BEYOND POINT ANCHORS: PATCH-AWARE ROTARY ENCODING FOR MASKED POINT CLOUD REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Point cloud patch tokens summarize local neighborhoods, yet anchor-only rotary encoding derives positional phases solely from normalized anchor coordinates. We introduce Patch-Aware Rotary Encoding (PaRE), which incorporates local patch geometry into rotary positional interactions through additive phase residuals derived from measured anchor-to-centroid offsets. The resulting relative phases combine anchor displacement with differences in centroid offsets, complementing the geometry encoded in token content. PaRE preserves the original anchors, fixed frequencies, and anchor-based normalization, and uses shared, bounded, frequency-wise residual coefficients that add only 32 learnable parameters in our configuration. With point-only pre-training on ShapeNet in a masked latent prediction framework, PaRE achieves strong performance across diverse downstream classification and segmentation tasks, including 90.77% overall accuracy on ScanObjectNN's PB_T50_RS setting. These results demonstrate the effectiveness of patch-aware rotary interactions for learning transferable representations through masked point cloud pre-training. Code is provided in the appendix.

1 INTRODUCTION

Masked point cloud modeling learns transferable 3D representations by predicting missing information from visible observations. Point-BERT predicts discrete patch tokens (Yu et al., 2022), Point-MAE reconstructs masked point patches (Pang et al., 2022), and Point-JEPA predicts in representation space (Saito et al., 2025). Despite different targets, these methods represent local neighborhoods as patch tokens. Alongside the prediction target, the geometric information used to define positional interactions between patch tokens during encoding is an important design choice.

A patch token summarizes a neighborhood, whereas an anchor-only rotary phase depends on normalized anchor coordinates. With the anchors and their normalization reference fixed, changes in surrounding points leave this phase unchanged. Local geometry can affect attention through token content, but it does not directly enter the anchor-only rotary operator. Figure 1 compares these encoding schemes and shows how PaRE incorporates patch geometry into relative rotary phases.

Existing work incorporates geometric cues through relative-coordinate attention in Point Transformer (Zhao et al., 2021), prediction of masked patch-center positional embeddings in PCP-MAE (Zhang et al., 2024), and coordinate-based PointROPE in LitePT (Yue et al., 2026). We ask a complementary question: *can measured local patch statistics complement anchor displacement directly in relative rotary phases while preserving the original sampled anchors?*

We introduce **Patch-Aware Rotary Encoding (PaRE)**, which augments the anchor-based phase component with a residual derived from each patch's anchor-to-centroid offset. Building on RoPE (Su et al., 2024), PaRE normalizes this first-order statistic, projects it onto fixed rotary frequency vectors, and scales the projections with bounded, frequency-wise coefficients. The coefficients are input-independent and shared across tokens, attention heads, and encoder layers, adding 32 learnable parameters in our configuration. PaRE retains the original anchors, fixed axis-frequency assignment, and anchor-based normalization. Its relative phase combines anchor displacement with differences in centroid offsets, complementing the geometry encoded in token content.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

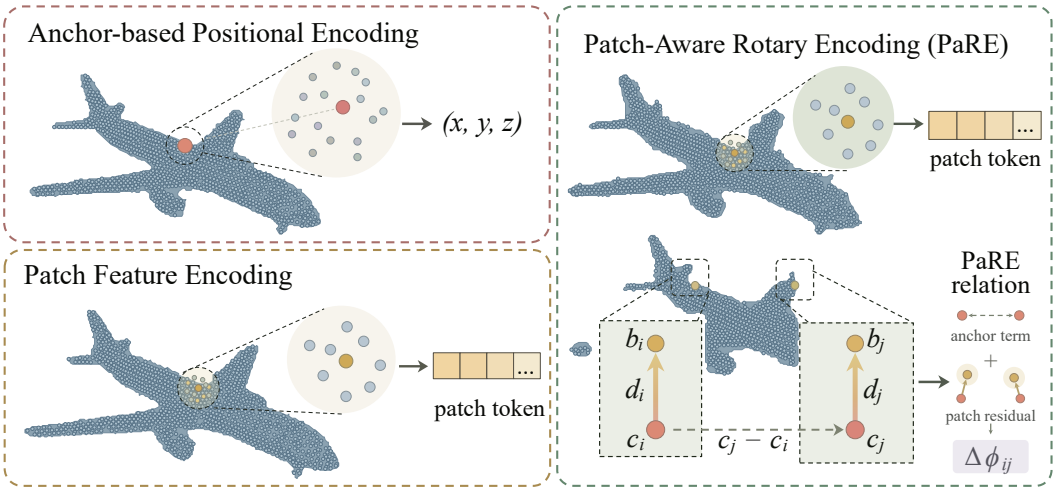


Figure 1: Motivation for PaRE: enriching anchor-based rotary relations with patch geometry.

We study PaRE within a bidirectional masked latent prediction framework, using only point coordinates for pre-training on ShapeNet (Chang et al., 2015). The Anchor-only baseline uses the same framework without the patch phase residual. Both variants share patch grouping, additive anchor embeddings, the prediction objective, and the target-generation procedure throughout pre-training.

PaRE achieves 95.70%, 93.80%, and 90.77% overall accuracy on the OBJ_BG, OBJ_ONLY, and PB_T50_RS settings of ScanObjectNN (Uy et al., 2019), respectively, exceeding Anchor-only by 1.38, 1.03, and 0.76 percentage points. It also outperforms No RoPE, which omits query/key rotations, on all three settings. On ModelNet40 (Wu et al., 2015), PaRE obtains 93.8% accuracy with 1,024 points without voting. We further evaluate transfer to few-shot classification, ShapeNetPart part segmentation (Yi et al., 2016), and S3DIS Area-5 semantic segmentation (Armeni et al., 2016).

Our contributions are threefold: (i) we characterize a granularity gap between neighborhood-based patch tokens and anchor-only rotary phases: with anchors and their normalization reference fixed, local geometric changes leave the phases unchanged; (ii) we introduce PaRE, which preserves the anchor phase and its fixed frequencies while adding centroid-offset phase residuals controlled by bounded, frequency-wise coefficients, and derive how relative phases combine anchor displacement with differences in centroid offsets; and (iii) we demonstrate gains over Anchor-only and No RoPE on all three ScanObjectNN settings, with controlled ablations supporting patch-offset correspondence, frequency-wise local residuals, and retention of pre-trained coefficients during fine-tuning.

