Tri-Efficient Transfer Learning for Point Cloud Videos

Yiding Sun[1*], Dongxu Zhang[1*], Jihua Zhu[1†], Haozhe Cheng[1], Zhengqiao Li[2], Pengcheng Li[3], Chaowei Fang[1], Yonghao Dong[1], and Lin Chen[1]

[1] School of Software Engineering, Xi'an Jiaotong University, China
sunyiding@stu.xjtu.edu.cn, zhujh@xjtu.edu.cn
[2] School of Information and Communication Engineering, Xi'an Jiaotong University
[3] SIGS, Tsinghua University, China
[*]Equal contribution. [†]Corresponding author.

Abstract. While point cloud foundation models have significantly advanced point cloud video understanding, existing parameter-efficient fine-tuning (PEFT) methods still suffer from two critical limitations: prohibitive annotation costs for large-scale point cloud datasets and severe memory bottlenecks. In this paper, we aim to mine richer supervision signals from existing data rather than blindly scaling datasets. A further key principle is that the memory footprint of fine-tuning must be drastically reduced compared to full fine-tuning, which remains elusive for current PEFT techniques. Driven by these challenges, we identify three core desiderata: data-, parameter-, and memory efficiency, and present PoinTriE, a unified framework that excels along all three dimensions. For pre-training, pseudo-motion trajectories are synthesized via rigid transformations, paired with text corpora and 2D projections derived from raw point clouds. We then propose a Geometric-Motion Duality Network optimized via multimodal contrastive learning, rigid rotation prediction, and motion distribution divergence to produce dense self-supervision. During fine-tuning, we freeze the pretrained backbone and only update a lightweight Spatio-temporal Side Network built with LoRA units. Equipped with a gradient flow masking strategy, PoinTriE simultaneously reduces memory consumption and parameter overhead. Extensive experiments confirm that PoinTriE establishes new state-of-the-art results on action recognition and semantic segmentation tasks.

Keywords: Point Cloud Videos · Transfer Learning · Efficient Learning

1 Introduction

Fine-tuning pretrained point cloud foundation models (PFMs) has been proven remarkably effective in adapting to versatile scenarios, especially those in dynamic domains [3, 4, 52]. However, catastrophic forgetting and prior knowledge disruption occur frequently, particularly as model sizes continue to grow [22,35]. To overcome these challenges, Parameter-Efficient Fine-Tuning (PEFT) methods [53,61,69], such as PointCSA [31], have emerged as more promising alternatives, garnering significant attention.

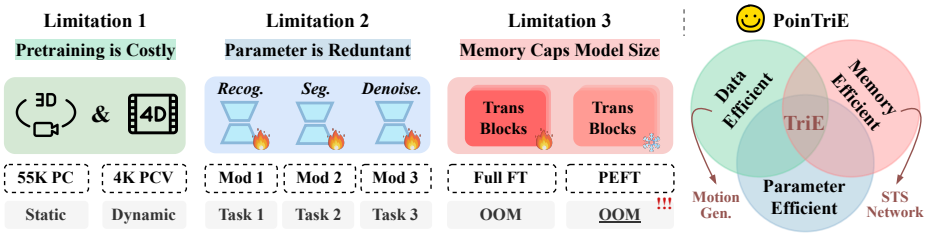

Fig. 1: Three key limitations existing in point cloud video learning. L1: PC and PCV data is two orders of magnitude smaller than images. L2: Using task-specific dedicated models leads to severe parameter redundancy. L3: Full fine-tuning often causes out-of-memory errors, and conventional PEFT methods still struggle to resolve this issue. To address these limitations, we propose PoinTriE, a unified framework that achieves strong performance along the three critical axes of efficiency.

Instead of updating all model parameters directly, PointCSA inserts a learnable temporal adapter into the intertwined feature space of Transformer [50] blocks, bridging the gap between static and dynamic domains while keeping the PFM frozen. This method significantly reduces the number of parameters requiring fine-tuning, thereby reducing computational overhead. Follow-up research has further enhanced this pipeline by improving aspects such as adapter architecture [46, 61], local geometric interaction [24], and backbone model scale [10].

Despite these advancements, most of the prior art still relies on additive structures, *i.e.*, trainable adapter parameters within the pretrained model across various downstream tasks. However, as depicted in Fig. 1, recent findings from NLP (Natural Language Processing) have revealed the redundancy of intermediate caches retained by each pretrained block during backpropagation in terms of GPU memory, underscoring the need to optimize the fine-tuning method to eliminate the substantial memory footprint [14, 47]. On the other hand, unlike images and language [36], collecting large-scale point cloud data requires enormous storage and computing resources [5], let alone dynamic point cloud videos (PCV) [30]. This implies that indiscriminately using a fixed training and fine-tuning paradigm may not fully tap into the potential of PFMs, motivating us to *efficiently* adjust the existing paradigm to better accommodate each task.

In this paper, we naturally consider data-, parameter-, and memory-efficiency as the ideal optimization objectives, which we refer to as tri-efficient transfer learning (TriETL). As the focus pivots towards the TriETL problem, several new research questions emerge. First, since the underlying nature of PCV learning is still underexplored, the definition of TriETL remains unclear. For instance, defining it as a single-stage process or a two-stage process leads to a critical consideration, namely whether the fine-tuning network has the privilege to intervene in the effectiveness of customized PFMs. This leads us to our first key question: (1) How can we appropriately and formally define TriETL? (§3.1). Second, once the definition of TriETL is established, a more essential question emerges. (2) How can we design an intuitive pipeline derived from this definition? (§3.2, 3.3).

To address these questions, we first theoretically analyze the TriETL problem in 3D vision. Our findings confirm the significance of TriETL and further reveal two key insights: rather than blindly expanding the scale of the dataset, it is better to carefully explore the intrinsic supervision within existing data and its variants. Moreover, the negative impact of intermediate cache occupancy on memory persists in the point cloud domain. Based on these insights, we propose PoinTriE, a novel two-stage PFM optimized for our objectives. Building on the definition, we utilize rigid body transformation to generate pseudo-motion trajectories, which comprise a series of variants derived from the same point cloud data. The Geometric-Motion Duality Network (GMD Net) first maps trajectory features and performs contrastive learning with their 2D projections and text corpora, while the subsequent dual-branch head assists in motion prior injection from the perspectives of regression and uncertainty. When it comes to fine-tuning, we freeze the backbone and adaptively learn the Spatio-temporal Side Network (STS Net) for each task in a ladder-side manner using a gradient flow masking strategy, thereby unlocking the full potential of PFM and low-rank linear transformation (LoRA) [20] units. Our main contributions are fourfold.

- To the best of our knowledge, we are the first to investigate the TriETL problem in 3D vision and provide the corresponding theoretical proofs.
- Building upon our insights into point cloud video learning, we propose a dedicated TriETL framework, termed PoinTriE, which can provide a baseline for further research from either a unified or an independent perspective.
- We present the GMD Network for pretraining, which introduces pseudo-motion trajectories to augment under-utilized static point cloud data for multi-modal contrastive learning. Then we allocate LoRA units to STS blocks via the gradient flow masking strategy, resulting in significantly fewer trainable parameters and lower GPU memory usage.
- Extensive experiments demonstrate that PoinTriE exhibits superior performance, *e.g.*, it attains 94.37% average accuracy on MSR-Action3D and 84.11% mIoU on Synthia 4D, with gains of 1.05pp and 0.95pp over baselines.

