MaskCLR: Attention-Guided Contrastive Learning for Robust Action Representation Learning (Supplementary Material)

Mohamed Abdelfattah* Mariam Hassan Alexandre Alahi

École Polytechnique Fédérale de Lausanne (EPFL)
firstname.lastname@epfl.ch

1. Details of datasets

In Table 1, we summarize the number of classes, as well as the number of samples in each subset of the datasets used in our experiments. We describe the datasets below.

NTU RGB+D. NTU RGB+D [6, 9] is lab-collected, large-scale action recognition dataset which has two versions: NTU-60 (60 classes) and NTU-120 (120 classes.) NTU-60 contains 57K videos while NTU-120 is an extension of it that contains 114K videos. The datasets are split in three ways: X-Sub (for both), X-View (for NTU-60), and X-Set (for NTU-120), in which human subjects, camera views, and camera setups are different, respectively. While 3D skeletons from sensors are provided in this dataset, we extract the 2D skeletons by applying the three different pose estimators (see Sec. 2) directly on the RGB videos.

Kinetics400. Kinetics400 [3] is a large scale video-based action recognition dataset with 400 action classes and 300k videos. The videos are 10s long extracted from YouTube which makes the dataset a challenging one due to the diversity in quality of videos, number of people, and background noise in each video. Furthermore, the dataset is not human-centric, meaning that in many frames, the human scales are only partly visible, hard to recognize, or nonexistent. Therefore, the extracted skeletons are of bad quality due to the numerous failure cases of the SOTA pose estimators under these conditions. For this reason, the highest top-1 accuracy achieved on Kinetics400 in skeleton-based action recognition is still far behind its RGB-based counterpart.

Table 1. **Action recognition datasets.**

Dataset	#Classes	#Train	#Val.	Total
NTU60-XSub [9]	60	40K	17K	57K
NTU60-XView [9]	60	38K	19K	57K
NTU120-XSub [6]	120	63K	51K	114K
NTU120-XSet [6]	120	54K	60K	114K
Kinetics400 [3]	400	250K	50K	300K

2. Pose Extraction

Pose estimation is a critical step that largely affects the final recognition accuracy, yet the importance of which is mostly overlooked in previous literature. Poses retrieved from sensor readings or existing pose estimators are used to train and test skeletal action recognition models without strong justification behind the pose extraction method. To the best of our knowledge, there's no consensus among the research community on a fixed set of skeletons to test action recognition performance. Furthermore, due to the large volume of research, it is not feasible to conduct a comprehensive study on which models work best on which poses and for which datasets. We, therefore, argue for the need for skeletal action recognition models that are generic to the type of pose estimator. We highlight the importance of reporting the model performance on poses extracted with multiple pose estimators instead of only one. To that end, we leverage three pose estimators of different levels of performance: ViT-Pose (SOTA) [11] (High Quality, HQ), HRNet [10] (Medium Quality, MQ), and OpenPifPaf [4] (Low Quality, LQ).

Table 2. **Quality of utilized pose estimators based on the AP score on COCO test-dev set.**

Pose Estimator	Type	AP	Pose Quality
ViTPose [11]	Top-Down	81.1	HQ
HRNet [10]	Top-Down	77.0	MQ
OpenPifPaf [4]	Bottom-Up	71.9	LQ

We leverage 2D poses instead of 3D ones because in general they are of higher quality [2]. As shown in Table 2, the selected pose estimators have different types and pose qualities, assigned according to their reported AP score on the

[*] Corresponding author.

COCO test-dev [5]. While Top-Down methods outperform Bottom-Up methods on standard benchmarks, we highlight the importance of experimenting with both to demonstrate the generalization of skeletal action recognition. Following previous literature [2], we store the extracted keypoints in the 17-joint coco format in coordinate triplets (x, y, c), where (x, y) is the joint coordinates and c is the joint confidence score. In Table 3, we report some metrics reflecting the percentage of keypoints and people that were undetected by each pose estimator. The percentage of missing keypoints is the number of missed keypoints within the detected poses, divided by the actual number of joints in these poses. The percentage of missing people indicate the number of undetected people divided by the total number of people in each dataset.

Table 3. **Assessment of pose estimators in terms of undetected joints and people.**

Pose Estimator	NTU60		NTU120	
	XSub	XView	XSub	XSet
Percentage of missing keypoints				
ViTPose [11]	0.0	0.0	0.0	0.0
HRNet [10]	0.0	0.0	0.0	0.0
OpenPifPaf [4]	7.7	6.9	6.5	7.5
Percentage of missing people				
ViTPose [11]	1.2	0.01	0.01	0.04
HRNet [10]	0.1	0.1	0.12	0.13
OpenPifPaf [4]	0.53	0.29	5.4	5.8

3. Transformer Backbones

In this section, we briefly summarize the details of the transformer backbones utilized in our experiments.