2 Related Work

2.1 Point Cloud Pre-training

Point cloud pre-training varies in prediction targets, prediction order, and sources of supervision. Point-BERT predicts discrete tokens for masked point patches (Yu et al., 2022), Point-MAE reconstructs masked patches in coordinate space (Pang et al., 2022), and Point-M2AE extends masked autoencoding to hierarchical, multi-scale representations (Zhang et al., 2022). MaskPoint discriminates masked object points from sampled noise (Liu et al., 2022). Point-JEPA predicts contextualized features from a momentum-updated target encoder (Saito et al., 2025), while MaskClu combines masked prediction of cluster assignments and cluster centers with instance-level representation alignment across masked views (Ren et al., 2026). Beyond masked prediction, PointGPT autoregressively generates spatially ordered point patches (Chen et al., 2023). ACT transfers knowledge from cross-modal teachers (Dong et al., 2023), while I2P-MAE uses pre-trained 2D models to guide masking and feature reconstruction (Zhang et al., 2023). We study PaRE within a single-modal masked latent prediction framework, using stop-gradient targets from the same online tokenizer.

The prediction objective and target-generation procedure are held fixed across positional variants to examine how local patch geometry enters positional interactions between tokens during encoding.

2.2 Local geometry and positional cues

Geometric cues enter point cloud networks through local feature aggregation, attention conditioning, and prediction targets. PointNet++ hierarchically aggregates local neighborhoods (Qi et al., 2017b), while PointCNN learns $\mathcal{X}$-transformations for convolutional feature aggregation (Li et al., 2018). Point Transformer uses relative coordinates to condition attention and transform features (Zhao et al., 2021). Geometric supervision provides another route: MaskFeat3D predicts surface normals and surface variations (Yan et al., 2024), while GeoMAE includes centroid, normal, and curvature prediction (Tian et al., 2023). PCP-MAE predicts positional embeddings of masked patch centers, replacing directly supplied embeddings in the decoder (Zhang et al., 2024). Point-MaDi combines center diffusion with patch diffusion conditioned on predicted centers and latent features (Xiao et al., 2025). PaRE uses measured anchor-to-centroid offsets to construct phase residuals in encoder attention, retaining the original anchors and additive anchor embeddings. This first-order geometric signal complements learned token content without adding a geometric reconstruction target.

2.3 Rotary positional encoding

RoFormer introduces RoPE, using query and key rotations to encode relative position in attention (Su et al., 2024). Heo et al. (2024) examine axial formulations for vision transformers and introduce learnable mixed frequencies. For point clouds, LitePT provides parameter-free PointROPE driven by three-dimensional coordinates (Yue et al., 2026). Selective RoPE introduces learned, input-dependent rotations for sequence modeling (Movahedi et al., 2026). For patch tokens, local neighborhood statistics provide positional cues beyond the anchor coordinates used to locate each token. PaRE adds centroid-offset phase residuals while retaining the fixed axis-frequency assignment and anchor-based normalization. It learns bounded, frequency-wise residual coefficients shared across tokens, heads, and encoder layers. These coefficients are input-independent model parameters; patch dependence comes from measured geometry. The learned coefficients thus modulate local phase residuals at each frequency while preserving the original anchor phase component.

3 Method

PaRE augments anchor-based rotary phases with patch centroid-offset residuals controlled by bounded, frequency-wise coefficients. This lets local patch geometry directly participate in positional interactions while preserving the original anchors. Figure 2 presents the overall pipeline, patch-aware rotary encoding, and relative geometric interpretation of pairwise attention.

3.1 Patch tokens and local geometry

Given a point cloud $\boldsymbol{X} \in \mathbb{R}^{N\times 3}$, we select G anchors $\{\boldsymbol{c}_i\}_{i=1}^{G}$ by farthest point sampling and gather S nearest neighbors $\{\boldsymbol{x}_{i,s}\}_{s=1}^{S}$ per anchor (Pang et al., 2022). The patches may overlap. We define each anchor-centered patch and its centroid offset, the mean displacement from the anchor, as

$$\mathcal{P}_i = \{\boldsymbol{x}_{i,s} - \boldsymbol{c}_i\}_{s=1}^{S}, \qquad \boldsymbol{d}_i = \frac{1}{S}\sum_{s=1}^{S}(\boldsymbol{x}_{i,s} - \boldsymbol{c}_i). \tag{1}$$

The centroid is $\boldsymbol{b}_i = \frac{1}{S}\sum_{s=1}^{S}\boldsymbol{x}_{i,s}$, so $\boldsymbol{d}_i = \boldsymbol{b}_i - \boldsymbol{c}_i$ (Figure 2, top). A shared PointNet-style tokenizer (Qi et al., 2017a) produces $\boldsymbol{Z} = T_\psi(\mathcal{P}_1, \ldots, \mathcal{P}_G)$, where $\boldsymbol{z}_i \in \mathbb{R}^C$ encodes patch i.

With anchors and their normalization reference fixed, anchor-only rotary phases remain unchanged as local support varies. PaRE introduces $\boldsymbol{d}_i$ into the phase to complement the anchor with a first-order patch statistic. This statistic supplements the tokenizer's richer content features but does not fully describe patch geometry, since distinct point distributions can share the same centroid offset.

