MotionMAE: A Kinematics-and-Phase-Aware Motion Masked Autoencoder for Motion Generation

Paper ID 4141

ABSTRACT

Recently, masked autoencoding has advanced NLP and computer vision tasks. In this paper, we explore this learning paradigm for 3D motion data to benefit motion generation tasks and propose a Motion Masked AutoEncoder (MotionMAE). Our idea is to randomly mask part of the motion data and reconstruct the missing motion. To this end, we first propose a spatial-temporal residual motion encoder to learn the complex spatial-temporal dependencies over diverse action types. Specifically, we use a spatial-temporal prototype operation to enhance the generalization and get more detailed information by residual blocks. Second, we design a kinematics-and-phase-aware decoder to reconstruct the masked information, which helps to keep motion spatial kinematics and motion regularity. Extensive experiments on the motion in-betweening task (LAFAN1) and the motion prediction task (Human 3.6M) demonstrate that MotionMAE achieves state-of-the-art (SOTA) performance under multiple evaluation settings. For example, on LAFAN1, it is more than 20% better than the previous SOTA.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**; • **Applied computing** → **Media arts**.

KEYWORDS

human motion, masked autoencoder, motion generation, phase aware

ACM Reference Format:
Paper ID 4141. 2023. MotionMAE: A Kinematics-and-Phase-Aware Motion Masked Autoencoder for Motion Generation. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (MM'22).* ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXXX

Recently, inspired by the successful masked autoencoding strategy in NLP [6, 18], some computer vision works [3, 15, 31] also apply such a strategy to images or videos by reconstructing the masked tokens and obtain impressive results. Instead of predicting the tokens, MAE [15] proposes to directly reconstruct raw pixels with impressive performance achieved, which provides a new self-supervised pre-training paradigm. VideoMAE [31] extends MAE to reconstruct masked video patches.

In this paper, we explore the masked autoencoding strategy on 3D human motion data. However, the extension is not trivial with three challenges. First, different from the dense and temporally redundant video data, skeleton-based motion data is sparse and changes quickly between time steps. A single frame from a video clip may contain the most information of this clip, but we cannot know what a motion sequence is when only a single pose is given. Therefore, directly applying previous image/video encoders to motion data may not obtain discriminative-enough representations. Second, the spatial joints of the skeleton-based motion data have special connections, and they need to meet kinematic constraints, which are hard to learn directly from the data. If not processed properly, errors will be accumulated from parent joints to children joints. Third, human motions (for example, walking and running) have their own motion characteristics [29], such as the relationship between the locations of different body parts, and the location of a certain body part at a certain moment. However, the existing deep learning-based methods for motion generation tasks often capture the temporal dependency by directly mining the data without considering the temporal motion characteristics, leading to the problem of over-smoothing or freezing [20, 33].

Based on the above analysis, we present MotionMAE, a masked autoencoder (MAE) for 3D human motion modeling. Unlike VideoMAE [31], which reconstructs the patches of a video, we choose to recover the information of a motion sample. In detail, we achieve the reconstruction at two levels, spatial level and temporal level. For the spatial level, we predict the joint belonging to a motion frame and make them satisfy human kinematics. For the temporal level, we regress the features on the phase manifold to keep the motion temporal regularity. Moreover, we propose a special encoder to get more discriminative information. Taking advantage of the encoder and the two reconstruction objectives, our model can learn discriminative motion representation in a self-supervised manner from unlabeled 3D human motion data. MotionMAE can correctly predict the masked joints and infer holistic reconstructions. It is pre-trained and fine-tuned on LAFAN1 and Human3.6M. The experimental results and comprehensive ablations demonstrate its effectiveness. The main contributions of this work are as follows:

- We design a spatial-temporal residual encoder to learn the entire spatial-temporal dependence of the motion, which helps to enhance the scalability of our network across multi-type motions.

MM'22, October 29–November 03, 2023, Ottawa, Canada
© 2023 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06...$15.00
https://doi.org/XXXXXXX.XXXXXXX

- We adapt the masked autoencoding-based pre-training strategy to 3D human motion modeling, where a kinematics-and-phase aware decoder is designed, with the kinematics constraints and temporal regularity.
- Our model achieves new state-of-the-art on the human motion in-betweening and motion prediction tasks. We also conduct extensive ablations to better understand the model.

