ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10887969

PointActionCLIP: Preventing Transfer Degradation in Point Cloud Action Recognition with a Triple-Path CLIP

Wei Tao[1,2*], Shenglin He[1*], Xiaoyang Qu[2], Jiguang Wan[1], Jianzong Wang[2]
[1]Huazhong University of Science and Technology, Wuhan, China
[2]Ping An Technology (Shenzhen) Co., Ltd, Shenzhen, China

***Abstract*—Directly applying CLIP to point cloud action recognition can cause severe accuracy collapse. In this paper, we propose PointActionCLIP, which successfully prevents this transfer degradation with a triple-path CLIP, including the image path, the sequence path, and the label path. Specifically, the image path projects the 3D point cloud sequence onto a 2D image sequence and uses a visual encoder to extract its feature. It also captures the temporal feature of the image sequence with a temporal encoding transformer. The sequence path adopts a pre-trained sequence encoder to encode the original point cloud sequence to obtain its spatiotemporal feature. The label path encodes the candidate labels with a text encoder. Finally, we fuse the output of the three paths to obtain the predicted action label. Extensive experiments validate that PointActionCLIP outperforms state-of-the-art (SOTA) methods.**

***Index Terms*—point cloud, action recognition, triple-path, temporal encoding transformer, sequence encoder**

I. Introduction

CLIP [1] is an outstanding pre-trained language-text model which performs well on few-shot tasks [2]–[10]. In recent years, researchers have made breakthrough progress on single-frame point cloud classification [11]–[18] by applying CLIP. For example, PointCLIP [12] projects point clouds onto corresponding 2D images in different directions, extracts features using CLIP's visual encoder, and aligns features with textual labels utilizing an adapter. Building upon PointCLIP, PointCLIPv2 [11] enhances the projection method and employs prompts from GPT [19] to further reduce the feature gap between point clouds, images, and text. CLIP2Point [13] introduces a renderer to make point clouds resemble their accurate image representations more closely.

Traditional fully trained networks [20]–[25] cannot effectively handle point cloud action recognition tasks. As illustrated in Figure 1a, for an "unseen" action category, the network cannot make an accurate judgment on the sequence. The CLIP-based single-frame point cloud classification methods above provide a new paradigm for point cloud action recognition. However, directly using them for point cloud action recognition will cause severe transfer degradation.

Figure 1(b) shows the typical process of directly applying CLIP-based single-frame point cloud classification methods to point cloud action recognition tasks and its shortcomings. First, the 3D point cloud sequence is projected onto a 2D image sequence (For simplicity in drawing, we assume the number of views to be 1), which is then processed by a visual encoder to extract features. Next, candidate labels are processed through a text encoder to extract their feature. The similarities between the feature extracted from the 2D image sequence and the candidate labels are then calculated, and the label which has the highest similarity is exactly the predicted label.

This work was sponsored by the Key Research and Development Program of Guangdong Province under grant No. 2021B0101400003, the National Key Research and Development Program of China under Grant No.2023YFB4502701. The corresponding authors are Jiguang Wan from Huazhong University of Science and Technology (jgwan@hust.edu.cn) and Xiaoyang Qu from Ping An Technology (Shenzhen) Co., Ltd. (quxiaoy@gmail.com). *Equal Contribution.

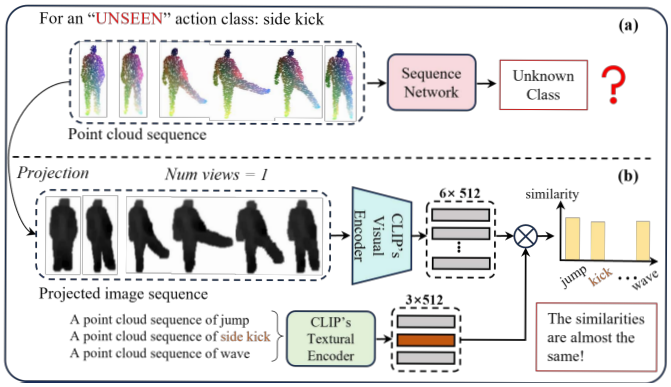

Fig. 1: (a) The traditional fully trained network cannot make an accurate judgment on an unseen label category. (b) The CLIP-based single-frame method is directly used for action recognition, where "side kick" is the true action label. However, the extracted feature has almost the same similarity to "side kick" as it does to other labels.