3.1. Vanilla Transformer

We adopt the implementation of the vanilla transformer in [1], which is originally designed for image classification. Similar to patching for images, we patchify the input joints individually using a fixed feature dimension $C_f = 512$. We add learnable spatial and temporal encodings before passing the result to the transformer blocks with depth $N = 5$. We also change the hidden dimension of the feed-forward network to 2048. The rest of the network architecture is almost unchanged compared to the original transformer.

3.2. STTFormer

The spatiotemporal tuples Transformer (STTFormer) [8] is an extension of the vanilla transformer in which the input skeleton sequence is partitioned across the temporal dimension into blocks of fixed size. Then spatiotemporal attention is employed to capture the correlations between different joints in consecutive frames. To improve performance on similar actions, a feature aggregation module is incorporated, which is a convolution operation on non-adjacent frames after the transformer encoder blocks. We use the model with the same hyperparameters, except that we the change the feature dimension from 256 to 512, and we set the depth to $N = 5$.

3.3. DSTFormer

The Dual-stream Spatio-temporal Transformer (DSTFormer) [12] is a transformer-based motion representation encoder, originally designed for 2D-to-3D pose lifting. DSTFormer consists of two streams of alternating spatial and temporal Multi-Head Self-Attention (MHSA) blocks that respectively model the spatial and temporal correlations in the input joints. The two streams are fused together with adaptive fusion weights to dynamically combine their learned information. We apply our framework on top of DSTFormer without changing any of its hyperparameters.

4. Attention Filters Diagnostics

We analyze the effect of finding the activated joints from different attention maps across three MHSA depth layers in the DSTFormer [12] backbone network. Our goal is to find the most important joints that lead to the action classification. We follow the black-box insertion/deletion metric proposed in RISE [7] for empirically evaluating the different attention maps. For the deletion metric, we incrementally delete the most important joints, as computed by the attention scores of a transformer layer, and measure the effect on the accuracy by computing the Area Under the Curve (AUC). On the other hand, the insertion metric is a complementary approach in which the most important joints are gradually introduced. Our results are shown in Figure 1 for different attention maps on NTU60-XSub [9]. The best attention map is determined by a lower deletion AUC score and a higher insertion AUC score. We find that the attention map from the last MHSA block $N = 5$ best reflects the most important joints. The last layer inherits information from all the preceding layers in learning attention parameters, and is therefore used in our approach to determine the most activated joints.

5. Noise visualization and class-wise scores

In Figure 2, we compare visualizations of standard and noisy skeletons. At noise $\sigma \leq 0.005$, we note that the resulting noisy skeletons are virtually indistinguishable from the original skeletons. Figure 3 shows the class-wise performance gained on noisy skeletons ($\sigma = 0.002$) from training DSTFormer [12] with our MaskCLR approach. Particularly in low-motion actions (*e.g,* drink water, pointing, etc), MaskCLR obtains considerable performance gains, up to