1 RELATED WORK

Our main goal is to pre-train representations for human motions with a masked autoencoder, and demonstrate its effectiveness in downstream motion generation applications. Here we will review the most relevant works on masked encoding and motion generation.

Masked Encoding. Masked encoding is first demonstrated successfully in NLP [8], the idea is to mask part of the input sequence and train deep models to estimate the masked content. Many works [6, 26, 27] show that the pre-trained representations generated with this method benefit various downstream NLP applications.

Due to its effectiveness, this idea quickly draws much attention from the computer vision community. For example, iGPT [7] proposes to operate on sequences of pixels and predict the unknown pixels, while ViT [9] reshapes an image into a sequence of patches, and then adapts a Transformer to process the patches. Recently, BEiT [3] tokenizes the input images into discrete visual tokens first and then recovers the masked discrete tokens, while MAE [15] encourages the model to reconstruct those missing pixels directly without the tokenization. VideoMAE [31] further extends MAE from image to video by reconstructing spatial-temporal masked patches with a high mask ratio and achieves SOTA results on several downstream tasks. For 3D data analysis, PointBERT [32] generalizes the idea of masked encoding into 3D point clouds by devising a masked point reconstruction task to pre-train a Transformer.

In this paper, we extend MAE [15] to human motion modeling. The reconstruction objectives include kinematics constraints and motion regularity.

Motion Generation. Motion generation refers to generating plausible motions from given inputs, such as previous poses [24, 30], previous and future poses [10, 14, 20], and semantic information [1, 12, 16]. These methods are sometimes termed as motion prediction, motion in-betweening (MIB) and motion synthesis, respectively. We concentrate on motion generation solely from motion modality.

Some studies adopt Recurrent Neural Networks (RNNs) to address motion prediction problems. Fragkiadaki et al. [11] propose a recurrent encoder-decoder model by incorporating nonlinear encoder and decoder networks before and after recurrent layers. To make reliable future predictions, Tang et al. [30] model motion context by summarizing the historical human motion with respect to the current prediction within a recurrent prediction framework. Recently, Martinez et al. [25] propose a seq2seq Transformer encoder and decoder model with Graph Convolutional Networks (GCNs) before and after the Transformer block. Zhong et al. [33] propose a Spatio-Temporal Gating-Adjacency GCN (GAGCN) to learn the complex spatial-temporal dependencies over diverse action types. Bouazizi et al. [5] learn spatial-temporal 3D body pose dependencies by sequentially mixing both modalities.

Motion in-betweening is a task that generates realistic motion from the given past and future poses. Harvey et al. [14] demonstrate studio-quality MIB results based on their motion transition network [13]. Also, they leverage the least-squares generative adversarial network (LSGAN) [23] to make generated motions more natural. Kaufmann et al. [19] employ a convolutional autoencoder by representing motion data in a matrix that can be interpreted as an image and show that a non-autoregressive method could produce comparable results in MIB without degrading visual results. Kim et al. [20] propose a controllable motion generation method on top of the MIB framework. Duan et al. [10] design a trainable mixture embedding module that models temporal information and encodes different key-frame combinations in a unified form.

Most of the previous works model the temporal and spatial information between individual joints, which cannot get discriminative-enough feature differences for different motion sequences. And we design a spatial-temporal residual encoder to learn the motion representation by obtaining the entire spatial-temporal dependence of the motion.

2 OUR METHOD

In this section, we first formulate the human motion generation problem, and then describe the details of our MotionMAE framework (Fig. 1), which consists of a spatial-temporal residual encoder and a kinematics-and-phase-aware decoder.