However, the point cloud action sequence has temporal features, while projecting each frame in the sequence onto a 2D image would result in the loss of them. This leads to poor similarity between the extracted features and the candidate action labels. In Figure 1b, this manifests as nearly identical similarity between the feature of each candidate action label and the point cloud sequence, making the model hard to distinguish the correct label.

To address this problem, this paper proposes PointActionCLIP, which is a triple-path structure CLIP. The first path is the label path, where the action label names are placed into handcrafted templates and then encoded by CLIP's pre-trained text encoder to obtain the label feature. The second path is the image path. we project every single frame point cloud in the original point cloud sequence onto a depth map. Then, we use the original visual encoder in CLIP to extract the features of the projected images sequence. We further design a temporal encoding transformer, which merges the extracted feature at various time steps within the image sequence. We call the output of the temporal encoding transformer the image feature, and we use an adapter to align the modalities of the image feature and the label feature. The third path is the sequence path. We use a pre-trained point cloud sequence encoder to extract the spatiotemporal feature from the original point cloud sequence, which is called the sequence feature. Then, we also use an adapter to align the sequence feature with the label feature. Image feature can reflect the morphological characteristics of the action represented by the point cloud sequence, while sequence feature reflect the spatiotemporal characteristics of the original point cloud sequence. Finally, we fuse the output of the three paths to obtain the predicted action label.

To sum up, our contributions can be summarized as:

- This paper proposes a new triple-path structure CLIP, called PointActionCLIP, which can effectively avoid the transfer degra-

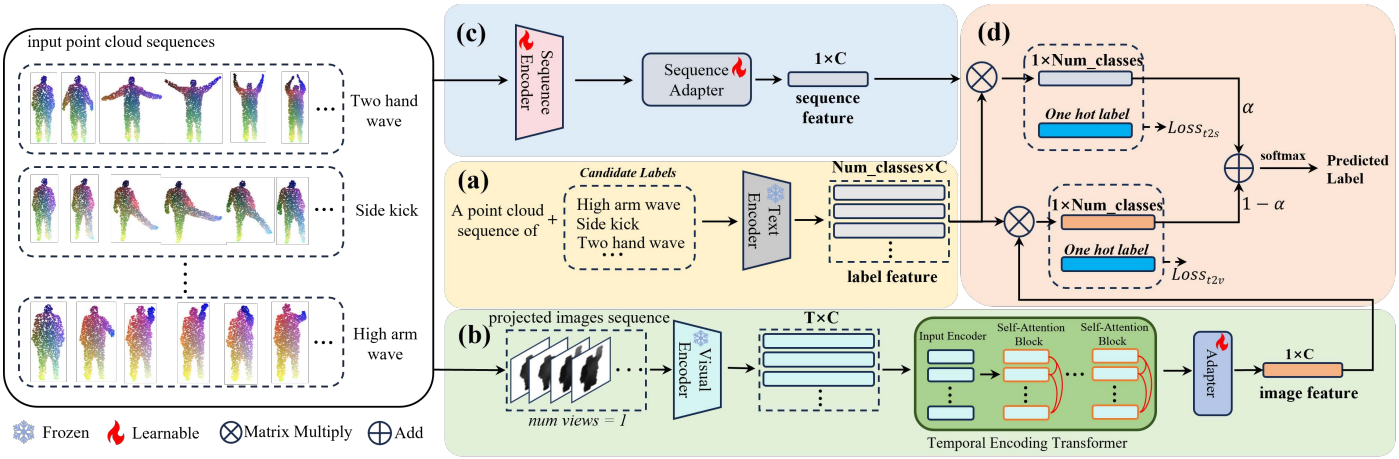

Fig. 2: The overall architecture of our PointActionCLIP. Our PointActionCLIP is a triple-path structure, each of which processes the original point cloud sequence, the projected images, and the label prompts, respectively. (a) The sequence path uses a pre-trained point cloud sequence encoder to extract the spatiotemporal feature from the original point cloud sequence. (b) The image path projects every single frame point cloud onto a depth map ((For simplicity in drawing, we assume the number of views to be 1), extracts features through CLIP's visual encoder, and uses a temporal encoding transformer to merge temporal features. (c) The label path uses CLIP's text encoder to obtain the text feature. (d) We fuse the output of the three paths to get the predicted action label.