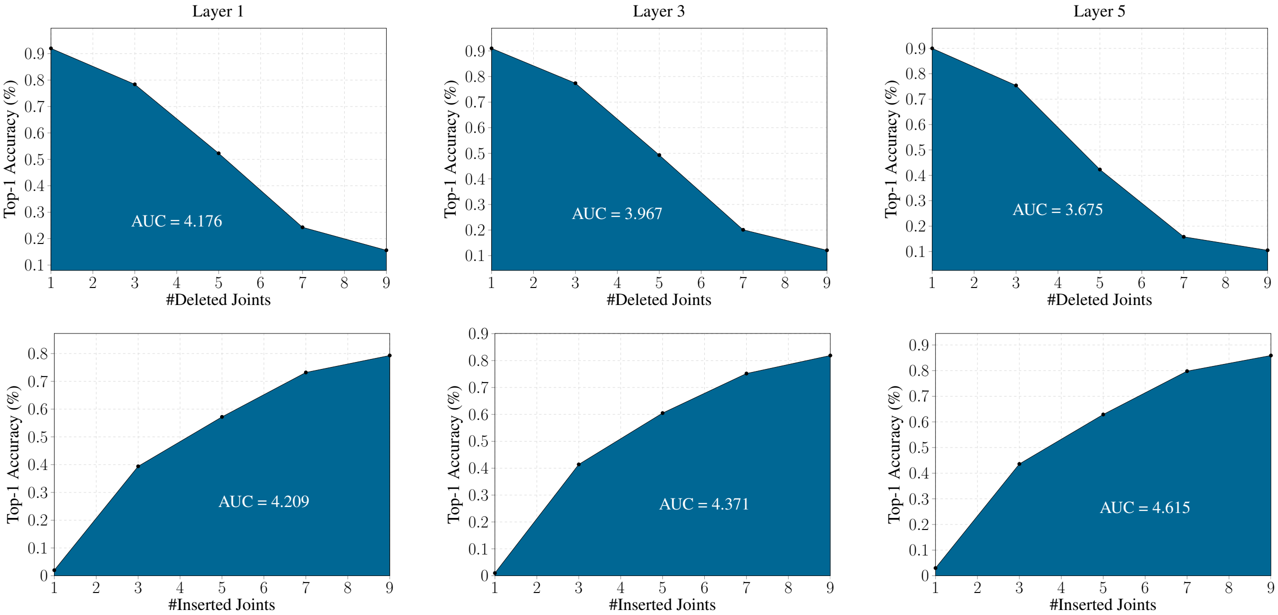


Figure 1. **Deletion (top) and insertion (bottom) metrics for attention maps of three layers.**

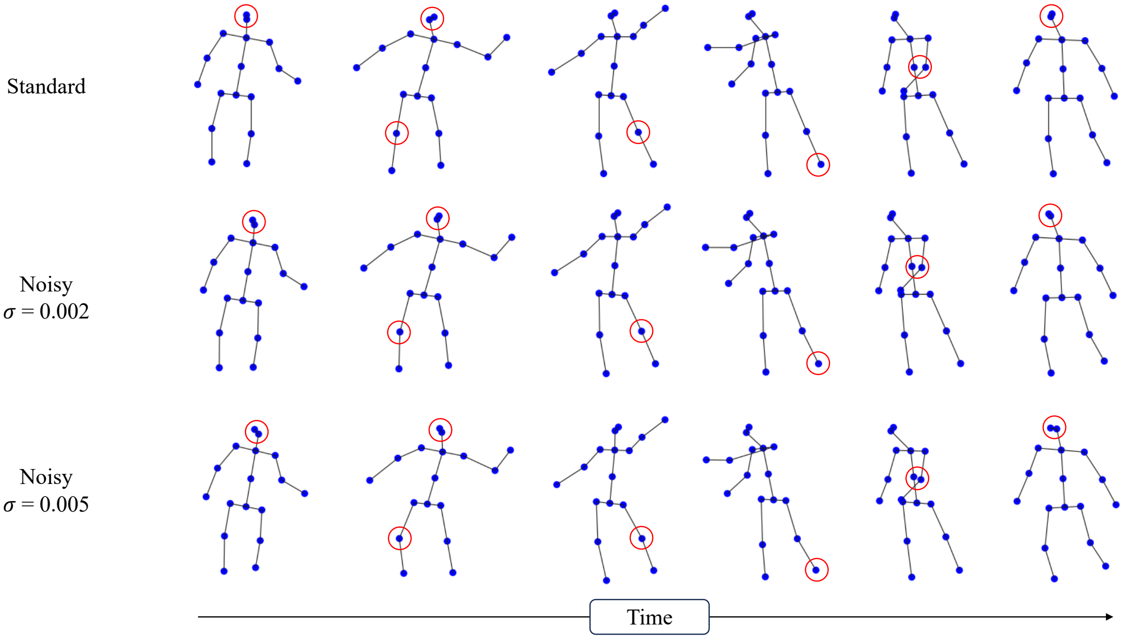


Figure 2. **Visualization of standard and noisy versions of action "throw" from NTU60-XSub.** Noise is sampled from a Gaussian Distribution $X \sim \mathcal{N}(0,\, \sigma^2)$ and introduced on all joints across time. At $\sigma = 0.002$, the noisy skeletons are virtually indistinguishable from the standard ones. The red circles reflect subtle differences in joint positions.

28.4 percentage points for "pointing." MotionBERT, which is used with the same DSTFormer backbone, only captures low-level joint motion patterns, which might be disrupted by noise. In contrast, MaskCLR encodes the pose information from the previously unactivated joints, which are important in differentiating fine-grained actions. Further, the penalty term ω used in our class contrastive loss $\mathcal{L}_{cc}$ penalizes the distance between ambiguous actions, which helps mitigate the confusion in some of the aforementioned fine-grained actions. The MLCL approach adopted in MaskCLR helps the model capture the high-level actions semantics instead of low-level joint variations, boosting the model performance on standard skeletons and robustness to perturbed ones.

