MAPLE: Masked Pseudo-Labeling autoEncoder for Semi-supervised Point Cloud Action Recognition

Xiaodong Chen*
cxd1230@mail.ustc.edu.cn
University of Science and Technology of China

Wu Liu†
liuwu1@jd.com
JD Explore Academy

Xinchen Liu
liuxinchen1@jd.com
JD Explore Academy

Yongdong Zhang
zyd73@ustc.edu.cn
University of Science and Technology of China

Jungong Han
jungong.han@aber.ac.uk
Aberystwyth University

Tao Mei
tmei@jd.com
JD Explore Academy

ABSTRACT

Recognizing human actions from point cloud videos has attracted tremendous attention from both academia and industry due to its wide applications like automatic driving, robotics, and so on. However, current methods for point cloud action recognition usually require a huge amount of data with manual annotations and a complex backbone network with high computation cost, which makes it impractical for real-world applications. Therefore, this paper considers the task of semi-supervised point cloud action recognition. We propose a Masked Pseudo-Labeling autoEncoder (**MAPLE**) framework to learn effective representations with much fewer annotations for point cloud action recognition. In particular, we design a novel and efficient **De**coupled **s**patial-**t**emporal Trans**Former** (**DestFormer**) as the backbone of MAPLE. In DestFormer, the spatial and temporal dimensions of the 4D point cloud videos are decoupled to achieve an efficient self-attention for learning both long-term and short-term features. Moreover, to learn discriminative features from fewer annotations, we design a masked pseudo-labeling autoencoder structure to guide the DestFormer to reconstruct features of masked frames from the available frames. More importantly, for unlabeled data, we exploit the pseudo-labels from the classification head as the supervision signal for the reconstruction of features from the masked frames. Finally, comprehensive experiments demonstrate that MAPLE achieves superior results on three public benchmarks and outperforms the state-of-the-art method by 8.08% accuracy on the MSR-Action3D dataset. [1]

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**; **Activity recognition and understanding**.

*This work is done when Xiaodong Chen is an intern at JD Explore Academy.
†Wu Liu is the corresponding author.
[1]See the project on www.xiaodongchen.cn/MAPLE/.

MM '22, October 10–14, 2022, Lisboa, Portugal.

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3547892

KEYWORDS

Point Cloud Action Recognition, Semi-supervised Learning, Autoencoder, Vision Transformer

ACM Reference Format:
Xiaodong Chen, Wu Liu, Xinchen Liu, Yongdong Zhang, Jungong Han, and Tao Mei. 2022. MAPLE: Masked Pseudo-Labeling autoEncoder for Semi-supervised Point Cloud Action Recognition. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3503161.3547892

1 INTRODUCTION

Point cloud videos, compared with 2D RGB videos, contain richer visual and geometric information for action recognition. Researchers from academia and industry have recently focused on point cloud action recognition due to its wide potential applications in autonomous driving, industrial manufacturing, robotics, and so on [14, 16, 17, 23, 32, 40]. With the development of deep learning techniques such as deep neural networks and the transformer [35], significant progress has been made in this task [20, 25]. However, the high computation cost and the requirement of large-scale annotated data hinder the practical application of point cloud action recognition.

To recognize human actions in point cloud videos, the mainstream methods are divided into three categories. The first one [36] is to convert the point cloud video into a series of ordered voxels and then apply traditional grid-based convolutions to these voxels. The second type of method [20, 26] is to model and track local points with pointnet-based [25] models such as MeteorNet [20]. However, these two types of methods suffer from low computational efficiency and point-tracking errors [20] respectively. To address these problems, He et al. [6] proposed the third method that extracts short-term local features by 4D convolutions and models long-term global information with the transformer. Nevertheless, the transformer-based methods usually require large-scale labeled data for training as the transformer has a larger model capacity [39]. Although a large amount of point cloud videos can be easily obtained, labeling point cloud often needs much more cost on manual annotations compared to 2D RGB videos, which hinders the application of these methods. Therefore, this paper focuses on the task of semi-supervised point cloud action recognition, which aims to reduce the reliance on manual labels in point cloud action recognition using a more efficient model.

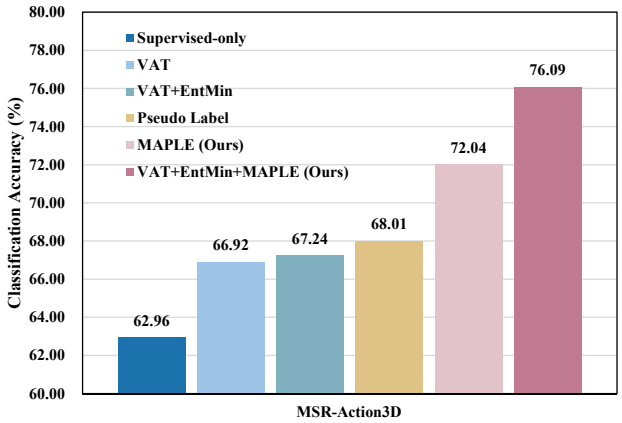


Figure 1: Comparisons of different semi-supervised methods, i.e., VAT [24], EntMin [7], and Pseudo Label [13] on the MSR-Action3D dataset in terms of classification accuracy.

Although the accuracy of current methods has been greatly improved, designing an annotation and computation-efficient framework for point cloud action recognition still faces several challenges. First of all, due to the noises and ambiguity of point clouds, how to learn discriminative features and model the spatial-temporal patterns from the point clouds is a great challenge. In image classification, researchers have studied combining techniques of CNNs with the transformer to improve the capability while reducing the computation complexity of the transformer models. For example, Swin-Transformer [22] greatly enhances the capacity of the transformer while improving its efficiency by the shifted windows, which demonstrates the potential of the transformer in the modality of RGB images. However, such models are limited on the point cloud action recognition task due to the irregularity of the point clouds.

The other challenge is how to reduce the dependence on manual annotations while preserving the capability of the learned feature representations through an appropriate learning paradigm. A common and effective learning framework is semi-supervised learning, which has rich applications in the field of image recognition and video understanding. Besides, Self-Supervised Learning (SSL) is also a powerful learning framework that exploits the generalizable representations from unlabeled data. In particular, some recent research on the field of SSL [4, 9, 10] has shown excellent results, yet self-supervision alone is still insufficient due to its limited practical applicability. To solve this dilemma, Zhai et al. [1] propose a new learning framework that combines self-supervised learning and semi-supervised learning and becomes a new paradigm in the semi-supervised field. However, limited by the invariance and unordered properties [25] of the point clouds, such methods cannot be directly applied to point cloud action recognition.

To overcome these challenges, we propose a novel learning framework named Masked Pseudo-Labeling autoEncoder (**MAPLE**) for point cloud action recognition. It introduces an autoencoder into the semi-supervised point cloud action recognition task. We also design an efficient transformer-based model named **De**coupled **s**patial-**t**emporal Trans**Former** (**DestFormer**) for this new learning framework. Based on this DestFormer backbone, we design an encoder-decoder structure for MAPLE. It consists of a spatial extractor for learning short-term global features of actions, a temporal encoder for learning long-term action information, and a temporal decoder for feature reconstruction. To learn action information from the unlabeled action sequences, we reconstruct the masked action sequence with a highly masking ratio (e.g. 75 %) during the training process. However, directly reconstructing the video action sequence tends to result in the non-convergence of the model training and a decrease in classification performance. Inspired by the knowledge distillation [11], we implicitly reconstruct the masked input sequence with the pseudo-label generated by the classification head to avoid this situation. Besides, to exploit the potential of MAPLE, we further combine MAPLE with the classical semi-supervised learning methods to improve the performance of semi-supervised point cloud action recognition.

