with high accuracy and offers a good trade-off between staying skeleton-agnostic and achieving good reconstruction. While more joints improve reconstruction accuracy, they also make the mapping to a skeleton more restrictive.

To further compress the representation temporally, we parameterize the trajectories via cubic Bézier curves with symmetric tangents. From empirical evaluation (see Appendix A.2), we have seen that it is sufficient to place a Bézier control point on every sixth frame to achieve good reconstruction quality while maintaining fine-grained control over the motion. To obtain ground truth Bézier curves, we employ optimization-based curve fitting using the parameters described in Section 3.2.

For each joint $j \in J_{sub}$, we define its 3D position at time i as P_j^i. Its position trajectory, P_j, is then given by P_j^i for $i \in 1, \ldots, N$, where N is the window length. We can then stack all position trajectories as $P = [P_j | j \in J_{sub}]$. We similarly define the tangents T, such that P and T define the full set of Bézier curves. Lastly, orientations O are represented as 6D vectors [Zhou et al. 2019][1].

Since the data only relies on a characteristic subset of joints, multiple data sets can be used or potentially even combined, although we do not explicitly show training on combinations of datasets. To do so one can use a simple scaling and an alignment of the pose to compute fixed orientation offsets.

Lastly, we incorporate a foot contact label, as is common in motion models to model foot contacts to reduce foot skating artifacts [Tevet et al. 2022]. For the Bézier control points of the feet, we make use of contact labels generated for the dense motion trajectory at the corresponding time step. For details on labeling the foot contacts, we refer to [Zhang et al. 2023b].

Data Alignment. As we are targeting the inbetweening use case in this work, the start and end frames for J_{sub} are always known. To further incorporate sparse joint-wise constraints, we cannot work in a parent local space as is the case for [Qin et al. 2022], a

[1]The usage of orientations is mostly used to ensure temporal stability when a frame-based IK-module is used. An IK-module that also models time would not require the orientations.

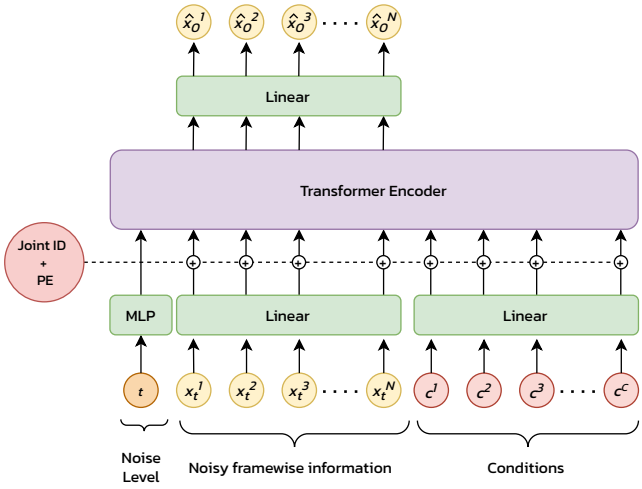


Figure 3: Model architecture of the BMM. The noisy Bézier control points and conditions are fed to a transformer encoder via independent linear encoding layers and decoded to the denoised Bézier control points. All tokens receive the positional encoding of the respective frame they represent. Furthermore, a one-hot vector representing the joint ID—or trajectory ID—is embedded with a linear layer, and added to the corresponding condition tokens.

root-velocity space offered by the popular HumanML3D dataset [Zhang et al. 2023b], or a root-relative space [Cohan et al. 2024; Karunratanakul et al. 2023], since the root might not be known ahead of time. Therefore, we work in a global coordinate frame and reduce the problem complexity by translating all samples such that the hip joint starts at the origin. The height of the root is not adjusted. We omitted a further alignment such that the hip joint faces in its local forward direction, similar to [Qin et al. 2022]. While this can be helpful, it adds complexity to know this transform at test time and we have observed sufficient results without it.

4.2 Bézier Motion Model

As outlined in Figure 2, the next step in our method is the generative model, which generates trajectories (or motion curves) for the predefined set of characteristic joints J_{sub}. For the generative process, we employ a diffusion model, that maps pure noise to a set of Bézier curves. Thus we refer to this model as Bézier Motion Model (BMM). The generated trajectories can be further conditioned based on 3D locations of the Bézier points that the individual joints should go through at the stated frames.

Model design. We make use of a simple transformer-encoder-based architecture, shown in Figure 3, which is based on MDM [Tevet et al. 2022]. However, we extend the input with condition tokens to allow for spatial constraints. Similar to MDM, the noise level is provided in a separate input token. Moreover, we stack the position, tangent, and orientation trajectories to create N frame-wise tokens $x_0^i = [P_j^i,\ T_j^i,\ O_j^i | j \in J_{sub}]$ with $i \in 1, \ldots, N$. These tokens are then noised according to the current denoising step t to give x_t^i.

The conditions, c^i, consisting of target joint positions, are encoded with separate linear layers. To enable the encoder to associate the conditions with the trajectories of the different joints in J_{sub}, a one-hot vector representing the joint ID is added. Lastly, the temporal ordering is represented with the typical additive sinusoidal positional encoding (PE), corresponding to the frame the input belongs to.

The model design is kept intentionally close to the standard MDM design because our work focuses not on model improvements but on the benefits of a factorized approach. Improvements to the model design choices or better inference sampling schemes are concurrent to this work. For further implementation details, please see Appendix C.

Loss function. During training, we supervise the predicted Bézier parameters, namely positions $\hat{P}$, and tangents $\hat{T}$, using an L2 loss function against fitted ground truth trajectories P and T respectively. Similarly, the predicted orientations $\hat{O}$ are supervised using an L2 loss against ground truth O. To better match the constraints, an additional L2 loss is used for the constrained joints:

$$\mathcal{L}_c = \left\| C \otimes (P - \hat{P}) \right\|_2, \tag{4}$$

where C is a binary constraints mask identifying which constraints are present and $\otimes$ is element-wise multiplication.

Additionally, we generate dense positions $\hat{P}_d$ by sampling the Bézier curves given by $\hat{P}$ and $\hat{T}$ using Equation (3). We then use finite differences on $\hat{P}_d$ to estimate velocities $\hat{V}$. We use these dense trajectories for dense position and velocity losses against ground truth P_d and V_d respectively.

As seen in prior works [Qin et al. 2022; Tevet et al. 2022], an additional loss is required to reduce foot skating and improve ground contact. We use a contact loss similar to [Tevet et al. 2022], consisting of an L2 loss on predicted foot contact labels and regularizing the L2 norm of the feet velocities at the frames with ground contact. For more details and weighting on the different loss terms, see Appendix C.1.

4.3 IK-module for Pose Completion

In the next step, the Bézier control points are mapped to dense trajectories of J_{sub}. For the spatial position, the Bézier curves are evaluated on uniform samples between the control points according to Equation (3), and the orientations are computed with spherical interpolation. In principle, one could have also modeled the orientations as a 6D Bézier curve, however, for our experimentation a simple interpolation was deemed sufficient. From those dense trajectories, the full pose is reconstructed by using a learned Inverse Kinematics system. Compared to the skeleton-agnostic BMM, the IK-module is skeleton-specific. We showcase that one could use several systems to map the sparse set of joints to a full pose, ranging from simple frame-based models [Agrawal et al. 2023; Oreshkin et al. 2022], to temporal models [Qin et al. 2022], and to another diffusion model trained with dense conditions [Cohan et al. 2024], or even a traditional rig. To exemplify the generative power of our BMM, we stick to a simple frame-based model in most experiments, namely a pre-trained ProtoRes model [Oreshkin et al. 2022]. Later, in Section 6.4, we showcase how other models can be used.