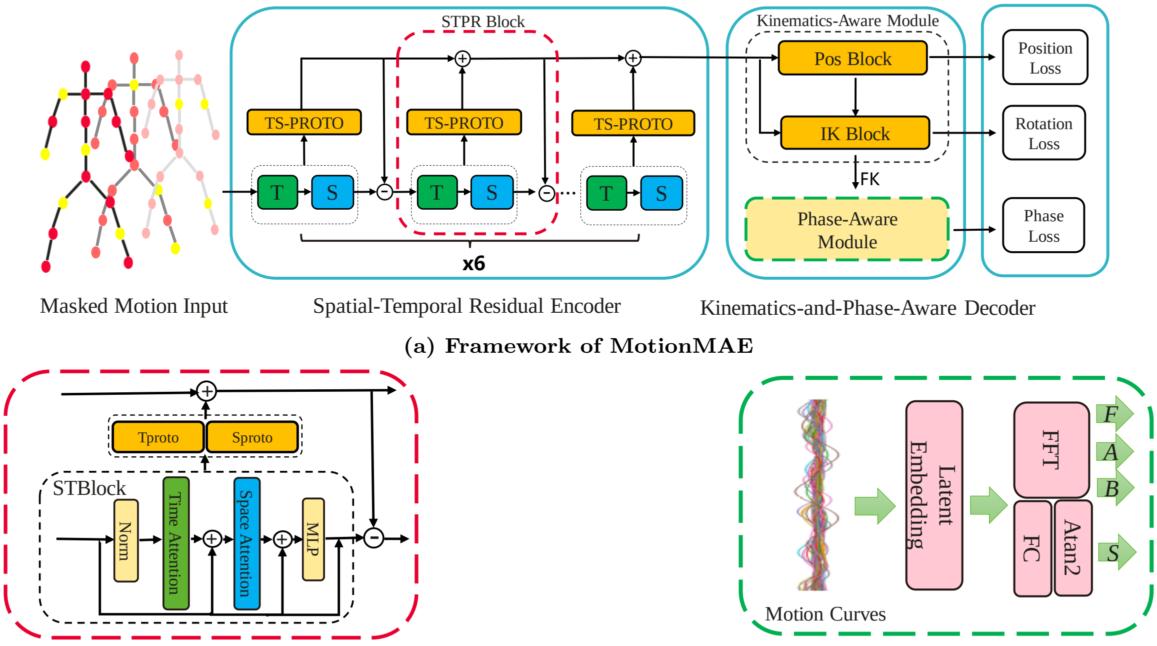
2.1 Problem Formulation

In this study, we represent a joint position vector $p \in \mathrm{R}^3$ in Euclidean space and adopt the 6D representation [34] $q \in \mathrm{R}^6$ as the rotational representation of each joint. Positions and rotations at specific key-pose are concatenated and embedded into a feature space of dimensionality F.

Skeleton-based human motion generation is to predict the future motion sequence given the historical (prediction task) or both the historical and the future (MIB task). Let the given historical motion sequence be $X_{1:T_1} = \{x_1, x_2, ..., x_{T_1}\}$ with T_1 frames, and the given future motion sequence $X_{T_1+t:T_1+t+T_2} = \{x_{T_1+t+1}, x_{T_1+t+2}, ..., x_{T_1+t+T_2}\}$ with T_2 frames (if it exists). Then, the generated motion sequence of the t time steps is $X_{T_1+1:T_1+t} = \{x_{T_1+1}, x_{T_1+2}, ..., x_{T_1+t}\}$, where x is usually represented as the 3D coordinates and/or joint angles of the J body joints.

2.2 Spatial-Temporal Residual Encoder

ST-Embeddings. Following most works in Transformers [6, 27], we first embed the motion sample into an embedding



(a) Framework of MotionMAE

(b) STPR Block

(c) Phase-Aware Module

Figure 1: Overview of the proposed MotionMAE network. We use the spatial-temporal residual encoder to learn the spatial-temporal dependence of the motion sequence, and then use the kinematics-and-phase-aware decoder to generation motion. We first randomly mask some joints in the motion sequence and feed it to the encoder, obtaining spatial-temporal features. Then, the kinematics module gets spatial human motion kinematics constraint and the phase module gets human motion temporal regularity.

space. In our work, the embeddings of the input sequence are $\bar{E}_t^j \in R^F, j = 1, ..., J, t = 1, ..., T$. Each of the $\bar{E}_t^j$ is added with a learnable spatial (position) embedding and a learnable temporal embedding. For simplicity, we still use $\bar{E}_t^j$ to represent the embedding after the addition. We randomly mask some joints in the sequence, and represent them as learnable embeddings $\widehat{E}_t^j \in R^F, j \in \{1, ..., J\}, t \in \{1, ..., T\}$.

