# NeMF: Neural Motion Fields for Kinematic Animation

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

We present an implicit neural representation to learn the spatio-temporal space of kinematic motions. Unlike previous work that represents motion as discrete sequential samples, we propose to express the vast motion space as a continuous function over time, hence the name *Neural Motion Fields (NeMF)*. Specifically, we use a neural network to learn this function for miscellaneous sets of motions, which is designed to be a generative model conditioned on a temporal coordinate $t$ and a random vector $\boldsymbol{z}$ for controlling the style. The model is then trained as a Variational Autoencoder (VAE) with motion encoders to sample the latent space. We train our model with diverse human motion dataset and quadruped dataset to prove its versatility, and finally deploy it as a generic motion prior to solve task-agnostic problems and show its superiority in different motion generation and editing applications, such as motion interpolation, in-betweening, and re-navigating.

## 1 Introduction

Motion synthesis and editing is a core problem in animation and game production, as well as in emerging applications like artificial agents. Traditional algorithms have limited capability of automatically producing convincing motions with high diversity and complexity. With the recent availability of large-scale motion capture data, more interest has shifted to deep learning-based methods. People have adapted various deep learning technologies such as Recurrent Neural Networks [28], Reinforcement Learning [23, 29] and Normalizing Flows [9] for skeletal motion generation. However, those methods are designed as an auto-regressive process that depends on its own past values and some stochastic terms. With this constraint, those methods have to sequentially predict the motion at discrete time steps. Consequently, they cannot just directly infer the motion at certain frames since they must predict the past motion first. Moreover, with this temporally asymmetric design, it is always hard for those methods to incorporate the control or editing of future frames and, thus, it is difficult to apply them to tasks like motion in-betweening.

Inspired by the recent success in neural radiance fields (NeRF) for novel view synthesis [27], we introduce *Neural Motion Fields (NeMF)* to model the spatio-temporal space of kinematic motions. Given that a motion sequence consists of different poses at different time steps, we represent a motion sequence as a continuous function $f : t \mapsto f(t)$ which parameterizes the entire sequence by the temporal coordinate $t$. This function can be approximated by a multilayer perceptron network (MLP) whose parameters are optimized by minimizing the reconstruction loss between the generated and ground truth motion.

The NeMF function defined above overfits one motion sequence. To extend it to general motion synthesis, as illustrated in Figure 1, we further introduce a random vector $\boldsymbol{z}$ as the conditioning variable to the function as $f(t; \boldsymbol{z})$, which now encodes the mapping from the normal distribution to the manifold spanned by all plausible motions. By varying $\boldsymbol{z}$, we expect $f$ to output motion with different styles, and by varying $t$, we obtain the pose at arbitrary time. We train this generative NeMF model in the form of a Variational Autoencoder (VAE), where motion sequences are fed to

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

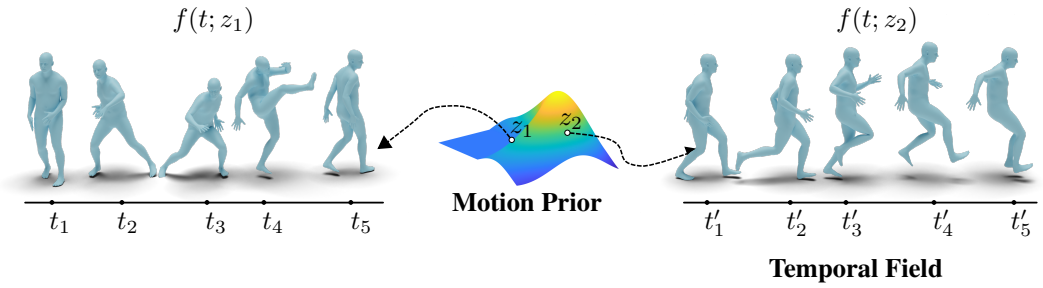



Figure 1: Overview of generative NeMF, which consists of a motion prior and a continuous temporal field. By sampling the motion prior, NeMF is able to infer a variety of motions with different styles. By sampling the temporal field, NeMF is able to generate motion at arbitrary frames.

convolutional encoders first to obtain the sampling of $\boldsymbol{z}$, and then $\boldsymbol{z}$ is passed to an MLP decoder for the reconstruction of the whole motion. To the best of our knowledge, our paper presents the first continuous and generative implicit motion representation that can sample different types of motion at an arbitrary time step.

This neural motion representation can serve as a generic motion prior to solve task-agnostic problems. Following Li et al. [20] who trained a hierarchical human motion prior and applied it in different tasks, we demonstrate utilizing a trained NeMF in various applications, including motion interpolation, motion in-betweening and motion re-navigating. We formulate those tasks as optimization problems to find the latent variable $\boldsymbol{z}$ that minimize the target energies and restore the complete plausible motion sequences. Although we did not train this NeMF model for any of these tasks specifically, it achieves equivalent or superior performance than task-specific alternatives in our experiments.

In summary, our contributions are as follows:

- We propose NeMF that represents the continuous motion field as a function over time and design a VAE architecture to train a generative NeMF.
- We validate that our model can reconstruct and synthesize motion with superior quality and diversity than state-of-the-art neural motion priors.
- We demonstrate NeMF in various offline motion editing and creation tasks.