Frame-based IK-module. The ProtoRes model can map a subset of joint handles to a full pose. These handles can be 3D position, global orientations, or look-at targets. Due to the frame-based nature of the model, there is no notion of temporal consistency when evaluating two independent sets of handles, besides their similarity in the input domain. When only the positional handles are used, one can observe that the poses exhibit some temporal jitter in the orientations of the end-effectors. We found that providing ProtoRes with additional orientation handles alleviates this issue, so we always add the predicted orientations from the BMM as orientation handles. Overall, the input to ProtoRes consists of the six positions and six rotation handles and its output is the full pose consisting of the root position and orientations on all joints.

5 Evaluation

In the following, the main metrics, baseline comparisons, and implementation details are explained that are used for the quantitative evaluation discussed in Section 5.4.

5.1 Evaluation Metrics

The main objective of our model is that it can be reliably used by an artist by using sparse joint-level constraints. The generated motion should hit the constraints precisely while looking natural.

To measure the accuracy of the user-specified constraints, we compute the L2 distance of the sparse constraints and refer to this metric as the handles error.[2] This measures the error only on intermediate constraints and excludes the first and last frames. We evaluate this metric for two ways of sampling the intermediate constraints. When the constraints are sampled directly from a data clip, we refer to this as in-distribution. While this best captures how well the model has learned to reconstruct the data, it does not necessarily reflect how well the model can stick to user-specified constraints, as those may go out of the data distribution. To quantify such out-of-distribution samples (OOD), we also add random noise to the sampled constraints to emulate such user-specified variations. The noise is sampled from a uniform distribution of ±16cm per axis.

To measure the naturalness of the generated motion we compute FID on motion embeddings, which quantifies how close the generated data is to the original data distribution. Additionally, as it is common for generative motion models, we also report Diversity and Multimodality [Tevet et al. 2022].

Lastly, to quantify the quality of the generated motion, in particular for locomotion, we report a foot skating metric. For more details on these metrics, see Appendix B.

5.2 Dataset and Implementation Details

We evaluate our model performance against the start of the art on the publicly available AMASS [Mahmood et al. 2019] and LaFan1 [Harvey et al. 2020] datasets. Additionally, we used a small internal dataset to test the data limits of training a motion model compared to an IK-module. As we do our predictions in a simple world-space representation, we do not make use of the data format of HumanML3D [Guo et al. 2022], but instead use the unconverted AMASS part of it for training and evaluation. After inference, results are converted to HumanML3D's representation in order to compute the FID metric in Section 5.4 using embeddings generated by their feature extractor. We downsample AMASS to 20 FPS for all listed metrics to do this. The LaFan1 dataset primarily contains locomotion data with some dance and other miscellaneous motions. We use subjects 1 through 4 as training data and subject 5 as validation data. Lastly, our internal dataset includes 45 minutes of locomotion, skipping, and yoga motions recorded at 30 FPS.

For retargeting results shown in Section 6.4 and in the supporting video, the skeletons of different datasets must be aligned to the same pose and scaled appropriately. Then, fixed orientation offsets can be computed, which allows to transform the trajectory predictions to the appropriate orientation spaces. Lastly, we train our Bézier model for a 31-frame window and additionally add zero to four uniformly sampled sparse intermediate constraints for each sequence.

For recovering the full pose from the predicted trajectories, we use our implementation of ProtoRes [Oreshkin et al. 2022] and SKEL-IK [Agrawal et al. 2023] models. Both learned Inverse Kinematics models are implemented without any modifications.

5.3 Baseline Comparisons

As detailed in Section 2.1, while methods such as Guided Motion Diffusion [Karunratanakul et al. 2023], OmniControl [Xie et al. 2023], and DNO [Karunratanakul et al. 2024] are capable of using sparse constraints, they are either restricted to taking constraints on only the root joint or rely on expensive optimization that does not allow for interactive rates. As we show in the supplementary video, in the case of OmniControl, the produced motions have unrealistic behavior when using sparse constraints with larger jitters to match the constraints.[3] Hence, we omit these types of models from quantitative comparisons.

CondMDI [Cohan et al. 2024], which is designed for inbetweening with additional full intermediate frames, comes closest to our use case. Therefore, we adapt CondMDI for inbetweening with sparse joint-wise constraints for our comparison. To do so, we use the same global representation as in our BMM instead of their root-relative representation. We refer to this adaption as CondMDI$^{\mathrm{G}}$.

Additionally, we evaluate against two ablated versions of our model. The first ablation predicts the orientation of all joints and all time steps, thus it neither uses a Bézier representation nor does it factorize the motion space. To allow this model also to use inpainting, analogous to BMM or CondMDI, the data is represented as a pair of positions and orientation, where the constrained positions can be inpainted during inference. We use the same transformer architecture as for our Bézier model and refer to this ablation as BMM$^{-IK,-B}$ model. A second ablation, BMM^{-B}, does not use a Bézier points but predicts the dense 3D trajectories for J_{sub}.

5.4 Quantitative Evaluation

In Table 1, we report the results of our quantitative comparison of BMM with the three different baselines for both the LaFan1 dataset as well as the AMASS dataset. The values are averaged for 3k samples on LaFan1 and 10k samples on AMASS. In the latter

[2] We use the terminology handles, since we regard the constraints as 3D control handles a user can interact with similarly to an IK constraint.

[3] This was evaluated on a pre-trained model that was not specifically trained for this sparse use case.

Table 1: Results of our quantitative evaluation of BMM against three baselines. Validation for LaFan1 is done for 3k samples, and 10k samples on AMASS to ensure a proper FID computation. For BMM and BMM^{-B}, the metrics are computed after the IK-module for which we use a ProtoRes model trained on the specific dataset.

Dataset	Model Type	Handles Error [cm] ↓	OOD Handles Error [cm] ↓	Foot Skating Ratio ↓	FID ↓	Diversity →	Multimodality ↑
LaFan1	LaFan1 GT	-	-	0.1209	-	-	-
	CondMDI^{G}	8.6809	15.4506	0.2489	-	-	-
	BMM$^{-IK,-B}$	14.8722	20.1980	0.2861	-	-	-
	BMM^{-B}	3.7742	6.8788	0.2347	-	-	-
	BMM	**0.8914**	**1.7090**	**0.2176**	-	-	-
AMASS	AMASS GT	-	-	0.1089	0.0043	1.5797	-
	CondMDI^{G}	12.8676	19.8571	0.1902	0.2178	1.4035	1.2122
	BMM$^{-IK,-B}$	14.5924	20.7648	0.2708	**0.1873**	1.4048	1.2098
	BMM^{-B}	4.0008	6.6274	**0.1138**	0.2024	1.5275	1.4283
	BMM	**1.1573**	**1.8384**	0.1709	0.1888	**1.5492**	**1.6382**

case, we computed FID (see Section 5.2 for details). As LaFan1 uses a different frame rate (30 FPS) compared to the feature extractor (20 FPS), FID, Diversity, and Multimodality are not computed.