We conduct extensive experiments with our MAPLE framework for semi-supervised point cloud action recognition on three widely-used datasets: MSR-Action3D (MSR3D) [14], NTU RGB+D 60 (NTU60) [29], and NTU RGB+D 120 (NTU120) [15]. As shown in Fig. 1, we make remarkable progress in the mainstream datasets compared to previous methods, e.g., VAT [24], EntMin [7], and Pseudo Label [13]). The MAPLE framework achieves the new state-of-the-art performance of the semi-supervised point cloud action recognition.

In summary, the contributions of this paper are three-fold:

- We present one of the first attempts toward semi-supervised point cloud action recognition which aims to learn efficient action representations from massive point cloud videos with fewer manual annotations.
- We design a **De**coupled **s**patial-**t**emporal Trans**Former**, named **DestFormer**, as the backbone of our semi-supervised learning framework, which decouples the spatial and temporal dimensions of the 4D point cloud videos for achieving a more efficient and effective self-attention.
- We propose a Masked Pseudo-Labeling autoEncoder (**MAPLE**) framework for learning a generalizable and discriminative classifier through reconstructing motion features of masked frames from the available action frames.

2 RELATED WORK

Point cloud action recognition is a popular topic of video understanding in computer vision, which aims to help the machines understand the 3D world. Based on the characteristics of previous methods, three main categories have been distinguished. The first one is mainly based on voxels obtained from point clouds. 3D dynamic voxel (3DV) [36] brought the voxelization of point clouds into point cloud action recognition, via temporal rank pooling. It learned both action information through voxels and appearance through point cloud to encode the temporal information. The second type of method is directly performed on the original point cloud with pointnet-based [25] models. For example, pointnet++ [26] borrowed the idea of local receptive fields to extract the spatial information of point clouds. MeteorNet [20] further constructed the concept of spatial-temporal neighborhoods based on pointnet++ and determined the neighborhoods with direct grouping or chained-flow grouping. The last category adopts the data-hungry transformer-based model in point cloud action recognition. P4Transformer [6]

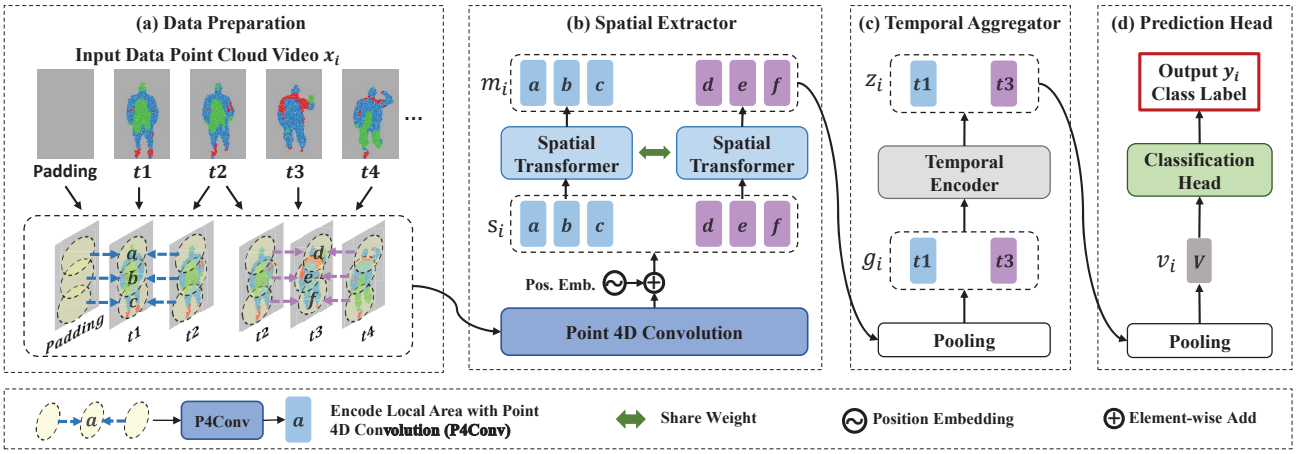


Figure 2: The detail of DestFormer. (a) *Data Preparation*: we construct some local areas (e.g. "a") on adjacent frames (e.g. "t1", "t2") from the input x_i as what P4Conv [6] do. (b) *Spatial Extractor*: we adopt P4Conv for modeling short-time local information and feed the output s_i frame by frame into a spatial transformer for extracting the merged local feature m_i. (c) *Temporal Aggregator*: we generate the short-term global feature g_i through the pooling layer and aggregate the long-term global information with the temporal encoder. (d) *Prediction Head*: we project the global feature v_i into label space via the classification head.

directly modeled the action and appearance information of the whole video while effectively discarding the requirement of point-tracking used in MeteorNet and the complex calculations of voxelization. Similarly, PST^2 [37] captured the spatial-temporal context information with the Spatial-temporal self-attention module. In this paper, our DestFormer belongs to the last category, but has less Floating-point Operations (FLOPs), powerful model capability, and less annotation dependence.

Semi-Supervised Learning is an important research topic in the field of pattern recognition and machine learning, which learns knowledge from the datasets including the much more set of unlabeled data and fewer labeled data. The theory and algorithms of semi-supervised learning were first summarized by Chapelle [2] in 2006 and Zhu [38] in 2008. The semi-supervised learning methods can be divided into two categories: the inductive methods and the transductive methods. The inductive method [18, 30, 33] usually constructs a classifier for predicting the label of the whole dataset, including both the labeled and unlabeled data. By way of illustration, Grandvalet and Bengio [7] optimized the pseudo label generated by unlabeled data with conditional entropy minimization. Miyato et al. [24] added the small perturbations to the original input and constrained the output of unlabeled data with regularization. Another transductive method [12, 19, 31] was always performed on the graph-based model. Different from the inductive methods, the transductive methods never produce a classifier for prediction. It usually defines a graph for all input data and encodes the relationship between the pairwise data points.

Self-Supervised Learning completely abandons the reliance on manual labels by adopting the input itself as supervision, thus making great progress on representation learning in the last few years. Through Liu's research [21] on SSL, its main methods can be divided into three categories: generative SSL, contrastive SSL, and generative-contrastive SSL. The generative SSL trains a generator consisting of an encoder and decoder to reconstruct the input data. Its represent research in natural language processing is GPT [27] and BERT [5], which predict the discarded content with the partially abandoned input sequence. In the field of computational vision, especially in the area of image classification and image generation, PixelCNN [34], VQ-VAE-2 [28], and MAE [9] successfully used the whole input image as the self-supervised target. Compared to generative SSL, the motivation of contrastive SSL is to measure the similarity of different inputs (e.g., mutual information maximization and instance discrimination). Its influential work includes MoCo [4, 10], BYOL [8], and SimCLR [3]. As for the generative-contrastive SSL, most works focus on learning knowledge from unlabeled data with generative adversarial networks.

This paper links the inductive semi-supervised algorithm (e.g. Pseudo Label [13]) with the generative self-supervision (e.g. MAE [9]). Our MAPLE uses the short-term pseudo label instead of the short-term features as the reconstruction target. By this means, it can achieve the stable reconstruction of masked frames and learn generalizable features from unlabeled point cloud videos.

3 THE PROPOSED MAPLE FRAMEWORK

In this section, we declare the detailed framework of our MAPLE. Our MAPLE consists of a Decoupled spatial-temporal TransFormer (DestFormer) backbone, as shown in Fig. 7. The DestFormer takes the point cloud videos x_i as input and adopts the spatial extractor, temporal aggregator and prediction head back-to-back for extracting the global feature v_i and predicting the final class label y_i. On this basis, our MAPLE builds a masked autoencoder learning framework for semi-supervised action recognition, as shown in Fig. 3. It consists of an encoder-decoder structure and implicitly reconstructs the masked input feature with the pseudo-label generated by the classification head. Before describing our MAPLE in detail, we first declare the necessary notations and definitions of the semi-supervised point cloud action recognition task.