dation issue in point cloud action recognition.

- PointActionCLIP employs the image path to capture the morphological characteristics of the action represented by the point cloud sequence.
- PointActionCLIP uses the sequence path to capture the spatiotemporal characteristics of the original point cloud sequence.
- Extensive experiments on various standard datasets prove that PointActionCLIP has better performance than current state-of-the-art (SOTA) point cloud action recognition methods.

II. Method

A. Architecture Overview

The overall architecture of our PointActionCLIP is shown in Figure 2. Our PointActionCLIP is a triple-path structure. For the input point cloud sequences, we first input several candidate action labels to the label path. Then, we input these point cloud sequences into the image path and the sequence path, respectively. Finally, we fuse the output of the three paths to obtain the predicted action label.

B. The Label Path

We place M action label names into a handcrafted template: "a point cloud sequence of [CLASS]" to form several label prompts, which are then encoded by CLIP's pre-trained textual encoder to get the final label feature $\mathcal{F}_{label} \in \mathbb{R}^{M\times C}$, where C is the output dimension of the text encoder for each label prompt.

C. The Image Path

Given the input point cloud dataset $\mathcal{S} = \{S^i\}_{i=1}^{|\mathcal{S}|}$, where $S^i \in \mathbb{R}^{3\times P\times T}$ represents a point cloud sequence, the P and T represent the number of points in each frame and the number of frames in the point cloud sequence respectively. Initially, We project each frame in the point cloud sequence S^i onto a depth map. The projected images path is denoted as $G^i = \{g_1^i, g_2^i, \ldots, g_T^i\}$. Then, we use CLIP's visual encoder to extract the feature of this projected images sequence. We denote the extracted feature as $\mathcal{F}_G^i \in \mathbb{R}^{T\times C}$. This feature does not contain the temporal information of the sequence, so we design an additional temporal encoding transformer to enable it to understand the temporal information of the entire sequence. Next, we will introduce the temporal encoding transformer in detail.

The temporal encoding transformer consists of an input encoder and several self-attention blocks. For the t_{th} image F_t^i in the projected image sequence G_i, the input encoder uses its position subscript t to encode its temporal information, where $1 \le t \le T$. The length of the encoded vector is C, where the j_{th} component is:

$$PV_{t,j}^i = \begin{cases} \sin(\omega_j t), & \text{if } j \equiv 0 \pmod 2 \\ \cos(\omega_j t), & \text{if } j \equiv 1 \pmod 2 \end{cases} \tag{1}$$

where $\omega_j = \frac{1}{10000^{2j/C}}$ and $j = 0, 1, 2, \ldots, C$. Through the above formula, we can get our position vector $P^i \in \mathbb{R}^{T\times C}$ as:

$$P^i = (\{(PV_{0,0}^i), \ldots, (PV_{0,C}^i)\}, \ldots, \{(PV_{T,0}^i), \ldots, (PV_{T,C}^i)\}) \tag{2}$$

We add this position vector element-wise to $\mathcal{F}_G^i$ to get the encoded feature I^i, which is the input of the following self-attention module. Then, we use self-attention block [26] to process I^i. Self-attention can help each image in the projected images sequence learn its relationships with images from different positions in the sequence, thus capturing the implicit temporal information within the sequence. We pass I^i through several self-attention blocks and get the final output called the image feature.

To align the modalities of 2D images and text, We further add a learnable adapter after the temporal encoding transformer to process the image feature. This adapter consists of two linear layers and a LayerNorm. We denote the aligned image feature as $\mathcal{F}_{image}^i \in \mathbb{R}^{1\times C}$.