BMM outperforms the other baselines in terms of handles accuracy. This is even more pronounced in the out-of-distribution (OOD) case, which is crucial for a model that is used in an interactive motion authoring tool. Due to the overwrite mechanism of the BMM, outlined in Appendix A.3, the handles accuracy is bounded by the accuracy of the trained IK-module. By looking at the FID, we also see that the accuracy of our method does not come at the cost of creating unnatural motion and still represents the data distribution well.

The ablation of BMM^{-B}, which predicts dense 3D curves for J_{sub}, performs significantly better than the other baselines that rely on reconstructing a full FK-skeleton. This confirms our intuition that it is harder for diffusion models to learn to accurately hit sparse constraints, when there is an indirection between condition and prediction space, here via the FK-chain. By comparing BMM^{-B} with BMM, we see that the Bézier curves representation also adds a benefit for hitting the constraints. Partially, this improvement comes from the overwriting mechanism. But as we report in Appendix A.3, even without the overwriting, BMM has a more accurate handles loss. Additionally, overwriting cannot be applied for BMM^{-B} without introducing artifacts and for BMM it does not result in a worse FID. Interestingly, we also observe that BMM^{-B} exhibits improved foot skating on AMASS. We suspect that this indicates that for accurate foot contacts, it is helpful to have a denser resolution than the six frames of the Bézier curves.

Lastly, the comparison of BMM$^{-IK,-B}$ and CondMDI^{G}indicates that the UNet architecture of CondMDI could be better than the transformer architecture of BMM. However, since the conditioning mechanism in the transformer offers more flexibility on how we can condition the model, we did not further investigate the usage of a UNet architecture for BMM and leave this for future work.

6 Results

In this section, we will cover the qualitative results of our approach.

6.1 Handles Accuracy

For influencing the motion with sparse joint-wise handles the constraints must be hit accurately while also creating smooth and natural motion. Due to the overwrite mechanism and the temporal smoothness of the Bézier curves, our model can create natural curves that go directly through the user constraints. On the other hand, Figure 4 shows how the baseline CondMDI^{G} struggles to hit the constraints accurately and even creates visual artifacts such as sharp peaks in the curves. These artifacts showcase the problem of the diffusion model to respect the sparse constraints in the generation process. For CondMDI^{G}, we also experimented with increasing the loss weighting on the constraints but found that it led to negligible improvements in handle error while decreasing the realism of the motion through more pronounced peaks in the curves.

Figure 4: Comparison of handles accuracy for the CondMDI^{G} (left) baseline and our BMM (right).

6.2 Influence of Intermediate Handles

The effect of editing the intermediate constraints on the curves and the pose can be best observed in the supplementary video. However, Figure 5 still clearly depicts the effects of moving a single intermediate handle, the hip. The position of the hip not only changes the timing of where the pose is in space but also heavily affects the overall type of motion. Even from a single visualized pose, one can see that the character is in a sprinting position when the edited hip is close to the starting frame (top row). When the hip is in the middle, the pose resembles more of a jogging pose, and when the hip is close to the final frame (last row), the character appears to be decelerating, with the center of mass behind the legs.

The fact that the speed or type of the motion can be inferred from a single pose highlights that even a frame-based IK-module

Figure 5: Comparison of motion curves when the middle root handle is moved in space (highlighted in white) to 6 different locations between the start and end. All images depict the same time step in the animation. The closer the root is moved to the start, the less time the character has to reach the final frame in the remaining frames, which changes the motion from a sprint (top) to a run (middle) and finally to a casual walk (bottom).

can create motion when the curves are descriptive enough of the motion.

Furthermore, the influence of moving a single joint is also distinctly different from the CondMDI$^{\text{G}}$ and BMM$^{-IK,-B}$ that rely on a representation of orientations. In that case, the influence is much more locally confined to the curve of that joint, which indicates again that the model struggles to propagate the relevant correlations properly across the kinematic chain. BMM and BMM^{-B}, which directly predict in the 3D space, have a much more global influence.

6.3 Editing Process

Figure 6 depicts how a running motion, generated without intermediate constraints, can be edited to become a running motion with a jump in the middle. This is easily done by moving just a few control points. First, the hip is adjusted to create a basic jump. Secondly, the leading leg is raised to better overcome the obstacle. Lastly, the arms are raised in mid-air for artistic reasons.

The editing process works at interactive rates and the user directly gets feedback from the shape of the curves and the reconstructed pose. Due to the overwrite mechanism of the Bézier points, the points are hit accurately and do not change with the addition of new joint-wise constraints. Compared to our baseline models, this ensures a non-destructive workflow, which is essential for real-world usage. During the editing process, the user can set different seeds to the diffusion process to explore other generated samples. For more editing results we refer to the supplementary video.

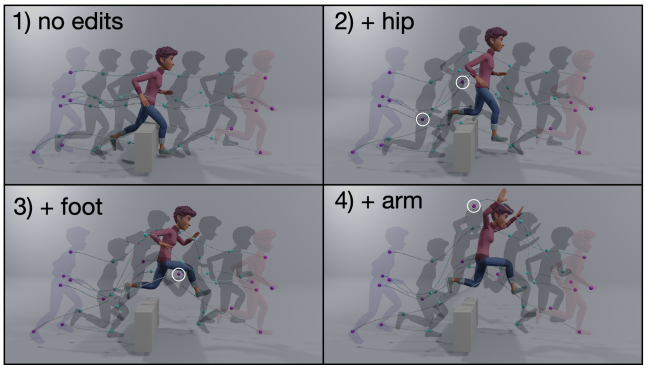


Figure 6: Visualization of an iterative editing process with the predicted Bézier curves. A generated running motion is converted into a run and a jump by simply moving a few handles (highlighted in white).

6.4 Modularity of IK-Modules

The factorization allows the use of a variety of skeletons and IK-modules to go from the Bézier curves to the final motion.

Different skeletons. In Figure 7, we showcase this flexibility for four different skeletons that use the same Bézier curves. All characters employ a ProtoRes model trained on their respective data. In the case of the frog character, a traditional IK rig is used to drive the reversed legs of the character, for which there is no data to train a pose model (see Appendix D.4 for more details). The motion curves are faithfully mapped to all characters, despite different skeletal hierarchies and proportions.

In the supplementary video, we showcase another type of skeleton with variable bone length. In this case, an IK-module similar to SMPL-IK [Voleti et al. 2022] is used to map to the full pose. Since this can be done on a per-frame basis, the bone length of the model can even be animated during the motion, which would be hard to achieve in a traditional way of retargeting. See Appendix D.3 for further details on this setting.

Different models. Besides different skeletons, one can also employ different types of IK-modules. Besides the ProtoRes model, we also tested SKEL-IK as another frame-based model. This model employs the additional notion of a base pose that should be conserved. By conditioning on the respective previously generated pose over time, this model also offers a way to add natural smoothness to predictions despite its frame-based nature. We observed that in this case, it is possible to omit any orientation handles as inputs, which further simplifies the application of the model across different characters.