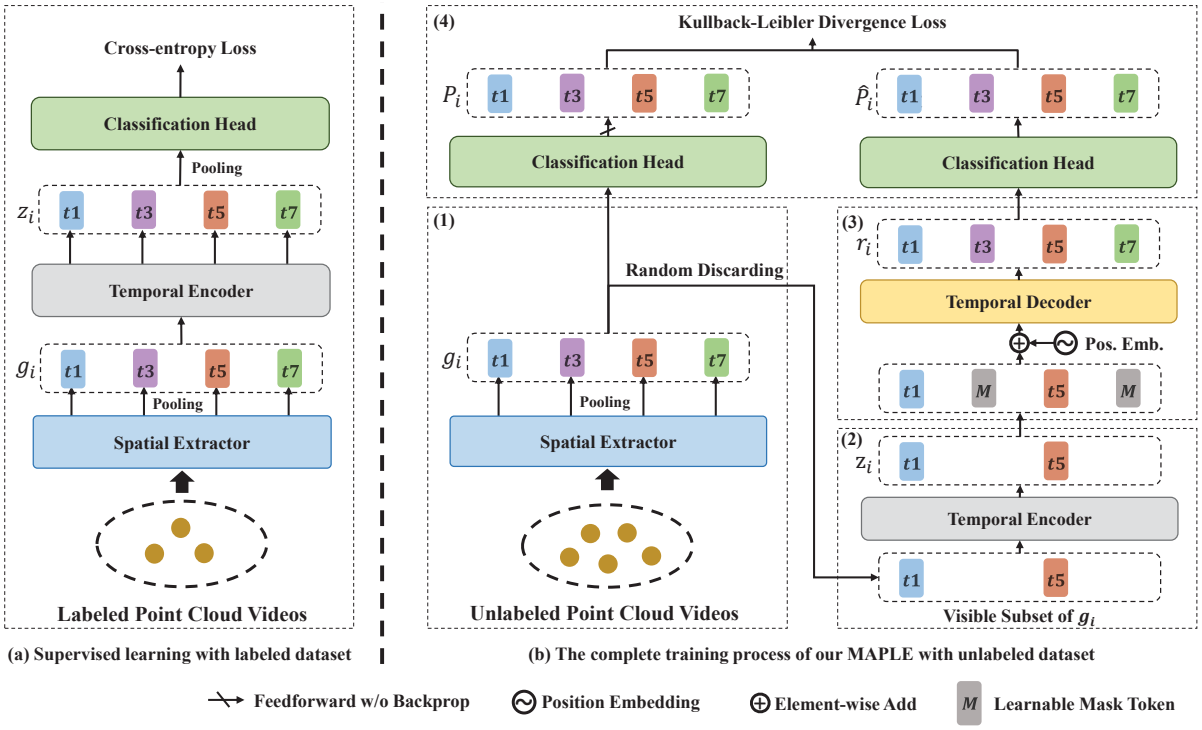


Figure 3: The detail of our MAPLE. (*a*) Adopting our DestFormer backbone and the cross-entropy loss for the supervised training. (*b*) The complete training process of MAPLE: (1) The spatial extractor encodes the input video as the short-term global feature g_i. (2) After randomly discarding the short-term global feature g_i, the temporal encoder projects the visible subset of g_i as the latent representation z_i. (3) The temporal decoder is responsible for reconstructing r_i from the latent representation z_i and the mask tokens M. (4) The classification head generates the pseudo-label P_i and $\hat{P}_i$ as our reconstruction target. Note that the modules here with the same colors share weights.

3.1 Preliminary

The task of point cloud action recognition is, given a point cloud video of humans, to predict the human behavior and actions in the video. Semi-supervised point cloud action recognition is consistent with the point cloud action recognition task in the inference phase, but usually adopts a different paradigm in the training phase as follows.

1) We have a dataset D, which contains a labeled subset $D_l = \{(x_i, y_i)\}$ and an unlabeled subset $D_u = \{x_j\}$. Both D_l and D_u are sampled i.i.d. from the same distribution $p(x)$ and in general the size of D_l is smaller than the size of D_u. Let $x_i = \{X_t \in \mathbb{R}^{(3+C)\times N}\}_{t=1}^{T}$ denote the matrix sequence of a point cloud video, where N indicates the number of points in each frame, T indicates the number of frames in this video, $\mathbb{R}^3$ and $\mathbb{R}^C$ indicates the spatial coordinates and features dimension of one point. It is worth noting that there are no point cloud features (ie., $C = 0$) in the given dataset (NTU RGB+D 60 [29], NTU RGB+D 120 [15] and MSR-Action3D [14]).

2) The training of model $f_\theta(\cdot)$ has an optimization function of the following form:

$$\min_{\theta} L_l(D_l, \theta) + \alpha L_u(D_u, \theta), \tag{1}$$

where L_l is the loss function (e.g., cross-entropy loss, mean squared error loss, Hinge loss, etc.) for classification of the labeled dataset D_l, and L_u is the optimization objective designed for unlabeled dataset D_u (the design of this function varies from paper to paper, and we discuss our designed L_u in later subsection.), α is the positive scalar weight for L_u and θ is the learnable parameters of $f_\theta(\cdot)$.

3.2 Decoupled Spatial-temporal TransFormer

This subsection presents our backbone for point cloud action recognition named Decoupled spatial-temporal TransFormer (DestFormer). The design of DestFormer is based on P4Transformer [6], and its main purpose is to serve as a basic backbone for the semi-supervised learning framework MAPLE and learn discriminative motion representations with fewer annotations. As shown in Fig. 7, the DestFormer consists of four parts: data preparation, spatial extractor, temporal aggregator, and prediction head. Note that embedding features in figures with different colors (e.g. "t1" and "t3") correspond to the different keyframes of the input action sequence.

Data Preparation. For the input point cloud video x_i, we construct some local areas (e.g. "a","b","c") on adjacent frames (e.g. "t1","t2") as what Point 4D Convolution (P4Conv) [6] do. The calculation of the local areas is based on the Farthest Point Sampling (FPS) algorithm [26], and the exhaustive calculation process is declared in [6].

Spatial Extractor (SE). SE is designed to extract the short-term global feature $g_i = \{G_{\zeta\cdot(t-1)+1} \in \mathbb{R}^D\}_{t=1}^{T/\zeta}$ from the local areas. We first extracts the short-term local feature $s_i = \{S_{\zeta\cdot(t-1)+1} \in$

$\mathbb{R}^{D\times(N/\kappa)}\}_{t=1}^{T/\zeta}$ through P4Conv [6], where D is the dimension of short-term local feature, $\kappa \geq 1.0$ denotes the spatial scaling rate and $\zeta \geq 1.0$ denotes the spatial scaling rate. P4Conv plays the role of aggregating the local information between adjacent ζ frames. After that, we feed the short-term local feature s_i frame by frame into the Spatial Transformer (ST) modules for extracting the merged short-term local feature $m_i = \{M_{\zeta\cdot(t-1)+1} \in \mathbb{R}^{D\times(N/\kappa)}\}_{t=1}^{T/\zeta}$ which aggregate the information of different spatial part.