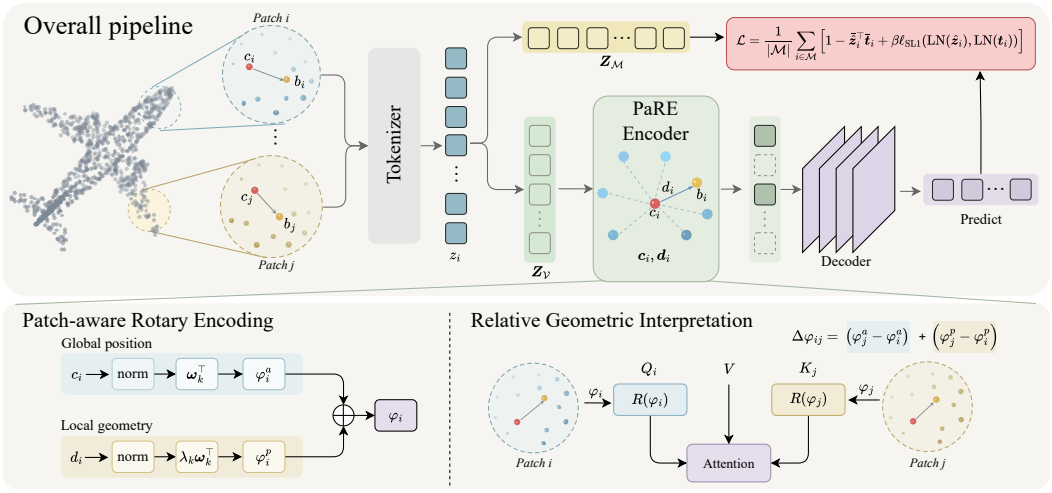


Figure 2: Overview of PaRE. Top: patch tokenization and masked latent prediction. Bottom left: patch-aware rotary encoding. Bottom right: relative geometric interpretation of attention.

3.2 Patch-aware rotary encoding

Anchor-referenced normalization. The anchor and offset branches in Figure 2 (bottom left) share the same anchor-derived normalization radius. For a nonempty encoder index set $\mathcal{I}$, we retain the anchor-only reference center and radius, both computed from the original anchors indexed by $\mathcal{I}$:

$$\bar{\boldsymbol{c}}_{\mathcal{I}} = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \boldsymbol{c}_i, \qquad r_{\mathcal{I}} = \max\left\{\epsilon, \max_{i\in\mathcal{I}} \|\boldsymbol{c}_i - \bar{\boldsymbol{c}}_{\mathcal{I}}\|_2\right\}. \tag{2}$$

$\epsilon > 0$ is a numerical safeguard. We use $\tilde{\boldsymbol{c}}_i = (\boldsymbol{c}_i - \bar{\boldsymbol{c}}_{\mathcal{I}})/r_{\mathcal{I}}$ and $\tilde{\boldsymbol{d}}_i = \boldsymbol{d}_i/r_{\mathcal{I}}$. Only anchors are centered; the reference uses original anchors rather than centroid-shifted positions. Offsets use the same radius. During masked pre-training, $\mathcal{I}$ indexes visible tokens, and PaRE uses only their offsets.

Patch-dependent phases. For an even head dimension d_h, let $K = d_h/2$ be the number of paired rotary slots. We retain the fixed axis-frequency assignment $\boldsymbol{\omega}_k = 2\pi\vartheta^{-k/K}\boldsymbol{e}_{1+(k \bmod 3)}$, where $k = 0, \ldots, K-1$, $\vartheta = 100$, and $\boldsymbol{e}_a$ is Cartesian basis vector a. PaRE defines rotary phases:

$$\varphi_{i,k} = \underbrace{\boldsymbol{\omega}_k^\top \tilde{\boldsymbol{c}}_i}_{\varphi^{\mathrm{a}}_{i,k}} + \underbrace{\lambda_k \boldsymbol{\omega}_k^\top \tilde{\boldsymbol{d}}_i}_{\varphi^{\mathrm{p}}_{i,k}}, \qquad \lambda_k = \lambda_{\max}\tanh(g_k). \tag{3}$$

$\varphi^{\mathrm{a}}_{i,k}$ is the anchor phase and $\varphi^{\mathrm{p}}_{i,k}$ is the patch residual. Both terms share the fixed frequency vector $\boldsymbol{\omega}_k$, with λ_k weighting the offset projection. Figure 2 omits k from phase labels. We set $\lambda_{\max} = 0.5$ and initialize λ_k to 0.25 using $g_k = \operatorname{arctanh}(0.25/0.5)$. The bound constrains the residual coefficient, not the phase angle. Because the coefficients may differ across slots and take either sign, PaRE is not restricted to rotary encoding at a single convex combination of the anchor and centroid.

A single vector $\boldsymbol{g} = (g_k)_{k=0}^{K-1}$ is shared across tokens, heads, and encoder layers. With $C = 384$ and six heads, $K = 32$, so PaRE adds 32 learnable parameters. These are input-independent; local patch geometry enters through $\boldsymbol{d}_i$. Patch grouping and the additive anchor embedding remain unchanged.

3.3 Relative geometric interpretation

For the token pair illustrated in Figure 2 (bottom right), let $\boldsymbol{Q}_i, \boldsymbol{K}_j \in \mathbb{R}^{d_h}$ denote the query and key vectors for tokens $i, j \in \mathcal{I}$ in one attention head. Their rotated versions are $\boldsymbol{Q}'$ and $\boldsymbol{K}'$. Layer and head indices are suppressed. For rotary slot k, let $\boldsymbol{Q}_{i,k}, \boldsymbol{K}_{j,k} \in \mathbb{R}^2$ denote the corresponding feature pairs, and $\boldsymbol{Q}'_k, \boldsymbol{K}'_k$ their rotated counterparts. Following RoPE (Su et al., 2024), let $\boldsymbol{R}(\varphi)$ denote a two-dimensional rotation. We apply $\boldsymbol{Q}'_k = \boldsymbol{R}(\varphi_{i,k})\boldsymbol{Q}_{i,k}$ and $\boldsymbol{K}'_k = \boldsymbol{R}(\varphi_{j,k})\boldsymbol{K}_{j,k}$,

while the value features $\boldsymbol{V}$ remain unrotated. In Figure 2 (bottom right), $\boldsymbol{R}(\varphi_i)$ denotes the block-diagonal rotation formed from $\boldsymbol{R}(\varphi_{i,k})$ over all K slots; the notation for token j is analogous. Using $\boldsymbol{R}(\varphi_{i,k})^\top \boldsymbol{R}(\varphi_{j,k}) = \boldsymbol{R}(\varphi_{j,k} - \varphi_{i,k})$, the pre-softmax attention logit is