6. Failure Analysis on NTU60-XSub

Table 4. **Top 5 confusing pairs of actions in NTU60-XSub.**

First Action	Second Action	$S_{MB} \downarrow$	$S_{MCLR} \downarrow$
Top 5 confusing pairs for MotionBERT (standard skeletons)			
play with phone/tablet	read	67	**17**
play with phone/tablet	write	55	**40**
typing on a keyboard	write	**54**	61
rub two hands together	clap	54	**31**
take a selfie	pointing to sth.	56	**19**
Top 5 confusing pairs for MotionBERT at $\sigma = 0.005$			
jump up	hopping	271	**265**
shake head	nod head/bow	275	**251**
kicking something	hand waving	275	**264**
wear jacket	tear up paper	274	**262**
pickup	drop	274	**261**
Top 5 confusing pairs for MaskCLR (standard skeletons)			
take off glasses	wear on glasses	51	**50**
typing on a keyboard	write	**54**	61
writing	reading	45	**44**
take off a shoe	wear a shoe	**44**	**44**
take off a hat/cap	put on a hat/cap	48	**42**
Top 5 confusing pairs for MaskCLR at $\sigma = 0.005$			
wipe face	shake head	**256**	277
touch person's pocket	giving sth. to person	**271**	**271**
kick person	punching person	**265**	274
point finger at person	pat on back of person	269	**266**
take off jacket	wear jacket	267	**266**

On standard skeletons, MaskCLR combined with DSTFormer [12] backbone achieves 93.9% on NTU60-XSub [9] dataset, outperforming previous SOTA methods. Further, MaskCLR is relatively robust to noisy skeletons, obtaining 93.9% and 92.5% respectively when skeletons are perturbed with Gaussian noise $\mathcal{N}(0, \sigma^2)$ at $\sigma = 0.002$ and $\sigma = 0.005$ across the spatiotemporal dimensions. To understand the failure cases of our model, we highlight the five most confusing classes that are incorrectly classified by MotionBERT [12] and MaskCLR, sharing the same DSTFormer backbone. Following [2], we define the confusion score S for a pair of classes i and j as $S = n_{ij} + n_{ij}$, where n_{ij} is the number of samples of class i that are misclassified as j. The total number of pairs of classes in a dataset is $(\texttt{num\_classes}(\texttt{num\_classes} - 1))/2$. For NTU60-XSub with 60 action classes, there's 1770 pairs of classes. In Table 4, we report S_{MB} for the top five most confusing pairs for MotionBERT and the corresponding S_{MCLR} for the same pairs for MaskCLR under standard and noisy ($\sigma = 0.005$) skeletons (with skeletons extracted with HRNet [10]). We note that the five most confusing pairs account for 22.4% and 36.4% of the failure cases of MotionBERT and MaskCLR respectively on the standard skeletons. Notably, MaskCLR achieves smaller confusion scores for most of the top confusing actions for MotionBERT under both standard and noisy skeletons (top half of Table 4). On the more challenging action pairs that are most confusing for MaskCLR, our framework mispredicts fewer confusing samples than baseline MotionBERT on the standard and noisy skeletons of most action classes (bottom half of Table 4). This suggests that activating more informative joints and using contrastive losses as an optimization task enhances the classification performance of the model, particularly on the fine-grained and confusing actions.

Further, we observe that the most confusing pairs, for both MotionBERT and MaskCLR, are very similar in action semantics. For example, "play with phone/tablet" is a very challenging class because there's no sufficient joint movements that can form a pattern unique to this action. Additionally, the human pose in this action is virtually not different from "read" or "write." This motivates the need for information fusion from other modalities such as RGB images. If the model recognizes a "phone/tablet" from the RGB image, it can help clear the confusion with other classes. MaskCLR can be applied on multi-modality fusion in a similar fashion by masking the most activated regions in an input image and adopting MLCL on the encoded feature representations. We leave this for future work.

7. More Experiments

7.1. Spatial Noise

In Figure 4, we experiment with *spatial only* noise drawn from a Gaussian distribution $X \sim \mathcal{N}(0, \sigma^2)$, effectively introducing a random shift in joint positions that is constant across frames. MotionBERT-R denotes *Robust* training of MotionBERT [12] with the same DSTFormer backbone and 15% random joint masking. MaskCLR consistently shows the strongest robustness even against highly perturbed skeletons ($\sigma \geq 0.005$). Our Multi-Level Contrastive Learning (MLCL) approach boosts the model robustness against variations by mapping the input skeletons into a disentangled feature space. Furthermore, the Attention-Guided Probabilistic Masking (AGPM) strategy expands the set of discriminative joints, helping the model capture the high-level semantics of actions instead of low-level variations.