Additionally, we tested two temporal models that have been trained on dense 3D condition trajectories: TwoStage [Qin et al. 2022] and our adapted version of CondMDI$^{\text{G}}$. Both models hit the generated curves accurately and thereby also the user-controlled conditions. In the case of CondMDI$^{\text{G}}$, this also supports our claim that the sparseness of the conditions is a major problem for the diffusion model to hit constraints accurately. When provided with a dense condition signal, it is easier for the generative model to use

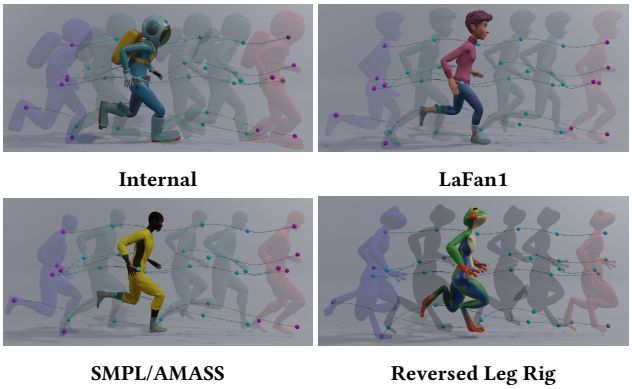


Figure 7: Bézier motion curves applied to different skeletons. For the first three skeletons, a ProtoRes models is used as the IK-module. The Reversed Leg Rig uses traditional IK controllers on the lower body. The start and end frame come from a validation clip of LaFan1.

the information and satisfy the constraints. Interestingly, we observe very little variability in the generated motions for CondMDI^{G} trained on dense conditions, indicating that the motion is sufficiently determined by the Bézier motion curves and J_{sub}. Similar to what we observe for SKEL-IK, motion models trained on dense conditions do not require orientation conditions to produce smooth motions. However, using a motion model or even an additional diffusion process introduces computational complexity that can significantly slow down the overall system.

6.5 Training with Small Dataset

To demonstrate the advantage of our factorized approach and a purely frame based pose model, we also trained an IK-module on a 10 minute subset of our internal data. This contains a variety of poses but little locomotion. As a comparison, we also trained BMM on this limited data.

As can be seen from the supplementary video, BMM trained on the small internal dataset cannot create realistic-looking motions for poses taken from an evaluation clip of the same dataset. On the other hand, when the pose model is applied on top of BMM trained on LaFan1, the curves and reconstructed full-body motion look much more realistic for this character, indicating that the IK-module is still sufficient for such a case of limited data.

7 Discussion

We hypothesized that diffusion models struggle with conditioning on sparse constraints in an inbetweening workflow due to the sparseness of the condition and the discrepancy between condition signal and output domain. By factorizing out the IK task from the generative process, our motion model, BMM, is directly regressing global positions and orientations. Additionally, our two stage approach decreases the sparsity of conditions by only working with J_{sub} in the generative stage. Working with Bézier representation for motion curves further decreases the condition sparsity. With comparisons, we showed that these design choices result in significantly better accuracy.

Due to our factorization, BMM is skeleton-agnostic as it is only aware of the leaf joints, i.e. the hands, the feet, the neck, and the root joint. The rest of the skeleton hierarchy is only known to the secondary IK-module. We show that working in this reduced skeleton space is sufficient for most motions and the final motion can be faithfully recovered using a number of IK-modules, ranging in complexity from simple traditional rigs to even other generative motion models.

Besides reducing the condition sparsity, working in a skeleton-agnostic space has several advantages. One can easily transfer the motion from one character to another, enabling animating characters with very little to no data or possibly even training on mixtures of datasets from different skeletons. However, a simple skeleton-agnostic approach might not be sufficient when working with characters of significantly different proportions or different topologies such as quadrupeds. Moreover, there might be some artistic controls, such as foot roll, that cannot be modeled in our restricted definition of a characteristic skeleton, J_{sub}. One could also introduce auxiliary joints to J_{sub}, e.g. the knee or the toes to reintroduce a foot roll, while still being largely skeleton-agnostic for the generative step. As we have seen in Image generation and Natural Language Processing, leveraging large datasets unlocks true generative powers of diffusion models. We hope that future work will explore more skeleton-agnostic approaches that can train on mixtures of datasets with varied proportions and even topologies.

Our use of Bézier curves also provides post-inference benefits. Firstly, the constraints can be inpainted even after the last step of the inference without introducing any discontinuity in the predicted curves. Secondly, animators are already familiar with this type of control for modeling motion. They are sparse and simpler to control compared to editing dense motion. And lastly, their widespread usage in any 3D animation tools simplifies the integration into such software and into the artists' workflow.

7.1 Limitations

While our approach offers several advantages over current approaches, it also comes with some limitations.

We showed that our model can generate realistic motion from a highly sparse set of conditions. However, similar to other learned methods, our model fails when the conditions are very far from the training data distribution. We include an example of one such failure in our supporting video. The ground truth depicts jumping down from a higher ground and rolling before standing. As both moving between different ground heights and rolling are ill-represented in the training data, our method struggles to generate realistic motion. However, the smooth interpolation characteristics of Bézier curves allow the animators to more easily clean up the initial prediction, indicating better usability.

While our representation helps with hitting constraints, it also lowers the theoretical upper bound on the final motion quality. As we show in Appendix A.1 and A.2, we select our J_{sub} and Bézier stride parameters to offer a good balance between motion quality and compression. Despite this upper limit, the generated motion quality is still comparative to the current state of the art.

While the control over the Bézier curves is closer to what an animator is familiar with, the factorization also implies that in

our implementation the generator can only be conditioned on the modeled joints, and only on frames that have a Bézier control point. Thus, to influence the other joints, one would either need to extend the conditioning method in the generator or address this in the IK-module. For a purely frame-based IK-module, such a condition is challenging because the influence is not automatically propagated over time. To address this shortcoming, it would require additional mechanisms that go beyond the scope of this work.

8 Conclusion and Future Work

While there has been rapid progress in the field of generative motion modeling, it is heavily constrained by the current representation of the datasets and their underlying skeleton. In contrast to the image domain, where data is already represented in one unified representation, namely RGB pixel values, motion always depends on a specific skeleton. This limits the scalability of data and also the usage of the trained models. We believe that similar to other domains, there is a significant benefit to overcoming this lack of alignment in the motion domain. We hope that our investigation towards a more unified motion space will inspire future research in this direction.

In the future, we envision addressing the fixed time interval in the Bézier handles, to give more freedom of where the conditions can be placed. Additionally, as animators set control points for their curves in very specific ways, it would significantly help adoption in the industry, if the timing could also be predicted simultaneously.

References

Kfir Aberman, Peizhuo Li, Dani Lischinski, Olga Sorkine-Hornung, Daniel Cohen-Or, and Baoquan Chen. 2020. Skeleton-aware networks for deep motion retargeting. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 62–1.

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.

Andreas Aristidou and Joan Lasenby. 2011. FABRIK: A fast, iterative solver for the Inverse Kinematics problem. *Graphical Models* 73, 5 (2011), 243–260.

Autodesk Inc. 2023. Autodesk Maya - a 3D modeling, animation, and rendering software. http://www.autodesk.com/maya Version 2023, Autodesk, Inc., San Rafael, CA.