We plan to make our implementation available upon acceptance.

## 2 Related Work

To clarify our design choices, we categorize the kinematic motion modeling into two types: time series models and space-time models. The former approach views motions as the kinematic pose evolving over time, which is often the choice for interactive applications where the future depends on unknown factors such as real-time user inputs. The latter is our approach, which is suitable for offline applications where we know what to expect for the overall motion within some time range.

### 2.1 Time Series Models

Time series models are often formulated as an auto-regressive model that predicts the future based on current and past observations. The predictions are fed into the model again to make further predictions recursively. GPDM [37] modeled such dynamics of human locomotion over time with Gaussian process. Recently people studied various deep learning-based auto-regressive architectures, including Recurrent Neural Networks [4, 26, 42], Reinforcement Learning Networks, Neural ODE [15], Transformers [1, 21, 30] and other attention models [25]. Many works demonstrated success in real-time interactive kinematic character controls [12, 33–35, 40]. The auto-regressive approach is also suitable for embracing the uncertainty of future predictions with generative models, for example, in the form of VAE as in HuMoR [31] and [23], and in the form of Normalizing Flows as in MoGlow [9].

In applications, Harvey et al. [8] used LSTM [10] with positional encoding to achieve the state-of-the-art result for the motion in-betweening task. But it is limited to a fixed in-betweening setting where some consecutive past frames need to be given as the seed. One of the key applications of our model is also motion in-betweening, but our method has no constraint on the setting and can generate the missing motion both between motion clips and between sparse keyframes.

### 2.2 Space-Time Models

Different from time series models, another approach is to directly model the spatio-temporal kinematic state itself. Such models are often used in combination with a time series model, as in [19] where the dynamic control policy is learned over the GPLVM-based spatial kinematic prior. One of our inspirations, Motion Fields [18], can be seen as a variant of this approach where the time-series dynamics are learned over the spatial kinematic model represented directly by the data and a hand-crafted distance function. More recently, space-time neural network models overcame the scalability issue in previous methods due to the availability of large-scale datasets [11, 13, 14, 38, 41]. Zhou et al. [44] demonstrated long-term motion in-betweening by learning the space-time kinematic prior without supervision using Generative Adversarial Networks (GAN) [6]. Kaufmann et al. [17] also proposed a convolutional network for motion-inbetweening. Our method NeMF is also a space-time model. Compared to the aforementioned works, NeMF has the advantage of being able to edit any frames in the motion sequence while maintaining fidelity and the source style.

### 2.3 Implicit Neural Representations

Implicit neural representations explosively gained popularity with the success of SIREN [32] and NeRF [27], which achieved state-of-the-art results in many tasks including novel view synthesis, geometry reconstruction, and solving differential equations. Among different tasks, the core idea of neural implicit representations is to build a *continuous function* that maps the spatial or temporal coordinates to any signal output at that location while maintaining high-frequency details. Inspired by their success, our key insight is to interpret time as the parameter of a *motion function* and learn the landscape of the spatio-temporal kinematic manifold as an implicit motion field parameterized by temporal coordinates.

## 3 Formulation

In this section, we will first give the definition and notation of motion representation, then present the formulation of NeMF for single motion sequence followed by the extended version of the generative NeMF for the entire motion space.

### 3.1 Motion Representation

Similar to Zhou et al. [44] and Li et al. [20], we divide the motion into two parts: *local motion*, which contains the pose of the skeleton relative to the root at time $t$, and *global motion*, which is the global translation of the root joint. Following Fussell et al. [5], we represent the local motion at time $t$ as a matrix $\mathbf{X}_t$ composed of joint positions $\mathbf{x}_t^p \in \mathbb{R}^{J\times 3}$, velocities $\dot{\mathbf{x}}_t^p \in \mathbb{R}^{J\times 3}$, rotations $\mathbf{x}_t^r \in \mathbb{R}^{J\times 6}$ in 6D rotation form [43], and angular velocities $\dot{\mathbf{x}}_t^r \in \mathbb{R}^{J\times 3}$:

$$\mathbf{X}_t = (\mathbf{x}_t^p \quad \dot{\mathbf{x}}_t^p \quad \mathbf{x}_t^r \quad \dot{\mathbf{x}}_t^p) \in \mathbb{R}^{J\times 15}, \tag{1}$$

where $J$ is the number of joints. We then factor out the root orientation $\mathbf{r}_t^o \in \mathbb{R}^6$ from local motion by multiplying the inverse of the root transform to each quantity in $\mathbf{X}_t$. In this way, all poses will be transformed to a local space with the same facing direction, thus making them easier to learn for our network. For global motion, based on previous work [20, 44], we use a neural network to predict the velocity of the root joint $\dot{\mathbf{r}}_t \in \mathbb{R}^3$ as well as the height of the root joint $\mathbf{r}_t^h \in \mathbb{R}$ from $\mathbf{X}_t$. We provide the details of the global motion predictor in the supplemental.

### 3.2 Neural Motion Fields

Unlike most other works that represent motion as a discrete sequential process, we represent motion as a continuous vector field of kinematic poses in the *temporal* domain. We define this motion field

as a function that maps temporal coordinates $t$ to local pose parameters and root orientation:

$$f : t \mapsto (\mathbf{x}_t^r, \mathbf{r}_t^o). \tag{2}$$

The function $f$ can be approximated by a neural network with parameters $\theta$ that can be optimized by minimizing the reconstruction loss between the generated and ground truth motion, where the ground truth poses can be considered as discrete samples of $f$ at integer time steps.