Encoder. For motion data, some works [10, 20] adapt the ViT backbone and flatten one pose into a 1D vector, which can get the benefit from the Transformer architecture. To better get the spatial information, space-time attention is employed in motion generation [2]. However, the skeleton-based motion data is sparse on the spatial domain (only tens of joints). And on the temporal domain, the motion changes similarly over time slices. Therefore, it is difficult to distinguish motion sequences with individual frames or individual joints.

Thus, we specially design an encoder consisting of two blocks to get better motion representation. The first extracts temporal and spatial information adapting the joint space-time attention Transformer in [4]. The second implements our spatial-temporal-prototype-residual (STPR) block, which is described in detail below.

Let the input sequence after embedding be $E = \{E_t^j\}_{t=1,...,T}^{j=1,...,J}$, with E_t^j being $\bar{E}_t^j$ or $\widehat{E}_t^j$. The output $G = \{G_t^j\}_{t=1,...,T}^{j=1,...,J}$ $= STBlock(E)$ of the space-time Transformer block (STBlock) (see Fig. 1) is fed into a $Tproto$ layer and a $Sproto$ layer, which calculate the means along the temporal and spatial dimensions, respectively:

$$Tproto(G^j) = \frac{1}{T} \sum\nolimits_{t=0}^{T} G_t^j, \tag{1}$$

$$Sproto(G_t) = \frac{1}{J} \sum\nolimits_{j=0}^{J} G_t^j. \tag{2}$$

Since the motion is hard to be represented with individual frames or individual joints, the $Tproto$ and $Sproto$ layers (prototype) respectively combine the joints in the temporal and spatial domains into the entire representation. Such layers can better aggregate the features of a motion sequence and maintain the feature difference from other sequences.

2.3 Kinematics-and-Phase-Aware Decoder

Kinematics-Aware Module. Different from images and videos, the spatial joints of the skeleton-based motion data need to meet kinematic constraints which means the joint positions are obtained by rotating the joints according to their parent-child relationship and bone lengths. Therefore, it is crucial that the network can learn these constraints to predict suitable joint locations for the motion generation task. Thus, we design a kinematics-aware module as follows.

The kinematics-aware module has a position block and an inverse kinematics (IK) block. Both are three fully-connected layers. The position block accepts the encoded pose embeddings and produces 3D position predictions of all joints. Since the position block produces predictions without the skeleton constraints, its predictions do not respect bone lengths. The inverse kinematics block is designed to accept the input consisting of the concatenation of the encoder-generated pose embeddings and the output of the position block. Effectively, the initial poses generated by the position block are used to condition the inverse kinematics block. Then, the forward kinematics (FK) operation [14] applies skeleton kinematic equations to the local joint rotations and the global root position to get all the global joint information. The L1 loss is used for both the output of the position block as the position loss and the output of the inverse kinematics (IK) block as the rotation loss (see Fig. 1).

This module constructs human kinematics constraints by designing the position block and the inverse kinematics block in the reconstruction process, so that the network can learn better spatial dependencies of the skeleton-based human body, thereby improving the results of motion generation.

Phase-Aware Module. The existing deep learning-based methods for motion generation tasks often face the problem of over-smoothing or freezing. Motion data such as walking and running have their own laws of motion, including the relationship between the locations of different body parts, and the location of a certain body part at a certain moment. Existing methods like RNN-based models or Transformer-based models capture the temporal dependency by directly mining the data, and it may be hard to model temporal motion characteristics (motion regularity). In the temporal domain, a motion sequence can be regarded as a transition among slices of periodic and non-periodic motion. In the spatial domain, it can be seen as a combination of periodic motions of different body parts. Recently, Starke et al. [29] propose to decompose the motion into a combination of multiple periodic signals, which constructs a phase space to model the periodicity of the motion itself.