D. The Sequence Path

We input the original point cloud sequence S^i to a sequence encoder to capture its spatiotemporal feature. We choose the PST-Transformer as the network backbone of our sequence encoder, which is pre-trained on other point cloud sequence datasets (For example, if it is tested on the MSR-Action3D dataset, then it is pre-trained on the NTU RGB+D dataset). We call the output of the sequence encoder the sequence feature, which is denoted as $\mathcal{F}_S^i \in \mathbb{R}^{1\times D}$, where D is the output dimension of the sequence encoder. Through the sequence encoder, we can directly capture the spatiotemporal feature from the original point cloud sequence, avoiding the influence of image projection.

TABLE I: Accuracy performance (%) comparison between PointActionCLIP and SOTA 3D point cloud sequence networks on MSR-Action3D, NTU RGB+D 60 and NTU RGB+D 120. Our PointActionCLIP shows consistent superiority to other models under 4, 8, and 16-shot settings.

Setting	Method	MSR-Action3D	NTU RGB+D 60		NTU RGB+D 120	
			Cross-view	Cross-subject	Cross-view	Cross-subject
4-shot	PSTNet++ [27]	77.92	78.19	73.64	73.59	70.13
	SequentialPointNet [28]	76.30	78.55	73.10	75.52	70.25
	P4Transformer [29]	69.15	71.16	66.13	68.61	64.72
	PST-Transformer [30]	70.50	72.55	68.44	69.75	66.80
	KiNet [31]	79.80	79.88	74.32	75.02	70.21
	3DInAction [32]	80.47	80.32	75.03	76.14	71.95
	PointActionCLIP	**81.69**	**83.07**	**77.36**	**79.87**	**76.17**
8-shot	PSTNet++ [27]	85.75	89.15	84.16	83.63	79.87
	SequentialPointNet [28]	85.94	88.35	84.35	84.54	80.73
	P4Transformer [29]	84.61	86.99	82.07	82.54	78.79
	PST-Transformer [30]	87.52	89.98	84.89	86.41	81.49
	KiNet [31]	83.84	89.77	84.25	86.23	81.47
	3DInAction [32]	86.20	90.33	85.21	87.43	82.38
	PointActionCLIP	**90.67**	**93.23**	**87.95**	**89.53**	**84.43**
16-shot	PSTNet++ [27]	87.40	89.86	85.77	86.29	82.38
	SequentialPointNet [28]	87.59	90.06	86.96	86.49	81.57
	P4Transformer [29]	86.24	88.67	83.65	85.15	80.30
	PST-Transformer [30]	88.21	90.69	85.56	87.09	82.13
	KiNet [31]	91.92	93.79	87.23	88.84	84.12
	3DInAction [32]	88.22	91.34	86.45	88.72	83.46
	PointActionCLIP	**92.42**	**95.02**	**89.65**	**91.25**	**86.06**

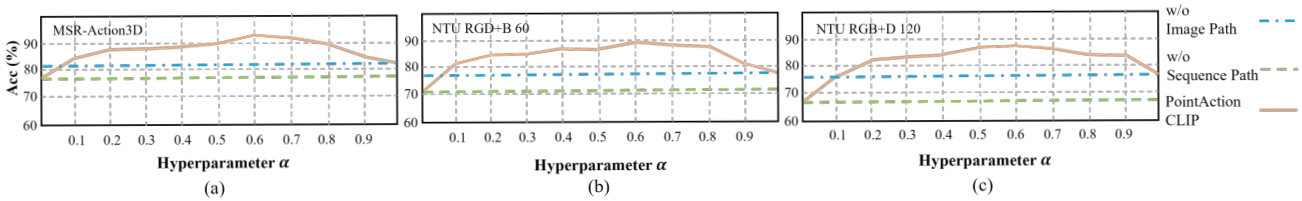

Fig. 3: The respective recognition accuracy of each path and the impact of different α settings for the fusion accuracy on MSR-Action3D, NTU RGB+D 60 and NTU RGB+D 120.