$$a_{ij} = \frac{(\boldsymbol{Q}')^\top \boldsymbol{K}'}{\sqrt{d_h}} = \frac{1}{\sqrt{d_h}} \sum_{k=0}^{K-1} \boldsymbol{Q}_{i,k}^\top \boldsymbol{R}(\Delta\varphi_{ij,k}) \boldsymbol{K}_{j,k}, \tag{4}$$

where

$$\begin{aligned} \Delta\varphi_{ij,k} &= \varphi_{j,k} - \varphi_{i,k} \\ &= (\varphi^{\mathrm{a}}_{j,k} - \varphi^{\mathrm{a}}_{i,k}) + (\varphi^{\mathrm{p}}_{j,k} - \varphi^{\mathrm{p}}_{i,k}) \\ &= \frac{\boldsymbol{\omega}_k^\top (\boldsymbol{c}_j - \boldsymbol{c}_i)}{r_{\mathcal{I}}} + \lambda_k \frac{\boldsymbol{\omega}_k^\top (\boldsymbol{d}_j - \boldsymbol{d}_i)}{r_{\mathcal{I}}}. \end{aligned} \tag{5}$$

For each query, a softmax over encoder keys converts logits into weights for aggregating unrotated values. The relative phase combines anchor displacement with differences in local centroid offsets. Setting all $\lambda_k = 0$ recovers Anchor-only, not No RoPE, and a common offset added to all $\boldsymbol{d}_i$ cancels from the relative phase. This cancellation holds for the rotary factors with fixed $\boldsymbol{Q}_i$ and $\boldsymbol{K}_j$; it does not establish invariance of the complete encoder to changes in patch content or to 3D rotations.

3.4 Masked latent prediction

Visible-token encoding. We use a shared masked latent prediction framework. All patches are tokenized before masking, with BatchNorm statistics shared across patches. Given random masked indices $\mathcal{M}$ and visible indices $\mathcal{V} = \{1, \ldots, G\} \setminus \mathcal{M}$, with $0 < |\mathcal{M}| < G$, we partition $\boldsymbol{Z}$ into $\boldsymbol{Z}_{\mathcal{V}}$ and $\boldsymbol{Z}_{\mathcal{M}}$. In the overall pipeline of Figure 2 (top), $\boldsymbol{Z}_{\mathcal{V}}$ enters the encoder, while $\boldsymbol{Z}_{\mathcal{M}}$ provides the corresponding stop-gradient latent targets. The encoder processes visible tokens bidirectionally:

$$\boldsymbol{H}_{\mathcal{V}} = E_\theta(\{\boldsymbol{z}_i + e_{\mathrm{enc}}(\boldsymbol{c}_i)\}_{i \in \mathcal{V}}; \boldsymbol{\Phi}_{\mathcal{V}}), \tag{6}$$

where e_{enc} is the additive anchor embedding and $\boldsymbol{\Phi}_{\mathcal{V}} = (\varphi_{i,k})_{i \in \mathcal{V},\, 0 \le k < K}$ is the phase array for $\mathcal{I} = \mathcal{V}$. The visible-only restriction applies to phase computation, not to the pre-mask tokenizer.

Masked-patch prediction and targets. We restore the visible encoder outputs $\boldsymbol{h}_i$ to their original patch indices, setting $\boldsymbol{u}_i = \boldsymbol{h}_i$ for $i \in \mathcal{V}$ and $\boldsymbol{u}_i = \boldsymbol{m}$ otherwise, where $\boldsymbol{m}$ is a shared learned mask token. The masked features $\boldsymbol{Z}_{\mathcal{M}}$ are not inserted into the decoder input. A four-layer bidirectional decoder uses a separate additive anchor embedding and no RoPE. The latent predictor P_η, shown as "Predict" in Figure 2 (top), maps decoder outputs at masked positions to latent predictions:

$$\hat{\boldsymbol{z}}_i = P_\eta\Big(\big[D_\xi\big(\{\boldsymbol{u}_j + e_{\mathrm{dec}}(\boldsymbol{c}_j)\}_{j=1}^{G} \big) \big]_i \Big), \qquad i \in \mathcal{M}. \tag{7}$$

The latent targets $\boldsymbol{t}_i = \mathrm{sg}(\boldsymbol{z}_i)$, $i \in \mathcal{M}$, come from the same online tokenizer, with stop-gradient applied only to the target branch. Only predictions at masked positions are matched to their corresponding targets, without a momentum teacher, next-token shift, or coordinate reconstruction.

Objective and transfer. All positional variants share the same masked latent prediction objective, combining cosine distance with a Smooth-L_1 term:

$$\mathcal{L} = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \left[1 - \bar{\hat{\boldsymbol{z}}}_i^\top \bar{\boldsymbol{t}}_i + \beta\, \ell_{\mathrm{SL1}}(\mathrm{LN}(\hat{\boldsymbol{z}}_i), \mathrm{LN}(\boldsymbol{t}_i)) \right], \tag{8}$$

where $\bar{\boldsymbol{v}} = \boldsymbol{v} / \max(\|\boldsymbol{v}\|_2, \epsilon)$, LN denotes non-affine layer normalization across channels, and ℓ_{SL1} is the channel-averaged Smooth-L_1 loss with transition parameter one. We set $\beta = 0.05$ and average losses over the batch. For downstream transfer, we discard the decoder, mask token, and latent predictor, and attach task-specific heads to the tokenizer and PaRE encoder. The learned positional coefficients are retained and fine-tuned by default; Section 4.4 examines alternative initializations.

Table 1: ScanObjectNN classification (OA, %). Best results are in bold; – denotes unreported results.