In [29], the phase manifolds are learned by a periodic autoencoder (PAE). We borrow this idea for our task. As shown in Fig. 1, in this module, the input motion curves are calculated the same as [29]. Then, a differentiable Fast Fourier Transform (FFT) layer is used to compute amplitude (A), frequency (F), and Bias (B) by the latent curves learned by a lower-dimensional embedding. To obtain the timing parameter, a separate fully-connected (FC) layer is learned for each latent curve to predict only the signed phase shifts (S) at the central frame. From the learned parameters A, F, B and S, along with the known time window τ, it is possible to reconstruct a parameterized latent space L in form of multiply periodic functions using the parameterization function:

$$L = A\sin(2\pi(F\tau - S)) + B. \tag{3}$$

Then, L is decoded to map back to the input motion curves while learning on the training set. After learning of this module on a motion dataset, the periodic parameters A, F, B, and S, can be computed per frame. A sample at frame t on the phase manifold P is computed by:

$$P^t = \{A^t\sin(2\pi S^t), A^t\cos(2\pi S^t)\}. \tag{4}$$

The features (P^{gt}) on P of the training data are extracted for all frames by Eq. (4) as the supervision information. Then, the motion curves after FK are fed into this phase-aware module during the training of MotionMAE, which is supervised by P^{gt} with the MSE loss (Phase Loss in Fig. 1). Since the amplitude and frequency of each phase channel can alter over time, the phase-aware module can encode non-periodic motion as well as periodic motion. Given a motion sequence, the periodic features smoothly shift over the phase manifold. The features on P well describe the timing of the frames within the input motion and correctly align motions with time. Those alignments help MotionMAE to model the laws of motion (motion regularity) and benefit the motion generation tasks.

2.4 Masking Strategy

Masking Strategy and Masking Ratio. Each joint in each frame is masked according to a pre-defined probability. Given a motion sequence, we find that this unstructured space-time masking is more effective than structured masking strategies, such as space-only, time-only, and block-wise masking. As neighboring joints in space or in time are coherent, with a high probability, space-only or time-only masking retains too little information and yields a difficult pre-training task. For masking ration, the optimal masking ratio is related to the information redundancy of different task data. With unstructured random masking, BERT [8] uses a masking ratio of 15% for language and MAE [15] uses a ratio of 75% for images, suggesting that images are more information-redundant than language. The optimal masking ratio for videos is 90% in VideoMAE [31]. Natural videos are more information redundant than images because of their temporal coherence. Our empirical results on skeleton-based motion show that the optimal masking ratio is 50%.

3 EXPERIMENTS

3.1 Datasets and Metrics

The datasets used in our experiments are LAFAN1 [14] for the motion in-betweening task and Human 3.6M [17] for the motion prediction task.

LAFAN1. LAFAN1 is a high-quality motion dataset. It contains 15 actions such as walking, dancing, fighting, jumping, etc, with approximately 4.6 hours long from five actors.

Human 3.6M. Human 3.6M is the most popular dataset in the research of motion prediction. It has 3.6 million 3D poses, consisting of 15 motion categories from 7 subjects. We down-sample the frame rate to 25Hz. Following [22], we use subjects 1, 6, 7, 8, and 9 for training, subject 11 for validation and subject 5 for testing.

	L2Q ↓			L2P ↓			NPSS ↓		
Length	5	15	30	5	15	30	5	15	30
Zero Velocity	0.56	1.10	1.51	1.52	3.69	6.60	0.0053	0.0522	0.2318
SLERP Interpolation	0.22	0.62	0.98	0.37	1.25	2.32	0.0023	0.0391	0.2012
TG [14]	0.17	0.42	0.69	0.23	0.65	1.28	0.0022	0.0258	0.1328
HHM-VAE [10]	0.24	0.54	0.94	-	-	-	-	-	-
SSMCT [14]	0.14	0.36	0.61	0.22	0.56	1.10	0.0016	0.0234	0.1222
CMIB [20]	0.14	0.35	0.59	0.24	0.58	1.19	-	-	-
MotionMAE	**0.10**	**0.29**	**0.51**	**0.11**	**0.42**	**0.84**	**0.0013**	**0.0208**	**0.1157**

Table 1: Results on the motion in-betweening task on the LAFAN1 dataset.

	MAE ↓						MPJPE ↓					
Length	80	160	320	400	560	1000	80	160	320	400	560	1000
Res-GRU [24]	0.36	0.67	1.02	1.15	-	-	25.3	46.8	78.2	89.9	108.2	139.4
Conseq2seq [21]	0.38	0.68	1.01	1.13	1.35	1.82	16.6	33.5	62.0	73.5	92.1	126.8
HRI [22]	0.27	0.52	0.82	0.94	1.14	1.57	10.4	22.1	46.5	57.5	76.6	112.3
STSGCN [28]	0.24	0.39	0.59	0.66	0.79	1.09	10.2	17.3	33.5	38.9	51.7	77.3
GAGCN [33]	0.24	0.38	0.54	0.65	0.74	1.02	10.1	16.9	32.5	38.5	50.0	72.9
Mixture [5]	0.20	0.34	0.55	0.63	0.78	1.08	9.0	13.2	26.9	33.6	-	71.6
MotionMAE	**0.18**	**0.31**	**0.55**	**0.60**	**0.71**	**1.03**	**8.5**	**11.2**	**25.1**	**31.6**	**43.8**	**70.2**

Table 2: Results on the motion prediction task on the Human 3.6M dataset.

