SKELETONMAE: SPATIAL-TEMPORAL MASKED AUTOENCODERS FOR SELF-SUPERVISED SKELETON ACTION RECOGNITION

Wenhan Wu[1], Yilei Hua[2], Ce Zheng[3], Shiqian Wu[2], Chen Chen[3], Aidong Lu[1]
[1]Department of Computer Science, University of North Carolina at Charlotte, USA
[2]School of Information Science and Engineering, Wuhan University of Science and Technology, China
[3]Center for Research in Computer Vision, University of Central Florida, USA
{wwu25,alu1}@uncc.edu; {hyl1997,shiqian.wu}@wust.edu.cn;
cezheng@knights.ucf.edu; chen.chen@crcv.ucf.edu

ABSTRACT

Self-supervised skeleton-based action recognition has attracted more attention in recent years. By utilizing the unlabeled data, more generalizable features can be learned to alleviate the overfitting problem and reduce the demand for massive labeled training data. Inspired by the MAE [1], we propose a spatial-temporal masked autoencoder framework for self-supervised 3D skeleton-based action recognition (SkeletonMAE). Following MAE's masking and reconstruction pipeline, we utilize a skeleton-based encoder-decoder transformer architecture to reconstruct the masked skeleton sequences. A novel masking strategy, named Spatial-Temporal Masking, is introduced in terms of both joint-level and frame-level for the skeleton sequence. This pre-training strategy makes the encoder output generalizable skeleton features with spatial and temporal dependencies. Given the unmasked skeleton sequence, the encoder is fine-tuned for the action recognition task. Extensive experiments show that our SkeletonMAE achieves remarkable performance and outperforms the state-of-the-art methods on both NTU RGB+D 60 and NTU RGB+D 120 datasets.

***Index Terms*—** Masked autoencoder, Skeleton action recognition

1. INTRODUCTION

Human Action Recognition is a fundamental research topic in computer vision, which aims to understand human behaviors and distinguish actions. With the booming development of deep learning and human pose estimation methods, human skeleton data can be efficiently extracted as a high-level but light-weighted representation, which draws great attention to human behavior and action analysis. Thus, 3D skeleton-based action recognition has become an important research field in human action recognition.

Most recent methods focus on full-supervised learning algorithms to build their frameworks: methods based on Convolutional Neural Networks (CNN) [2], methods based on Recurrent Neural Networks (RNN) [3], methods based on Graph Convolution Networks (GCN) [4] and methods based on Transformer [5] are widely applied in skeleton action recognition and lead to very good results. However, fully supervised action recognition is liable to overfitting. Also, it requires massive labeled training data, which is expensive and time-consuming. To alleviate these issues, *self-supervised learning* methods, which utilize unlabeled data to learn data representations, have been increasingly prevalent in skeleton action recognition. Some self-supervised approaches consider pretext tasks for skeleton representation learning using unlabeled skeleton data, such as motion reconstruction [6] and jigsaw puzzle [7]. However, such pretext-based methods focus on local features such as joint correlation and skeleton scale in the same frame, and have not fully explored the temporal information. Recently, several works [8, 9] train the contrastive-based model by constructing the skeleton sequences in different views by data augmentation and positive-negative pairs. Although these contrastive learning-based methods emphasize high-level context information, they heavily rely on the number of contrastive pairs in the joints for extracting skeleton features and ignore the joint correlation information among different frames.

Recently, a new self-supervised learning approach named masked autoencoders (MAE) [1] demonstrates a strong generalization capability with remarkable performance in computer vision tasks. MAE masks a large proportion of the input image and then forces the model to learn a generalizable representation by using only the unmasked proportion to reconstruct the original image. However, MAE can not be directly utilized for self-supervised skeleton action recognition because the Vision Transformer