Similarly, to align modalities of the 3D point cloud and text, we add a sequence adapter to further process the cloud feature. This adapter has the same network structure as the adapter in the image path, including two linear layers and a LayerNorm. We denote the aligned sequence feature as $\mathcal{F}^i_{sequence} \in \mathbb{R}^{1\times C}$.

E. Triple Path Fusion

Finally, we fuse the three paths to obtain the final predicted action label. Specifically, we compute the similarity between the aligned image feature feature and the label feature. The similarity formula is as follows:

$$\mathrm{sim}^i_{image} = \frac{\mathcal{F}^i_{image} \cdot \mathcal{F}_{label}}{||\mathcal{F}^i_{image}|| \times ||\mathcal{F}_{label}||} \tag{3}$$

where $||\mathcal{F}^i_{image}||$ and $||\mathcal{F}^i_{label}||$ denotes the L2 norms of the projected images feature and the label prompt feature, respectively. We use the cross entropy loss function to calculate the loss between sim^i_{image} and the one hot action labels vector, which is denoted as $Loss_{t2v}$. Similarly, we compute the similarity between the aligned sequence feature and the label feature as $\mathrm{sim}^i_{sequence}$. We also use the cross entropy loss function to calculate the loss between $\mathrm{sim}^i_{sequence}$ and the one hot action labels vector, which is denoted as $Loss_{t2s}$.

sim^i_{image} pays more attention to the overall temporal characteristics of the point cloud sequence, while $\mathrm{sim}^i_{sequence}$ pays more attention to the difference of the point cloud frames belonging to various action categories. These two similarities are complementary. During training, we train the image feature and the sequence feature separately using different loss functions so that they can be aligned to the final label prompt word simultaneously. During inference, we use a hyper-parameter α to weight and add these two similarities as follows:

$$\mathrm{logits}^i = \alpha \mathrm{sim}^i_{sequence} + (1-\alpha)\mathrm{sim}^i_{image} \tag{4}$$

Our final label probability vector will be obtained by a softmax function:

$$p^i = \left\{p^i_j\right\}_{j=1}^{M} = \frac{\exp(\mathrm{logits}^{\mathrm{i}}_{\mathrm{j}})}{\sum_{m=1}^{M}\exp(\mathrm{logits}^{\mathrm{i}}_{\mathrm{m}})} \tag{5}$$

where the label with the highest probability in p^i will be chosen as the predicted action label.

III. Experiment

A. Experiment Setup

Dataset. We use two widely-used standard public datasets: MSR-Action 3D [33] and NTU RGB+D [34]. MSR-Action 3D contains 567 videos and 23k frames, belonging to 20 different action categories. The action categories cover common actions in daily life, such as shaking hands, waving, running, jumping, etc. NTU RGB+D 60 [34] is a commonly used human action recognition dataset, including 56,880 videos composed of 60 action classes. NTU RGB+D 120 [35] is an extended version of NTU RGB+D 60 with an additional 60 extra action classes and contains 114,480 videos.

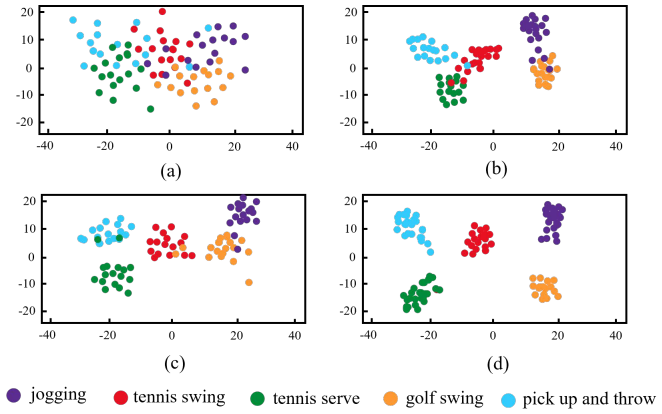

Fig. 4: UMAP figure of feature distributions on MSR-Action3D. (a) Directly using the CLIP-based single-frame point cloud classification method. (b) PointActionCLIP without image path. (c) PointActionCLIP without sequence path. (d) PointActionCLIP.