Metrics. Our model is trained on both 3D coordinate representation and angle-based representation. Thus, we evaluate the results on both the 3D coordinate errors and the angle errors. Following previous work [14, 33], we use L2Q, L2P and NPSS for motion in-betweening same as [14], and MAE and MPJPE for motion prediction same as [33].

3.2 Training Setting

MotionMAE is implemented in PyTorch using the Adam optimizer. The initial learning rate for pre-training and fine-tuning are both 0.0001 with a linearly decreasing schedule. We first pre-train MotionMAE with the masking strategy under a fixed frame length (64), and then fine-tune it with the special input setting according to the task (For example, only the past ten and the future one frames are given for fine-tuning in the MIB task.) same as [14]and [22]. The pre-training and fine-tuning are conducted on the single dataset for different tasks because of the different skeleton topologies between these two datasets.

3.3 Comparisons with State-of-the-Art Methods

Motion In-Betweening. We test MotionMAE on the motion in-betweening task on LAFAN1. Our model is compared with zero-velocity, interpolation, and other SOTA methods. The zero-velocity baseline interpolates the missing frames with the latest frame. The interpolation baseline model uses LERP [14] for joint positions and SLERP [14] for joint rotations. Tab. 1 shows the evaluation results on the test set of LAFAN1. It can be seen that MotionMAE outperforms all the state-of-the-art models over all time horizons by a large margin more than 20%.

Motion Prediction. We test MotionMAE on the motion prediction task on Human 3.6M. Tab. 2 presents the MPJPE and MAE comparison against the state-of-the-art methods. The results indicate that our model again outperforms them.

Visual Comparison. We visualize the motion sequences generated by MotionMAE and two state-of-the-art methods in Fig. 2, and more visualizations can be seen in our supplementary material.

3.4 Ablation Study

Encoder Design. We test three structures as the encoder: (1) the ViT-like Transformer same as [9], (2) the spatial-temporal attention (STA) Transformer [4], and (3) our spatial-temporal residual encoder (STR) shown in Fig. 1. Note that the three models all use our kinematic-and-phase-aware decoder. The results are shown in Fig. 3. Compared with ViT, STA with added spatial attention is not improved much. In contrast, our STR works much better, which is due to the fact that STR has learned different levels of information from coarse to fine in the shallow to deep blocks.

Masking Strategies. The masking ratios between 10% and 80% are tested with spatial-temporal masking, which is the one used in this work. In Fig. 4, different masking ratios result in obvious performance variations. Interestingly, the ratio 50% is always the best for different metrics. So, it is used in MotionMAE in all the experiments.

Decoder Design. We test four structures as the decoder: (1) a simple network with three fully-connected layers (B) which directly predicts rotations, (2) the kinematics-aware