(ViT) [10] architecture is used in MAE [1] to process the image input. Different from the image that does not contain temporal information, human skeleton sequences are extracted from videos with high information density, which contains fruitful semantic information: at the spatial level, joint features contain the relationships among different joints in the same frame; in temporal level, frame features represent the movements of the same joint from different frames. Moreover, The masking strategy in MAE only focuses on the spatial domain. When processing the human skeleton sequences data, a spatial-temporal masking strategy is needed. To address these issues, we introduce a novel skeleton-based masked autoencoder named **SkeletonMAE** for self-supervised skeleton spatial-temporal representation learning: 1) the masked input sequences are generated from the original skeleton sequences, which contain joints coordinates (spatial) information and frames (temporal) information; 2) with spatial-temporal masking strategy and encoding-decoding rule, SkeletonMAE gains reconstruction sequences from masked sequences, where the spatial and temporal information is well processed by the transformer-based encoder and decoder (transformers have great potential for spatial-temporal representation learning with long-term sequence data).

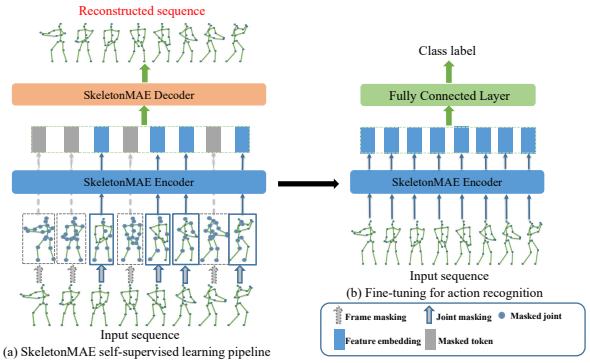

Figure 1: (a) The overall pipeline of the SkeletonMAE. During pre-training. The transformer-based encoder only deals with visible skeleton tokens, and the transformer-based decoder is utilized for skeleton reconstruction, then we only use the SkeletonMAE encoder during the fine-tuning. (b) The end-to-end fine-tuning procedure for skeleton action recognition.

The framework of SkeletonMAE is presented in Fig. 1. Specifically, the whole SkeletonMAE pipeline is designed with the following principles. During pre-training, a spatial-temporal masking strategy (with pre-set frame-masking and joint masking ratios) is employed to mask out part of the input skeleton sequence in both frame-level and the joint-level (Sec. 3.1). In order to find the best trade-off point for spatial-temporal representation learning, we discuss the roles of joint-masking and frame-masking ratios and find the best ratio combination. The encoder is applied to learn the generalizable feature representation while the decoder is designed to reconstruct the missing skeletons. Since we are dealing with the skeleton sequences, we utilize skeleton-based spatial-temporal transformer [5] as our network backbone. During the fine-tuning stage, we only use the encoder with a simple output layer to predict the actions. The action recognition results show that our approach outperforms the state-of-the-art self-supervised learning methods without extra data. To summarize, we make the following contributions: (1) We propose a simple and efficient skeleton-based masked autoencoder architecture, which aims to learn comprehensive and generalizable skeleton feature representations. (2) To have a better understanding of the skeleton masking methods, we explore different masking methods and develop a novel spatial-temporal masking for skeleton data at both joint-level and frame-level. At the same time, we validate the proper combination of the joint-masking ratio and frame-masking ratio. (3) We evaluate our model on NTU-RGB+D 60 and NTU-RGB+D 120 datasets, and extensive experimental results show that SkeletonMAE achieves state-of-the-art performance under self-supervised settings.

2. RELATED WORK

2.1. Self-supervised skeleton-based action recognition

Self-supervised learning aims to extract feature representations without using labeled data and achieves promising performance in image-based and video-based representation learning [11, 12]. More self-supervised representation learning approaches adopt the so-called contrastive learning manner [13, 14] to boost their performance. Inspired by contrastive learning architectures, recent skeleton representation learning works have achieved some inspiring progress in self-supervised skeleton action recognition. MS^{2}L [7] introduced a multi-task self-supervised learning framework for extracting joint representations by using motion prediction and jigsaw puzzle recognition. CrosSCLR [8] developed a contrastive learning-based framework to learn both single-view and across-view representations from skeleton data. Following CrosSCLR, AimCLR [9] exploited an extreme data augmentation strategy to add extra hard contrastive pairs, which aims to learn more general representations from skeleton data.