7.2. Shifted Joints

To further evaluate the robustness of our approach, we randomly shift different numbers of joints in the input skeleton sequence and report the effect on accuracy. More specifically, we shift 1, 3, 5, and 10 joints (selected randomly) in the input skeleton sequence to a random position within the skeleton bounding box. Shifted joints are commonly observed in the output of pose estimators [4, 10, 11]. As shown in Figure 5, shifted joints cause rapid drop in the accuracy of SOTA methods. In contrast, MaskCLR exhibits the lowest drop in accuracy, notably surpassing baselines MotionBERT [12]

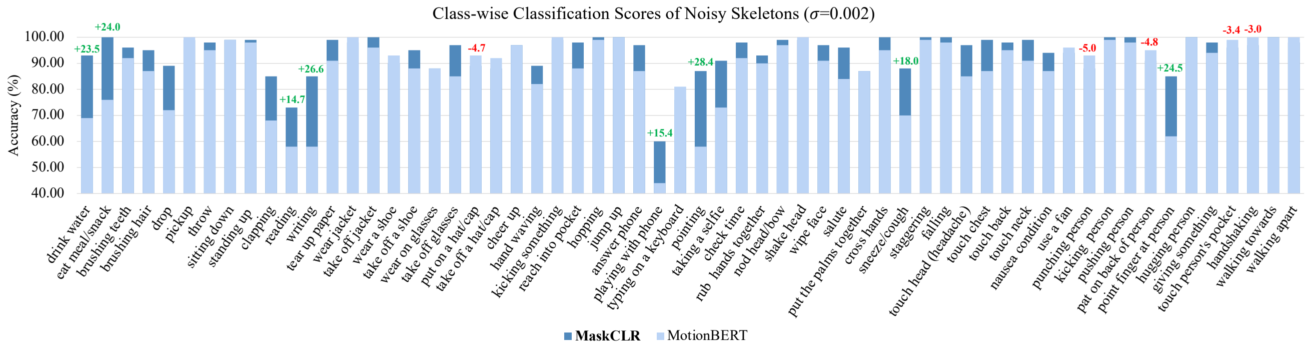


Figure 3. **Class-wise accuracy on NTU60-XSub under Gaussian noise** ($\sigma = 0.002$)**.** MaskCLR improves the classification performance in most classes, especially in subtle actions such drink water, reading, writing, etc. Our approach exploits the pose information from previously unactivated joints to reduce the confusion between low-motion action classes (Best viewed in color.)

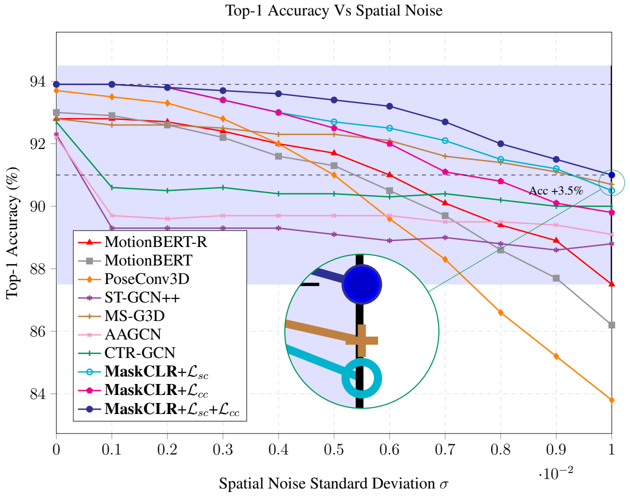


Figure 4. Top 1 accuracy on NTU60-XSub against spatial only noise. While the performance of current methods drops rapidly with noise, MaskCLR (with DSTFormer [12] backbone) shows the lowest drop in accuracy. The two contrastive losses individually contribute to enhancing the model robustness to noise.

(same backbone) by **18** percentage points when the number of shifted joints is 10.