Temporal Aggregator (TA). TA consists of a pooling layer and a transformer-based [35] Temporal Encoder (TE). we first prepare the short-term global feature g_i from the merged short-term local feature m_i through the pooling layer (e.g. maximum pooling). Then we aggregate the long-term global z_i from the short-term global feature g_i with our TE module.

Prediction Head. Following the TA module is the pooling layer (e.g. maximum pooling) and classification head, which consists of Layer Normalization layers (LayerNorm), linear layers, and Gaussian Error Linear Units (GELUs). Its role is to project the global feature v_i into the label space and generate the corresponding pseudo labels for classification.

3.3 Masked Pseudo-labeling Autoencoder

This subsection elaborates our Masked Pseudo-Labeling autoEncoder (MAPLE) framework for semi-supervised point cloud action recognition. As shown in the left part of Fig. 3, we adopt our DestFormer backbone and the cross-entropy loss for the supervised training with labeled point cloud videos. The right part of Fig. 3 shows the complete training process of our MAPLE with unlabeled point cloud videos. Similar to the reconstruction process of autoencoders, the spatial extractor encodes the point cloud videos as the short-term local embedding feature g_i. The temporal encoder of our MAPLE projects the visible subset of embedding feature g_i into latent space Z, and the temporal decoder is responsible for reconstructing from the latent representation z_i and the learnable mask tokens M. However, different from classical autoencoders, our MAPLE implicitly reconstructs the original signal through the pseudo-label generated from the classification head, rather than reconstructing the original signal itself. We introduce the training process in detail as follows:

Masking the short-term global feature g_i that is extracted from the Spatial Extractor (SE) modules is the first step of our framework. Like what ImageMAE does in [9], we directly discard a subset (e.g., 50%) of the original short-term global feature g_i with random sampling. The motivation of masking is to help the model efficiently understand the order of actions via reconstructing the complete action sequence from the mutilated one.

Temporal Encoder (TE) is a lightweight transformer that only contains several self-attention blocks in the second step of our MAPLE. We directly feed the masked short-term global features g_i into the TE module without adding positional embedding, since its temporal positional embedding is already added when fed into the SE module.

Temporal Decoder (TD) is also a lightweight transformer that is used to reconstruct the removed embedding feature g_i in the third step of our MAPLE. Before feeding the latent representation z_i into the TD modules, we first insert the shared and learnable mask token M at the position of the original abandoned features and then add the new temporal positional embedding to the full set of sequences. Note that the shared mask without new temporal positional embedding cannot reconstruct the action information at different times.

Algorithm 1 The training process of our MAPLE.

Stage 1: Pre-training with labeled dataset D_l (corresponding to the left part of Fig. 3).
Initialization: the network parameters of DestFormer θ; basic learning rate η; the labeled batch size b_l; the supervised cross-entropy loss L_l.
repeat
  t = 1 ... max iteration num:
    fetch mini-batch d_l from D_l;
    compute loss L_l on d_l;
    update $\theta^t = \theta^{t-1} - \eta \bigtriangledown L_l$.
until stable accuracy and loss in the validation set.

Stage 2: Training of MAPLE with unlabeled dataset D_u (corresponding to the left part of Fig. 3).
Initialization: positive scalar weight α for unsupervised loss L_u ; unlabeled batch size b_u, where $b_u \geq b_l$.
repeat
  t = 1 ... max iteration num:
    fetch mini-batch d_l from D_l and d_u from D_u;
    compute loss $L = L_l + \alpha L_u$ on d_l and d_u;
    update $\theta^t = \theta^{t-1} - \eta \bigtriangledown L$.
until stable accuracy and loss in the validation set.

Reconstruction Target. As shown in the final step of our MAPLE in Fig. 3 (b), the target of reconstruction is calculated with pseudo-label P_i instead of the original feature g_i. We feed both the original feature g_i and the reconstructed feature r_i into the classification head to obtain their corresponding pseudo-label P_i and $\hat{P}_i$. Note that the original pseudo-label P_i is generated without backprop for maintaining the stability of the training process. Following this target, our unsupervised loss can be defined with the Kullback-Leibler divergence:

$$L_u = L_{maple} = \frac{1}{|D_u|} \sum_{x_i \in D_u} KL(f(P_i|x_i)||f(\hat{P}_i|x_i)), \tag{2}$$

where KL is the function of Kullback-Leibler divergence, $|D_u|$ is the size of the unlabeled dataset, $f(\cdot)$ is the model, P_i is the pseudo-label generated from the original feature g_i, $\hat{P}_i$ is the reconstructed pseudo-label generated from the reconstructed feature r_i. *Algorithm* 1 and *Algorithm* 2 present the training and inference process of our MAPLE, respectively.

Compare to reconstructing the original feature, reconstructing the pseudo-label not only improves the performance of classification but also makes the training stage more stable. We compare these two strategies in detail in section 4.5.

Algorithm 2 The inference process of our MAPLE.

Initialization: the DestFormer model $f(\cdot)$ without the temporal decoder; the best-trained network parameters θ.
repeat
t = 1 ... final test batch:
fetch mini-batch d_t from test dataset D_t;
calculate the accuracy on d_t;
finished.
Calculate the accuracy on the whole test dataset D_t.

4 EXPERIMENTS

To show the effectiveness of our DestFormer and MAPLE, we first evaluate the supervised-only performance and computational efficiency of our DestFormer. Then we compare our MAPLE with the semi-supervised baseline algorithms and further combine these leading algorithms with our MAPLE to obtain superior classification performance. At last, we investigate the choices of masking rate, the depth of temporal decoder, and the irreplaceability of pseudo-label.

4.1 Dataset

Our experiments are performed on three main human action recognition datasets: MSR-Action3D [14], NTU RGB+D 60 [29], and NTU RGB+D 120 [15].

MSR-Action3D [14] dataset captured with Kinect v1 depth camera, which contains 567 videos and 23k frames in total (270 videos for training and 297 videos for testing). This dataset contains twenty actions: high arm wave, horizontal arm wave, and so on. For our semi-supervised point cloud action recognition, 7.5%, 15.0%, 22.5%, 30.0%, and 37.5% of training videos of each action are selected for the labeled dataset D_l and the rest for the unlabeled dataset D_u. More detailed information is available in the supplementary material.

NTU RGB+D 60 [29] is a large dataset that was captured with Kinect v2 depth camera. It consists of 56K videos and 4M frames captured from 80 views and with 40 performers. Sixty action categories and two types of evaluation (i.e. cross-subject and cross-view) are defined in this dataset. In this paper, we evaluate our model with a cross-subject setting. For our semi-supervised point cloud action recognition task, 5%, 10%, 20%, 30%, and 40% of training videos of each action are selected for the labeled dataset D_l.

NTU RGB+D 120 [15] is an extension of NTU RGB+D 60 and the largest dataset for human action recognition. It consists of 114K videos and 8M frames captured from 155 views and with 106 performers. The dataset captured by Kinect v2 depth camera has the modalities of RGB, Depth, 3DJoints, and IR. One hundred and twenty action categories and two types of evaluation (i.e. cross-subject and cross-setup) are defined on this dataset. To harmonize with the above dataset, we still evaluate our model with a cross-subject setting and select the same percentage of the labeled dataset D_l as NTU RGB+D 60.

Dataset	Backbone	Ratio of Labeled Data				
		7.5%	15.0%	22.5%	30.0%	37.5%
MSR3D [14]	P4Transformer	61.95	77.10	80.47	83.16	85.85
	DestFormer	62.96	77.44	81.14	83.84	86.53
Dataset	Backbone	Ratio of Labeled Data				
		5%	10%	20%	30%	40%
NTU60 [29]	P4Transformer	45.21	57.20	68.41	73.98	77.26
	DestFormer	46.80	59.63	70.03	74.98	78.16
NTU120 [15]	P4Transformer	30.38	40.34	48.66	53.28	56.94
	DestFormer	36.09	47.75	58.05	62.56	65.31

Table 1: Comparison of the supervised-only action recognition accuracy (%) between P4Transformer [6] and our DestFormer on three benchmark dataset.