Method	OBJ_BG	OBJ_ONLY	PB_T50_RS
Supervised learning			
PointNet (Qi et al., 2017a)	73.30	79.20	68.00
DGCNN (Wang et al., 2019)	82.80	86.20	78.10
PointNeXt (Qian et al., 2022)	91.90	91.00	88.10
PCM (Zhang et al., 2025a)	–	–	88.40
SPM (Wu et al., 2025)	90.20	89.50	84.20
PointLearner (Qu et al., 2026)	–	–	89.80
3DSMT (Zhou et al., 2026)	90.80	89.70	90.40
Single-modal self-supervised pre-training			
Point-MAE (Pang et al., 2022)	90.02	88.29	85.18
Point-M2AE (Zhang et al., 2022)	91.22	88.81	86.43
MaskFeat3D (PointNeXt) (Yan et al., 2024)	92.70	92.00	88.60
PointDif (Zheng et al., 2024)	93.29	91.91	87.61
Point-FEMAE (Zha et al., 2024)	95.18	93.29	90.22
PointMamba (Liang et al., 2024)	94.32	92.60	89.31
Point-MAE + CSeP (Zha et al., 2025)	91.32	90.30	86.64
PointMamba + CSeP (Zha et al., 2025)	91.95	90.59	86.12
RI-MAE (Su et al., 2025)	91.90	89.20	85.80
Point-JEPA (Saito et al., 2025)	93.30	90.30	86.90
SI-Mamba (Bahri et al., 2025)	94.32	91.91	89.10
Point-PQAE (Zhang et al., 2025b)	95.00	93.60	89.60
PointGAC (Li et al., 2025)	92.60	91.40	89.40
PreDifPoint (Qiu et al., 2025)	93.42	92.25	87.96
AsymDSD-S (Leijenaar & Kasaei, 2025)	94.32	91.91	90.53
Point-MaDi (Xiao et al., 2025)	95.52	93.46	89.52
ZigzagPointMamba (Diao et al., 2025)	94.15	92.10	88.65
MaskClu (Ren et al., 2026)	93.59	92.04	89.28
PaRE (ours)	**95.70**	**93.80**	**90.77**
Cross-modal self-supervised pre-training			
TAP (Wang et al., 2023)	90.36	89.50	85.67
Joint-MAE (Guo et al., 2023)	90.94	88.86	86.07
ACT (Dong et al., 2023)	93.29	91.91	88.21
I2P-MAE (Zhang et al., 2023)	94.15	91.57	90.11
UniPre3D (Wang et al., 2025)	92.60	92.08	87.93
PointSD (Chen et al., 2025)	95.18	93.63	90.08
DR-Point (Fei et al., 2026)	89.51	88.97	84.66
KR-MAE (Jin et al., 2026)	94.45	91.91	89.28

4 EXPERIMENTS

We evaluate PaRE on point cloud classification and segmentation and examine patch-aware geometry, positional parameterization, and pre-training and coefficient transfer through comparisons.

4.1 EXPERIMENTAL SETUP

We pre-train PaRE on ShapeNet (Chang et al., 2015) using point coordinates only. Transfer is evaluated on ScanObjectNN (Uy et al., 2019) and ModelNet40 (Wu et al., 2015) for classification, ShapeNetPart (Yi et al., 2016) for part segmentation, and S3DIS Area-5 (Armeni et al., 2016) for semantic segmentation. We report overall accuracy (OA) for classification, using OBJ_BG, OBJ_ONLY, and PB_T50_RS on ScanObjectNN. For ShapeNetPart, we report class-average and instance-average IoU (Cls. mIoU and Inst. mIoU); for S3DIS, we report mean class accuracy (mAcc) and mean IoU (mIoU). All metrics are reported in percent, unless otherwise explicitly stated.

Table 2: ModelNet40 classification (%). (a) Few-shot accuracy (mean ± std. over 10 evaluations). (b) OA with 1,024 input points, without voting. Best values per column, including ties, are bold.

(a) Few-shot classification

Method	5-way		10-way	
	10-shot	20-shot	10-shot	20-shot
Supervised learning				
PointNet (rand)	52.0 ± 12.2	57.8 ± 15.5	46.6 ± 13.5	35.2 ± 15.3
DGCNN (rand)	31.6 ± 9.0	40.8 ± 14.6	19.9 ± 6.5	16.9 ± 4.8
Single-modal self-supervised learning				
Point-BERT	94.6 ± 3.1	96.3 ± 2.7	91.0 ± 5.4	92.7 ± 5.1
MaskPoint	95.0 ± 3.7	97.2 ± 1.7	91.4 ± 4.0	93.4 ± 3.5
Point-MAE	96.3 ± 2.5	97.8 ± 1.8	92.6 ± 4.1	95.0 ± 3.0
Point-M2AE	96.8 ± 1.8	98.3 ± 1.4	92.3 ± 4.5	95.0 ± 3.0
PointMamba	96.9 ± 2.0	$\mathbf{99.0} \pm 1.1$	93.0 ± 4.4	$\mathbf{95.6} \pm 3.2$
PointGPT-S	96.8 ± 2.0	98.6 ± 1.1	92.6 ± 4.6	95.2 ± 3.4
SI-Mamba	96.4 ± 2.7	98.5 ± 1.5	92.0 ± 5.1	95.1 ± 3.6
MaskClu	96.5 ± 1.6	98.3 ± 1.1	92.9 ± 3.3	95.2 ± 2.3
PaRE (ours)	$\mathbf{97.0} \pm 2.0$	98.9 ± 0.5	$\mathbf{93.6} \pm 3.6$	$\mathbf{95.6} \pm 2.6$
Cross-modal self-supervised learning				
ACT	96.8 ± 2.3	98.0 ± 1.4	93.3 ± 4.0	$\mathbf{95.6} \pm 2.8$
I2P-MAE	$\mathbf{97.0} \pm 1.8$	98.3 ± 1.3	92.6 ± 5.0	95.5 ± 3.0

(b) Standard classification

Method	OA
Supervised learning	
PointNet	89.2
DGCNN	92.9
Single-modal self-supervised learning	
Point-BERT	92.7
MaskPoint	92.6
Point-MAE	93.2
Point-M2AE	93.4
PointMamba	93.6
PointGPT-S	93.3
Point-MaDi	**93.8**
ZigzagPointMamba	93.2
PaRE (ours)	**93.8**
Cross-modal self-supervised learning	
ACT	93.2
I2P-MAE	93.7

Table 3: Dense prediction on ShapeNetPart and S3DIS Area 5 (%). Best results are bold.