2 Related Work

2.1 Static&Dynamic Point Cloud Pretraining

Within the domain of point cloud pretraining, self-supervised strategies are primarily divided into contrastive and generative frameworks. The contrastive framework aims to maximize agreement between different augmentations of the same point cloud while minimizing similarity across distinct samples. Initially proposed for 2D images [7, 36, 38, 51], it has been effectively extended to 3D data [17, 54, 62, 63] and 4D data [15, 39, 41, 66]. For instance, PointContrast [57] was an early milestone, followed by CPR [42], PointCMP [41], and PointCPSC [43], which improved feature extraction via various priors. Multi-modal extensions further enrich this line of work. CrossPoint [1], CrossVideo [28] and VG4D [10] leverage modality correspondences to boost discriminability,

while RECON [35], PTM [8] and HyperPoint [45] embed contrastive objectives into generative pipelines to mitigate overfitting in Transformer-based methods. On the other hand, several advances focus on masked reconstruction. Inspired by MAE [19], PointMAE [33] and PointM2AE [64] simplify training to capture geometric details. Subsequent works, such as JointMAE [16], PointGPT [6], and PointDif [67], extend the concepts of multi-modal integration, GPT, and Diffusion frameworks to pretraining, respectively. Despite the effectiveness of MaST-Pre [40] and M2PSC [18] in 4D representations, we notice that existing methods possess two main limitations: (1) They require cumbersome and time-consuming data collection, which is not feasible for most laboratories and companies. (2) All of them learn motion directly from the decoded features. As noted in PointRAE [27] and PointSRA [55], such pretext tasks inadvertently push the encoder to learn shortcuts, leading to poor transferability for motion-aware downstream 4D tasks. To tackle these limitations, we turn our attention to the field of static point clouds, where relatively mature metadata is already available [5, 49]. Our encoder retains a strong capability in modeling spatial structures while emphasizing the capture of morphological priors to ensure the efficacy of transfer learning.

2.2 Efficient Transfer Learning

As highlighted above, PFMs are iterating at an unprecedented pace in terms of structure, method, and applications. This rapid evolution has fueled the PEFT community to become equally dynamic. Under the topic of efficient transfer learning (ETL), PEFT can be finely categorized into four schemes: selective, additive, prompt, and reparameterization. 3D-specific methods, such as IDPT [60], DAPT [69], PointPEFT [48], and GAPrompt [2], have achieved efficient adaptation to the unique requirements of 3D vision. PointCSA [31] pioneered the bridging of 3D and 4D domains via additive adapters, making cross-modal tuning possible. PointATA [46] decomposed fine-tuning into two stages, narrowing inter-domain discrepancy while mitigating overfitting risk. Nevertheless, parameter efficiency is not the community's sole priority as data and memory efficiency are equally indispensable. LST [47] and DTL [14] have set a precedent in the fields of language and 2D vision. They utilize side tuning to fuse intermediate information from the backbone network, thereby saving substantial memory. In this work, we hypothesize that the pretraining data scale and memory demands for special 4D scenarios may also play a crucial role, leading us to simultaneously incorporate data- and memory-efficiency into our focus. Furthermore, we extend TriETL to PFM architecture and demonstrate its practical usage on multiple high-dimensional vision tasks for the first time.

3 Method

3.1 Analysis and Formulation of TriETL

We first explore the underlying nature of TriETL and its formal definition to address question (1). Intuitively, TriETL can be defined in two distinct forms:

a single-stage process (where data-, parameter-, and memory-efficiency are all achieved during pretraining) or a two-stage process (where these three objectives are allocated to the pretraining and fine-tuning processes, respectively). These two definitions differ in whether the fine-tuning network is capable of undertaking some of these optimization objectives. If we can prove that these three properties may be mutually exclusive under the single-stage condition, we can thereby prove by contradiction that multi-stage optimization is the appropriate definition.

Given the well-established scaling laws for upstream pretraining performance in prior work, we adopt a simple yet empirically validated formulation as the theoretical foundation for our analysis. Specifically, in visual transfer learning scenarios with limited labeled data, the downstream task error rate E satisfies the following power-law scaling relation, which is adapted and generalized from Yang et al. (2025) [58]:

$$E\left(\mathcal{D}_p, \mathcal{M}, \mathcal{D}_f\right) = E_\infty + \frac{\mathcal{D}_p^{-\alpha}}{\lambda_p} + \frac{\mathcal{M}^{-\beta}}{\lambda_m} + \frac{\mathcal{D}_f^{-\gamma}}{\lambda_f}, \tag{1}$$

where all symbols are rigorously defined as follows. $E_\infty \in \mathbb{R}_+$ is a constant representing the irreducible error of the downstream task and a larger E indicates worse performance in downstream transfer learning. $\mathcal{D}_p$ denotes the pretraining data size (*corr.* data efficiency), $\mathcal{M}$ represents the model size (*corr.* parameter and memory efficiency), $\mathcal{D}_f$ denotes the fine-tuning data size, and λ determines the characteristic scale of each dimension.

To simplify the theoretical analysis, we define the constant term $\mathcal{C}_f = \frac{\mathcal{D}_f^{-\gamma}}{\lambda_f}$, which absorbs the fixed fine-tuning data contribution. We first introduce two axioms that underpin our subsequent theorem, formalizing the inherent trade-offs between model parameters and memory usage:

Axiom 1 (Parameter-Memory Monotonicity). For any pretrained model, the memory usage $O(\mathcal{M})$ is a strictly increasing, continuously differentiable function of the model size $\mathcal{M}$, *i.e.*, $O(\mathcal{M}) = j \cdot \mathcal{M} + o(\mathcal{M})$ for some constant j, where $o(\mathcal{M})$ is the lower-order term and negligible for large $\mathcal{M}$. Thus, a reduction in $\mathcal{M}$ implies a strict decrease in $O(\mathcal{M})$, and vice versa.

Axiom 2 (Dual-Constraint Performance Degradation). Let $\mathcal{D}_p \leq \mathcal{D}_{p,\max}$ and $\mathcal{M} \leq \mathcal{M}_{\max}$ denote constraints on pretraining data size and model size, respectively. Then, the downstream error $E(\mathcal{D}_p, \mathcal{M}, \mathcal{C}_f)$ is strictly larger than the error achieved when at least one constraint is relaxed.

Theorem (Constrained Pretraining Feasibility). Let $E_\tau \in \mathbb{R}_{++}$ be the target downstream error threshold, satisfying $E_\tau \geq E_\infty$. If the pretraining data size is constrained to a finite upper bound $\mathcal{D}_p = \mathcal{D}_p^*$ (*i.e.*, $\mathcal{D}_p \leq \mathcal{D}_p^* < +\infty$), then the model size $\mathcal{M}$ must be unconstrained to ensure that the feasible region of $\mathcal{M}$ is non-empty.

Proof. We proceed via three key steps: error simplification, inequality derivation, and limit analysis.

Step 1: Error Function Simplification. Define the constant term $\mathcal{C}_p^* = \frac{\mathcal{D}_p^{-\alpha,*}}{\lambda_p}$. Fix the pretraining data size and substitute $\mathcal{C}_p^*$ and $\mathcal{C}_f$ into Eq. (1), the downstream error simplifies to a univariate function of $\mathcal{M}$:

$$E(\mathcal{M}) = E_\infty + \mathcal{C}_p^* + \mathcal{C}_f + \frac{\mathcal{M}^{-\beta}}{\lambda_m}, \tag{2}$$

where $E(\mathcal{M})$ is strictly decreasing in $\mathcal{M}$, since $\frac{\partial E(\mathcal{M})}{\partial \mathcal{M}} = -\frac{\beta \mathcal{M}^{-(\beta+1)}}{\lambda_m} < 0$.