Backbone	Depth	GFLOPs	Inference Time (ms)
P4Transformer [6]	5	85.6	865
DestFormer	4+3	60.5	665

Table 2: Comparison of the time complexity (GFLOPs) and average inference time (ms) between P4Transformer [6] and Our DestFormer.

4.2 Implementation Details and Approaches

This subsection presents the training hyperparameters and implementation details of our DestFormer and MAPLE.

Network Structure. The DestFormer $f(\cdot)$ are introduced in Section 3.2. By default, The spatial scaling rate κ of P4Conv is set to 2 and the spatial scaling rate ζ is set to 32. The spatial transformer is designed with 4 self-attention blocks and the temporal encoder is designed with only 3 self-attention blocks. Each black spatial transformer and temporal encoder contains 8 heads. As the Temporal Decoder used in MAPLE, it consists of 8 self-attention blocks to strengthen its ability for reconstruction.

MAPLE Training. In the whole process of training, the basic learning rate η is set to 0.01. The warm-up strategy is used for the first 10 epochs with the initial $\eta = 10^{-6}$ and the decreased learning rate η of the final 5 epochs is set to 0.001. The mini-batch of D_l and D_u is set to 14 for the MSR-Action3D dataset, and 32 for NTU RGB+D 60 and NTU RGB+D 120. The masking ratio is set to 75% for all datasets. In Step 1 (Pre-training) of our training, the DestFormer is trained on the labeled dataset of MSR-Action3D, NTU RGB+D 60 and NTU RGB+D 120 with the epoch of 40, 20, and 20, respectively. In Step 2 (Training of MAPLE), we set the positive scalar weight α as 0.5 for MSR-Action3D, and 0.2 for NTU RGB+D 60 and NTU RGB+D 120.

Compared Approaches. In the following section 4.4, we use leading semi-supervised learning algorithms that have proven to be generally effective as our compared approaches:

1) *Supervised-only.* We train the model only with labeled dataset D_l. The performance of the best-trained model is used as the lower bounds of semi-supervised learning.

2) *Pseudo Labels [13].* The main idea is to further train the model with the pseudo hard labels of unlabeled data. This algorithm can be summarized in two steps as follows. First, we get the pre-training model with the supervised-only method, then we predict the pseudo

hard labels of unlabeled data. Finally, the model can be retrained with these hard labels.

3) *Virtual Adversarial Training (VAT) [24].* It is inspired by adversarial learning and its regularization only needs unlabeled data. In the training process, it first adds small adversarial perturbation ϵ_{vat} to the unlabeled data for changing the final prediction, and then forces the model $f_\theta(\cdot)$ against this type of perturbation with the following consistency loss:

$$L_{vat} = \frac{1}{|D_u|} \sum_{x_i \in D_u} KL(f_\theta(x_i) \,||\, f_\theta(x_i + \triangle x_i)), \tag{3}$$

$$where \ \triangle x_i = \underset{\delta \ s.t. \ |\delta|_2 = \epsilon_{vat}}{\arg\max} KL(f_\theta(x_i) \,||\, f_\theta(x_i + \triangle x_i)). \tag{4}$$

4) *Conditional Entropy Minimization (EntMin) [7].* This approach encourages the model to output the confident pseudo labels y for unlabeled input data. In other words, the predictions y closed to the one-hot vector are encouraged. The conditional entropy minimization loss can be defined as:

$$L_{entmin} = \frac{1}{|D_u|} \sum_{x_i \in D_u} \sum_{y \in Y} -f_\theta(y|x_i) \log f_\theta(y|x_i). \tag{5}$$

Note that the EntMin is almost not used alone for semi-supervised learning because the model can easily increase the weights of the classification head to generate a confident prediction. It always adopt with the VAT loss, i.e. $L_u = \alpha_{vat} L_{vat} + \alpha_{entmin} L_{entmin}$, where α_{vat} and α_{entmin} are the positive scalar weight for loss of VAT and EntMin, respectively.

4.3 Supervised-only Performance

To demonstrate the validity of our spatial-temporal backbone, this subsection compares the action recognition performance and computational efficiency of our DestFormer and the P4transformer model [6] with the supervised-only setting on MSR-Action3D, NTU RGB+D 60 and NTU RGB+D 120 datasets. The action recognition accuracy (%) of each backbone on three benchmark datasets is listed in Table 1. The time complexity (GFLOPs) and average inference time (ms) of each point cloud video are listed in Table 2.

In Table 1, we observe that our DestFormer has less annotation dependence and better classification performance in the supervised-only setting. Especially on the NTU RGB+D 120 dataset, our DestFormer model generally obtains a greater than 5.7% increase in action recognition accuracy.

In Table 2, we notice that our DestFormer is more efficient in computational complexity, which obtain about 30% and 23% decrease for time complexity (GLOPs) and inference time (ms), respectively.

4.4 Evaluation of Semi-supervised Methods

In this section, we first evaluate our MAPLE method by comparing it with leading semi-supervised methods (e.g. Pseudo Label, VAT, and EntMin) for semi-supervised point cloud action recognition on three mainstream datasets. Then we further combined our MAPLE method with those methods (VAT+EntMin+MAPLE) and obtain better performance for semi-supervised point cloud action recognition. Specifically, we use L_{vat} and L_{entmin} as unsupervised loss functions L_u in the early training stage until the model has almost stabilized and then adopt the L_{maple} as the unsupervised loss function. Please refer to the supplementary materials for the detailed training process of "VAT+EntMin+MAPLE".

Method	Ratio of Labeled Data				
	7.5%	15.0%	22.5%	30.0%	37.5%
supervised-only	62.96	77.44	81.14	83.84	86.53
Pseudo Label [13]	68.01	80.64	81.65	85.19	88.05
VAT [24]	66.92	80.47	81.14	85.19	86.53
VAT + EntMin [7]	67.24	81.48	83.84	85.94	87.29
MAPLE (Ours)	72.04	82.15	84.85	87.04	89.40
VAT+EntMin+MAPLE (Ours)	**76.09**	**84.85**	**86.20**	**87.21**	**89.56**

Table 3: Comparison of the results on MSR-Action3D.

Method	Ratio of Labeled Data				
	5%	10%	20%	30%	40%
supervised-only	46.80	59.63	70.03	74.98	78.16
Pseudo Label [13]	47.24	61.96	72.14	76.74	79.15
VAT [24]	46.80	59.95	70.92	75.77	78.47
VAT + EntMin [7]	47.07	62.20	72.59	77.25	79.33
MAPLE (Ours)	48.78	60.61	71.05	75.72	78.61
VAT+EntMin+MAPLE (Ours)	**50.63**	**62.98**	**73.01**	**77.57**	**79.96**

Table 4: Comparison of the results on NTU RGB+D 60.