(a) ShapeNetPart

Method	Cls. mIoU	Inst. mIoU
PointNet	80.4	83.7
PointNet++	81.9	85.1
PointCNN	84.6	**86.1**
Point-MAE	84.2	**86.1**
PaRE (ours)	**84.9**	**86.1**

(b) S3DIS Area 5

Method	mAcc	mIoU
PointNet	49.0	41.1
PointNet++	–	53.5
PointCNN	63.9	57.3
Point-MAE	69.9	60.8
PaRE (ours)	**71.5**	**60.9**

During pre-training, we sample 1,024 points per shape and form 64 patches with 32 neighbors each. The 12-layer bidirectional Transformer encoder uses 384-dimensional features and six heads. The masking ratio is 0.6; all positional variants share the masked latent prediction objective. ModelNet40 classification uses 1,024 points without voting. Few-shot accuracy is reported as mean ± standard deviation over 10 few-shot splits. The appendix provides implementation and optimization details.

4.2 Downstream transfer

ScanObjectNN. With point-only pre-training, PaRE achieves 95.70%, 93.80%, and 90.77% OA on OBJ_BG, OBJ_ONLY, and PB_T50_RS (Table 1). It achieves the highest OA among the listed methods in three settings. Compared with the strongest baseline in each setting, PaRE improves OA by 0.18, 0.17, and 0.24 percentage points over Point-MaDi, PointSD, and AsymDSD-S, respectively.

ModelNet40. With 1,024 points and without voting, PaRE achieves 93.8% OA, matching Point-MaDi and exceeding Point-MAE by 0.6 percentage points (Table 2(b)). For few-shot classification (Table 2(a)), its accuracies are 97.0%, 98.9%, 93.6%, and 95.6% for 5-way 10-shot, 5-way 20-shot, 10-way 10-shot, and 10-way 20-shot, respectively. Among the listed methods, PaRE achieves the best or joint-best mean in three settings and exceeds Point-MAE by 0.6-1.1 points across all four.

Table 4: Positional ablations on ScanObjectNN (OA, %). Best reported results are in bold.

(a) Local geometry

Variant	Setting	Coeff.	OBJ_BG	OBJ_ONLY	PB_T50_RS
No RoPE	None	0	94.84	93.12	90.42
Anchor-only	None	0	94.32	92.77	90.01
Fixed offset	Fixed 0.25	0	94.66	**93.80**	89.94
Centroid rotary	Fixed 1.0	0	93.80	93.46	89.45
Shuffled offset	Shuffled	32	94.49	93.29	89.87
PaRE (ours)	Matched	32	**95.70**	**93.80**	**90.77**

(b) Positional parameterization

Variant	Component	Coeff.	OBJ_BG	OBJ_ONLY	PB_T50_RS
Scalar additive	Additive PE	1	94.49	92.60	89.49
Scalar rotary	Local residual	1	94.15	92.94	90.46
Learned frequency	Full phase	32	94.66	92.77	90.56
PaRE (ours)	Local residual	32	**95.70**	**93.80**	**90.77**

Table 5: Pre-training and transfer on ScanObjectNN (OA, %). Best reported results are in bold.

ID	Pre-training	Fine-tuning	Init.	OBJ_BG	OBJ_ONLY	PB_T50_RS
C0	Anchor-only	Anchor-only	–	94.32	92.77	90.01
C1	Anchor-only	PaRE	Init 0.25	94.15	92.77	89.76
C2	Fixed offset	PaRE	Init 0.25	94.15	92.77	89.35
C3	PaRE	PaRE	Reset 0.25	94.32	93.63	89.76
C4	PaRE	**PaRE**	Carry	**95.70**	**93.80**	**90.77**

Dense prediction. On ShapeNetPart, PaRE achieves 84.9% Cls. mIoU and 86.1% Inst. mIoU (Table 3), improving over Point-MAE by 0.7 percentage points in Cls. mIoU while matching its Inst. mIoU. On S3DIS Area-5, PaRE obtains 71.5% mAcc and 60.9% mIoU, exceeding Point-MAE by 1.6 and 0.1 percentage points, respectively. These results support transfer of the learned representation to both object-level part segmentation and scene-level semantic segmentation.

4.3 Positional ablations

Patch-aware geometry. Table 4(a) compares positional variants that share the same masked latent prediction objective during pre-training and retain their positional configurations during fine-tuning. Anchor-only omits the patch residual; No RoPE removes query/key rotations but retains additive anchor embeddings. Fixed offset and Centroid rotary use fixed residual coefficients of 0.25 and 1, respectively. PaRE exceeds Anchor-only by 1.38, 1.03, and 0.76 percentage points on OBJ_BG, OBJ_ONLY, and PB_T50_RS, respectively. Shuffling offsets among patches within each object (visible patches only during pre-training) reduces OA by 1.21, 0.51, and 0.90 points, supporting patch-offset correspondence. PaRE also outperforms No RoPE and Centroid rotary on all three settings, and Fixed offset except for a tie on OBJ_ONLY. Anchor-only also underperforms No RoPE on all three settings, showing that anchor-based rotations alone do not improve performance in this setup.

Positional parameterization. Table 4(b) compares scalar additive and rotary modulation, full-phase scaling, and PaRE's bounded, frequency-wise local residual. Coeff. counts additional learnable positional coefficients relative to Anchor-only. PaRE outperforms both scalar variants on all three settings. Both PaRE and Learned frequency add 32 learnable positional coefficients, but PaRE achieves 1.04, 1.03, and 0.21 points higher OA on OBJ_BG, OBJ_ONLY, and PB_T50_RS, respectively. These results favor the tested local-residual parameterization over full-phase scaling.