Similar to NeRF [27], we train an MLP with positional encoding of $t$ to fit a given motion sequence. Considering a sequence with $T$ frames, we first convert the 6D rotations $\mathbf{x}_t^r$ and $\mathbf{r}_t^o$ to rotation matrices $\mathbf{R}_t$ and $\mathbf{R}_t^o$ with the Gram-Schmidt-like process described in [43], and then compute the geodesic distance to measure the rotational difference:

$$\mathcal{L}_{\text{rot}} = \sum_{t=1}^{T} \arccos \frac{\operatorname{Tr}\left(\mathbf{R}_t (\hat{\mathbf{R}}_t)^{-1}\right) - 1}{2}, \quad \mathcal{L}_{\text{ori}} = \sum_{t=1}^{T} \arccos \frac{\operatorname{Tr}\left(\mathbf{R}_t^o (\hat{\mathbf{R}}_t^o)^{-1}\right) - 1}{2}. \tag{3}$$

We then evaluate the $L_1$ loss on local joint positions obtained from forward kinematics (FK) [36], which regularizes the skeletal topology to obtain better result [28]:

$$\mathcal{L}_{\text{pos}} = \sum_{t=1}^{T} \|\mathbf{x}_t^p - \hat{\mathbf{x}}_t^p\|_1. \tag{4}$$

The reconstruction loss is then expressed as a weighted sum of the above terms with weighting factors $\lambda_{\text{rot}}$, $\lambda_{\text{ori}}$, and $\lambda_{\text{pos}}$:

$$\mathcal{L}_{\text{rec}} = \lambda_{\text{rot}} \mathcal{L}_{\text{rot}} + \lambda_{\text{ori}} \mathcal{L}_{\text{ori}} + \lambda_{\text{pos}} \mathcal{L}_{\text{pos}}, \tag{5}$$

and we train the network parameters $\theta$ to minimize this loss function.

### 3.3 Generative Neural Motion Fields

Although the neural motion field defined above can represent a motion sequence in a compact way, it cannot generate a variety of motions. To change to a different motion sequence, the entire network needs to be re-trained from scratch, which severely limits its application. Therefore, in this section, we extend NeMF to a generative model which can represent the entire motion space instead of a specific motion sequence.

First of all, we introduce a conditioning variable $\boldsymbol{z}$ to the input of $f$, thus parameterizing the entire *spatio-temporal* kinematics space as a function:

$$f : (t, \boldsymbol{z}) \mapsto (\mathbf{x}_t^r, \mathbf{r}_t^o), \tag{6}$$

where $\boldsymbol{z}$ defines the spatial location on the manifold and $t$ controls the temporal evolution of the sequence.

We then propose a variational autoencoder to formulate Equation 6 as a latent variable model with Gaussian distribution. Based on our motion representation, the VAE contains two separate convolutional motion encoders to learn and parameterize the posterior distribution of latent variables $\boldsymbol{z}_l$ and $\boldsymbol{z}_g$, which control the local motion and root orientation respectively:

$$q_{\phi_1}(\boldsymbol{z}_l \mid \mathbf{X}) = \mathcal{N}(\boldsymbol{z}_l; \mu_\phi(\mathbf{X}), \sigma_\phi(\mathbf{X})), \quad q_{\phi_2}(\boldsymbol{z}_g \mid \mathbf{r}^o) = \mathcal{N}(\boldsymbol{z}_g; \mu_\phi(\mathbf{r}^o), \sigma_\phi(\mathbf{r}^o)), \tag{7}$$

where $\mathbf{X}$ and $\mathbf{r}^o$ are the concatenation of all $\mathbf{X}_t$ and $\mathbf{r}_t^o$ within the same sequence.

The concatenation of $\boldsymbol{z}_l$ and $\boldsymbol{z}_g$ forms the final representation of the latent variable $\boldsymbol{z}$, which then serves as the conditioning variable to the MLP decoder to produce local pose parameters $\mathbf{x}^r$ and root orientation $\mathbf{r}^o$ for each time step, thus, defining the output probability distribution $p_\theta(\mathbf{x}^r, \mathbf{r}^o \mid \boldsymbol{z}_l, \boldsymbol{z}_g)$.

In addition to the probabilistic perspective, our model can also be interpreted as learning a *motion prior* from input sequences, where each latent variable on the prior corresponds to a natural motion sequence. The MLP decoder then approximates a mapping function between the motion prior and pose space, conditioned on the latent variables $\boldsymbol{z}_l$, $\boldsymbol{z}_g$, thus, allowing the navigation in the pose space to have a consistent motion style.