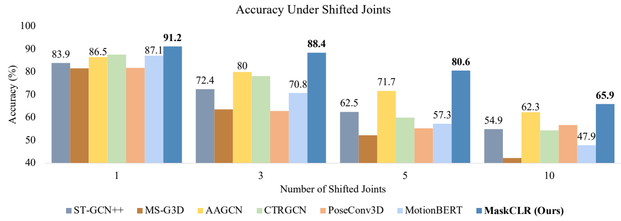
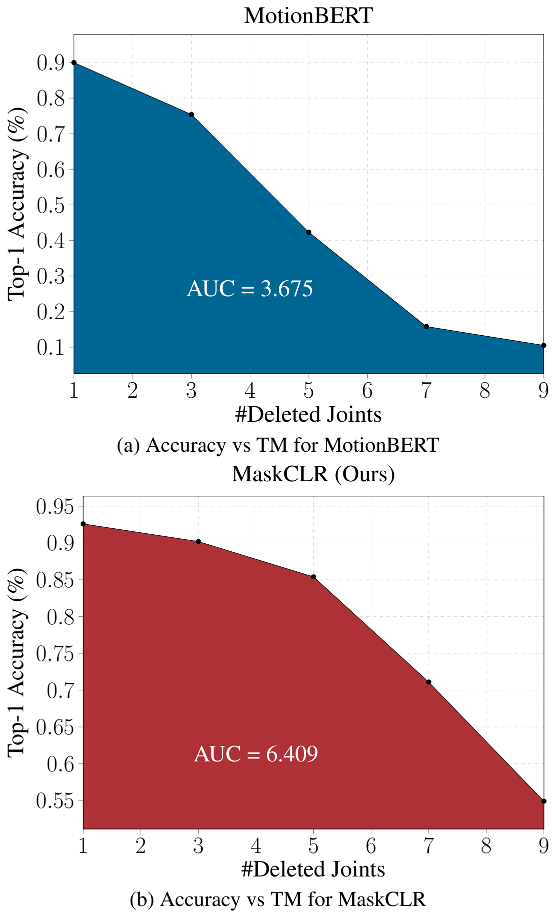


Figure 5. **Top-1 accuracy vs the number of shifted joints.**



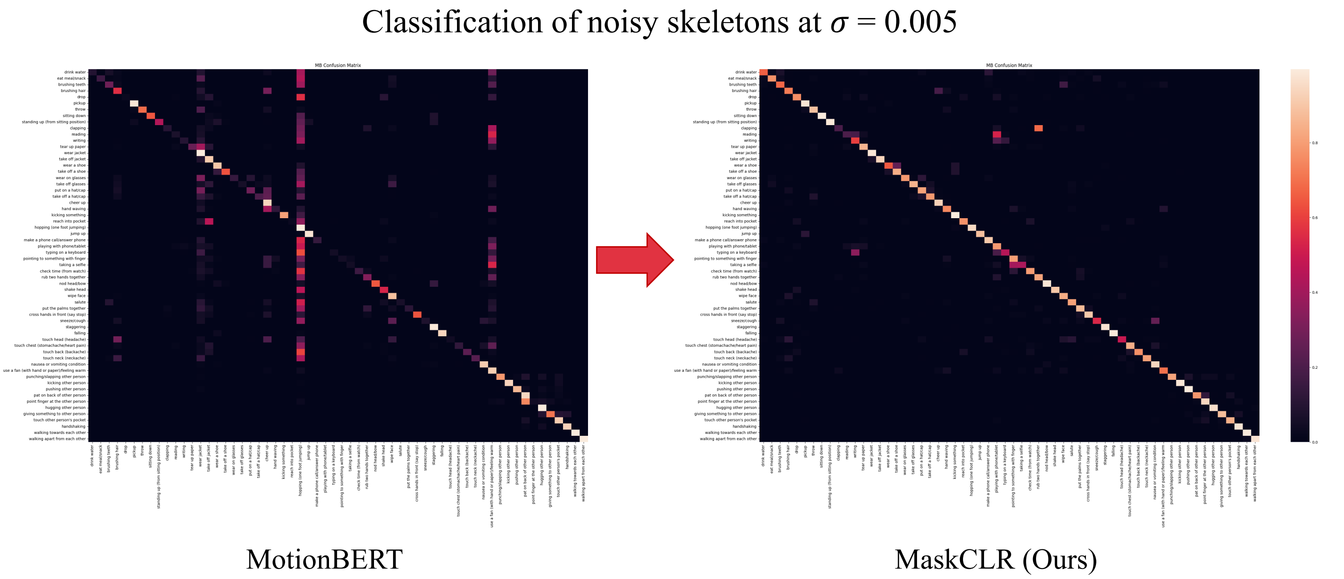
(a) Accuracy vs TM for MotionBERT

(b) Accuracy vs TM for MaskCLR

Figure 6. **Deletion score for baseline MotionBERT [12] and MaskCLR.**

7.3. Targeted Masking (TM)

In Figure 6, we compare the deletion AUC, described in section 4, for baseline MotionBERT [12] and MaskCLR based