Baselines. We compare PointActionCLIP with 6 SOTA point cloud action recognition methods: PSTNet++ [27], SequentialPointNet [28], P4Transformer [29], PST-Transformer [36], KiNet [31] and 3DInAction [32], where P4 Transformer and PST-Transformer are based on self-attention mechanism, and the other four methods are based on traditional convolutional networks.

Implementation Details. Since the number of frames and points in each sequence in the datasets is different, we need to preprocess them. For each sequence, we sample 24 frames equidistantly. For each frame of the point cloud in the sequence, we sample 2048 points using the Farthest Point Sampling algorithm. Besides, in the image path, we project each frame in the point cloud sequence to obtain a depth image of size 224×224. We choose ResNet-101 [37] and PST-Transformer [30] as the network backbones of our visual encoder and sequence encoder, respectively.

B. Performance

As shown in Table I, we present the performance of our PointActionCLIP and compare it with the baseline methods mentioned above. As we can see, our triple-path structure PointActionCLIP achieves the best results on the three datasets, especially in the 4-shot setting. Specifically, our method improves the accuracy by 5.20%, 4.24%, and 3.98% on average compared to other methods in the 4-shot, 8-shot, and 16-shot settings, respectively. This is because our method captures the temporal feature of the projected 2D images sequence and the spatio-temporal feature of the original point cloud sequence. Additionally, our method utilizes CLIP, which performs exceptionally well in few-shot tasks. Therefore, when there are fewer training samples, PointActionCLIP can perform much better than other methods. As the number of training samples increases, the lead of our method decreases slightly, but it still has the highest recognition accuracy among all methods.

C. Ablation study

We examine the recognition accuracy of our method without the image path and without the sequence path, and the accuracy of the whole method with different parameter α. As illustrated in Figure 3, on the three datasets, the recognition accuracy of our method without the image path is always 3% - 7% higher than that of the method without the sequence path. This is because for point cloud action recognition tasks, sometimes the depth maps projected by different action categories are roughly the same, which means that the spatial structure information of the original point cloud is more important to the final result.

TABLE II: Accuracy performance (%) of different network backbones of the sequence encoder on MSR-Action 3D.

Method	PSTNet	P4Transformer	**PST-Transformer**
w/o Image Path	75.85	76.63	**81.23**
w Image Path	87.10	86.24	**92.42**

TABLE III: Accuracy performance (%) of different network backbones of the visual encoder on MSR-Action 3D. RN50 and ViT-B/32 denote ResNet-50 and vision transformer [38] where each patch has a size of 32×32, respectively. RN50×16 is a variant of ResNet-50 with 16 times more computations.

Method	RN50	**RN101**	ViT/32	ViT/16	RN50x4	RN50x16
w/o Sequence Path	76.63	**78.53**	75.49	76.88	77.07	77.36
w Sequence Path	91.18	**92.42**	89.34	90.47	89.32	91.04

To further visualize the influence of the image path and the sequence path in PointActionCLIP, we present the feature distribution of the inference results on the MSR-Action3D dataset using UMAP [39] (a visualization algorithm) in Figure 4. Figure 4(a) shows the results of directly using the CLIP-based single-frame point cloud classification method, where the feature vectors of different categories are completely mixed together, indicating that the model is unable to distinguish the true action labels to which these point cloud sequences belong. Figure 4(b) and Figure 4(c) respectively show the cases of PointActionCLIP without the image path and PointActionCLIP without the sequence path. From the figures, it can be seen that without either of the two paths, PointActionCLIP can somewhat distinguish the true action labels of these point cloud sequences, but it still cannot achieve complete separation, as some sequences are misclassified into other labels (corresponding to a few scattered points of a different color in densely packed regions of a certain color in the figures). Figure 4(d) shows the case of the complete PointActionCLIP, where it can be seen that these point cloud sequences are fully distinguished. These experiments demonstrate the effectiveness and necessity of our image path and sequence path.