During training, we consider the modified variational lower bound:

$$\begin{aligned} \log p_\theta(\mathbf{x}, \mathbf{r}^o) \geq \mathbb{E}_{q_{\phi_1}, q_{\phi_2}}[\log p_\theta(\mathbf{x}, \mathbf{r}^o \mid \boldsymbol{z}_l, \boldsymbol{z}_g)] - \mathcal{D}_{\text{KL}}(q_{\phi_1}(\boldsymbol{z}_l \mid \mathbf{X}) \parallel p(\boldsymbol{z}_l)) \\ - \mathcal{D}_{\text{KL}}(q_{\phi_2}(\boldsymbol{z}_g \mid \mathbf{r}^o) \parallel p(\boldsymbol{z}_g)), \end{aligned} \tag{8}$$

where the expectation term measures the reconstruction error of the decoder, and the KL divergence $\mathcal{D}_{\text{KL}}$ regularizes the encoders' outputs to be near $p(\boldsymbol{z}_l)$ and $p(\boldsymbol{z}_g)$, which are $\mathcal{N}(0, 1)$ in our scenario. Thus, we formulate the loss function $\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}}$ with the weight $\lambda_{\text{KL}}$ to approximate Equation 8, and the network parameters $(\phi_1, \phi_2, \theta)$ are optimized during training to minimize the loss.

### 3.4 Applications

After training the generative NeMF, we can employ it as a generic motion prior to solve different tasks via latent space optimization.

**Motion In-betweening** is a long-standing animation creation problem of generating motion in the interval between two clips or sets of keyframes. Our insight is, given a sparse set of observations, we can search in the latent space of NeMF to approximate the whole continuous underlying motion. Thus, we define the energy function as the reconstruction loss on given frames $t \in \mathcal{T}$, which measures the differences on joint rotations, joint positions, root orientations and root translations:

$$\boldsymbol{z}_l^*, \boldsymbol{z}_g^* := \underset{\boldsymbol{z}_l, \boldsymbol{z}_g}{\arg\min}\, \lambda_{\text{rot}}\mathcal{L}_{\text{rot}} + \lambda_{\text{ori}}\mathcal{L}_{\text{ori}} + \lambda_{\text{pos}}\mathcal{L}_{\text{pos}} + \lambda_{\text{trans}}\mathcal{L}_{\text{trans}}, \tag{9}$$

where $\mathcal{L}_{\text{trans}}$ weighted by $\lambda_{\text{trans}}$ evaluates the $L_1$ loss on the root joint positions $\mathbf{r}_t$ predicted from our standalone global motion predictor. In our experiments, $\boldsymbol{z}_l$ and $\boldsymbol{z}_g$ are initialized using the SLERP data to facilitate convergence.

**Motion Re-navigating** is a task where we redirect a reference motion with a new trajectory while preserving the style. More formally, the goal is to generate a motion that 1) looks similar to the given exemplar; and 2) follows the given trajectory as closely as possible. To formulate the energy function, we first borrow ideas from time-series analysis by introducing the soft dynamic time warping metric ($\mathcal{L}_{\text{sDTW}}$) [3] to measure the similarity between joint positions in the canonical frame [31]. Then, we project the predicted root joints onto the ground plane and compute the $L_1$ loss between them and the given 2D trajectory. Last but not least, an additional regularization term is introduced to ensure that the generated and reference motion have a similar angle between their forward direction $\mathbf{f}_t$ and the tangent direction $\mathbf{t}_t$ on the trajectory, where the similarity is determined by $\mathcal{L}_{\text{sDTW}}$ as well. In total, the energy function we try to minimize can be expressed as:

$$\boldsymbol{z}_l^*, \boldsymbol{z}_g^* := \underset{\boldsymbol{z}_l, \boldsymbol{z}_g}{\arg\min} \sum_{t=1}^{T} \lambda_{\text{sim}}\mathcal{L}_{\text{sDTW}}(\mathbf{x}_t^p, \hat{\mathbf{x}}_t^p) + \lambda_{\text{traj}}\|\mathbf{r}_t^{\text{proj}} - \hat{\mathbf{r}}_t^{\text{proj}}\|_1 + \lambda_{\text{angle}}\mathcal{L}_{\text{sDTW}}(\mathbf{f}_t \cdot \mathbf{t}_t, \hat{\mathbf{f}}_t \cdot \hat{\mathbf{t}}_t), \tag{10}$$

where $\lambda_{\text{sim}}$, $\lambda_{\text{traj}}$ and $\lambda_{\text{angle}}$ are three weighting factors introduced to balance different terms. In our experiments, $\boldsymbol{z}_l$ are initialized from the encoder's output, while $\boldsymbol{z}_g$ are optimized from scratch.

## 4 Experiments

We train our model on the AMASS dataset [24] for most of the experiments. After processing, we have 20 hours of human motion sequences at 30 fps for training and testing. We additionally train our model for the reconstruction experiments on a quadruped motion dataset [40], which contains 30 minutes of dog motion capture at 60 fps. Details are provided in the supplemental.

### 4.1 Sanity Test

#### 4.1.1 Motion Reconstruction

We first perform a sanity test to validate our single-motion NeMF model described in Section 3.2, where we test the reconstruction capability of NeMF with different lengths of motions.

For AMASS data, we pick 16 different motion sequences for each length and report the reconstruction errors in Table 1. For evaluation metrics, we report mean rotation error (**MRE**, $^\circ$) and mean position error (**MPE**, $cm$) for each joint, which measure the geodesic distance between joint rotations and Euclidean distance between root-aligned joint positions. For the root joint, we further report its orientation error (**MOE**, $^\circ$), which measures the geodesic distance on the root orientation. From the

results shown in Table 1, we can observe that a simple MLP with positional encoding is able to to achieve very low reconstruction error (lower than $4mm$ position error and $0.7°$ rotation error) for sequence lengths varying from 32 to 512 frames.