2.2. Masked autoencoding

Masked autoencoding [15] is a well-structured self-supervised learning model for general representation learning, and successfully applied in BERT [16], one of the most famous self-supervised frameworks in natural language processing (NLP). The BERT model is simple and straightforward – remove part of the sequence data with the masked tokens, predict the removed parts and calculate the loss between prediction and ground-truth data. As

a result, the reconstruction sequence works well for the training of the generalizable models. Inspired by masked autoencoders and BERT, He et al. [1]. design a scalable self-supervised masked autoencoder (MAE) for computer vision tasks. With the same core concept as BERT, MAE masks parts of the image patches and rebuilds them for pre-training. Compared with the original MAE, there are two main spotlights in our proposed SkeletonMAE: 1) a skeleton-based transformer encoder-decoder framework, the encoder processes the unmasked tokens and the decoder reconstructs the original skeleton sequence; 2) a spatial-temporal masking strategy for both joint and frame-level features. Following the main idea of MAE, we propose SkeletonMAE for self-supervised skeleton action recognition.

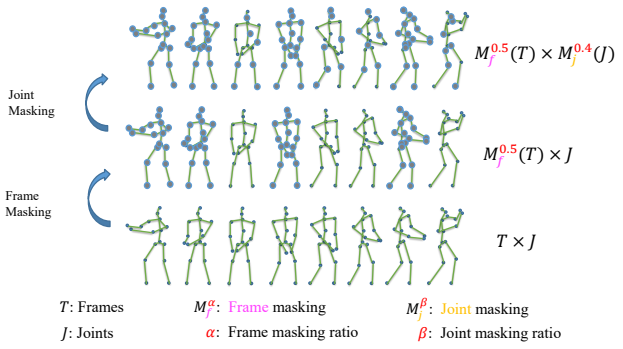

Figure 2: Illustration of the spatial-temporal masking pipeline. Based on the pre-set frame-masking ratio (α) and joint-masking ratio (β), we first adopt frame masking (*i.e.* removing an entire skeleton frame) in skeleton sequence (*e.g.*, $\alpha = 0.5$), and randomly mask the joints in joint-level (*e.g.*, $\beta = 0.4$).

3. METHODOLOGY

In this section, we first design a spatial-temporal masking strategy for skeleton data in Sec. 3.1. Next, we analyze our SkeletonMAE for action recognition in Sec. 3.2. Finally, we present our fine-tuning procedure in Sec. 3.3.

3.1. Spatial-temporal masking strategy

We propose a spatial-temporal masking method for a portion of the skeleton sequence input, the pipeline of our masking strategy is illustrated in Fig. 2.

Temporal-masking method. Fig. 2 shows our masking method at the frame level. Based on the pre-set frame-masking ratio, a portion of the frames are randomly removed and their indices are stored, the remaining frames are then processed by the spatial-masking method at the joint level.

Spatial-masking method. As shown in Fig. 2, after implementing the temporal masking method in all the input frames, the rest frames are then processed via spatial masking strategy. And based on the pre-set joint-masking ratio, we randomly mask part of the joints in every unmasked frame. It is worth noting that the indices of the masked joints are not fixed in this randomly spatial-masking method, which means that the same joints in different frames may be masked or not. This simple approach is illustrated in Fig. 3(b). Besides this masking method, we also introduce a joint masking strategy with fixed indices, which is shown in Fig. 3(c). The joints with the same indices in different frames are all masked or not based on the joint-masking ratio. We conduct experiments to compare these two masking strategies in Sec. 4.3.

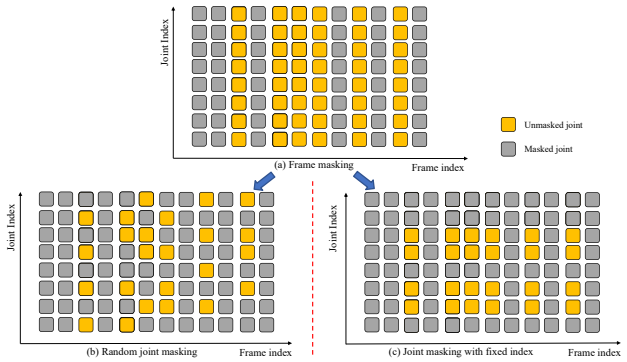

Figure 3: Illustration of two masking strategies. (a) The frame-masking is first implemented and then: (b) randomly mask the joints in the spatial level; (b) mask the joint with the fixed index.