Blender Online Community. 2023. Blender - a 3D modelling and rendering package. http://www.blender.org Blender Foundation, Stichting Blender Foundation, Amsterdam.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. 2024. Video generation models as world simulators. (2024). https://openai.com/research/video-generation-models-as-world-simulators

Setareh Cohan, Guy Tevet, Daniele Reda, Xue Bin Peng, and Michiel van de Panne. 2024. Flexible Motion In-betweening with Diffusion Models. *arXiv preprint arXiv:2405.11126* (2024).

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. 2024. Scaling Rectified Flow Transformers for High-Resolution Image Synthesis. arXiv:2403.03206 [cs.CV] https://arxiv.org/abs/2403.03206

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. 2022. Generating Diverse and Natural 3D Human Motions From Text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 5152–5161.

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.

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.

Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. 2023. Guided motion diffusion for controllable human motion synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2151–2162.

Jehee Lee and Sung Yong Shin. 1999. A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. 39–48.

Sunmin Lee, Taeho Kang, Jungnam Park, Jehee Lee, and Jungdam Won. 2023. Same: Skeleton-agnostic motion embedding for character animation. In *SIGGRAPH Asia 2023 Conference Papers*. 1–11.

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.

OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL] https://arxiv.org/abs/2303.08774

Boris N. Oreshkin, Florent Bocquelet, Félix G. Harvey, Bay Raitt, and Dominic Laflamme. 2022. ProtoRes: Proto-Residual Network for Pose Authoring via Learned Inverse Kinematics. arXiv:2106.01981 [cs.CV]

Boris N Oreshkin, Antonios Valkanas, Félix G Harvey, Louis-Simon Ménard, Florent Bocquelet, and Mark J Coates. 2023. Motion In-Betweening via Deep Δ-Interpolator. *IEEE Transactions on Visualization and Computer Graphics* (2023).

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

Sigal Raab, Inbar Gat, Nathan Sala, Guy Tevet, Rotem Shalev-Arkushin, Ohad Fried, Amit H. Bermano, and Daniel Cohen-Or. 2024. Monkey See, Monkey Do: Harnessing Self-attention in Motion Diffusion for Zero-shot Motion Transfer. arXiv:2406.06508 [cs.CV] https://arxiv.org/abs/2406.06508

Yeongho Seol, Carol O'Sullivan, and Jehee Lee. 2013. Creature features: online motion puppetry for non-human characters. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 213–221.

Hyun Joon Shin, Jehee Lee, Sung Yong Shin, and Michael Gleicher. 2001. Computer puppetry: An importance-based approach. *ACM Transactions on Graphics (TOG)* 20, 2 (2001), 67–94.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*. PMLR, 2256–2265.

Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H. Bermano. 2022. Human Motion Diffusion Model. arXiv:2209.14916 [cs.CV]

Vikram Voleti, Boris Oreshkin, Florent Bocquelet, Félix Harvey, Louis-Simon Ménard, and Christopher Pal. 2022. Smpl-ik: Learned morphology-aware inverse kinematics for ai driven artistic workflows. In *SIGGRAPH Asia 2022 Technical Communications*. 1–7.

Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. 2023. Omnicontrol: Control any joint at any time for human motion generation. *arXiv preprint arXiv:2310.08580* (2023).

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Yong Zhang, Hongwei Zhao, Hongtao Lu, Xi Shen, and Ying Shan. 2023b. Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 14730–14740.

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

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 MOTIVATION OF DESIGN CHOICES

In the following section, we will go into more details on how we justify our design choices for BMM.

A.1 Sufficiency of Sparse Handles

To verify that a sparse set of constraints is sufficient to accurately reconstruct the original motion, we compared the reconstruction quality that ProtoRes can achieve for different joint configurations of J_{sub}. In Table 2, the reconstruction quality for LaFan1 validation data is shown. From this, we determined that using positions and orientations of root, neck, hands, and feet provide the best trade-off between reconstruction quality and joint sparseness. The latter of which is important for using our method with different skeletons and to achieve a larger reduction of the prediction space.

For visualizations of different subsets of joints and also for omitting the orientation constraints, we refer to the supplementary video.

Table 2: Reconstruction error of a pre-trained ProtoRes model for different configurations of subset skeleton. For all joints, position and orientation handles are provided from the LaFan1 validation data. Error is measured as Mean Per Joint Positional Error (MPJPE) in cm.

Joints used as constraints	MPJPE ↓
All joints	0.9943
Root, Neck, Hands, Feet, Elbows, Knees	1.1610
Root, Neck, Hands, Feet	1.2802
Root, Hands, Feet	2.8380
Hands, Feet	6.5878

A.2 Stride between Bézier Control Points

To determine a good value for the temporal discretization between Bézier control points, we evaluated the reconstruction for several settings, as shown in Figure 8. We find that even with larger Bézier strides, such as 6-9, the reconstruction quality is mostly dominated by the lower bound of the error from the ProtoRes model. This emphasizes the power of Bézier curves for compressing motion data. However, bigger strides come at the downside of reduced controllability. We found that a stride of six offers a good trade-off between reconstruction quality and controllability.

A.3 Overwrite for Bézier Curves

The compressed representation of time with Bézier curves not only leads to a smaller input size but also to the ability to employ a simple overwriting mechanism to accurately match the 3D conditions. As shown in Figure 9, the temporal smoothness of the Bézier curves guarantees a smooth final trajectory, even when there are slight prediction errors. When using dense 3D trajectories such a direct overwrite cannot be done without introducing artifacts. Diffusion models, like CondMDI, often employ such an overwrite mechanism also during inference, such as for inpainting. While this reduces the artifact a bit, it does not fully get rid of it. Additionally, any inference based mechanisms, like overwriting or guidance, to help with prediction error, can also be employed in the case of Bézier curves.

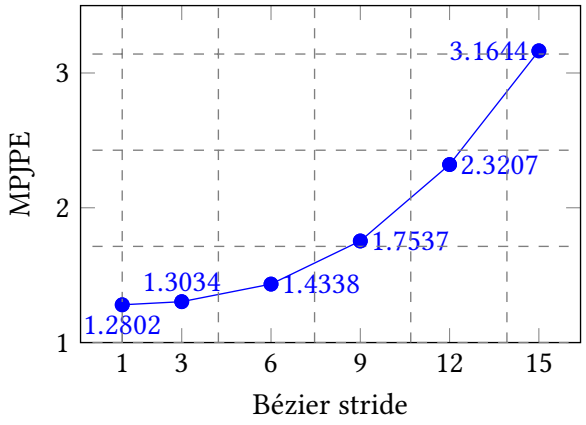


Figure 8: Reconstruction error for different strides of Bézier control points. The error is computed as Mean Per Joint Positional Error (MPJPE) in cm after pose completion with a ProtoRes model on a LaFan1 validation data.

Table 3: Comparison of BMM with and without overwriting on AMASS.