Step 2: Inequality Derivation for Feasibility. We require $E(\mathcal{M}) \leq E_\tau$. Substituting Eq. (2) into this inequality yields:

$$E_\infty + \mathcal{C}_p^* + \mathcal{C}_f + \frac{\mathcal{M}^{-\beta}}{\lambda_m} \leq E_\tau. \tag{3}$$

Combine the constant terms by defining $\Gamma = E_\tau - \left(E_\infty + \mathcal{C}_p^* + \mathcal{C}_f\right)$. By the definition of irreducible error, $E_\infty + \mathcal{C}_p^* + \mathcal{C}_f > E_\infty$, so Γ can be either positive or negative. Isolating the term involving $\mathcal{M}$ in Eq. (3), we obtain:

$$\mathcal{M}^{-\beta} \leq \lambda_m \cdot \Gamma. \tag{4}$$

Step 3: Limit Analysis and Feasibility. Note that $\mathcal{M} \in \mathbb{R}_{++}$ and $\beta, \lambda_m \in \mathbb{R}_{++}$, so $\mathcal{M}^{-\beta} = \frac{1}{\mathcal{M}^\beta} > 0$ for all finite $\mathcal{M}$. We take the limit of $E(\mathcal{M})$ as $\mathcal{M} \to +\infty$. By the limit property of negative power functions, we obtain:

$$\lim_{\mathcal{M}\to+\infty} E(\mathcal{M}) = E_\infty + \mathcal{C}_p^* + \mathcal{C}_f + \lim_{\mathcal{M}\to+\infty} \frac{\mathcal{M}^{-\beta}}{\lambda_m} = E_\infty + \mathcal{C}_p^* + \mathcal{C}_f. \tag{5}$$

Define $L = E_\infty + \mathcal{C}_p^* + \mathcal{C}_f$. By the Sign-Preserving Property of Limits [37], for any $\zeta > 0$, there exists $\mathcal{M}^* \in \mathbb{R}_{++}$ such that for all $M \geq M^*$, $|E(M) - L| < \zeta$. Choosing $\zeta = \Gamma$, we obtain $E(\mathcal{M}) < L + \Gamma = E_\tau$ for all $\mathcal{M} \geq \mathcal{M}^*$. The feasible region of $\mathcal{M}$ is defined as $\mathbb{M} = \{\mathcal{M} \in \mathbb{R}_{++} \mid E(\mathcal{M}) \leq E_\tau\}$. From the above analysis, $\{\mathcal{M} \in \mathbb{R}_{++} \mid \mathcal{M} \geq \mathcal{M}^*\} \subseteq \mathbb{M}$, which implies $\mathbb{M} \neq \emptyset$.

Remark. Theorem highlights a fundamental trade-off between pretraining data efficiency and parameter efficiency in limited transfer learning, which indicates that the above attributes cannot be simultaneously optimized in constrained pretraining. Motivated by this trade-off, we formally define the TriETL problem as a two-stage optimization framework.

3.2 Data-Efficient 3D Pretraining

Pseudo Motion Generation. To ensure that pseudo-motion adheres to the physical motion patterns of real PCVs, we predefine a set of rigid transformations. Given an input point cloud P, we first construct local point cloud patches via Farthest Point Sampling (FPS) and K-Nearest Neighbors (KNN) clustering. For the m-th rigid transformation $g_m^r \in SE(3)$, we define the transformation as:

$$g_m^r(\mathbf{p}_i) = R_m \mathbf{p}_i + t_m, \tag{6}$$

where $R_m \in \mathbb{R}^{3\times3}$ is a rotation matrix with rotation angles constrained to $\theta_m \in [-30°, 30°]$. $t_m \in \mathbb{R}^3$ is a translation vector whose components follow the uniform distribution $t_{m,x}, t_{m,y}, t_{m,z} \sim \mathcal{U}(-0.2, 0.2)$ *(cf Supp E)*.

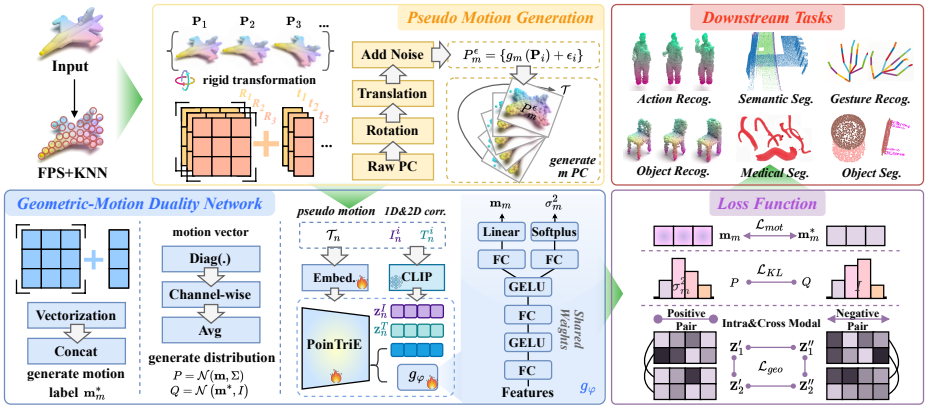

Fig. 2: The overall pretraining framework of PoinTriE. The input first undergoes rigid transformations to augment the data scale. Motion prediction and KL divergence enhance sensitivity to fine-grained dynamic changes. Cross-modal alignment subsequently injects additional semantic priors to support various downstream tasks.

For a single 3D point cloud P, we apply M independent rigid transformations and add per-point Gaussian noise $\epsilon_i \sim \mathcal{N}(0, \sigma^2 I)$, resulting in the noisy pseudo-motion point clouds $P_m^\epsilon = \{g_m^r(\mathbf{p}_i) + \epsilon_i \mid \mathbf{p}_i \in P\}$. By aggregating all pseudo-motion samples derived from the same input point cloud, we construct the final pseudo-motion trajectory set $\mathcal{T} = \{P, P_1^\epsilon, P_2^\epsilon, \ldots, P_M^\epsilon\}$.

Geometric-Motion Duality Network (GMD Net). For the pseudo-motion trajectory $\mathcal{T} = \{P, P_1^\epsilon, P_2^\epsilon, \ldots, P_M^\epsilon\}$ generated from a single-frame point cloud P, we divide the pseudo-motion samples $\{P_1^\epsilon, \ldots, P_M^\epsilon\}$ into the positive sample set as:

$$\mathcal{S}_{\text{pos}} = \{P_1^\epsilon, P_2^\epsilon, \ldots, P_M^\epsilon\}. \tag{7}$$

We take other point cloud and their trajectories within the batch as the negative sample set. We use our 3D point cloud encoder $f_\theta(\cdot)$ to encode point cloud features, obtaining sample-wise global features. For cross-modal alignment, we encode the image input I_i and text corpora T_i separately as follows:

$$\mathbf{z}_n^P = f_\theta(P_n), \quad \mathbf{z}_n^I = f_I(I_n), \quad \mathbf{z}_n^T = \text{Avg}(f_T(T_n)), \tag{8}$$

where $f_I(\cdot)$ is the image encoder, $f_T(\cdot)$ is the text encoder, and $\text{Avg}(\cdot)$ denotes the global average pooling of text features *(cf Supp A.1)*.

Meanwhile, we vectorize the real rigid transformation parameters corresponding to the m-th pseudo motion to construct the real motion label,

$$\mathbf{m}_m^* = [\text{vec}(R_m); t_m] \in \mathbb{R}^{12}, \tag{9}$$

where $\text{vec}(\cdot)$ is the matrix column vectorization operation. This label has a one-to-one correspondence with the pseudo-motion point cloud P_m^ϵ.

In this manner, we introduce two Gaussian distributions to respectively model the true motion distribution and the predicted motion distribution. The true motion distribution is $\mathcal{Q} = \mathcal{N}(\mathbf{m}_m^*, I)$, where the covariance matrix is the identity matrix I, indicating that the true motion value is definite and unambiguous. The predicted motion distribution is $\mathcal{P} = \mathcal{N}(\mathbf{m}_m, \Sigma_m)$, where $\mathbf{m}_m$ denotes the predicted motion vector and Σ_m is the predicted diagonal covariance matrix.