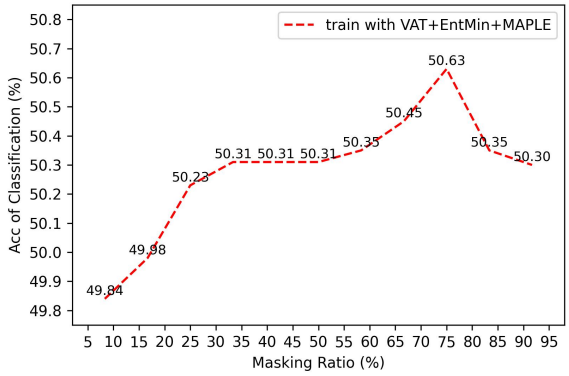
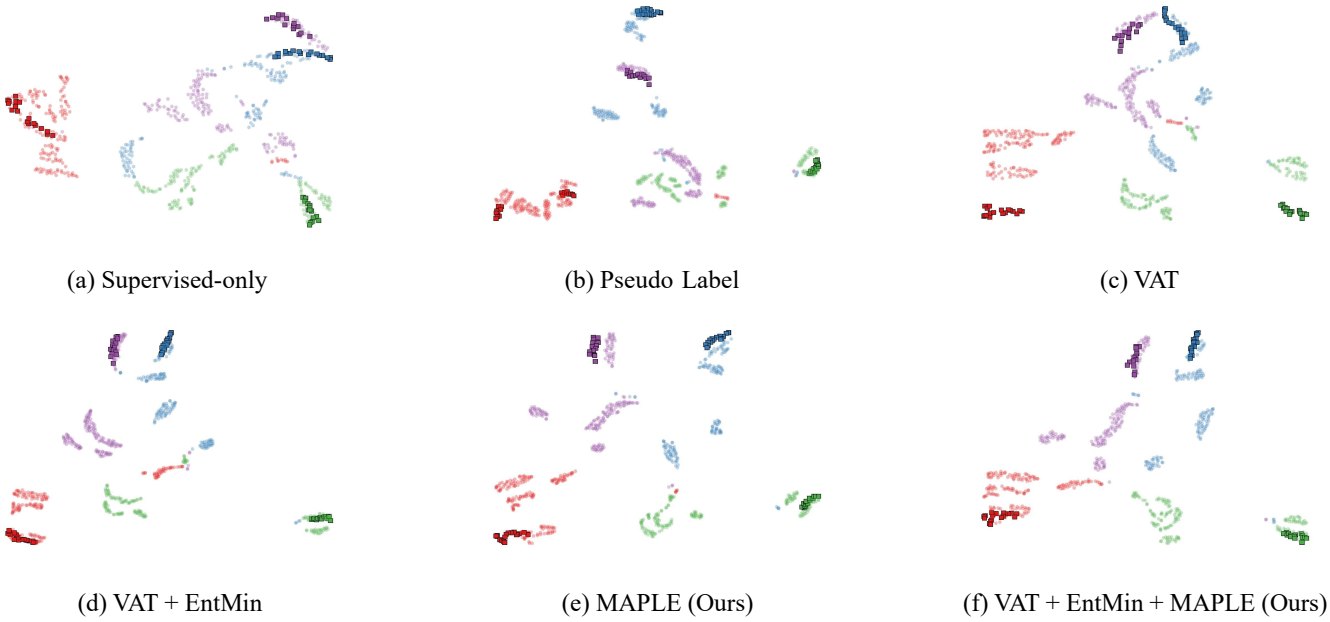


Figure 4: The accuracy of classification on NTU RGB+D 60 5% labeled dataset with different masking ratios. The 75% masking ratio of reconstruction achieves peak accuracy.

The results on MSR-Action3D and NTU RGB+D 60 datasets are shown in Tables 3 and 4 respectively. The results on NTU RGB+D 120 are shown in the supplementary material. From the semi-supervised results of each semi-supervised method, we can find that our MAPLE method is effective for semi-supervised learning and slightly outperforms the previous methods' performance under most settings. We also observe that our MAPLE method can be combined with other leading semi-supervised methods, and obtain significant performance increases under each setting. Especially on the 7.5% labeled MSR-Action3D setting, it brings significant improvement in action recognition performance (+8.08% Acc).



(a) Supervised-only (b) Pseudo Label (c) VAT

(d) VAT + EntMin (e) MAPLE (Ours) (f) VAT + EntMin + MAPLE (Ours)

Figure 5: The t-SNE visualization of different approaches on the MSR-Action3D dataset. The squares with the black border indicate the labeled data, and other dots indicate the unlabeled ones. Note that different colors denote different classes.

4.5 Ablation Study and Visualization

We investigate the effectiveness of our proposed MAPLE method on benchmark datasets in this section. We first analyze the influence of the masking ratio and the depth of the temporal decoder, then illustrate the importance of reconstruction with pseudo-label during the training process. At last, we show the feature distributions of each method to prove the effectiveness of our MAPLE method.

Masking ratio. Fig. 4 shows the accuracy of classification on NTU RGB+D 60 5% labeled dataset with different masking ratios. The high masking ratio (75%) of reconstruction achieves the peak of classification performance, which is as high as the masking ratio of Image MAE [9]. This phenomenon is the exact opposite of the natural language processing field, whose best masking ratio is 15% typically. However, this behavior verifies our hypothesis that most frames of the action sequence are redundant and we can reconstruct the complete action sequence from the small residual part of the sequence.

Depth of Temporal Decoder. We investigate the effect of decoder depth on the final classification performance and the result on NTU RGB+D 60 5% labeled dataset with different depth of temporal decoder. For more detail, please refer to the supplementary materials.

Reconstruction with Pseudo-Label. To demonstrate the importance of implicit reconstruction through pseudo-label, we show the results with and without pseudo-label on the MSR-Action3D dataset in the supplementary materials.

The t-SNE visualization. To further explore the mechanism of MAPLE, we visualize the feature distributions of the labeled and unlabeled video sequences of the MSR-Action3D training dataset with t-SNE in Fig. 5. From the t-SNE visualization, we can find that the model trained with only supervised action sequences is often hard to distinguish the decision boundaries of the unlabeled action sequence. Although the benchmark methods (Pseudo Label, VAT, and EntMin) have better distributions, there are still some outliers that make the decision boundaries ambiguous. Compared to previous approaches, our MAPLE and "VAT+EntMin+MAPLE" form tighter data clusters and clear decision boundaries which benefit from the semi-supervision learning with our masked autoencoder. To summarize, the visualization shows that combining semi-supervised learning with masked pseudo-labeling autoencoder is possible to learn numerous action concepts from unlabeled point cloud videos and improves the performance of action recognition.

5 CONCLUSION

In this paper, we present a Masked Pseudo-Labeling autoEncoder (**MAPLE**) framework with an effective transformer-based **De**coupled **s**patial-**t**emporal Trans**Former** (**DestFormer**) backbone to learn discriminative representations with much fewer annotations for the semi-supervised point cloud action recognition task. The MAPLE framework exploits the reconstruction of the masked features from the available frames to learn the numerous action concepts from unlabeled action sequences. Moreover, we combine our MAPLE with the classical semi-supervised methods to learn more generalizable features and establish the state-of-the-art performances of the semi-supervised point cloud action recognition task. We hope that our MAPLE framework can inspire the research of autoencoder on point cloud sequence in the future.

ACKNOWLEDGMENTS

This research was supported by the National Key R&D Program of China under Grant No. 2020AAA0103800.

REFERENCES

[1] Lucas Beyer, Xiaohua Zhai, Avital Oliver, and Alexander Kolesnikov. 2019. S4L: Self-Supervised Semi-Supervised Learning. In *ICCV*. 1476–1485.

[2] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien (Eds.). 2006. *Semi-Supervised Learning*.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML*. 1597–1607.

[4] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. 2020. Improved Baselines with Momentum Contrastive Learning. *CoRR* (2020).

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*. 4171–4186.

[6] Hehe Fan, Yi Yang, and Mohan S. Kankanhalli. 2021. Point 4D Transformer Networks for Spatio-Temporal Modeling in Point Cloud Videos. In *CVPR*. 14204–14213.