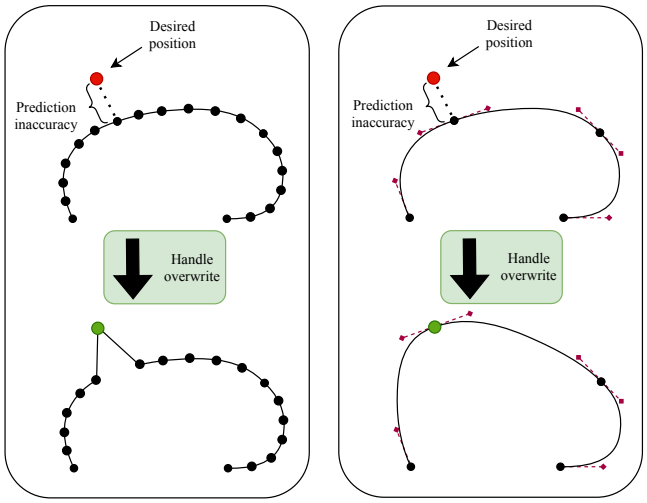
Model	Handles Error [cm] ↓	OOD Handles Error [cm] ↓	Foot Skating Ratio ↓	FID
BMM	**1.1573**	**1.8384**	0.1709	**0.1888**
BMM w/o overwrite	2.6455	4.6216	**0.1556**	0.1932

Overall, the overwriting scheme guarantees that the generated curves will exactly hit the constraints and all constraint errors are accredited to the IK-module. In the case of a learned model, the latter can still introduce some discrepancy between the visualized pose and the motion curves. However, as pose based models keep improving, this discrepancy will also go down.

B EVALUATION METRICS

Fréchet Inception Distance (FID). FID is a measure of distribution similarity. First proposed in [Heusel et al. 2017], it is used to measure the performance of generative models by comparing features of produced values to ground truth features, after feeding them through a feature extractor. As proposed in [Tevet et al. 2022], we use the motion feature extractor of [Guo et al. 2022b]. Since we do not use HumanML3D directly, we instead split up the combined evaluation and test set of AMASS into non-overlapping 31-frame segments and subsequently convert them to a 263-dimensional representation with the conversion script from [Guo et al. 2022b]. We do the same for generated motions and encode everything with their feature extractor. The mean and covariance of the encoded features are then computed and used to obtain FID as

$$FID(\mu_{\text{Pred}}, \Sigma_{\text{Pred}}) = ||\mu_{\text{GT}} - \mu_{\text{Pred}}||_2^2 + tr\left(\Sigma_{\text{GT}} + \Sigma_{\text{Pred}} - 2(\Sigma_{\text{GT}}\Sigma_{\text{Pred}})^{\frac{1}{2}}\right). \quad (5)$$



(a) Dense 3D (b) Bézier curves

Figure 9: Schematic comparison of overwriting a slight prediction error for trajectories parameterized as (a) dense 3D points or (b) or Bézier curves. The temporally smooth nature of the latter helps to avoid artifacts.

Handles Error. The handles error measures the mean L2-norm between predicted joint trajectories and the desired locations. Note that we only compute this on intermediate frames and not the first and last frames, to get a better representation of sparse constraint hitting.

Foot Skating Ratio. Similar to [Zhang et al. 2021], we consider a motion to exhibit foot skating if both heels are within 5 cm of the floor and move laterally at more than 7.5 cm/s. This metric then represents the fraction of frames on which these conditions are met, i.e. skating occurs.

Diversity. Diversity measures the variance of encoded motions, which should ideally be close to that of ground truth. Like [Tevet et al. 2022], this is obtained by randomly sampling two subsets $\{v_1, ..., v_{S_d}\}$, $\{v_1^{'}, ..., v_{S_d}^{'}\}$ of equal size (we choose $S_d = 1000$) from generated motions and estimating the variance with

$$\text{Diversity} = \frac{1}{S_D} \sum_{i=1}^{S_d} ||v_i - v_i^{'}||_2. \tag{6}$$

Multimodality. This metric measures variance under fixed conditions, or how varied output motions are when providing the model with the same constraints. Similar to diversity, it is computed as

$$\text{Multimodality} = \frac{1}{C \times S_m} \sum_{c=1}^{C} \sum_{i=1}^{S_m} ||v_{c,i} - v_{c,i}^{'}||_2, \tag{7}$$

where we first pick $C = 100$ conditions at random from GT and then sample $S_m = 20$ times per condition. For each condition c, the output is split into two subsets $\{v_{c,1}, ..., v_{c,S_m/2}\}$, $\{v_{c,1}^{'}, ..., v_{c,S_m/2}^{'}\}$.

C IMPLEMENTATION DETAILS

C.1 Training Hyperparameters

In Table 4, the hyperparameters for training BMM are listed. Additionally, the noise-dependent loss scaling method proposed in [Hang et al. 2024] was applied to improve convergence speed.

Table 4: Hyperparameters of BMM

Hyperparameter	Value
Learning rate	1e-4
Optimizer	AdamW w/ AMSGrad
Weight decay	1e-2
Batch size	64
Latent dim	512
FF Layer dim	1024
# Heads	4
# Layers	8
Noise scheduler	DDIM [Song et al. 2022]
Noise schedule	Cosine [Nichol and Dhariwal 2021]
Diffusion steps	1000

C.2 Loss functions

The training loss consists of the following terms:

Bézier point position loss $\mathcal{L}_b$. L2 loss between predicted and GT Bézier positions. This can be represented as:

$$\mathcal{L}_b = \left\| P - \hat{P} \right\|_2 \tag{8}$$

Bézier point tangent loss $\mathcal{L}_t$. L2 loss between predicted and GT Bézier tangents. This can be represented as:

$$\mathcal{L}_t = \left\| T - \hat{T} \right\|_2 \tag{9}$$

Orientation loss $\mathcal{L}_o$. L2 loss between predicted and GT orientations on Bézier control points.

$$\mathcal{L}_o = \left\| O - \hat{O} \right\|_2 \tag{10}$$

Dense position loss $\mathcal{L}_p$. L2 loss between positions sampled from the Bézier curves and GT positions.

$$\mathcal{L}_p = \left\| P_d - \hat{P}_d \right\|_2 \tag{11}$$

Velocity loss $\mathcal{L}_v$. L2 loss between predicted velocities, via finite differences, and GT velocities.

$$v_{d_j}^{\,i} = P_{d_j}^{\,i} - P_{d_j}^{\,i-1} \tag{12}$$

$$\mathcal{L}_v = \left\| V_d - \hat{V}_d \right\|_2, \tag{13}$$

where V_d is a velocity vector containing all $v_{d_j}^{\,i}$ for $j \in J_{sub}$ and $i \in 1, \ldots N$. We define $\hat{V}_d$ similarly using the predicted positions $\hat{P}_d$.

Bézier point conditions loss $\mathcal{L}_c$. L2 loss between predictions and input conditions on Bézier positions.

$$\mathcal{L}_c = \left\| C \otimes (P - \hat{P}) \right\|_2, \tag{14}$$

Foot contact labels loss $\mathcal{L}_{gc}$. L2 loss on binary contact labels.

$$\mathcal{L}_{gc} = \left\| GC - \widehat{GC} \right\|_2 , \tag{15}$$

where GC are binary ground contact labels.

Foot skating loss $\mathcal{L}_{skate}$. L2 loss on velocities from Bézier points with predicted foot contact label to the respective next discretized frame.

$$\mathcal{L}_{\text{skate}} = \sum_{j \in J_{Feet}} \left\| \widehat{GC}_j \otimes \hat{V}_j \right\|_2 , \tag{16}$$

where J_{Feet} includes the left and right feet joints.
The relative weightings of the loss terms are listed in Table 5.