Loss Function. The total loss of GMD Net is composed of the geometric invariance loss $\mathcal{L}_{\text{geo}}$ and the motion consistency loss $\mathcal{L}_{\text{mot}}$. The geometric invariance loss comprises the intra-modal point cloud contrastive loss and the cross-modal alignment loss. First, for different pseudo-motion samples of the same input point cloud, we encourage the encoder to output similar global representations. Given a positive sample pair $\mathbf{z}_n^{t_1}$ and $\mathbf{z}_n^{t_2}$, the contrastive loss is defined as:

$$\mathrm{l}\left(n, t_1, t_2\right) = -\log \frac{\exp\left(s\left(\mathbf{z}_n^{t_1}, \mathbf{z}_n^{t_2}\right)/\tau\right)}{\sum_{\substack{k=1 \\ k\neq n}}^{B} \exp\left(s\left(\mathbf{z}_n^{t_1}, \mathbf{z}_k^{t_1}\right)/\tau\right) + \sum_{k=1}^{B} \exp\left(s\left(\mathbf{z}_n^{t_1}, \mathbf{z}_k^{t_2}\right)/\tau\right)}, \tag{10}$$

where B is the batch size, τ is the temperature coefficient, and $s(\cdot, \cdot)$ denotes the cosine similarity function. Furthermore, the intra-modal geometric invariance loss is given by:

$$\mathcal{L}_{\text{intra}} = \frac{1}{2B} \sum_{n=1}^{B} \left[\mathrm{l}\left(n, t_1, t_2\right) + \mathrm{l}\left(n, t_2, t_1\right)\right]. \tag{11}$$

The cross-modal loss aligns 3D point cloud, image, and text features into a unified semantic space. It leverages the vision-language prior of the pretrained CLIP [36] model to enhance the semantic consistency of geometric representations. We compute the weighted sum of all modal pairs with InfoNCE [34] loss:

$$\mathcal{L}_{\text{geo}} = \mathcal{L}_{\text{intra}} + \mathcal{L}_{(I,P)} + \mathcal{L}_{(S,P)} = \mathcal{L}_{\text{intra}} + \mathcal{L}_{\text{cross}}. \tag{12}$$

On the other hand, we employ a lightweight dual mapping network g_φ to simultaneously predict the motion vector and its associated uncertainty. This network concurrently outputs the predicted motion vector and the diagonal elements of the predicted covariance.

We use the L2 regression loss to enforce numerical consistency between the predicted motion vector $\mathbf{m}_m$ and the ground-truth motion label $\mathbf{m}_m^*$. To enhance the model's robustness against noise and point cloud perturbations, we constrain the predicted motion distribution to approximate the true motion distribution via KL divergence. The overall motion consistency loss is thus defined as:

$$\mathcal{L}_{\text{mot}} = \frac{1}{M} \sum_{m=1}^{M} \left(\eta_1 \left\|\mathbf{m}_m - \mathbf{m}_m^*\right\|_2^2 + \eta_2 \mathbb{D}_{\text{KL}}\left(\mathcal{N}\left(\mathbf{m}_m, \Sigma_m\right) \| \mathcal{N}\left(\mathbf{m}_m^*, I\right)\right)\right), \tag{13}$$

where $\eta_1 = 1$ and $\eta_2 = 0.1$, $\Sigma_m = \text{diag}\left(\sigma_{m,1}^2, \ldots, \sigma_{m,12}^2\right)$ is the predicted diagonal covariance matrix (*cf Supp B.2*).

These two types of information collectively form the supervision source for geometric-motion duality learning. The total loss of GMD Net is formulated as:

$$\mathcal{L}_{\text{GMDP}} = \mathcal{L}_{\text{geo}} + \delta \mathcal{L}_{\text{mot}}, \tag{14}$$

where δ is a coefficient that balances the supervision of geometry and motion.

3.3 Parameter- and Memory-Efficient 4D Transfer Learning

Despite the significant reduction in trainable parameters achieved by PEFT methods, the decrease in GPU memory usage is not proportional, rendering the fine-tuning of large-scale PFMs still challenging. To simplify the analysis, we assume the input PCV sample is $\mathbf{h}^{(0)} \in \mathbb{R}^{F\times N\times 3}$, where F is the length of the video and N is the number of points per frame. We exclude bias terms in the hidden layers. For the l-th layer, the linear transformation, activation value and the output can be expressed as:

$$\mathbf{v}^{(l)} = \mathbf{W}^{(l)}\mathbf{h}^{(l-1)}, \quad \mathbf{h}^{(l)} = \phi_l\left(\mathbf{v}^{(l)}\right), \quad \mathbf{o} = \mathbf{h}^{(l)}, \tag{15}$$

where $\mathbf{W}^{(l)} \in \mathbb{R}^{h_l\times h_{l-1}}$ denotes the weight matrix. ϕ_l is an element-wise differentiable activation function.

Formally, we define the objective function as $\mathcal{J} = \mathcal{L}(\mathbf{o}, y) + \frac{\mu}{2}\sum_{k=1}^{l}\left\|\mathbf{W}^{(k)}\right\|_F^2$, where $\mathcal{L}(\cdot,\cdot)$ is the differentiable task loss, y denotes the label, $\mu > 0$ is the regularization coefficient, and $\|\cdot\|_F$ represents the Frobenius norm.

The gradient of the objective function with respect to the output layer can be uniformly expressed as:

$$\boldsymbol{\delta}^{(l)} = \frac{\partial \mathcal{J}}{\partial \mathbf{v}^{(l)}} = \frac{\partial \mathcal{L}}{\partial \mathbf{o}} \odot \phi_l'\left(\mathbf{v}^{(l)}\right), \quad \frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(l)}} = \boldsymbol{\delta}^{(l)}\left(\mathbf{h}^{(l-1)}\right)^\top + \mu\mathbf{W}^{(l)}, \tag{16}$$

where $\odot$ denotes the Hadamard product.

Notably, $\phi_l'(\mathbf{v}^{(l)})$ and the outer product $\boldsymbol{\delta}^{(l)}(\mathbf{h}^{(l-1)})^\top$ depend on the forward propagation intermediate variables $\mathbf{v}^{(l)}$ and $\mathbf{h}^{(l-1)}$, respectively. By induction, gradient computation for each layer $k \in \{1,\ldots,l\}$ relies on at least one forward intermediate variable, either $\mathbf{v}^{(k)}$ or $\mathbf{h}^{(k-1)}$. Thus, even when only a subset of parameters is updated, PEFT methods incur nearly identical memory cost as full fine-tuning due to the need to cache these intermediate variables.

To address this memory bottleneck, we propose our Spatio-Temporal Side Network. As illustrated in Fig. 3, the core idea of STS Net is to decouple the weight update of lightweight auxiliary modules from the backbone network. This design drastically reduces the intermediate variables $\mathbf{v}^{(l)}$ and $\mathbf{h}^{(l-1)}$ that must be cached for backpropagation. As a result, STS Net is not only parameter-efficient but also substantially reduces the GPU memory footprint for fine-tuning PFMs.