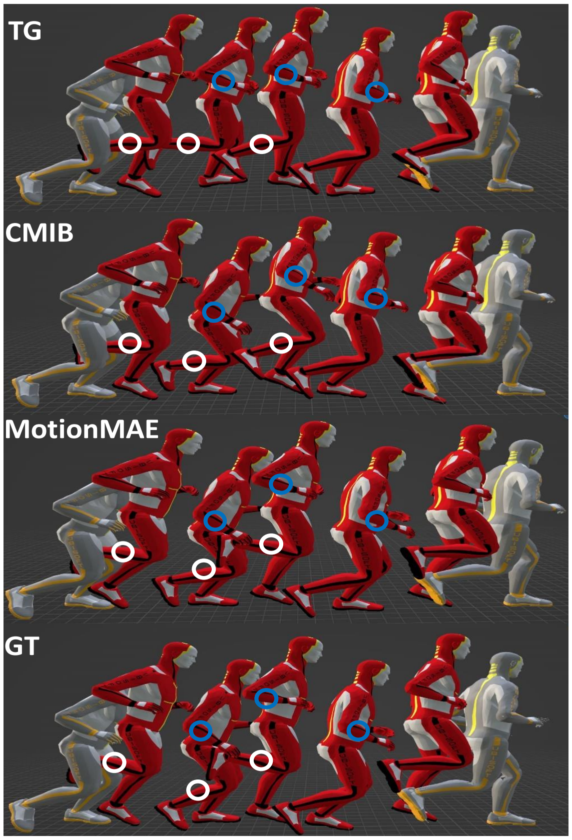


Figure 2: Visulization of generated motion sequences on a test input (jump) of LAFAN1.

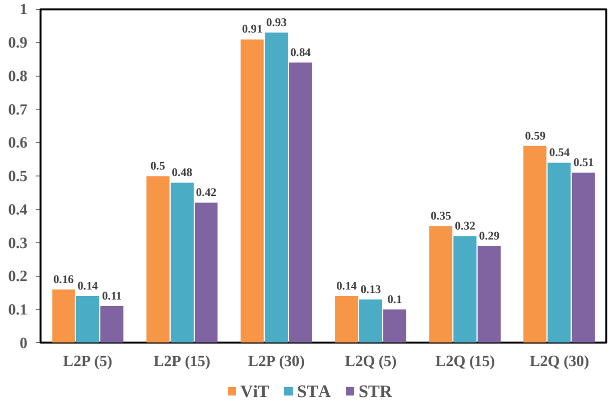


Figure 3: Results by three encoder structures on LAFAN1.

module (K), (3) B with the phase-aware module (B+P), and (4) the kinematics-and-phase-aware decoder (K+P) shown in Fig. 1. Note that the four models all use our spatial-temporal residual encoder and the mask ratio is set to 50%. The results are shown in Tab. 3. Compared with B, K gets significantly improved, indicating that our kinematics-aware module predicts unseen motion better and the human kinematics keeps the spatial information of body joints. We also find that B+P is similar to B in performance under the test length of 5. But under the test length of 15 and 30, B+P gets pretty good improvement, which implies that the phase-aware module has a noticeable effect on the in-betweening generation in long-frame prediction. Importantly, K+P performs better than all other models, which means the combination of K and P works well.

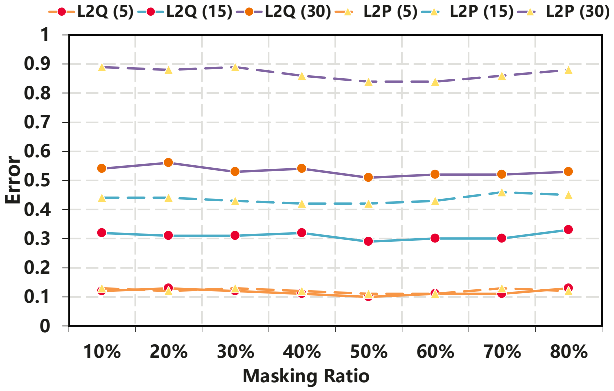


Figure 4: Results by MotionMAE with spatial-temporal masking (ST) on the LAFAN1 dataset.

	L2Q ↓			L2P ↓		
Length	5	15	30	5	15	30
B	0.12	0.35	0.57	0.16	0.54	1.05
K	0.10	0.33	0.56	0.11	0.50	0.98
B+P	0.13	0.31	0.54	0.14	0.48	0.90
K+P	0.10	0.29	0.51	0.11	0.42	0.84

Table 3: Results by four decoder structures on LAFAN1.

4 CONCLUSION AND FUTURE WORK

In this paper, we propose a novel method called MotionMAE for motion generation. We design a spatial-temporal residual encoder to learn the spatial-temporal dependence of the motion sequence. To learn the spatial constraint and the motion regularity, we further design a kinematics-and-phase-aware decoder. We first randomly mask some joints in the motion sequence and feed it to the encoder, obtaining spatial-temporal features. Then, the kinematics module gets the spatial human motion kinematics constraint and the phase module obtains the human motion temporal regularity. Our MotionMAE outperforms SOTA methods by a larger margin. In the future, we will extend MotionMAE to pre-training across different skeleton datasets and to motion generation with more diversity.