3.2. SkeletonMAE architecture

We describe the main components in SkeletonMAE, *e.g.*, encoder, decoder, reconstruction sequence, loss function, and fine-tuning pipeline for skeleton action recognition. The pipeline and SkeletonMAE structure are illustrated in Fig. 1. And more details are provided in Supplementary A.

SkeletonMAE encoder. Our encoder is based on STTFormer[5] and only processes the visible skeleton tokens. Given a skeleton sequence as input, we apply the frame-masking and joint-masking methods respectively. This spatially and temporally unmasked token is fed to the SkeletonMAE encoder, which maps the input to the spatial-temporal embedding features.

SkeletonMAE decoder. Our decoder also adopts the STTFormer structure. Same to the decoder in MAE, the spatial-temporal embedding features are processed in the SkeletonMAE decoder to reconstruct the original sequence. At the same time, in order to reserve the position information for reconstruction, positional embeddings are also introduced. The output of the decoder is the reconstructed sequence, which should be the same as the original sequence without masking.

Reconstruction. We use the mean squared error (MSE)

loss to measure the consequence of reconstruction. In this case, we compute the MSE loss between the original skeleton sequences and the reconstructed sequences as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}|S_i - S_i^*|^2, \quad (1)$$

where i is the index of frame, N is the number of samples, S is the input sequence, and S^* is the reconstructed sequence.

3.3. Fine-tuning for skeleton action recognition

In order to evaluate SkeletonMAE's ability to learn skeleton representations, we load the learned parameter weights obtained from pre-training to fine-tune the model with all the training data, then the label for each action is predicted with the recognition accuracy. The procedure of fine-tuning is shown in Fig. 1 (b). Different from the latest contrastive-based self-supervised skeleton action recognition methods [8, 9], which verify the model via linear evaluation protocol, we focus more on the skeletal generalization learning capabilities in the end-to-end fine-tuning scenarios.

4. EXPERIMENTS

4.1. Datasets

We evaluate our experiments on the following two most-used datasets: NTU-RGB+D 60 (NTU-60) dataset [17] and NTU-RGB+D 120 (NTU-120) dataset [18], and follow the evaluation protocols for the experimental evaluation: Cross-Subject (X-Sub) and Cross-View (X-View) protocols for NTU-60 dataset, Cross-Subject (X-Sub) and Corss-Set (X-Set) for NTU-120 dataset.

4.2. Experimental settings

Our experiments are performed on 8× A6000 GPUs with Pytorch framework implementation. Both our pre-training and fine-tuning models are trained by Adam optimizer [19] with a base learning rate of 0.005 and weight decay of 0.0001. The batch size is 64. The pre-training and fine-tuning epoch numbers are all set to 200. We also use a multi-step learning rate schedule for learning rate adjustment with gamma 0.1 and milestones are 60 epoch, 90 epoch, and 110 epoch. For fair comparisons among different methods, we limit the length of the skeleton sequence to 20 frames for all experiments.

Sequence division and patch embedding. In our research, we follow the patch embedding method in [5]. We first divide the original skeleton sequence into tuples. Then, since the skeleton data does not contain a large number of pixels and various noises like image data, we directly use a 1x1 Conv for patch embedding processing.

Masking settings. We implement our masking strategies before sequence division. As we discussed in Sec. 3.1, we first mask out a random subset of frames by the pre-set frame masking ratio and then mask out a random index of joints by the pre-set joint masking ratio. During experiments, we test several trials of the frame-masking ratio and joint-masking ratio, finding the best trade-off combination.

Pre-training. We choose MSE loss as pre-training loss and save the best model by the minimized validation loss.

Fine-tuning. As we discussed in Sec. 3.3, we use end-to-end fine-tuning for the end task. Moreover, we choose cross-entropy loss with label smoothing [20] as the fine-tuning loss with a smoothing rate of 0.1 and save the best model by the maximized validation accuracy.