We further train the NeMF model on an very long (4,336 frames and $73s$) quadruped motion sequence and visualize the result in supplemental. Our predicted motion is visually almost identical to the ground truth.

Table 1: Mean reconstruction errors of single-motion NeMF for motion of different lengths. Mean rotation error (°), mean position error ($cm$), and mean orientation error (°) are reported.

| | Sequence Lengths | | | | |
|---|---|---|---|---|---|
| Metrics | 32 | 64 | 128 | 256 | 512 |
| MRE | 0.610 | 0.482 | 0.381 | 0.369 | 0.379 |
| MPE | 0.314 | 0.249 | 0.213 | 0.192 | 0.170 |
| MOE | 0.465 | 0.340 | 0.344 | 0.321 | 0.300 |

#### 4.1.2 Temporal Sampling

Unlike other motion models, NeMF is theoretically guaranteed to generate smooth motion in arbitrary frame rates. We find that the dimension of the Fourier features generated by positional encoding plays a critical role in the generalization here.

Since the maximal frequency of the Fourier features is determined by a hyperparameter $L$ as illustrated in [27], we uniformly sample $L$ from 1 to 21 and train NeMF with these different setups. For each trained model, we respectively generate 30 and 60 fps motions, and report their mean per-joint velocity as the smoothness metric in Figure 2. Although a large value of $L$ does not harm the results with the training frame rate, such a model struggles to generalize to different frame rates. However, since positional encoding is crucial to achieve satisfactory reconstruction results, we set $L = 7$ throughout our experiments to balance the trade-off. In the supplemental video, we show that the generated motion remains smooth even sampled at 240 fps.

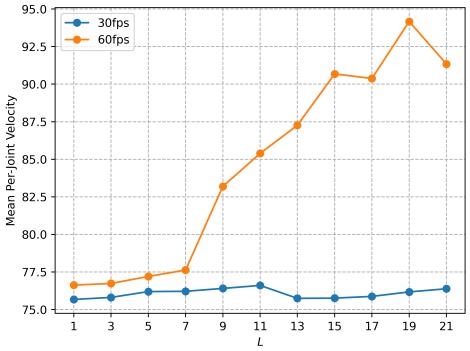



Figure 2: Mean per-joint velocity ($cm/s$) evaluated on the same motion with different $L$.

### 4.2 Generative NeMF

#### 4.2.1 Evaluation Metrics

We evaluate the reconstruction capability of our generative NeMF model through direct network inference and motion synthesis capability through latent space sampling. To better describe motion reconstruction, we further measure the Euclidean distance on the root joint's translation (**MTE**, $cm$) and global joint displacement on the final step (**FDE**, $cm$) [38]. For estimating the versatile motion quality, we introduce three metrics, namely Fréchet Inception Distance (**FID**), diversity and foot skating (**FS**, $cm$), following related papers [7, 23, 30, 40]. Generally speaking, a lower FID suggests a more natural result, a higher diversity indicates a more various result, and foot skating shows the accumulated drift of foot joints during contact. Please see our supplemental material for details.

#### 4.2.2 Ablation Study

We first take a look at the effect of the positional encoding function. By removing positional encoding, the input temporal coordinate $t$ will be passed directly to the MLP to infer the motion parameters. Results in the first row of Table 2 shows that the model without positional encoding produces higher errors in all the metrics we evaluate, thus demonstrating the necessity of positional encoding in terms of the fidelity of the results.

We further experiment on using single motion encoder instead of two separated ones to process both local motion and root orientation. In this simpler design, the root orientation and local motion are entangled in the same latent space. From the results we reported in the second row of Table 2,

though the simpler version produces a lower reconstruction error on the root orientation, it has worse performance on joint rotations, joint positions and root translations.

Table 2: Ablation study.

| | Motion Reconstruction | | | | Motion Synthesis | | |
|---|---|---|---|---|---|---|---|
| | **MRE** ↓ | **MPE** ↓ | **MOE** ↓ | **MTE** ↓ | **FID** ↓ | **Diversity** ↑ | **FS** ↓ |
| No Positional Encoding | 6.413 | 2.952 | 6.027 | 33.524 | 2.146 | 2.783 | 0.612 |
| Single Motion Encoder | 8.661 | 4.503 | **5.982** | 35.391 | 2.185 | **2.806** | 0.818 |
| Full Model | **5.988** | **2.870** | 6.157 | **29.692** | **2.073** | 2.774 | **0.573** |

#### 4.2.3 Latent Space Sampling

In our model, we can navigate in the latent space to control the motion styles and synthesize novel motion. To examine the smoothness of the latent space and see whether our model can blend different styles of motion in sequence-level, we linearly interpolate $\boldsymbol{z}$ from two existing motion clips and infer the motion sequences. Previous works like [39] demonstrate motion interpolation between similar motion patterns, such as from walking to zombie styled walking. However, we show in the supplemental videos that our model can blend the high-level perceptual style between much harder cases like jumping and punching.

Besides, since we disentangle latent space for root orientation and local motion, we also experiment to combine $\boldsymbol{z_l}$ and $\boldsymbol{z_g}$ from different motions. Through this, we can create interesting editing results like canceling the spinning motion of a pirouette jump (see supplemental).