[7] Yves Grandvalet and Yoshua Bengio. 2004. Semi-supervised Learning by Entropy Minimization. In *NeurIPS*. 529–536.

[8] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. 2020. Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning. In *NeurIPS*. 21271–21284.

[9] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. 2021. Masked Autoencoders Are Scalable Vision Learners. *CoRR* (2021).

[10] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR*. 9726–9735.

[11] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the Knowledge in a Neural Network. *CoRR* (2015).

[12] Tony Jebara, Jun Wang, and Shih-Fu Chang. 2009. Graph construction and *b*-matching for semi-supervised learning. In *ICML*. 441–448.

[13] Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*. 896.

[14] Wanqing Li, Zhengyou Zhang, and Zicheng Liu. 2010. Action recognition based on a bag of 3D points. In *CVPR Workshops*. 9–14.

[15] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C. Kot. 2020. NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* (2020), 2684–2701.

[16] Kun Liu, Wu Liu, Chuang Gan, Mingkui Tan, and Huadong Ma. 2018. T-C3D: Temporal Convolutional 3D Network for Real-Time Action Recognition. In *AAAI*. 7138–7145.

[17] Wu Liu, Qian Bao, Yu Sun, and Mei Tao. 2022. Recent Advances of Monocular 2D and 3D Human Pose Estimation: A Deep Learning Perspective. *ACM Computing Surveys (CSUR)* (2022).

[18] Wei Liu, Junfeng He, and Shih-Fu Chang. 2010. Large Graph Construction for Scalable Semi-Supervised Learning. In *ICML*. 679–686.

[19] Wei Liu, Jun Wang, and Shih-Fu Chang. 2012. Robust and Scalable Graph-Based Semisupervised Learning. *Proc. IEEE* (2012), 2624–2638.

[20] Xingyu Liu, Mengyuan Yan, and Jeannette Bohg. 2019. MeteorNet: Deep Learning on Dynamic 3D Point Cloud Sequences. In *ICCV*. 9245–9254.

[21] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Zhaoyu Wang, Li Mian, Jing Zhang, and Jie Tang. 2020. Self-supervised Learning: Generative or Contrastive. *CoRR* (2020).

[22] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *ICCV*. 9992–10002.

[23] Roberto Martin-Martin, Mihir Patel, Hamid Rezatofighi, Abhijeet Shenoi, JunYoung Gwak, Eric Frankel, Amir Sadeghian, and Silvio Savarese. 2021. JRDB: A dataset and benchmark of egocentric robot visual perception of humans in built environments. *TPAMI* (2021).

[24] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2019. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* (2019), 1979–1993.

[25] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *CVPR*. 77–85.

[26] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *NeurIPS*. 5099–5108.

[27] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *CoRR* (2018).

[28] Ali Razavi, Aäron van den Oord, and Oriol Vinyals. 2019. Generating Diverse High-Fidelity Images with VQ-VAE-2. In *NeurIPS*. 14837–14847.

[29] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. 2016. NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. In *CVPR*. 1010–1019.

[30] Razieh Sheikhpour, Mehdi Agha Sarram, Sajjad Gharaghani, and Mohammad Ali Zare Chahooki. 2017. A Survey on semi-supervised feature selection methods. *Pattern Recognit.* (2017), 141–158.

[31] Amarnag Subramanya and Partha Pratim Talukdar. 2014. *Graph-Based Semi-Supervised Learning*. Morgan & Claypool Publishers.

[32] Yu Sun, Wu Liu, Qian Bao, Yili Fu, Tao Mei, and Michael J. Black. 2022. Putting People in their Place: Monocular Regression of 3D People in Depth. (2022), 13243–13252.

[33] Isaac Triguero, Salvador García, and Francisco Herrera. 2015. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowl. Inf. Syst.* (2015), 245–284.

[34] Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. 2016. Pixel Recurrent Neural Networks. In *ICML*. 1747–1756.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NeurIPS*. 5998–6008.

[36] Yancheng Wang, Yang Xiao, Fu Xiong, Wenxiang Jiang, Zhiguo Cao, Joey Tianyi Zhou, and Junsong Yuan. 2020. 3DV: 3D Dynamic Voxel for Action Recognition in Depth Video. In *CVPR*. 508–517.

[37] Yimin Wei, Hao Liu, Tingting Xie, Qiuhong Ke, and Yulan Guo. 2022. Spatial-Temporal Transformer for 3D Point Cloud Sequences. In *WACV*. 657–666.

[38] Zhu X. (Ed.). 2008. *Semi-supervised learning literature survey: Department of Computer Sciences*.

[39] Yucheng Zhao, Guangting Wang, Chuanxin Tang, Chong Luo, Wenjun Zeng, and Zheng-Jun Zha. 2021. A Battle of Network Structures: An Empirical Study of CNN, Transformer, and MLP. *CoRR* (2021).

[40] Jinkai Zheng, Xinchen Liu, Wu Liu, Lingxiao He, Chenggang Yan, and Tao Mei. 2022. Gait Recognition in the Wild with Dense 3D Representations and A Benchmark. In *CVPR*. 20228–20237.

A ADDITIONAL EXPERIMENTAL RESULTS

In this supplementary material, we show more details about the semi-supervised datasets and our MAPLE algorithm.

A.1 Division of Semi-supervised Datasets

Our experiments are conducted on three benchmark datasets: MSR-Action3D (MSR3D), NTU RGB+D 60 (NTU60), and NTU RGB+D 120 (NTU120). As shown in Table 5, we divide each training dataset into labeled training dataset D_l and unlabeled training dataset D_u. As an illustration, we select 33 videos for each class (1980 videos in total) from NTU RGB+D 60 as the 5% labeled training dataset.

A.2 Training Process of VAT+EntMin+MAPLE

In this subsection, we describe the detailed training progress of "VAT+EntMin+MAPLE". As shown in Algorithm 3, we first pre-train our model with labeled dataset D_l for getting better initialization parameters. Then we adopt $\alpha_{vat}L_{vat} + \alpha_{entmin}L_{entmin}$ as the unsupervised loss with the unlabeled dataset D_u until the model achieve stable accuracy (approximately 10 to 15 epochs for this step). At last, we use the L_{maple} as our unsupervised optimization functions and continue training until the model almost converges.

Dataset	Division	Ratio of Labeled Data				
		7.5%	15.0%	22.5%	30.0%	37.5%
MSR3D	Labeled	20	40	60	80	100
	Unlabeled	250	230	210	190	170
Dataset	Division	Ratio of Labeled Data				
		5.0%	10.0%	20.0%	30.0%	40.0%
NTU60	Labeled	1980	3960	7920	11880	15840
	Unlabeled	38340	36360	32400	28440	24480
NTU120	Labeled	3120	6358	12416	17846	22810
	Unlabeled	60240	57002	50944	45514	40550

Table 5: The division of the semi-supervised datasets.

Method	Ratio of Labeled Data				
	5%	10%	20%	30%	40%
Supervised-only	36.09	47.75	58.05	62.56	65.31
Pseudo Label	36.18	48.25	58.33	62.85	65.55
VAT	35.90	47.74	58.29	62.56	65.75
VAT + EntMin	36.02	48.2	58.42	62.70	66.88
MAPLE (Ours)	**37.15**	48.56	58.59	63.18	65.84
VAT+EntMin+MAPLE (Ours)	36.91	**48.80**	**59.25**	**64.02**	**67.08**

Table 6: Comparison of the results on the NTU120 dataset.