Table 5: Weightings of loss terms

Loss	$\mathcal{L}_b$	$\mathcal{L}_t$	$\mathcal{L}_o$	$\mathcal{L}_v$	$\mathcal{L}_p$	$\mathcal{L}_c$	$\mathcal{L}_{gc}$	$\mathcal{L}_{\text{skate}}$
Weight	1.0	5.0	1.0	10.0	1.0	5.0	0.01	10.0

While we conducted several experiments on how to weigh those losses properly, the reported values are sufficient to reproduce the shown results, but may not be optimal. Additionally, we experimented with omitting the dense losses of $\mathcal{L}_p$ and $\mathcal{L}_v$, since the positions are already parameterized by the Bézier controls and may seem redundant with respect to $\mathcal{L}_b$ and $\mathcal{L}_t$. However, we observed the dense losses helped to improve foot skating metrics. This may indicate that for the feet the temporal discretization with the Bézier curves could benefit from a more dense resolution. This observation is also supported by the fact that the ablation BMM^{-B} has a better foot skating ratio as reported in Table 1.

C.3 CondMDI$^{\text{G}}$ Baseline

We follow the implementation of [Cohan et al. 2024] and make use of the unconditioned motion diffusion model of Guided Motion Diffusion [Karunratanakul et al. 2023] as the backbone of CondMDI. As outlined before, instead of predicting in a root-relative frame, we predict everything in world-space coordinates to allow for conditioning without requiring the root.

To keep the input close to the inbetweening setting of BMM, we condition the first and last frame only on the positions of the J_{sub}. Note this is different from the inbetweening scenario in [Karunratanakul et al. 2023], where all joints and orientations are given.

We tested several ways to improve the handles accuracy, such as using classifier-free guidance (CFG) [Ho and Salimans 2022] and weighting for the handles loss $\mathcal{L}_c$. Similarly as in the original implementation of CondMDI, using a CFG value of 2.5 improves handles accuracy. Hence, all reported results are evaluated with that setting. However, we observed that the poor accuracy is not simply a result of the weighting on the handles loss. Scaling $\mathcal{L}_c$ up by a factor of four did not result in satisfactory constraints accuracy, but rather led to more peaky motions.

C.4 IK-Modules

C.4.1 Protores. We train our implementation of [Oreshkin et al. 2022] without any modifications for 3600 epochs with a starting learning rate of 2×10^{-4} and decay of 0.5 every 300 epochs.

C.4.2 SKEL-IK. We train our implementation of [Agrawal et al. 2023] without any modifications following the learning parameters stated by the authors.

C.4.3 TwoStage. The original TwoStage model by [Qin et al. 2022] works in parent local space, predicting joint orientations and root translation. We adapted their implementation to predict world space positions and conditions on the 3D positions trajectories for J_{sub}. Hence, we refer to this model as TwoStage^{G}.

C.4.4 CondMDI^{G}. We adapt [Cohan et al. 2024]'s work according to Section 5.3 and train with the joint trajectories for all joints of J_{sub} provided. Note that one could also use their original model without any modification as the root trajectory is known for this application. We use the adapted version to be consistent with all our other results.

C.4.5 Variable bone length model. We adapt the ProtoRes model by [Oreshkin et al. 2022] to include bone proportions of the skeleton, similar to [Voleti et al. 2022]. By sampling a number of bone proportions during training, the converged model is able to generate a pose satisfying the given constraints for different character proportions.

D ADDITIONAL EXPERIMENTS

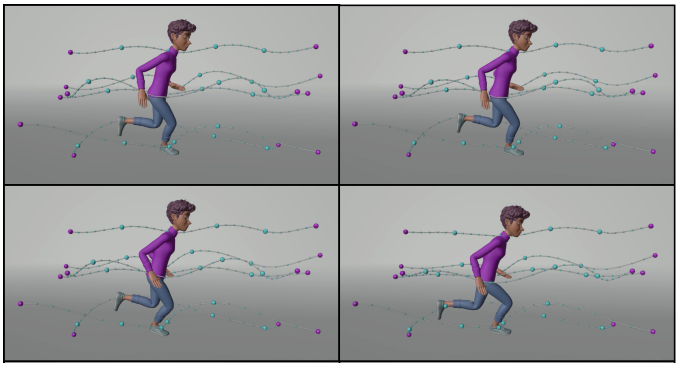
D.1 Inbetweening of Multiple Segments

In our supplementary video, we showcase the reconstruction for inbetweening for multiple sequential segments taken from a LaFan1 validation clip (with and without intermediate constraints). Those segments are evaluated independently since our use case focused on the controlling sparse constraints of a single inbetweening section. Therefore, visual artifacts in the transitions between the different segments can be expected since BMM is not designed for this type of scenario.

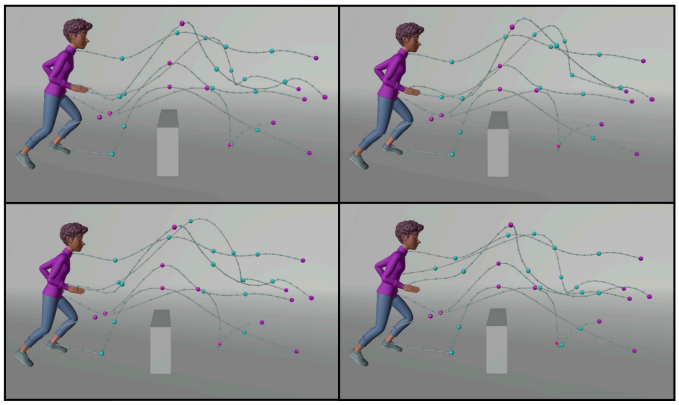
However, even though we focus in this work on the control of a single inbetweening section, the representation of Bézier curves also helps to counter some of the temporal inconsistencies at the boundaries compared to a dense trajectory model. Since the Bézier curves are naturally smooth and the model hits the conditions exactly, one can average the tangents or the predicted orientation on the boundaries to create more temporally consistent motion. Additionally, one can also further couple the inference steps by sharing information between the different boundary variables. In the video, we only averaged orientation handles across the boundary, but did not implement any of the other more sophisticated schemes.

D.2 Sampling Diversity

Figure 10 shows four samples for the same input condition in the unconstrained and constrained case after manual edits, cf. Figure 6. The model is trained on LaFan1 and the start and end frames are sampled from LaFan1 validation clip. For the unconstrained case, the variability is fairly low, which could indicate that the start and end frame already strongly determine the motion. Interestingly, the introduction of manual constraints seems to increase variability. This could potentially come from the fact that the handles put the motion into an underrepresented part of the motion space with large variability, compared to the running motion which might



(a) unconstrained



(b) with intermediate handles

Figure 10: Example of diversity of the sampled Bézier motion curves. a) for the unconstrained case b) a constrained case with intermediate handles to get a jumping motion, cf. Figure 6. While the constrained control points are hit perfectly by design, the unconstrained control points still show variability.

have less variable mode, due to the many locomotion samples in the LaFan1 data. For a better visualization of the sampling diversity, we defer to the supplementary video.

Since the inbetweening interval of 31 frames (~ 1 second) is not too long, it can also not be expected that the diversity can show too many different motions between the start and end poses. Nonetheless, sampling those curves in an interactive way, offers new opportunities to quickly explore motions in an animation workflow.