REFERENCES

[1] Hyemin Ahn, Timothy Ha, Yunho Choi, Hwiyeon Yoo, and Songhwai Oh. 2018. Text2action: Generative adversarial synthesis from language to action. In *ICRA*.

[2] Emre Aksan, Manuel Kaufmann, Peng Cao, and Otmar Hilliges. 2021. A spatio-temporal transformer for 3d human motion prediction. In *3DV*.

[3] Hangbo Bao, Li Dong, and Furu Wei. 2021. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254* (2021).

[4] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. 2021. Is space-time attention all you need for video understanding?. In *ICML*.

[5] Arij Bouazizi, Adrian Holzbock, Ulrich Kressel, Klaus Dietmayer, and Vasileios Belagiannis. 2022. MotionMixer: MLP-based 3D Human Body Pose Forecasting. *arXiv preprint arXiv:2207.00499* (2022).

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* (2020), 1877–1901.

[7] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. 2020. Generative pretraining from pixels. In *ICML*.

[8] J Devlin, MW Chang, K Lee, and KB Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL-HLT* (2019).

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[10] Yinglin Duan, Yue Lin, Zhengxia Zou, Yi Yuan, Zhehui Qian, and Bohan Zhang. 2022. A Unified Framework for Real Time Motion Completion. (2022).

[11] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. 2015. Recurrent network models for human dynamics. In *ICCV*.

[12] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. 2020. Action2motion: Conditioned generation of 3d human motions. In *ACM MM*.

[13] Félix G Harvey and Christopher Pal. 2018. Recurrent transition networks for character locomotion. In *SIGGRAPH Asia*.

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

[15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *CVPR*.

[16] Gustav Eje Henter, Simon Alexanderson, and Jonas Beskow. 2020. Moglow: Probabilistic and controllable motion synthesis using normalising flows. *ACM Transactions on Graphics* (2020), 1–14.

[17] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 2013. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence* (2013), 1325–1339.

[18] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics* (2020), 64–77.

[19] Manuel Kaufmann, Emre Aksan, Jie Song, Fabrizio Pece, Remo Ziegler, and Otmar Hilliges. 2020. Convolutional autoencoders for human motion infilling. In *3DV*.

[20] Jihoon Kim, Taehyun Byun, Seungyoun Shin, Jungdam Won, and Sungjoon Choi. 2022. Conditional Motion In-betweening. *arXiv preprint arXiv:2202.04307* (2022).

[21] Chen Li, Zhen Zhang, Wee Sun Lee, and Gim Hee Lee. 2018. Convolutional sequence to sequence model for human dynamics. In *CVPR*.

[22] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. 2020. History repeats itself: Human motion prediction via motion attention. In *ECCV*.

[23] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. 2017. Least squares generative adversarial networks. In *ICCV*.

[24] Julieta Martinez, Michael J Black, and Javier Romero. 2017. On human motion prediction using recurrent neural networks. In *CVPR*.

[25] Angel Martínez-González, Michael Villamizar, and Jean-Marc Odobez. 2021. Pose transformers (potr): Human motion prediction with non-autoregressive transformers. In *ICCV*.

[26] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. (2018).

[27] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. (2019).

[28] Theodoros Sofianos, Alessio Sampieri, Luca Franco, and Fabio Galasso. 2021. Space-time-separable graph convolutional network for pose forecasting. In *ICCV*.

[29] Sebastian Starke, Ian Mason, and Taku Komura. 2022. DeepPhase: periodic autoencoders for learning motion phase manifolds. *ACM Transactions on Graphics* (2022), 1–13.

[30] Yongyi Tang, Lin Ma, Wei Liu, and Weishi Zheng. 2018. Long-term human motion prediction by modeling motion context and enhancing motion dynamic. *arXiv preprint arXiv:1805.02513* (2018).

[31] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. 2022. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *arXiv preprint arXiv:2203.12602* (2022).

[32] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. 2022. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *CVPR*.

[33] Chongyang Zhong, Lei Hu, Zihao Zhang, Yongjing Ye, and Shihong Xia. 2022. Spatio-Temporal Gating-Adjacency GCN for Human Motion Prediction. In *CVPR*.

[34] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. 2019. On the continuity of rotation representations in neural networks. In *CVPR*.