Specifically, given a 3D backbone with C blocks, the forward computation can be written as:

$$\mathbf{o} = b_C\left(b_{C-1}\left(\ldots b_1\left(\mathbf{h}^{(\mathbf{0})}\right)\right)\right), \tag{17}$$

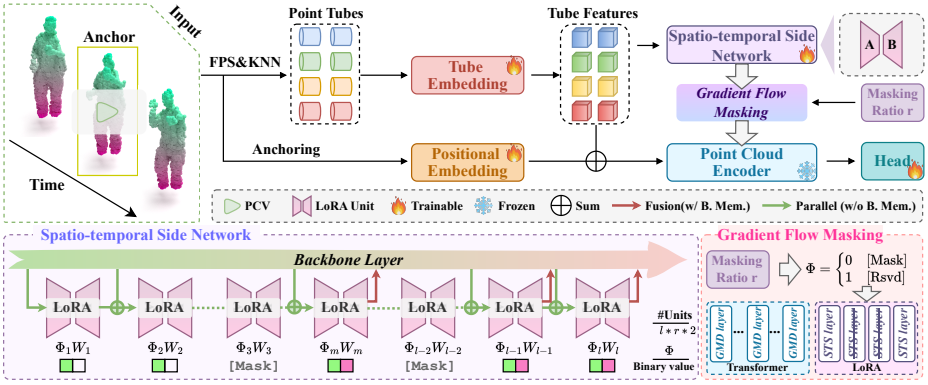

Fig. 3: The overall PCV fine-tuning framework in PoinTriE. First, anchor centers are sampled and embedded via FPS and KNN. Then, we freeze the backbone and insert the STS Net in parallel. At each block, the STS Net applies LoRA units to the input features to extract task-specific side information (green arrows). Starting from block m, this side information is fused back to the outputs of the block (red arrows) to adapt the backbone representations for downstream tasks. During fine-tuning, we adopt a gradient flow masking strategy to further reduce the parameters of the STS Net, which is controlled by masking ratio.

where b_i denotes the i-th backbone block *(cf Supp A.2/D.2)*.

STS Net comprises C low-rank linear transformation [20] matrices, with one matrix inserted into each block to extract task-specific information. Let $w_i = a_i c_i \in \mathbb{R}^{d\times d}$ be the low-rank weight matrix for the i-th block, where $a_i \in \mathbb{R}^{d\times d'}, c_i \in \mathbb{R}^{d'\times d}$, and $d' \ll d$. We initialize a_i from a uniform distribution and set c_i to zero. Empirically, we find that fully-equipped LoRA units suffer from high redundancy and tend to overfit on downstream datasets. To alleviate this issue, we further introduce a gradient flow masking strategy to reduce redundancy. This strategy randomly removes a subset of LoRA units, controlled by a predefined masking ratio, which improves model generalization and cross-modal transfer ability. Starting from the D-th block, the aggregated task-specific information $\mathbf{h}^{(i+1)}$ is used to adapt the backbone features $\mathbf{v}^{(i+1)}$ for downstream tasks via a residual adaptation.

$$\mathbf{h}'_{i+1} = \begin{cases} \Phi(\mathbf{h}_{i+1} + \phi(\mathbf{v}_{i+1})) & \text{if } i \geq D \\ \Phi\mathbf{h}_{i+1} & \text{otherwise} \end{cases}, \quad \Phi = \begin{cases} 0 & [\text{ Mask }] \\ 1 & [\text{ Rsvd }] \end{cases}. \tag{18}$$

4 Experiments

Extensive experiments are conducted on three point cloud video benchmarks: MSR-Action3D [23], SHREC'17 [44], and Synthia 4D [9]. Our backbone is first pretrained and then fine-tuned under the proposed TriETL paradigm. We evaluate the performance of our method on three downstream tasks including action

Table 1: Action recognition accuracy (%) on MSR-Action3D. SL: Supervised Learning; PD: Pretraining Data; Para: Tunable Parameters; Mem: Memory; FFT: Full Fine-tuning; MSR: MSR-Action3D; SN: ShapeNet; F: Frames; Avg: Average Performance Our method is highlighted in light blue. Same for subsequent tables.

Methods	Ref	Paradigm	PD	Para	Mem	8 F.	12 F.	16 F.	24 F.	Avg
MeteorNet [26]	ICCV	SL	N/A	N/A	N/A	81.14	86.53	88.21	88.50	86.09
PSTNet [13]	ICLR	SL	N/A	N/A	N/A	83.50	87.88	89.90	91.20	88.12
P4Trans [11]	CVPR	SL	N/A	N/A	N/A	83.17	87.54	89.56	90.94	87.80
Kinet [68]	CVPR	SL	N/A	N/A	N/A	83.84	88.53	91.92	93.27	89.39
PPTr [56]	ECCV	SL	N/A	N/A	N/A	84.02	89.89	90.31	92.33	89.13
LeaF [29]	ICCV	SL	N/A	N/A	N/A	84.50	N/A	91.50	93.84	N/A
PST-Trans [12]	TPAMI	SL	N/A	N/A	N/A	83.97	88.15	91.98	93.73	89.45
X4D-Scene [21]	AAAI	SL	N/A	N/A	N/A	86.47	N/A	92.56	93.90	N/A
MAMBA4D [25]	CVPR	SL	N/A	N/A	N/A	N/A	N/A	N/A	92.68	N/A
CPR [42]	AAAI	FFT	MSR.	100%	N/A	86.53	91.00	92.15	93.03	90.67
C2P [65]	CVPR	FFT	MSR.	100%	N/A	87.16	N/A	91.89	94.76	N/A
PointCMP [41]	CVPR	FFT	MSR.	100%	N/A	89.56	91.58	92.26	93.27	91.66
PointCPSC [43]	ICCV	FFT	MSR.	100%	N/A	88.89	90.24	92.26	92.68	91.01
MaST-Pre [40]	ICCV	FFT	MSR.	100%	N/A	N/A	N/A	N/A	94.08	N/A
PointCSA [31]	CVPR	PEFT	SN.	3.4%	26.4	90.72	92.39	93.38	94.77	92.81
PointATA [46]	TMM	PEFT	SN.	2.8%	28.5	91.28	92.68	94.07	95.28	93.32
PoinTriE	-	TriETL	SN.	2.2%	9.8	91.28	93.37	95.62	97.21	94.37

recognition, gesture recognition, and semantic segmentation. We further carry out comprehensive ablation studies to verify the effectiveness of each component within our method *(cf Supp B-D for more experiments and ablation studies)*.

4.1 Action Recognition

Dataset. We conduct experiments on MSR-Action3D [23], which contains 567 depth videos covering 20 action classes. Each video contains approximately 40 frames on average. Following previous work [31,46], we split the dataset into 270 training videos and 297 test videos. During fine-tuning, only point coordinates are utilized, and each frame is randomly sampled to 2048 points. Videos are partitioned into clips of 8, 12, 16, or 24 frames, and the final video-level accuracy is obtained by averaging clip-level predictions.

Comparison Results. To ensure a fair comparison, we primarily compare our method with those adopting similar training paradigms, specifically PointCSA [31] and PointATA [46]. As illustrated in Tab. 1, our method achieves the highest action recognition accuracy, surpassing both supervised and fully fine-tuned counterparts. This demonstrates that the proposed pretext tasks, which emphasize geometric invariance and motion consistency, enable the model to effectively capture spatio-temporal information in point cloud videos. Such a design also provides substantial benefits when applying static models to dynamic downstream tasks. Notably, PoinTriE maintains its superiority over existing methods under the 12, 16, and 24-frame settings, yielding an average

Table 2: Gesture recognition accuracy (%) on SHREC'17.

Methods	Reference	Paradigm	Acc.
PLSTM-b [32]	CVPR2020	SL	87.6
PLSTM-e [32]	CVPR2020	SL	93.5
PLSTM-P [32]	CVPR2020	SL	93.1
PLSTM-m [32]	CVPR2020	SL	94.7
PLSTM-l [32]	CVPR2020	SL	93.5
Kinet [68]	CVPR2022	SL	95.2
PointCMP [41]	CVPR2023	FFT	93.3
MaST-Pre [40]	ICCV2023	FFT	92.4
M2PSC [18]	ECCV2024	FFT	92.8
PointCSA [31]	CVPR2025	PEFT	95.2
PointATA [46]	TMM2026	PEFT	95.5
PoinTriE	-	TriETL	96.5

Table 3: 4D semantic segmentation results (%) on Synthia 4D.