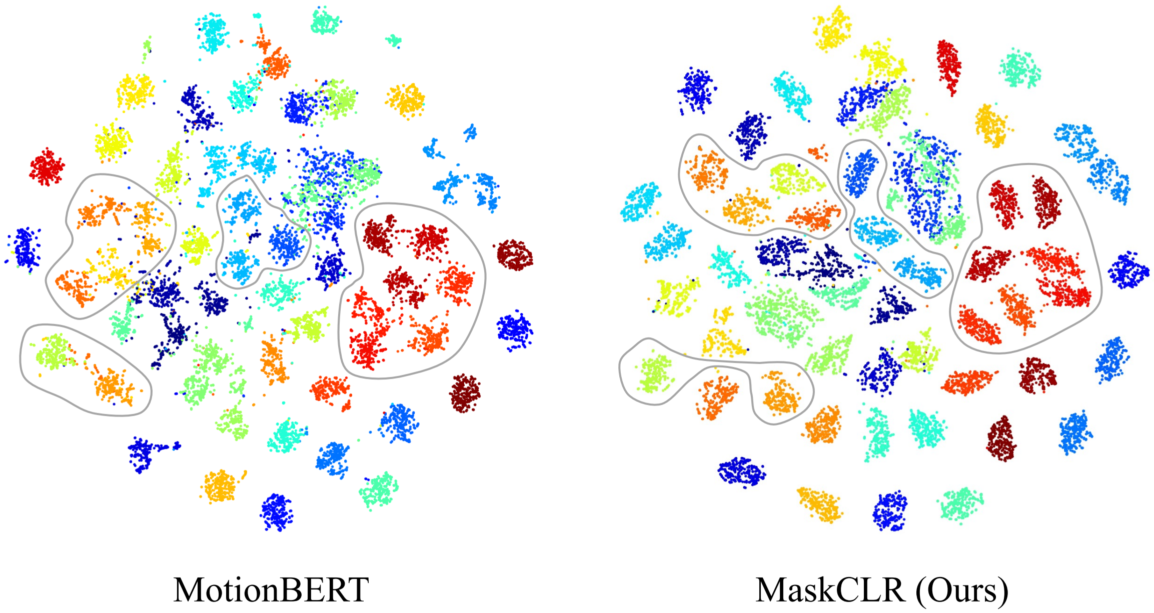


Classification of noisy skeletons at $\sigma = 0.005$

MotionBERT

MaskCLR (Ours)

Figure 7. **Confusion matrices of noisy skeletons from NTU60-XSub.** MaskCLR reduces the ratio of false positives and false negatives by establishing clearer decision boundaries between representations of different classes in the feature space.

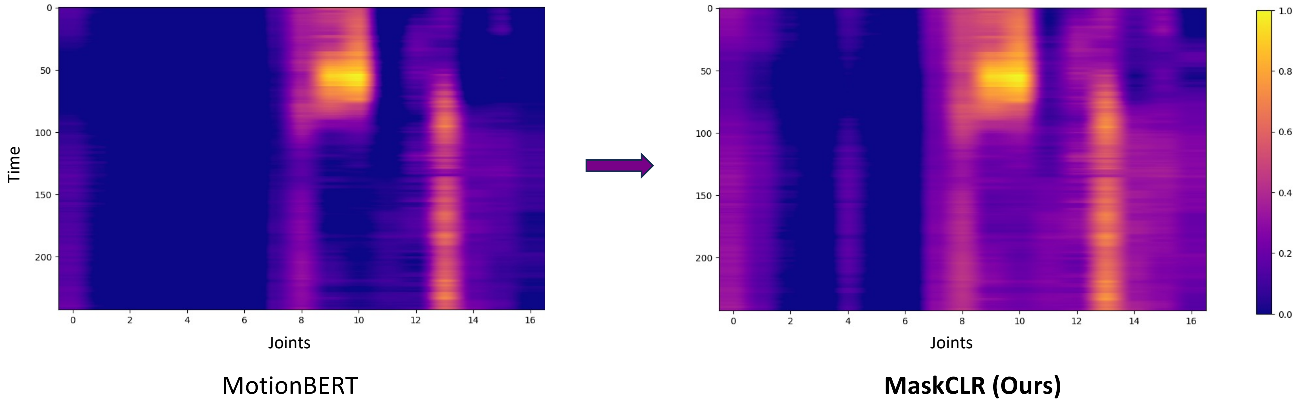


MotionBERT

MaskCLR (Ours)