4.3. Ablation study

Different masking strategies. After performing the same degree of random frame masking, we compare the masking strategies of masking the joints randomly with the method of keeping the same masked joint index over the entire sequence. The experimental results show that the pure random joint masking for visible frames is more helpful for the final fine-tuning result (in Table 1, we get our best fine-tuned recognition accuracy of 86.6% on X-sub using random masking strategy, which is 1.2% better than the best result of the masking method by fixing the joints indices). Moreover, we also conduct experiments that mask the inputs only in the frame level and only in the joint level. The overall results indicate that the random masking method outperforms the masking method with fixed joint indices, which means the model learns better features with a randomly generated input than the pre-defied input. Notably, the MAE experiment also shows that using a more random masking strategy is more beneficial to the final fine-tuning result.

Frame-masking ratio and joint-masking ratio. In spatial and temporal domains, we test several combinations of different frame-masking ratios and joint-masking ratios on SkeletonMAE. Following both joint index fixed and random masking strategies, we set the frame masking ratio 0.4, 0.5, and 0.6 respectively, for every decided frame masking ratio, we test different joint masking ratios (0.4, 0.5, and 0.6 respectively). As shown in Table 1, the final results on NTU-60 with X-Sub show that a frame-masking ratio of 0.4 and a joint-masking ratio of 0.5 work best in the masking method with fixed joints indices (85.4% accuracy). Using the random masking method, we achieve the best result (86.6% accuracy) in two combinations (0.5 joint-masking ratio with 0.5 or 0.4 frame-masking ratios). The qualitative analysis of these masking strategies is shown in Supplementary B.

Embedding dimension. Table 2 shows the ablation study on the embedding dimension of the decoder. We change the different embedding dimensions in the Skele-

method	frame-masking ratio	joint-masking ratio	NTU-60 X-Sub
fixed index	0.6	0.4	85.2
	0.6	0.5	84.9
	0.6	0.6	85.3
	0.5	0.4	85.3
	0.5	0.5	85.0
	0.5	0.6	84.8
	0.4	0.4	84.8
	0.4	0.5	**85.4**
	0.4	0.6	85.2
random	0.6	0.4	86.5
	0.6	0.5	86.0
	0.6	0.6	86.3
	0.5	0.4	86.3
	0.5	0.5	**86.6**
	0.5	0.6	85.7
	0.4	0.4	85.6
	0.4	0.5	**86.6**
	0.4	0.6	85.4
only joint	0	0.4	85.3
	0	0.5	85.2
	0	0.6	85.3
only frame	0.4	0	84.9
	0.5	0	82.1
	0.6	0	85.2

Table 1: Masking strategies with joint-masking ratio and frame-masking ratio. Specifically, there are two joint masking methods tested: fixed indices masking and randomly masking. Besides, masking only in joint level and masking only in frame level are also tested.

tonMAE decoder and find that the default setting with 256 dimension works better (86.6% accuracy) than the larger size (86.0% accuracy) and the small size (85.2% accuracy). We also observe that with the increasing size of the embedding dimension, the number of model parameters increases as well, when we set the dimension as 512, the parameters are 11 times larger than the parameters with dimension 128, which costs more time for training. So we choose 256 as the default embedding dimension for the following ablation studies.

embedding dimension	NTU-60 X-Sub	parameters(M)
128	85.2	3
256	**86.6**	11
512	86.0	33

Table 2: Ablation study on embedding dimension.

decoder depth	NTU 60 X-Sub
11	86.5
9	**86.6**
7	86.2
5	85.7

Table 3: Ablation study on decoder depth.

Decoder depth. Decoder depth represents the number of the STTFormer blocks. According to the last ablation experiment, we set the embedding dimension (the width of the decoder) as the default size of 256, and vary the decoder depth (11, 9, 7, and 5 blocks). As the results shown in Table 3, SkeletonMAE achieves the best result (86.6% accuracy) when the decoder depth is 9. The deep depth (11 blocks with 86.5% accuracy) and shallow depth (7 blocks with 86.2% accuracy and 5 blocks with 85.7% accuracy) perform worse. According to the results from the embedding dimension and decoder depth experiments, we finalize our default decoder configurations for the following experiments (256 embedding dimension and 9 blocks). The encoder and decoder settings are shown in Supplementary A.