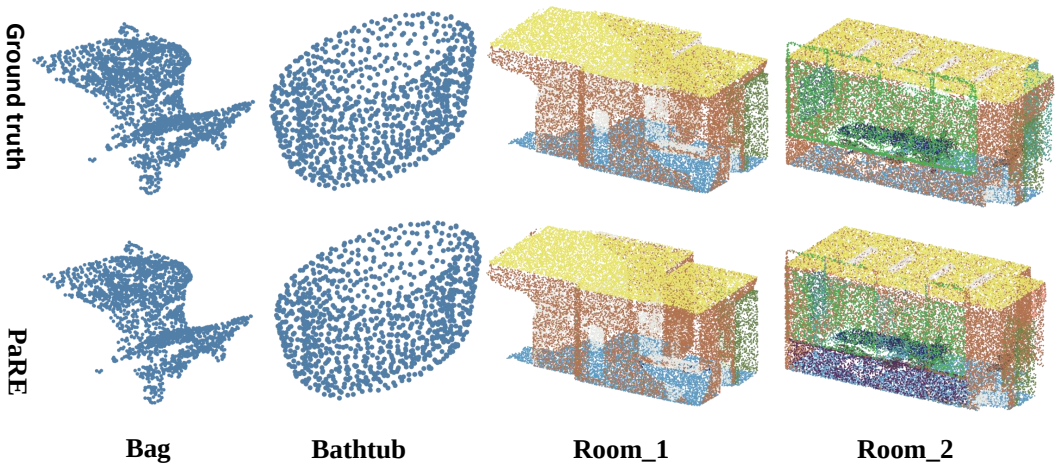


Figure 3: Qualitative examples on ScanObjectNN, ModelNet40, and S3DIS.

4.4 Pre-training and coefficient transfer

Table 5 examines pre-training encodings and coefficient transfer. C1-C4 share the PaRE fine-tuning architecture with trainable residual coefficients, whose effective values are initialized or reset to 0.25 in C1-C3. C3 and C4 share a PaRE pre-training checkpoint, but C4 retains its learned coefficients.

Introducing PaRE only during fine-tuning (C1) does not improve OA over Anchor-only (C0) on any setting. Under the common coefficient initialization, C2 matches C1 on OBJ_BG and OBJ_ONLY but is 0.41 points lower on PB_T50_RS. C3 exceeds C1 by 0.17 and 0.86 points on the first two settings, respectively, while matching it on PB_T50_RS. Retaining the coefficients (C4) improves OA over resetting them (C3) by 1.38, 0.17, and 1.01 points on OBJ_BG, OBJ_ONLY, and PB_T50_RS, respectively, supporting retention of the learned coefficients when transferring the backbone.

4.5 Qualitative visualization

Figure 3 presents selected object examples from the classification benchmarks and qualitative results for scene semantic segmentation. The first two columns show a bag from ScanObjectNN's PB_T50_RS setting and a bathtub from ModelNet40, respectively, with their category labels below. The last two columns compare ground-truth semantic labels with PaRE predictions for two S3DIS scenes. The predictions agree with the annotations over several large semantic regions.

5 Conclusion

We presented Patch-Aware Rotary Encoding (PaRE), which incorporates local patch geometry into rotary interactions while preserving the original sampled anchors. By adding centroid-offset phase residuals controlled by bounded, frequency-wise coefficients, PaRE combines anchor displacement with differences in local offsets in the relative phase. Rotary phases can therefore reflect neighborhood differences even when normalized anchor coordinates remain unchanged. This design makes local geometry explicit in the positional operator, complementing token content through measured patch offsets and input-independent coefficients shared across tokens, attention heads, and encoder layers. Under the same masked latent prediction objective, PaRE improves over Anchor-only on all three ScanObjectNN settings and supports transfer to standard and few-shot classification, part segmentation, and semantic segmentation. Controlled ablations support geometrically matched offsets, local-residual parameterization, and transferring learned coefficients with the backbone. These findings highlight the value of local patch statistics for learning transferable point cloud representations through patch-aware rotary interactions. Since centroid offsets capture only first-order information, exploring richer local geometric statistics is a natural direction for future work.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools solely to aid and polish the writing of text prepared by the authors, including improving grammar, phrasing, clarity, and readability. Generative AI tools were not used to generate research ideas, design the methodology or experiments, formulate mathematical claims, implement the proposed method, generate synthetic data, interpret experimental results, or draft substantive sections of the paper. All AI-assisted edits were reviewed and, where necessary, revised by the authors. The authors take full responsibility for the final content of this work, including its text, claims, and conclusions.

REFERENCES

Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1534–1543, 2016.

Ali Bahri, Moslem Yazdanpanah, Mehrdad Noori, Sahar Dastani, Milad Cheraghalikhani, Gustavo Adolfo Vargas Hakim, David Osowiechi, Farzad Beizaee, Ismail Ben Ayed, and Christian Desrosiers. Spectral informed mamba for robust point cloud processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11799–11809, 2025.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Guangyan Chen, Meiling Wang, Yi Yang, Kai Yu, Li Yuan, and Yufeng Yue. PointGPT: Auto-regressively generative pre-training from point clouds. In *Advances in Neural Information Processing Systems*, volume 36, pp. 29667–29679, 2023.

Yiyang Chen, Shanshan Zhao, Lunhao Duan, Changxing Ding, and Dacheng Tao. Harnessing text-to-image diffusion models for point cloud self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 26156–26166, 2025.

Linshuang Diao, Sensen Song, Yurong Qian, and Dayong Ren. ZigzagPointMamba: Spatial-semantic mamba for point cloud understanding. In *Advances in Neural Information Processing Systems*, volume 38, pp. 139255–139274, 2025.

Runpei Dong, Zekun Qi, Linfeng Zhang, Junbo Zhang, Jianjian Sun, Zheng Ge, Li Yi, and Kaisheng Ma. Autoencoders as cross-modal teachers: Can pretrained 2d image transformers help 3d representation learning? In *International Conference on Learning Representations*, 2023.

Ben Fei, Yixuan Li, Weidong Yang, Lipeng Ma, and Ying He. Toward a unified representation of multi-modal pre-training for 3-D processing. *IEEE Transactions on Visualization and Computer Graphics*, 32(2):2216–2229, 2026.

Ziyu Guo, Renrui Zhang, Longtian Qiu, Xianzhi Li, and Pheng-Ann Heng. Joint-MAE: 2d-3d joint masked autoencoders for 3d point cloud pre-training. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 791–799, 2023.

Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *European Conference on Computer Vision*, pp. 289–305. Springer, 2024.

Xianglong Jin, Zheng Wang, Rong Wang, and Feiping Nie. Reliable-view 2d-3d key-part aligned transformer with reinforced masking for 3d point cloud understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 5530–5538, 2026.