Methods	Reference	Frame	mIoU
3D MinkNet14 [9]	CVPR2019	1	76.24
4D MinkNet14 [9]	CVPR2019	3	77.46
MeteorNet-M [26]	ICCV2019	2	81.47
MeteorNet-L [26]	ICCV2019	3	81.80
PSTNet [13]	ICLR2021	3	82.24
P4Trans [11]	CVPR2021	1	82.41
P4Trans [11]	CVPR2021	3	83.16
PST-Trans [12]	PAMI2022	1	82.92
PST-Trans [12]	PAMI2022	3	83.95
MAMBA4D [25]	CVPR2025	3	83.35
PointATA [46]	TMM2026	1	82.92
PointATA [46]	TMM2026	3	84.06
PoinTriE	-	3	84.11

improvement of 1.05%, which attests to the robustness of our method across diverse temporal resolutions.

4.2 Gesture Recognition

Dataset. We utilize SHREC'17 [44] for gesture recognition. The dataset contains 2800 videos covering 28 gesture categories. Following [40], we split the dataset into 1960 training videos and 840 test videos.

Comparison Results. As illustrated in Tab. 2, our method outperforms FFT and SL methods by a substantial margin, achieving the accuracy of 96.5%. Since gesture recognition inherently demands fine-grained modeling of geometric semantics, existing PEFT methods heavily rely on powerful pre-trained backbones such as PointBERT [59], PointMAE [33] and PointGPT [6], which lack dedicated designs for joint feature learning. This limitation is precisely addressed by our GMD Net. By incorporating the STS Net in parallel, PoinTriE further boosts the performance of the backbone model. With fine-tuning on limited target-domain data, PoinTriE sheds new light on point cloud video learning.

4.3 Semantic Segmentation

Dataset. To verify that PoinTriE handles point-level tasks, we apply it to 4D semantic segmentation on Synthia 4D [9]. This dataset includes six driving sequences where both objects and cameras are in motion. Each frame provides four stereo RGB-D images captured from a vehicle-mounted platform. Following [25, 46], we reconstruct point cloud videos and use the standard train–validation–test split with 19888 training frames, 815 validation frames, and 1886 test frames. Consistent with [26], we use clips formed by three consecutive frames. While single-frame segmentation is feasible, modeling temporal correlations effectively enhances scene understanding, segmentation accuracy, and robustness to noise.

Table 4: Ablation studies of the pretraining objectives w/o fine-tuning.

Model	Motion	Contrastive		KL	MSR.
		Intra	Cross		
A0	✗	✓	✗	✗	83.50
A1	✓	✓	✗	✗	93.03
A2	✓	✓	✓	✗	94.42
A3	✓	✓	✓	✓	95.42

Table 5: Ablation studies of the fine-tuning components.

Model	Adapter	Schemes		Mask	MSR.
		Side	Additive		
B0	✗	✗	✗	✗	95.42
B1	✓	✗	✓	✗	96.16
B2	✓	✓	✗	✗	96.86
B3	✓	✓	✓	✓	97.21

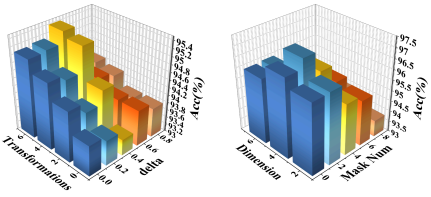

Fig. 4: Ablation studies of the pretraining *(left)* and fine-tuning *(right)* hyperparameters.

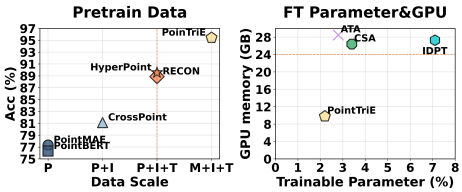

Fig. 5: Ablation studies of pretrained data *(left)*, GPU memory, and tunable parameter ratio *(right)*.

Comparison Results. We report performance using mean Intersection over Union (mIoU) in Tab. 3. Without adopting task-specific modules, PoinTriE consistently outperforms previous CNN, Transformer and Mamba models, achieving a 0.95 mIoU improvement over P4Transformer [11]. While our performance is comparable to existing PEFT methods, our framework offers superior practicality by drastically reducing GPU memory usage, which is essential for lab-scale experimental environments.

Visualization Samples. We present two segmentation results in Fig. 6. Our method demonstrates strong discriminative capability in scenarios with high semantic ambiguity, such as dense clusters of multiple objects.

4.4 Ablation Studies

Pretraining Objectives. Tab. 4 reports the impact of each pretraining objective on downstream tasks. The results indicate that static point cloud contrastive learning alone [57] fails to capture effective action patterns, leading to performance below that of the 4D-specific backbone. When the motion consistency objective is introduced, action recognition performance is substantially improved, which aligns with our core motivation. Incorporating KL divergence and more discriminative supervision, our method finally outperforms the 4D counterparts.

Fine-Tuning Components. After validating the effectiveness of our pretrained backbone, we further analyze the contribution of each component in the fine-tuning pipeline. As shown in Tab. 5, PointCSA [31] (B1), a representative

additive adapter, also yields improved performance with our pretrained backbone. This result demonstrates the strong versatility of PoinTriE. When the fine-tuning paradigm is replaced with the proposed side tuning, performance is further enhanced. We then regulate the parameters of STS Net via gradient flow masking to fully exploit the model capacity. These observations also verify the overfitting phenomenon discussed in PointATA [46].

Hyperparameters. We conduct a pairwise analysis on the hyperparameters used during pretraining and fine-tuning, with results reported in Fig. 4. In general, a larger number of rigid transformations consistently benefits task performance. However, an excessively large coefficient for motion consistency may introduce negative effects. Meanwhile, the masking ratio and LoRA rank should be carefully controlled to ensure the effectiveness of STS Net.

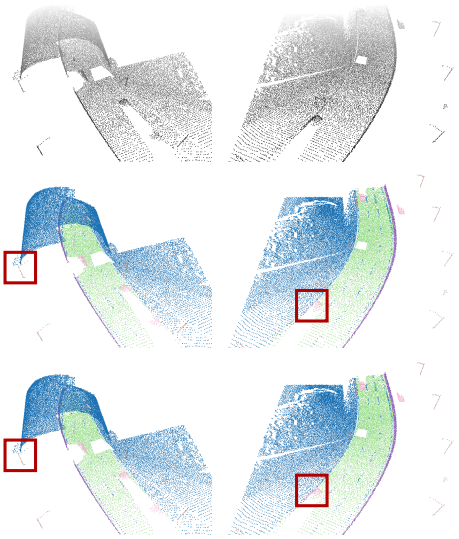

Fig. 6: Visualization of 4D semantic segmentation. Top: inputs. Middle: ground truth. Bottom: PoinTriE predictions. Red Box: Semantic details.

Training details. In Fig 5, we summarize the pretraining data used by several 3D backbones [1, 35, 45] and report their corresponding performance. It can be observed that larger pretraining datasets consistently yield improved performance across different backbones. PoinTriE leverages the ShapeNet dataset and its projections, further augmenting its scale with text corpora and pseudo-trajectories to achieve efficient data exploitation. More importantly, our fine-tuning pipeline consumes only about one-third of the memory required by existing PEFT methods [31, 46, 60], making it practical for most research settings.