Pre-training schedule. Normally, a longer pre-training schedule will give an improvement, thus in this ablation study, we increase the pre-training epoch from 50 epoch to 200 epoch, and test the best fine-tuned results at every 50 epoch. As it shown in Fig. 4, the best accuracy is 86.6%, so we select 200 epoch as the default pre-training epoch for the following experiments. It is worth noting that there is an impressive improvement (5.0%) between 50 epoch to 100 epoch, but a slight improvement (0.2%) between 150 epoch to 200 epoch, which means it is not cost-effective to keep increasing the pre-training epoch.

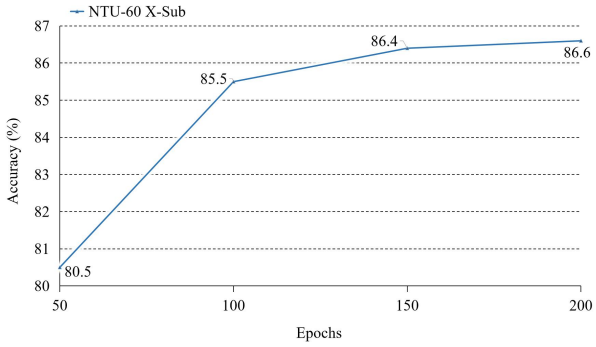

Figure 4: Ablation study on pre-training schedule.

4.4. Comparison with state-of-the-art

Self-supervised training. Notably, as we can see from Table 4, our SkeletonMAE outperforms the two latest self-supervised skeleton action recognition methods: CrosSCLR [8] and AimClR [9]. For a fair comparison, we replace their backbone networks (both of them use ST-GCN as the backbone) with STTFormer under the same settings. The results show that on NTU-60 dataset, our SkeletonMAE leads CrosSCLR 2.0% and AimCLR 2.7% on X-Sub, and also leads CrosSCLR 2.4% and AimCLR 2.5% under X-View protocol. As for the results on NTU-120 dataset, SkeletonMAE outperforms CrosSCLR by 1.8% and 1.2% on X-Sub and X-Set and also outperforms AimCLR by 2.2% and 1.9% on X-Sub and X-Set respectively. The results indicate that our SkeletonMAE not only achieves outperforming results on the small-size dataset but also on the large-size dataset.

		NTU-60		NTU-120	
method	backbone	X-Sub	X-View	X-Sub	X-Set
CrosSCLR[8]	ST-GCN	82.2	88.9	73.6	75.3
AimCLR[9]	ST-GCN	83.0	89.2	76.4	76.7
CrosSCLR[8]	STTFormer	84.6	90.5	75.0	77.9
AimCLR[9]	STTFormer	83.9	90.4	74.6	77.2
SkeletonMAE	STTFormer	**86.6**	**92.9**	**76.8**	**79.1**

Table 4: Fine-tuned results on NTU-60 and NTU-120 datasets.

Fewer labeled data training. In order to evaluate the ability of spatial-temporal feature learning in the fewer-data situation, we fine-tune our pre-trained SkeletonMAE model with only 5% and 10% labeled data on both NTU-60 and NTU-120 datasets. According to Table 5, our SkeletonMAE achieves 64.4% and 68.8% on NTU-60 X-Sub and X-View with only 5% fine-tuning data and surpasses CrossSCLR and AimCLR. Moreover, our SkeletonMAE

also performs better than CrossSCLR and AimCLR with 10% labeled data (73.0% and 76.9% on NTU-60 X-Sub and X-View respectively). Meanwhile, our SkeletonMAE achieves outperformed results on NTU-120 data with 5% (50.4% on X-Sub and 52.0% on X-Set) and 10% (61.8% on X-Sub and 62.5% on X-Set) labeled data, which demonstrates a better capability of generalizability learning of our approach under the extreme fine-tuning situation.