#### 4.2.4 Comparison with other Generative Models

We compare our method with other deep learning-based motion priors on motion reconstruction and synthesis in Table 3. We select HuMoR [31] and HM-VAE [20] to represent the "time series model" and "space-time model" respectively.

Table 3: Comparison of NeMF with other generative motion models.

| | Motion Reconstruction | | | | | Motion Synthesis | | |
|---|---|---|---|---|---|---|---|---|
| | **MRE** ↓ | **MPE** ↓ | **MOE** ↓ | **MTE** ↓ | **FDE** ↓ | **FID** ↓ | **Diversity** ↑ | **FS** ↓ |
| HuMoR [31] | 13.008 | 9.071 | 17.097 | **23.882** | 62.756 | 8.687 | 1.741 | 0.904 |
| HM-VAE [20] | 10.258 | 7.686 | 13.054 | 90.924 | 137.218 | 7.998 | 2.002 | 0.690 |
| Ours | **5.988** | **2.870** | **6.157** | 29.692 | **42.985** | **6.508** | **2.118** | **0.566** |

Although HuMoR can generate convincing motions, we observe two shortcomings of HuMoR: 1) in motion reconstruction, HuMoR's results will gradually diverge due to its auto-regressive prediction. From the quantitative results in Table 3 and qualitative results in Figure 3, divergence can be observed in both the large FDE and global translation difference respectively. 2) In motion synthesis, HuMoR tends to favor common motions like walking when inferring long sequences, thus yielding a relatively high FID and low diversity. As for our method, we can get rid of these artifacts since we handle the entire sequence at once instead of predicting the motion frame by frame.

Compared to HuMoR and our model, it is obvious that HM-VAE models an over-smoothed latent space, thus filtering high-frequency details in both the reconstructed and synthesized motions, making their results worse in both the reconstruction and synthesis task. In short, our model outperforms these state-of-the-art methods in most of the metrics reported in Table 3, as well as achieving the closest match of the ground truth motion visualized in Figure 3.

### 4.3 In-betweening Tasks

We compare our method with traditional and deep learning-based motion in-betweening methods. For quantitative evaluation, we use the FID and foot skating metrics only, since reconstruction errors cannot well reflect the quality of motion for highly-varied plausible solutions.

HM-VAE [20] HuMoR [31] Ours

Figure 3: Comparison of motion reconstruction with other generative motion models. Predicted motions are overlapped on top of the ground truth motion (yellow).

Table 4: Motion clips in-betweening.

| | FID | | |
|---|---|---|---|
| Length (frames) | 10 | 20 | 30 |
| SLERP | 0.027 | 0.170 | 0.455 |
| Inertialization [2] | 0.025 | 0.184 | 0.496 |
| RMI [8] | 0.264 | 0.362 | 0.609 |
| HM-VAE [20] | 0.137 | 0.387 | 0.675 |
| Ours | **0.024** | **0.141** | **0.365** |
| | **Foot Skating** | | |
| SLERP | 1.193 | 1.200 | 1.023 |
| Inertialization [2] | 1.237 | 1.234 | 1.151 |
| RMI [8] | 1.629 | 1.833 | 1.643 |
| HM-VAE [20] | 0.862 | 0.835 | 0.832 |
| Ours | **0.646** | **0.697** | **0.660** |

Table 5: Sparse keyframe in-betweening.

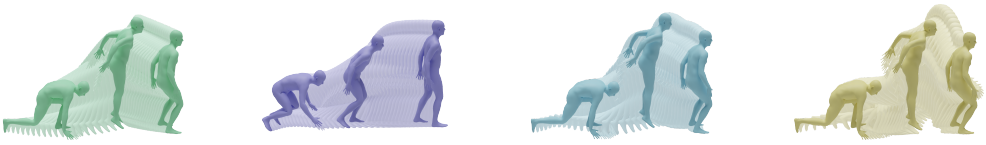
| | FID | | | |
|---|---|---|---|---|
| Length (frames) | 5 | 10 | 15 | 20 |
| SLERP | **0.032** | 0.305 | 0.713 | 1.191 |
| HM-VAE [20] | 2.300 | 2.370 | 2.434 | 2.423 |
| Ours | 0.085 | **0.302** | **0.612** | **0.879** |
| | **Foot Skating** | | | |
| SLERP | 0.868 | 1.224 | 1.278 | 1.193 |
| HM-VAE [20] | 0.892 | 0.837 | 0.841 | **0.751** |
| Ours | **0.719** | **0.826** | **0.837** | 0.804 |

**Motion Clips In-betweening.** In this task, we aim at generating motion in the interval from 10 frames (0.33 second) to 30 frames (1 second) between two clips. We compare with traditional local interpolation methods including SLERP and Inertialization [2], and deep learning-based methods including Robust Motion In-betweening (RMI) [8] and HM-VAE [20], the task-generic deep learning model. In the results reported in Table 4, our method outperforms all other alternatives quantitatively. We further show in Figure 4 that our model can be applied to real captured motions in AIST++ [22] to produce a natural one-second transition between two dancing footages.

Figure 4: Generating the transition (cyan) between AIST++ motion clips (yellow).