Figure 8. **t-SNE visualizations of feature space on NTU60-XSub.** The feature representations of our MaskCLR is better clustered and well-disentangled compared to that of MotionBERT [12]. Our multi-level contrastive learning approach minimizes the distance between similar input skeletons at both the sample and class levels, boosting the robustness of the model against noisy or incomplete skeletons and improving the overall classification accuracy. (Best viewed in color.)

on the attention map of the last layer $N = 5$. We note that targeted masking is more challenging than random masking since the occluded joints are the ones that contribute most to the classification prediction. MaskCLR outperforms MotionBERT by **2.7** in deletion AUC. Our targeted masking strategy helps the model explore a bigger set of discriminative joints, thus alleviating the dependency on a few number of joints to recognize actions.



MotionBERT

MaskCLR (Ours)

Figure 9. **Visualization of attention scores across time of class "drink water."** Using the same DSTFormer backbone, MaskCLR activates more discriminative joints over time compared to baseline MotionBERT [12]. (Best viewed in color.)

7.4. Qualitative Results

Confusion Matrix. In Figure 7, we visualize the confusion matrices of MotionBERT [12] and MaskCLR on NTU60-XSub under spatiotemporal noise $\sigma = 0.005$. We observe that MotionBERT misclassifies most actions into high-motion classes such as "wear a jacket" and "one foot jumping." One possible explanation is that the introduced noise causes *artificial* movements in skeleton joints. While such fluctuations do not change the overall action semantics, it introduces motion to *all* joints, which typically happens in high-motion actions. Hence, the model misclassifies the sequence into a high-motion action. Focusing on low-level joint variations leads to the accuracy deterioration of MotionBERT under noisy skeletons. Instead, MaskCLR aims at capturing the high-level action semantics by utilizing a larger number of informative joints, the holistic motion of which does not change under small amounts of noise. Additionally, the rich cross-sequence intrinsic information shared between skeleton sequences of the same class is exploited through our multi-level contrastive learning approach. Consequently, MaskCLR is better able to handle perturbed skeleton sequences, as reflected in the confusion matrix (Figure 7.)

t-SNE visualization of feature space. Figure 8 shows the t-SNE visualizations of the feature space of NTU60-XSub before the final classification layer of MotionBERT and MaskCLR. We observe the feature space of our method is better disentangled across most classes, which we attribute to the added sample- and class-level contrastive losses.

Visualization of attention scores across time. Our Attention-Guided Probabilistic Masking (AGPM) strategy is designed to increase the likelihood of masking the most activated joints. In this way, AGPM encourages the model to explore the informative joints that were previously unactivated by the backbone network. To validate our approach, we inspect the visualization of the attention scores from the final MHSA layer in the DSTFormer [12] network before and after applying our framework. In Figure 9, we show the visualization of attention scores for an example sample of class "drink water." We observe that MaskCLR achieves a richer attention map by activating more discriminative joints across time. Further, the joints that were previously activated in MotionBERT (DSTFormer backbone) remain activated in MaskCLR, suggesting that the encoded information from the model are not lost when applying our appraoch. Additionally, the unactivated regions from MaskCLR are also unactivated in MotionBERT. We attribute this to the absence of sufficient information in such joints to aid in recognizing the action.

References

[1] 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. *arXiv preprint arXiv:2010.11929*, 2020. 2

[2] Haodong Duan, Yue Zhao, Kai Chen, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2969–2978, 2022. 1, 2, 4

[3] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset, 2017. 1

[4] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Openpifpaf: Composite fields for semantic keypoint detection and spatio-temporal association. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):13498–13511, 2021. 1, 2, 4

[5] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In

Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pages 740–755. Springer, 2014. 2

[6] 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. *IEEE transactions on pattern analysis and machine intelligence*, 42(10): 2684–2701, 2019. 1

[7] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018. 2

[8] Helei Qiu, Biao Hou, Bo Ren, and Xiaohua Zhang. Spatio-temporal tuples transformer for skeleton-based action recognition. *arXiv preprint arXiv:2201.02849*, 2022. 2

[9] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019, 2016. 1, 2, 4

[10] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5693–5703, 2019. 1, 2, 4

[11] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *Advances in Neural Information Processing Systems*, 35:38571–38584, 2022. 1, 2, 4

[12] Wentao Zhu, Xiaoxuan Ma, Zhaoyang Liu, Libin Liu, Wayne Wu, and Yizhou Wang. Motionbert: A unified perspective on learning human motion representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 2, 4, 5, 6, 7