5 Conclusion

In this paper, we investigate the TriETL problem in point cloud video learning. Motivated by our insights into transfer learning, we formalize the TriETL co-design framework with theoretical proofs. Based on this formulation, we propose PoinTriE framework, which integrates the GMD Net and STS Net. Our method achieves superior performance with lower computational overhead across several point cloud video benchmarks. We note that to acquire rich supervisory signals, we have to fully exploit the original data and its variants, which results in longer pretraining time. Due to the complex analysis required to identify highly informative data, we leave this direction for future exploration.

Acknowledgements

This work was supported in part by NSFC (No. 62125305) , the Natural Science Basis Research Plan in Shaanxi Province of China (No. 2025JC-JCQN-091) and Technology Innovation Leading Program of Shaanxi (Program No. 2024QY-SZX-23).

References

1. Afham, M., Dissanayake, I., Dissanayake, D., Dharmasiri, A., Thilakarathna, K., Rodrigo, R.: Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 9902–9912 (2022)
2. Ai, Z., Liu, Z., Lei, Y., Cui, Z., Zou, X., Zhou, J.: Gaprompt: Geometry-aware point cloud prompt for 3d vision model. In: Int. Conf. Mach. Learn. pp. 796–808. PMLR (2025)
3. Bao, Y., Ding, T., Huo, J., Liu, Y., Li, Y., Li, W., Gao, Y., Luo, J.: 3D gaussian splatting: Survey, technologies, challenges, and opportunities. IEEE Trans. Circuits Syst. Video Technol. **35**(7), 6832–6852 (2025)
4. Bao, Y., Liao, J., Huo, J., Gao, Y.: Distractor-free generalizable 3D gaussian splatting. In: Int. Conf. Learn. Represent. (2026)
5. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
6. Chen, G., Wang, M., Yang, Y., Yu, K., Yuan, L., Yue, Y.: Pointgpt: Auto-regressively generative pre-training from point clouds. Adv. Neural Inform. Process. Syst. **36** (2024)
7. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: Int. Conf. Mach. Learn. pp. 1597–1607 (2020)
8. Cheng, H., Zhu, J., Hu, N., Chen, J., Yan, W.: Ptm: Torus masking for 3d representation learning guided by robust and trusted teachers. IEEE Trans. Circuit Syst. Video Technol. **34**(12), 12158–12170 (2024)
9. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 3075–3084 (2019)
10. Deng, Z., Li, X., Li, X., Tong, Y., Zhao, S., Liu, M.: Vg4d: Vision-language model goes 4d video recognition. In: IEEE Int. Conf. Robot. Autom. pp. 5014–5020. IEEE (2024)
11. Fan, H., Yang, Y., Kankanhalli, M.: Point 4d transformer networks for spatio-temporal modeling in point cloud videos. In: IEEE Conf. Comput. Vis. Pattern Recog. (2021)
12. Fan, H., Yang, Y., Kankanhalli, M.: Point spatio-temporal transformer networks for point cloud video modeling. IEEE Trans. Pattern Anal. Mach. Intell. **45**(2), 2181–2192 (2023)
13. Fan, H., Yu, X., Ding, Y., Yang, Y., Kankanhalli, M.: Pstnet: point spatio-temporal convolution on point cloud sequences. In: Int. Conf. Learn. Represent. (2021)
14. Fu, M., Zhu, K., Wu, J.: Dtl: Disentangled transfer learning for visual recognition. In: AAAI. vol. 38, pp. 12082–12090 (2024)

15. Guo, Z., Sun, Y., Zhang, D., Cheng, H., Liu, J., Zhu, J., Mian, A.S.: Mantis: Mamba-native tuning is efficient for 3d point cloud foundation models. arXiv preprint arXiv:2605.03438 (2026)
16. Guo, Z., Zhang, R., Qiu, L., Li, X., Heng, P.A.: Joint-mae: 2d-3d joint masked autoencoders for 3d point cloud pre-training. In: IJCAI. pp. 791–799 (2023)
17. Han, X., Sun, Y., Lu, C.: Rethinking regressor in 3d gaussian pretraining. In: Pattern Recognit. Comput. Vis. Springer Nature (2025)
18. Han, Y., Xu, C., Xu, R., Qian, J., Xie, J.: Masked motion prediction with semantic contrast for point cloud sequence learning. In: Eur. Conf. Comput. Vis. pp. 414–431. Springer (2024)
19. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 16000–16009 (2022)
20. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. Int. Conf. Learn. Represent. **1**(2), 3 (2022)
21. Jing, L., Xue, Y., Yan, X., Zheng, C., Wang, D., Zhang, R., Wang, Z., Fang, H., Zhao, B., Li, Z.: X4d-sceneformer: Enhanced scene understanding on 4d point cloud videos through cross-modal knowledge transfer. In: AAAI. vol. 38, pp. 2670–2678 (2024)
22. Li, P., Sun, Y., Cheng, H.: Pointdico: Contrastive 3d representation learning guided by diffusion models. In: IEEE Int. Joint Conf. Neural Netw. (2025)
23. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3d points. In: IEEE Conf. Comput. Vis. Pattern Recog. Worksh. pp. 9–14 (2010)
24. Liang, D., Feng, T., Zhou, X., Zhang, Y., Zou, Z., Bai, X.: Parameter-efficient fine-tuning in spectral domain for point cloud learning. IEEE Trans. Pattern Anal. Mach. Intell. (2025)
25. Liu, J., Han, J., Liu, L., Aviles-Rivero, A.I., Jiang, C., Liu, Z., Wang, H.: Mamba4d: Efficient 4d point cloud video understanding with disentangled spatial-temporal state space models. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 17626–17636 (2025)
26. Liu, X., Yan, M., Bohg, J.: Meteornet: Deep learning on dynamic 3d point cloud sequences. In: Int. Conf. Comput. Vis. (2019)
27. Liu, Y., Chen, C., Wang, C., King, X., Liu, M.: Regress before construct: Regress autoencoder for point cloud self-supervised learning. In: ACM Int. Conf. Multimedia. pp. 1738–1749 (2023)
28. Liu, Y., Chen, C., Wang, Z., Yi, L.: Crossvideo: Self-supervised cross-modal contrastive learning for point cloud video understanding. In: IEEE Int. Conf. Robot. Autom. pp. 12436–12442 (2024)
29. Liu, Y., Chen, J., Zhang, Z., Huang, J., Yi, L.: Leaf: Learning frames for 4d point cloud sequence understanding. In: Int. Conf. Comput. Vis. pp. 604–613 (2023)
30. Liu, Y., Liu, Y., Jiang, C., Lyu, K., Wan, W., Shen, H., Liang, B., Fu, Z., Wang, H., Yi, L.: Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 21013–21022 (2022)
31. Lv, B., Zha, Y., Dai, T., Yuerong, X., Chen, K., Xia, S.T.: Adapting pre-trained 3d models for point cloud video understanding via cross-frame spatio-temporal perception. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 12413–12422 (2025)
32. Min, Y., Zhang, Y., Chai, X., Chen, X.: An efficient pointlstm for point clouds based gesture recognition. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 5760–5769 (2020)