Finally, we conduct ablation experiments to prove the effectiveness of the image path and the sequence path given the parameter $\alpha = 0.6$. For the sequence path, we conduct ablation experiments with different backbones of sequence encoders. The experimental results are shown in Table II. It can be seen that, whether with or without the image path, PST-Transformer achieves the highest recognition accuracy. For the image path, as shown in Table III, we explore the impact of different network backbones of the visual encoder. The table illustrates that, whether with or without the sequence path, using RN101 as the visual encoder network backbone achieves the highest recognition accuracy.

IV. Conclusion

We propose a triple-path CLIP method for point cloud action recognition named PointActionCLIP to prevent CLIP's transfer accuracy degradation. Our method is a triple-path structure, where each path processes the original point cloud sequence, projected images sequence, and label prompts, respectively. Finally, we fuse the output of the three paths to achieve the predicted action label. We conduct extensive experiments on several standard point cloud action recognition datasets. Compared to other SOTA point cloud action recognition methods, our PointActionCLIP achieves the best performance in different settings.

REFERENCES

[1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.

[2] M. Ge, Y. Li, H. Wu, and M. Li, "Jm-clip: A joint modal similarity contrastive learning model for video-text retrieval," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3010–3014, IEEE, 2024.

[3] N. Liang, Y. Liu, W. Sun, Y. Xia, and F. Wang, "Ckt-rcm: Clip-based knowledge transfer and relational context mining for unbiased panoptic scene graph generation," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3570–3574, IEEE, 2024.

[4] J. Xiong, Y. Wang, and J. Zeng, "Clip-font: Sementic self-supervised few-shot font generation with clip," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3620–3624, IEEE, 2024.

[5] Z. Yin, J. Wang, J. Cao, Z. Shi, D. Liu, M. Li, X. Huang, Z. Wang, L. Sheng, L. Bai, *et al.*, "Lamm: Language-assisted multi-modal instruction-tuning dataset, framework, and benchmark," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[6] S. Shen, L. H. Li, H. Tan, M. Bansal, A. Rohrbach, K.-W. Chang, Z. Yao, and K. Keutzer, "How much can clip benefit vision-and-language tasks?," *arXiv preprint arXiv:2107.06383*, 2021.

[7] R. Mokady, A. Hertz, and A. H. Bermano, "Clipcap: Clip prefix for image captioning," *arXiv preprint arXiv:2111.09734*, 2021.

[8] Y. Liu, Y. Li, Z. Liu, W. Yang, Y. Wang, and Q. Liao, "Clip-based synergistic knowledge transfer for text-based person retrieval," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7935–7939, IEEE, 2024.

[9] Z. Lin, S. Geng, R. Zhang, P. Gao, G. De Melo, X. Wang, J. Dai, Y. Qiao, and H. Li, "Frozen clip models are efficient video learners," in *European Conference on Computer Vision*, pp. 388–404, Springer, 2022.

[10] Y. Rao, W. Zhao, G. Chen, Y. Tang, Z. Zhu, G. Huang, J. Zhou, and J. Lu, "Denseclip: Language-guided dense prediction with context-aware prompting," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18082–18091, 2022.

[11] X. Zhu, R. Zhang, B. He, Z. Guo, Z. Zeng, Z. Qin, S. Zhang, and P. Gao, "Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2639–2650, 2023.

[12] R. Zhang, Z. Guo, W. Zhang, K. Li, X. Miao, B. Cui, Y. Qiao, P. Gao, and H. Li, "Pointclip: Point cloud understanding by clip," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8552–8562, 2022.

[13] T. Huang, B. Dong, Y. Yang, X. Huang, R. W. Lau, W. Ouyang, and W. Zuo, "Clip2point: Transfer clip to point cloud classification with image-depth pre-training," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22157–22167, 2023.

[14] X. Yan, H. Zhan, C. Zheng, J. Gao, R. Zhang, S. Cui, and Z. Li, "Let images give you more: Point cloud cross-modal training for shape analysis," *Advances in Neural Information Processing Systems*, vol. 35, pp. 32398–32411, 2022.