Method	Ratio of Labeled Data				
	7.5%	15.0%	22.5%	30.0%	37.5%
MAPLE w/o pseudo-label	69.19	81.32	84.51	86.87	86.53
MAPLE with pseudo-label	**72.04**	**82.15**	**84.85**	**87.04**	**89.40**

Table 7: The results of MAPLE with and w/o pseudo-label on the MSR-Action3D dataset

A.3 Results on NTU RGB+D 120 Dataset

In this subsection, we additional evaluate our MAPLE by comparing it with leading semi-supervised methods on NTU RGB+D 120 dataset and show the results in Table 6. From the table, we can observe that previous semi-supervised methods (e.g. Pseudo Label, VAT and EntMin) are slightly effective for semi-supervised learning on this largest dataset of human action recognition. Our method outperforms the previous methods by about 1.0% classification accuracy under the most setting.

A.4 Reconstruction with Pseudo-Label

To demonstrate the importance of implicit reconstruction through pseudo-label, we show the $L2$ Norm of the reconstructed feature r_i with and without pseudo-label on the 5% labeled MSR-Action3D dataset in Fig. 6 and compare the results of MAPLE under each setting in Table 7. The loss function without pseudo-label can be defined with Mean-Squared Error (MSE) loss as follows:

$$L_u = L_{maple} = \frac{1}{|D_u|} \sum_{x_i \in D_u} MSE(f(g_i|x_i)||f(r_i|x_i)), \tag{6}$$

where MSE is the function of MSE loss, $|D_u|$ is the size of the unlabeled dataset, g_i is the original short-term global feature, and r_i is the reconstructed short-term global feature. Note that the exploding and vanishing problem is not the same as gradient exploding and

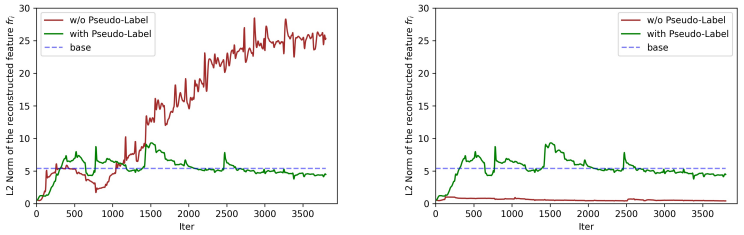
Algorithm 3 The training process of our VAT+EntMin+MAPLE.

Stage 1: Pre-training with labeled dataset D_l.
Initialization: the network parameters of DestFormer θ; basic learning rate η; the labeled batch size b_l; the supervised cross-entropy loss L_l.
repeat
  t = 1 ... max iteration num:
    fetch mini-batch d_l from D_l;
    compute loss L_l on d_l;
    update $\theta^t = \theta^{t-1} - \eta \bigtriangledown L_l$.
until stable accuracy and loss in the validation set.

Stage 2: Training of VAT+EntMin+MAPLE with unlabeled dataset D_u.
Initialization: positive scalar weight α_{vat}, α_{entmin} and α_{maple} for unsupervised loss L_{vat}, L_{entmin} and L_{maple}, respectively; unlabeled batch size b_u, where $b_u \geq b_l$.
first repeat
  t = 1 ... max iteration num:
    fetch mini-batch d_l from D_l and d_u from D_u;
    compute loss $L = L_l + \alpha_{vat}L_{vat} + \alpha_{entmin}L_{entmin}$;
    update $\theta^t = \theta^{t-1} - \eta \bigtriangledown L$.
until stable accuracy and loss in the validation set.
second repeat
  t = 1 ... max iteration num:
    fetch mini-batch d_l from D_l and d_u from D_u;
    compute loss $L = L_l + \alpha_{maple}L_{maple}$ on d_l and d_u;
    update $\theta^t = \theta^{t-1} - \eta \bigtriangledown L$.
until stable accuracy and loss in the validation set.

Action Accuracy (%)	high arm wave	horizontal arm wave	hammer	hand catch	forward punch
supervised-only	6.67	86.67	0.00	33.33	91.67
VAT+EntMin+MAPLE (Ours)	93.33	86.67	0.00	26.67	21.43
Action Accuracy (%)	high throw	draw x	draw tick	draw circle	hand clap
supervised-only	21.43	76.92	100.00	6.67	84.62
VAT+EntMin+MAPLE (Ours)	28.57	42.86	100.00	73.33	100.00
Action Accuracy (%)	two hand wave	side-boxing	bend	forward kick	side kick
supervised-only	100.00	35.71	20.00	100.00	100.00
VAT+EntMin+MAPLE (Ours)	100.00	100.00	80.00	100.00	100.00
Action Accuracy (%)	jogging	tennis swing	tennis serve	golf swing	pick up throw
supervised-only	100.00	86.67	93.33	46.67	66.67
VAT+EntMin+MAPLE (Ours)	100.00	93.33	100.00	73.33	93.33

Table 8: More details about the improvement of per class accuracy on the 7.5% labeled MSR-Action3D dataset.



(a) Exploding (b) Vanishing

Figure 6: The $L2$ Norm of the reconstructed feature r_i with and w/o pseudo-label in two common situations: (a) Exploding. (b) Vanishing. Note that the blue baseline denotes the average $L2$ Norm of the original feature g_i during the supervised-only training process.

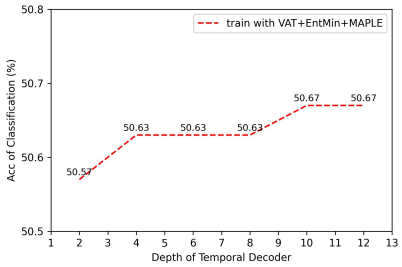


Figure 7: The accuracy of classification on NTU RGB+D 60 5% labeled dataset with different depth of temporal decoder.

gradient vanishing. It indicates the difference in the feature size under each training strategy.

From the figure and table, we can observe that training without pseudo-label lead to explosion or vanishment on the $L2$ Norm of the reconstructed features, which not only results in a significant decrease in the final classification performance but also make it hard to stably converge during the training process. The main reason for this occurrence can be obtained from our loss function 6 and the encoder-decoder structure of our MAPLE. The explosion situation (Fig. 6 (a)) happens when both the detached g_i (without backprop) and r_i are learnable features generated by our MAPLE and they share the same spatial extractor. When r_i tries to increase closer to the g_i, it often leads to an increase in the weights of the spatial extractor, which in turn leads to the increase of g_i. After hundreds or thousands of iterations, the $L2$ Norm of g_i and r_i become larger and larger, even tending to infinity. The vanishment situation (Fig. 6 (b)) happens when both the original g_i (with backprop) and r_i are learnable features generated by our MAPLE. The model tends to simply reduce the $L2$ Norm of g_i and r_i to achieve the reduction of loss function 6. After thousands of iterations, the $L2$ Norm of g_i and r_i become smaller and smaller, even tending to zero.

A.5 Depth of Temporal Decoder

We investigate the effect of decoder depth on the final classification performance and the result on NTU RGB+D 60 5% labeled dataset is shown in Fig. 7. From the figure we can know that the depth of the temporal decoder does not have a large impact on the performance of the final classification, the main reason is that the role of the decoder is only used to reconstruct the complete action sequence from the latent space, and has less relevance to the classification. Therefore, a decoder of shallow depth is sufficient to reconstruct the complete sequence.

A.6 Details about per class accuracy

To find out what type of actions are easily classified and which ones are tough, we show more details about the improvement of per class accuracy after training with our MAPLE in Table 8 on the 7.5% labeled MSR-Action3D dataset. From an overall perspective, it seems that simple repetitive actions, such as high arm wave (+86.67% accuracy), can greatly benefit from our MAPLE, while some complex actions, such as draw x (-34.06% accuracy), are hardly benefit from our MAPLE.