33. Pang, Y., Wang, W., Tay, F.E., Liu, W., Tian, Y., Yuan, L.: Masked autoencoders for point cloud self-supervised learning. In: Eur. Conf. Comput. Vis. pp. 604–621 (2022)
34. Park, S., Kwak, N.: Feature-level ensemble knowledge distillation for aggregating knowledge from multiple networks. In: ECAI (2020)
35. Qi, Z., Dong, R., Fan, G., Ge, Z., Zhang, X., Ma, K., Yi, L.: Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining. In: Int. Conf. Mach. Learn. (2023)
36. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: Int. Conf. Mach. Learn. pp. 8748–8763 (2021)
37. Rudin, W.: Principles of Mathematical Analysis. McGraw-Hill, 3 edn. (1976)
38. Shao, M., Meng, L., Lv, X., Wu, M., Chen, X., Zhang, Q., Liu, C., Qiao, Y., Dong, C.: Unidef: Universal defense against unauthorized image manipulation. In: IEEE/CVF Conf. Comput. Vis. Pattern Recogn. pp. 8631–8640 (2026)
39. Shao, M., Zhang, Q., Chen, X., Lv, X., Meng, L., Liu, C., Zhan, Q., Jian, L.: Wavelet-driven 3D anomaly detection under pose-agnostic and sparse-view. In: IEEE/CVF Conf. Comput. Vis. Pattern Recogn. pp. 43083–43092 (2026)
40. Shen, Z., Sheng, X., Fan, H., Wang, L., Guo, Y., Liu, Q., Wen, H., Zhou, X.: Masked spatio-temporal structure prediction for self-supervised learning on point cloud videos. In: Int. Conf. Comput. Vis. pp. 16580–16589 (2023)
41. Shen, Z., Sheng, X., Wang, L., Guo, Y., Liu, Q., Zhou, X.: Pointcmp: Contrastive mask prediction for self-supervised learning on point cloud videos. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 1212–1222 (2023)
42. Sheng, X., Shen, Z., Xiao, G.: Contrastive predictive autoencoders for dynamic point cloud self-supervised learning. In: AAAI. vol. 37, pp. 9802–9810 (2023)
43. Sheng, X., Shen, Z., Xiao, G., Wang, L., Guo, Y., Fan, H.: Point contrastive prediction with semantic clustering for self-supervised learning on point cloud videos. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 16515–16524 (2023)
44. de Smedt, Q., Wannous, H., Vandeborre, J.P., Guerry, J., Le Saux, B., Filliat, D.: SHREC'17 Track: 3D Hand Gesture Recognition Using a Depth and Skeletal Dataset. In: Proc. 10th Eurographics Workshop on 3D Object Retrieval. pp. 1–6 (2017)
45. Sun, Y., Cheng, H., Lu, C., Li, Z., Wu, M., Lu, H., Zhu, J.: Hyperpoint: Multimodal 3d foundation model in hyperbolic space. Pattern Recognition **173**, 112800 (2026)
46. Sun, Y., Zhu, J., Cheng, H., Lu, C., Yang, Z., Chen, L., Wang, Y.: Align then adapt: Rethinking parameter-efficient transfer learning in 4d perception. IEEE Trans. Multimedia (2026)
47. Sung, Y.L., Cho, J., Bansal, M.: Lst: Ladder side-tuning for parameter and memory efficient transfer learning. Adv. Neural Inform. Process. Syst. **35**, 12991–13005 (2022)
48. Tang, Y., Zhang, R., Guo, Z., Ma, X., Zhao, B., Wang, Z., Wang, D., Li, X.: Point-peft: Parameter-efficient fine-tuning for 3d pre-trained models. In: AAAI. vol. 38, pp. 5171–5179 (2024)
49. Uy, M.A., Pham, Q.H., Hua, B.S., Nguyen, T., Yeung, S.K.: Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In: Int. Conf. Comput. Vis. pp. 1588–1597 (2019)
50. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Adv. Neural Inform. Process. Syst. (2017)

51. Wang, J., Lin, C., Nie, L., Liao, K., Shao, S., Zhao, Y.: Digging into contrastive learning for robust depth estimation with diffusion models. In: Proceedings of the 32nd ACM International Conference on Multimedia. p. 4129–4137. MM '24, ACM (Oct 2024)
52. Wang, J., Lin, C., Sun, L., Cao, Z., Yin, Y., Nie, L., Yuan, Z., Chu, X., Wei, Y., Liao, K., et al.: Geometry-guided reinforcement learning for multi-view consistent 3d scene editing. arXiv preprint arXiv:2603.03143 (2026)
53. Wang, S., Liu, X., Kong, L., Xu, J., Hu, C., Fang, G., Li, W., Zhu, J., Wang, X.: Pointlora: Low-rank adaptation with token selection for point cloud learning. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 6605–6615 (2025)
54. Wang, Y., Sun, Y., Wang, Q., Li, P., Lu, C., Zhang, D.: Pointrft: Explicit reinforcement fine-tuning for point cloud few-shot learning. In: IEEE Int. Conf. Multimedia Expo (2026)
55. Wei, L., Lu, J., Cheng, H., Zhu, J., Zhang, K.: Point-sra: Self-representation alignment for 3d representation learning. AAAI (2026)
56. Wen, H., Liu, Y., Huang, J., Duan, B., Yi, L.: Point primitive transformer for long-term 4d point cloud video understanding. In: Eur. Conf. Comput. Vis. pp. 19–35. Springer (2022)
57. Xie, S., Gu, J., Guo, D., Qi, C.R., Litany, L.G.O.: Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In: Eur. Conf. Comput. Vis. (2020)
58. Yang, W., Wei, Q., Ma, C., Tan, W., Yan, B.: Scaling laws for data-efficient visual transfer learning. In: ACM Int. Conf. Multimedia. pp. 2968–2976 (2025)
59. Yu, X., Tang, L., Rao, Y., Huang, T., Zhou, J., Lu, J.: Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022)
60. Zha, Y., Wang, J., Dai, T., Chen, B., Wang, Z., Xia, S.T.: Instance-aware dynamic prompt tuning for pre-trained point cloud models. In: Int. Conf. Comput. Vis. pp. 14161–14170 (2023)
61. Zha, Y., Wang, Y., Guo, H., Wang, J., Dai, T., Chen, B., Ouyang, Z., Yuerong, X., Chen, K., Xia, S.T.: Pma: Towards parameter-efficient point cloud understanding via point mamba adapter. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 16976–16986 (2025)
62. Zhang, D., Sun, Y., Li, P., Liu, Y., Lin, H., Xu, H., Mu, X., Lin, L., Yan, W., Yang, N., et al.: Pointcot: A multi-modal benchmark for explicit 3d geometric reasoning. arXiv preprint arXiv:2602.23945 (2026)
63. Zhang, D., Wang, Y., Sun, Y., Xu, H., Fan, P., Zhu, J.: Cmhanet: A cross-modal hybrid attention network for point cloud registration. Neurocomputing (2026)
64. Zhang, R., Guo, Z., Gao, P., Fang, R., Zhao, B., Wang, D., Qiao, Y., Li, H.: Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. In: Adv. Neural Inform. Process. Syst. (2022)
65. Zhang, Z., Dong, Y., Liu, Y., Yi, L.: Complete-to-partial 4d distillation for self-supervised point cloud sequence representation learning. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 17661–17670 (2023)
66. Zhang, Z., Sun, Y., Fang, C., Cheng, H., Liu, J., Zhu, J., Mian, A.S.: Diffusion masked pretraining for dynamic point cloud. arXiv preprint arXiv:2605.03639 (2026)
67. Zheng, X., Huang, X., Mei, G., Hou, Y., Lyu, Z., Dai, B., Ouyang, W., Gong, Y.: Point cloud pre-training with diffusion models. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 22935–22945 (2024)

68. Zhong, J.X., Zhou, K., Hu, Q., Wang, B., Trigoni, N., Markham, A.: No pain, big gain: classify dynamic point cloud sequences with static models by fitting feature-level space-time surfaces. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 8510–8520 (2022)
69. Zhou, X., Liang, D., Xu, W., Zhu, X., Xu, Y., Zou, Z., Bai, X.: Dynamic adapter meets prompt tuning: Parameter-efficient transfer learning for point cloud analysis. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 14707–14717 (2024)