[15] R. Chen, Y. Liu, L. Kong, X. Zhu, Y. Ma, Y. Li, Y. Hou, Y. Qiao, and W. Wang, "Clip2scene: Towards label-efficient 3d scene understanding by clip," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7020–7030, 2023.

[16] X. Huang, Z. Huang, S. Li, W. Qu, T. He, Y. Hou, Y. Zuo, and W. Ouyang, "Frozen clip transformer is an efficient point cloud encoder," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 2382–2390, 2024.

[17] Z. Guo, R. Zhang, L. Qiu, X. Ma, X. Miao, X. He, and B. Cui, "Calip: Zero-shot enhancement of clip with parameter-free attention," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 746–754, 2023.

[18] Y. Zeng, C. Jiang, J. Mao, J. Han, C. Ye, Q. Huang, D.-Y. Yeung, Z. Yang, X. Liang, and H. Xu, "Clip2: Contrastive language-image-point pretraining from real-world point cloud data," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15244–15253, 2023.

[19] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[20] S. Zhao and S. Yan, "Bounding box-guided pseudo point clouds early-fusion and density optimize for 3d object detection," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4205–4209, IEEE, 2024.

[21] T. Hong, Z. Zhang, and J. Ma, "Pcsalmix: Gradient saliency-based mix augmentation for point cloud classification," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, IEEE, 2023.

[22] R. Li, X. Li, P.-A. Heng, and C.-W. Fu, "Pointaugment: an auto-augmentation framework for point cloud classification," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6378–6387, 2020.

[23] H. Wang, L. Yang, X. Rong, J. Feng, and Y. Tian, "Self-supervised 4d spatio-temporal feature learning via order prediction of sequential point cloud clips," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3762–3771, 2021.

[24] P. Kadam, H. Prajapati, M. Zhang, J. Xue, S. Liu, and C.-C. J. Kuo, "S3i-pointhop: So (3)-invariant pointhop for 3d point cloud classification," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, IEEE, 2023.

[25] Y. Wu, K. Song, X. Huang, and D. Zhang, "Mitigating intra-class variance in few-shot point cloud classification," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6330–6334, IEEE, 2024.

[26] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[27] H. Fan, X. Yu, Y. Yang, and M. Kankanhalli, "Deep hierarchical representation of point cloud videos via spatio-temporal decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 9918–9930, 2021.

[28] X. Li, Q. Huang, Z. Wang, Z. Hou, and T. Yang, "Sequentialpointnet: A strong parallelized point cloud sequence network for 3d action recognition," *arXiv preprint arXiv:2111.08492*, 2021.

[29] H. Fan, Y. Yang, and M. Kankanhalli, "Point 4d transformer networks for spatio-temporal modeling in point cloud videos," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14204–14213, 2021.

[30] H. Fan, Y. Yang, and M. Kankanhalli, "Point spatio-temporal transformer networks for point cloud video modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 2181–2192, 2022.

[31] J.-X. Zhong, K. Zhou, Q. Hu, B. Wang, N. Trigoni, and A. Markham, "No pain, big gain: classify dynamic point cloud sequences with static models by fitting feature-level space-time surfaces," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8510–8520, 2022.

[32] Y. Ben-Shabat, O. Shrout, and S. Gould, "3dinaction: Understanding human actions in 3d point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19978–19987, 2024.

[33] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," in *2010 IEEE computer society conference on computer vision and pattern recognition-workshops*, pp. 9–14, IEEE, 2010.

[34] A. Shahroudy, J. Liu, T.-T. Ng, and G. 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*, pp. 1010–1019, 2016.

[35] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 10, pp. 2684–2701, 2019.

[36] H. Fan, X. Yu, Y. Ding, Y. Yang, and M. Kankanhalli, "Pstnet: Point spatio-temporal convolution on point cloud sequences," *arXiv preprint arXiv:2205.13713*, 2022.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[38] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[39] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.