**Sparse Keyframe In-betweening.** In this task, a set of sparse keyframe skeleton poses are given every 5, 10, 15 or 20 frames. We compare our method with SLERP and HM-VAE, and report the quantitative results in Table 5. Both RMI and Inertialization are not applicable here because they both require multiple frames in the beginning.

From Table 5, SLERP has the best FID score for 5 frames interval, but becomes worse with increased interval length. Compared to SLERP and HM-VAE, ours have much better quantitative and qualitative results, especially for long intervals (Figure 5). In the onionskin images, though SLERP's result looks similar to ours, our result has more natural high-frequency details. HM-VAE fails to reconstruct the poses at the keyframes and generates over-smoothed results. Our result, although different from the ground truth, is visually plausible as in the supplemental video.

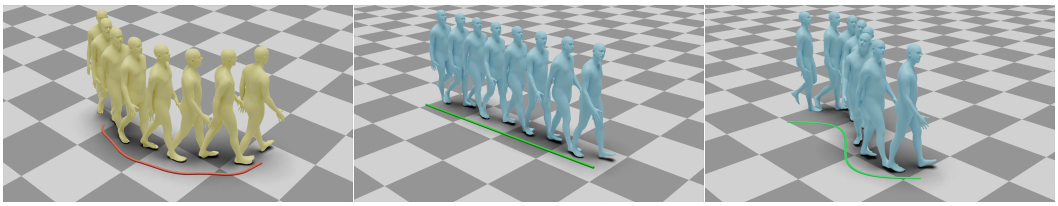


SLERP HM-VAE [20] Ours Ground Truth

Figure 5: Comparison of sparse keyframe in-betweening. The translucent in-between poses are generated from the opaque keyframes given every 20 frames.

### 4.4 Re-navigating Tasks

In this task, we experiment on redirecting the reference motion to different synthetic trajectories. Qualitative results are reported in Figure 6. The original motion style is well preserved in the results, and the character has a natural orientation while walking along the trajectory.



Reference motion Straight line Sinusoidal curve

Figure 6: Motion re-navigating from a reference walking motion (left) to a straight line (middle) and sinusoidal curve (right).

## 5 Conclusion

We propose an implicit neural motion representation that defines a continuous motion field over style and time. We design it to be a generative model for the whole motion space with a Gaussian prior. We design and test different network architectures and use the trained generative model as a motion prior for solving different tasks like motion interpolation, motion in-betweening and motion re-navigating.

### 5.1 Limitation and Future Work

In this paper, since we train the generative NeMF model in the form of a VAE, it has the limitation of not always giving sharp results when sampling the latent space. A promising direction is to design other generative architectures such as GAN, Normalizing Flows, and Diffusion Models for NeMF.

Besides, the optimization problems we formulated are analogous to the latent space optimization problems of StyleGAN [16]. How to transfer the algorithm of those successful applications designed for StyleGAN to our NeMF setup in the motion domain is another interesting direction to explore.

To keep our model task-agnostic, we chose to solve motion tasks by optimizing $\boldsymbol{z}$. But for practical applications, task-specific inference models may be preferred for performance consideration. To have an end-to-end inference model, a solution is to design and train the encoder to directly predict $\boldsymbol{z}$ with different input settings.

### 5.2 Broader Impact

Our model can be applied for kinematic motion creation in animation and game production. Due to the limitation of training data diversity, it cannot guarantee to generate uncommon motion and avoid generating offensive motion attributed to certain culture.

# References


[1] Emre Aksan, Manuel Kaufmann, Peng Cao, and Otmar Hilliges. A spatio-temporal transformer for 3d human motion prediction. In *2021 International Conference on 3D Vision (3DV)*, pages 565–574. IEEE, 2021.

[2] David Bollo. Inertialization: High-Performance animation transitions in 'gears of war'. *Proc. of GDC 2018*, 2018.

[3] Marco Cuturi and Mathieu Blondel. Soft-dtw: A differentiable loss function for time-series. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 894–903. JMLR.org, 2017.

[4] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE international conference on computer vision*, pages 4346–4354, 2015.

[5] Levi Fussell, Kevin Bergamin, and Daniel Holden. Supertrack: Motion tracking for physically simulated characters using supervised learning. *ACM Trans. Graph.*, 40(6), dec 2021.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[7] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20)*, 2020.

[8] Félix G. Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion in-betweening. *ACM Trans. Graph.*, 39(4), jul 2020.

[9] Gustav Eje Henter, Simon Alexanderson, and Jonas Beskow. Moglow: Probabilistic and controllable motion synthesis using normalising flows. *ACM Trans. Graph.*, 39(6), nov 2020.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, nov 1997.

[11] Daniel Holden, Oussama Kanoun, Maksym Perepichka, and Tiberiu Popa. Learned motion matching. *ACM Trans. Graph.*, 39(4):53:1–53:12, July 2020.

[12] Daniel Holden, Taku Komura, and Jun Saito. Phase-functioned neu-ral networks for character control. *ACM Trans. Graph*, 36, 2017.

[13] Daniel Holden, Jun Saito, and Taku Komura. A deep learning framework for character motion synthesis and editing. *ACM Trans. Graph.*, 35(4):1–11, July 2016.