D.3 Variable Skeleton Proportions

One can also apply the generated Bézier curves to characters of different proportions. To showcase this, we trained a ProtoRes model that can deal with variable bone lengths similarly to SMPL-IK [Voleti et al. 2022]. With such a learned system, we cannot only use a fixed set of bone length but even vary the bone length over time, see the supplementary video or Figure 11. Here the scales are varied for the arms, legs, and spine joints between factors of 0.5 and 1.5. In order to use such extreme proportions, we omitted the

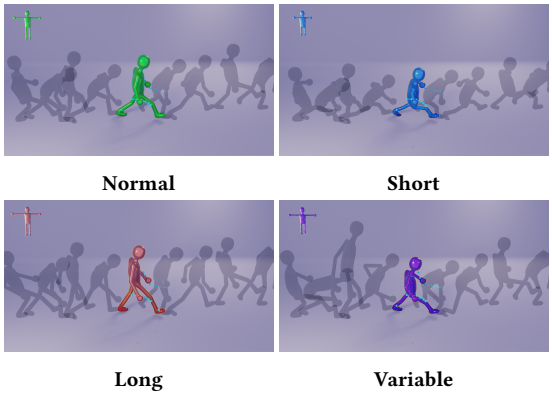


Figure 11: IK-module with variable skeleton proportions.

position handles of the hip and head to the IK-module. Otherwise, those position constraints would become too restrictive, e.g. scaling up the legs and arms could not result in an upright walking pattern, because the hip and head would force the person into a hunched-down pose.

While this experiment nicely highlights the power of factorizing the motion generation into a skeleton agnostic part and an IK-part, it also comes with its limitations when pushed to such extremes. Ideally, the generative models should also have some information about the overall skeleton proportions to better generate motion that looks authentic with those different proportions. We believe this could still be achieved in a mostly skeleton-agnostic way by feeding the proportions of the unified skeletons as additional inputs to the model: such as height of the legs, length of arms, or height from head to hip. However, exploring this extension goes beyond the scope of this work.

D.4 Using a Traditional IK-Rig

As shown in the supplementary video, one can also drive a traditional character Rig with the generated Bézier curves. We showcase this by driving the lower body of the frog character. The legs of this character are so-called reversed legs, that compared to a human skeleton, express two highly animated knee-like joints, one going forward and one backward. The legs are rigged with a multi-step IK-chain, whose positional target for the foot can be driven by the Bézier curves of the heel. For additional control of the leg orientation, we mapped the orientation of the feet to the pull vector of the IK chain.

Since the upper body shared the same skeleton as the LaFan1 skeleton, we applied a ProtoRes model on the upper body.

This example highlights the strength of using a representation of motion for which data are abundant, suitable for training large generative models, and a secondary system that can connect this representation to the specific character for which very little or no data exists. While this is traditionally done with a complex retargeting setup, we believe that our approach simplifies this by implicitly doing a simple yet effective end-effector retargeting under the hood.

Table 6: Evaluation of the number of inference steps.

Steps	Model	Handles error [cm] ↓	Foot Skating Ratio ↓	FID ↓
5	CondMDI^{G}	32.7625	0.2045	1.6199
	BMM	1.1559	0.1803	0.1974
10	CondMDI^{G}	16.7361	0.2157	0.2613
	BMM	1.1573	0.1709	0.1888
25	CondMDI^{G}	12.8676	0.1902	0.2178
	BMM	1.1594	0.1650	0.1855
50	CondMDI^{G}	12.5095	0.1859	0.2185
	BMM	1.1612	0.1632	0.1842
100	CondMDI^{G}	12.5071	0.1846	0.2193
	BMM	1.1642	0.1619	0.1839

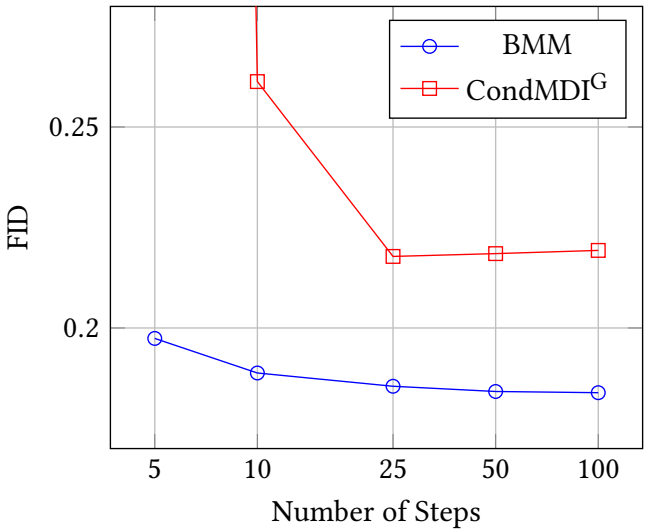


Figure 12: Comparison of FID for different number of inference steps for both BMM and CondMDI^{G}.

Table 7: Comparison of model sizes and computation times.

Model	Parameters	Epoch time	Inference time
CondMDI^{G}	224.6 m	~ 81 min	0.64 ± 0.03 s @ 25 steps
BMM	17.4 m	~ 19 min	0.08 ± 0.01 s @ 10 steps

D.5 Number of Diffusion Steps

For further comparison, we evaluated our BMM and CondMDI^{G} on varying numbers of diffusion steps. We report a reduced set of metrics in Table 6 and visualize FID in Figure 12. We observe that our BMM can operate well with as little as ten diffusion steps, whereas for CondMDI^{G}, at least 25 steps are required. Since inference time scales linearly with the number of diffusion steps, we tried to minimize this to ensure a faster and more interactive system. All reported metrics in Section 5.4 use this setting.

D.6 Computation Time Comparison

For an interactive system, it is crucial that the model can run fast enough. In Table 7, we report the model size and inference time for BMM and the CondMDI^{G} baseline. Epoch is the time during training a full epoch on AMASS. Inference time is the forward pass for a single motion that needs to be evaluated in the interactive demo. These numbers are recorded for an Intel i7-6700k CPU and Nvidia RTX 3090 GPU.

Due to the smaller representation space and the faster convergence during inference (see Figure 12), BMM can run significantly faster. This allows building the interactive setting, which we showcase in our supplementary video.

Besides the model evaluation, there is also a significant overhead in the demo due to the visualization of the curves and the Blender plugin, which is not yet optimized for speed. Thus this also accounts for some of the reduced frame rate during the recorded interactions. While the performance of the model is still away from seamless real-time rates, it does allow for a usable interactive experience. With the rapid improvements of diffusion models and inference schemes, we expect that the gap to real-time rates can be closed in the near future.

E FAILURE CASES

As for any data-driven model, very rare samples or out-of-distribution samples are harder to learn and hence lead to bad performance. As shown in the supplementary video, this can be the case when the start or end pose are very rare poses, such as the midway point of a somersault. In that case, the body is upside down and crouched together such that even the pose model struggles to create a nice-looking pose. The motion model struggles even more to create natural looking Bézier curves and even for 50% of additional constraints, sampled from ground truth, the motion exhibits artifacts. For such extreme motion, additional text conditions may also help to improve the model.

Other cases, where BMM struggles (in a less pronounced way) are the turning points in fast locomotion situations, where the root needs to turn by 180 degrees.