method	backbone	label fraction	NTU-60		NTU-120	
			X-Sub	X-View	X-Sub	X-Set
CrosSCLR[8]	STTFormer	5%	63.5	66.9	50.2	50.4
AimCLR[9]	STTFormer	5%	63.9	67.5	49.0	51.8
SkeletonMAE	STTFormer	5%	**64.4**	**68.8**	**50.4**	**52.0**
CrosSCLR[8]	STTFormer	10%	71.0	75.1	58.5	60.6
AimCLR[9]	STTFormer	10%	70.2	76.2	58.6	60.5
SkeletonMAE	STTFormer	10%	**73.0**	**76.9**	**61.8**	**62.5**

Table 5: Fine-tuned results with fewer labeled data on NTU-60 and NTU-120 datasets.

4.5. Visualization

We provide several visualization results of the reconstructed action sequences, due to the page limitation, the analysis and visualization are shown in Supplementary C.

5. CONCLUSION

We conduct a novel skeleton-based masked autoencoder named SkeletonMAE for self-supervised skeleton action recognition. In order to get a better skeleton representation learning, we apply a novel spatial-temporal masking strategy in pre-training for skeleton reconstruction. The roles of different frame-ratio and joint-ratio are also discussed and implemented. With comprehensive experiments on NTU-60 and NTU-120 datasets, we show outperformed results of SkeletonMAE for skeleton action recognition.

6. REFERENCES

[1] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick, "Masked autoencoders are scalable vision learners," in *CVPR*, 2022, pp. 16000–16009.

[2] Yong Du, Yun Fu, and Liang Wang, "Skeleton based action recognition with convolutional neural network," in *ACPR*. IEEE, 2015,2015, pp. 579–583.

[3] Hongsong Wang and Liang Wang, "Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks," in *CVPR*, 2017, pp. 499–508.

[4] Sijie Yan, Yuanjun Xiong, and Dahua Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *AAAI*, 2018.

[5] Helei Qiu, Biao Hou, Bo Ren, and Xiaohua Zhang, "Spatio-temporal tuples transformer for skeleton-based action recognition," *arXiv preprint arXiv:2201.02849*, 2022.

[6] Yi-Bin Cheng, Xipeng Chen, Dongyu Zhang, and Liang Lin, "Motion-transformer: self-supervised pre-training for skeleton-based action recognition," in *ACM Multimedia Asia*, 2021, pp. 1–6.

[7] Lilang Lin, Sijie Song, Wenhan Yang, and Jiaying Liu, "Ms2l: Multi-task self-supervised learning for skeleton based action recognition," in *ACM Multimedia*, 2020, pp. 2490–2498.

[8] Linguo Li, Minsi Wang, Bingbing Ni, Hang Wang, Jiancheng Yang, and Wenjun Zhang, "3d human action representation learning via cross-view consistency pursuit," in *CVPR*, 2021, pp. 4741–4750.

[9] Tianyu Guo, Hong Liu, Zhan Chen, Mengyuan Liu, Tao Wang, and Runwei Ding, "Contrastive learning from extremely augmented skeleton sequences for self-supervised action recognition," in *AAAI*, 2022, vol. 36, pp. 762–770.

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

[11] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov, "Unsupervised learning of video representations using lstms," in *ICML*. PMLR, 2015, pp. 843–852.

[12] Zelun Luo, Boya Peng, De-An Huang, Alexandre Alahi, and Li Fei-Fei, "Unsupervised learning of long-term motion dynamics for videos," in *CVPR*, 2017, pp. 2203–2212.

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, 2020, pp. 9729–9738.

[14] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*. PMLR, 2020, pp. 1597–1607.

[15] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol, "Extracting and composing robust features with denoising autoencoders," in *ICML*, 2008, pp. 1096–1103.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[17] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang, "Ntu rgb+ d: A large scale dataset for 3d human activity analysis," in *CVPR*, 2016, pp. 1010–1019.

[18] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot, "Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding," *TPAMI*, vol. 42, no. 10, pp. 2684–2701, 2019.

[19] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[20] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li, "Bag of tricks for image classification with convolutional neural networks," in *CVPR*, 2019, pp. 558–567.