[14] Daniel Holden, Jun Saito, Taku Komura, and Thomas Joyce. Learning motion manifolds with convolutional autoencoders. In *SIGGRAPH Asia 2015 Technical Briefs*, number Article 18 in SA '15, pages 1–4, New York, NY, USA, November 2015. Association for Computing Machinery.

[15] Boyan Jiang, Yinda Zhang, Xingkui Wei, Xiangyang Xue, and Yanwei Fu. Learning compositional representation for 4d captures with neural ode. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5340–5350, 2021.

[16] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[17] Manuel Kaufmann, Emre Aksan, Jie Song, Fabrizio Pece, Remo Ziegler, and Otmar Hilliges. Convolutional autoencoders for human motion infilling. In *2020 International Conference on 3D Vision (3DV)*, pages 918–927. IEEE, 2020.

[18] Yongjoon Lee, Kevin Wampler, Gilbert Bernstein, Jovan Popović, and Zoran Popović. Motion fields for interactive character locomotion. In *ACM SIGGRAPH Asia 2010 papers*, number Article 138 in SIGGRAPH ASIA '10, pages 1–8, New York, NY, USA, December 2010. Association for Computing Machinery.

[19] Sergey Levine, Jack M. Wang, Alexis Haraux, Zoran Popović, and Vladlen Koltun. Continuous character control with low-dimensional embeddings. *ACM Trans. Graph.*, 31(4), July 2012.

[20] Jiaman Li, Ruben Villegas, Duygu Ceylan, Jimei Yang, Zhengfei Kuang, Hao Li, and Yajie Zhao. Task-generic hierarchical human motion prior using vaes. 2021.

[21] Jiaman Li, Yihang Yin, Hang Chu, Yi Zhou, Tingwu Wang, Sanja Fidler, and Hao Li. Learning to generate diverse dance motions with transformer. *arXiv preprint arXiv:2008.08171*, 2020.

[22] Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++, 2021.

[23] Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel Van De Panne. Character controllers using motion VAEs. *ACM Trans. Graph.*, 39(4):40:1–40:12, July 2020.

[24] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pages 5442–5451, October 2019.

[25] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. History repeats itself: Human motion prediction via motion attention. In *European Conference on Computer Vision*, pages 474–489. Springer, 2020.

[26] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2891–2900, 2017.

[27] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[28] Dario Pavllo, David Grangier, and Michael Auli. Quaternet: A quaternion-based recurrent model for human motion. In *British Machine Vision Conference (BMVC)*, 2018.

[29] Xue Bin Peng, Angjoo Kanazawa, Jitendra Malik, Pieter Abbeel, and Sergey Levine. Sfv: Reinforcement learning of physical skills from videos. *ACM Transactions On Graphics (TOG)*, 37(6):1–14, 2018.

[30] Mathis Petrovich, Michael J. Black, and Gül Varol. Action-conditioned 3D human motion synthesis with transformer VAE. In *International Conference on Computer Vision (ICCV)*, pages 10985–10995, October 2021.

[31] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J. Guibas. Humor: 3d human motion model for robust pose estimation. In *International Conference on Computer Vision (ICCV)*, 2021.

[32] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. NeurIPS*, 2020.

[33] Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene interactions. *ACM Trans. Graph.*, 38(6):1–14, November 2019.

[34] Sebastian Starke, Yiwei Zhao, Taku Komura, and Kazi Zaman. Local motion phases for learning multi-contact character movements. *ACM Trans. Graph.*, 39(4):54:1–54:13, July 2020.

[35] Sebastian Starke, Yiwei Zhao, Fabio Zinno, and Taku Komura. Neural animation layering for synthesizing martial arts movements. *ACM Trans. Graph.*, 40(4):1–16, July 2021.

[36] Ruben Villegas, Jimei Yang, Duygu Ceylan, and Honglak Lee. Neural kinematic networks for unsupervised motion retargetting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[37] Jack M Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(2):283–298, February 2008.

[38] Ye Yuan and Kris Kitani. Dlow: Diversifying latent flows for diverse human motion prediction. In *European Conference on Computer Vision*, pages 346–364. Springer, 2020.

[39] M Ersin Yumer and Niloy J Mitra. Spectral style transfer for human motion between independent actions. *ACM Transactions on Graphics (TOG)*, 35(4):1–8, 2016.

[40] He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. Mode-adaptive neural networks for quadruped motion control. *ACM Trans. Graph.*, 37(4), jul 2018.

[41] Siwei Zhang, Yan Zhang, Federica Bogo, Marc Pollefeys, and Siyu Tang. Learning motion priors for 4d human body capture in 3d scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11343–11353, 2021.

[42] Yan Zhang, Michael J Black, and Siyu Tang. We are more than our joints: Predicting how 3d bodies move. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3372–3382, 2021.

[43] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019.

[44] Yi Zhou, Jingwan Lu, Connelly Barnes, Jimei Yang, Sitao Xiang, et al. Generative tweening: Long-term inbetweening of 3d human motions. *arXiv preprint arXiv:2005.08891*, 2020.

## Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.1.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 5.2.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See supplemental and the beginning of Section 4.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We fixed random seeds to 0 throughout our experiments.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See supplemental.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Our model is trained on the AMASS [24] dataset and a quadruped dataset from [40], we also use AIST++ [22] to test motion in-betweening.
   (b) Did you mention the license of the assets? [Yes] See supplemental.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]