Remco Leijenaar and Hamidreza Kasaei. Asymmetric dual self-distillation for 3d self-supervised representation learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 148230–148261, 2025.

Abiao Li, Chenlei Lv, Yuming Fang, Yifan Zuo, Jian Zhang, and Guofeng Mei. PointGAC: Geometric-aware codebook for masked point cloud modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 24989–24998, 2025.

Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. PointCNN: Convolution on x-transformed points. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Dingkang Liang, Xin Zhou, Wei Xu, Xingkui Zhu, Zhikang Zou, Xiaoqing Ye, Xiao Tan, and Xiang Bai. PointMamba: A simple state space model for point cloud analysis. In *Advances in Neural Information Processing Systems*, volume 37, pp. 32653–32677, 2024.

Haotian Liu, Mu Cai, and Yong Jae Lee. Masked discrimination for self-supervised learning on point clouds. In *European Conference on Computer Vision*, pp. 657–675. Springer, 2022.

Sajad Movahedi, Timur Carstensen, Arshia Afzal, Frank Hutter, Antonio Orvieto, and Volkan Cevher. Selective rotary position embedding. In *International Conference on Learning Representations*, volume 2026, pp. 10219–10247, 2026.

Yatian Pang, Wenxiao Wang, Francis E. H. Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *European Conference on Computer Vision*, pp. 604–621. Springer, 2022.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660, 2017a.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, volume 30, 2017b.

Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. PointNeXt: Revisiting pointnet++ with improved training and scaling strategies. In *Advances in Neural Information Processing Systems*, volume 35, pp. 23192–23204, 2022.

Chang Qiu, Feipeng Da, and Zilei Zhang. Feature extraction and representation of pre-training point cloud based on diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 26559–26568, 2025.

Kanglin Qu, Pan Gao, Qun Dai, and Yuanhao Sun. Point-focused attention meets context-scan state space: Robust biological visual perception for point cloud representation. In *International Conference on Learning Representations*, volume 2026, pp. 155663–155689, 2026.

Bin Ren, Xiaoshui Huang, Mengyuan Liu, Hong Liu, Fabio Poiesi, Nicu Sebe, and Guofeng Mei. Masked clustering prediction for unsupervised point cloud pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 8712–8720, 2026.

Ayumu Saito, Prachi Kudeshia, and Jiju Poovvancheri. Point-JEPA: A joint embedding predictive architecture for self-supervised learning on point cloud. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 7348–7357, 2025.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Kunming Su, Qiuxia Wu, Panpan Cai, Xiaogang Zhu, Xuequan Lu, Zhiyong Wang, and Kun Hu. RI-MAE: Rotation-invariant masked autoencoders for self-supervised point cloud representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7015–7023, 2025.

Xiaoyu Tian, Haoxi Ran, Yue Wang, and Hang Zhao. GeoMAE: Masked geometric target prediction for self-supervised point cloud pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13570–13580, 2023.

Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1588–1597, 2019.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics*, 38(5):1–12, 2019.

Ziyi Wang, Xumin Yu, Yongming Rao, Jie Zhou, and Jiwen Lu. Take-a-photo: 3d-to-2d generative pre-training of point cloud models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5640–5650, 2023.

Ziyi Wang, Yanran Zhang, Jie Zhou, and Jiwen Lu. UniPre3D: Unified pre-training of 3d point cloud models with cross-modal gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1319–1329, 2025.

Peixi Wu, Bosong Chai, Menghua Zheng, Wei Li, Zhangchi Hu, Jie Chen, Zheyu Zhang, Hebei Li, and Xiaoyan Sun. Efficient spiking point mamba for point cloud analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 26393–26403, 2025.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1912–1920, 2015.

Xiaoyang Xiao, Runzhao Yao, Zhiqiang Tian, and Shaoyi Du. Point-MaDi: Masked autoencoding with diffusion for point cloud pre-training. In *Advances in Neural Information Processing Systems*, volume 38, pp. 50302–50330, 2025.

Siming Yan, Yuqi Yang, Yuxiao Guo, Hao Pan, Peng-Shuai Wang, Xin Tong, Yang Liu, and Qixing Huang. 3d feature prediction for masked-autoencoder-based point cloud pretraining. In *International Conference on Learning Representations*, 2024.

Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics*, 35(6):1–12, 2016.

Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-BERT: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19313–19322, 2022.

Yuanwen Yue, Damien Robert, Jianyuan Wang, Sunghwan Hong, Jan Dirk Wegner, Christian Rupprecht, and Konrad Schindler. LitePT: Lighter yet stronger point transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24173–24184, 2026.

Yaohua Zha, Huizhen Ji, Jinmin Li, Rongsheng Li, Tao Dai, Bin Chen, Zhi Wang, and Shu-Tao Xia. Towards compact 3d representations via point feature enhancement masked autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 6962–6970, 2024.

Yixin Zha, Chuxin Wang, Wenfei Yang, and Tianzhu Zhang. Exploring semantic masked autoencoder for self-supervised point cloud understanding. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 2341–2349, 2025.

Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, and Hongsheng Li. Point-M2AE: multi-scale masked autoencoders for hierarchical point cloud pre-training. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27061–27074, 2022.

Renrui Zhang, Liuhui Wang, Yu Qiao, Peng Gao, and Hongsheng Li. Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21769–21780, 2023.

Tao Zhang, Haobo Yuan, Lu Qi, Jiangning Zhang, Qianyu Zhou, Shunping Ji, Shuicheng Yan, and Xiangtai Li. Point cloud mamba: point cloud learning via state space model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 10121–10130, 2025a.

Xiangdong Zhang, Shaofeng Zhang, and Junchi Yan. Pcp-mae: Learning to predict centers for point masked autoencoders. In *Advances in Neural Information Processing Systems*, volume 37, pp. 80303–80327, 2024.

Xiangdong Zhang, Shaofeng Zhang, and Junchi Yan. Towards more diverse and challenging pre-training for point cloud learning: Self-supervised cross reconstruction with decoupled views. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 28696–28706, 2025b.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16259–16268, 2021.

Xiao Zheng, Xiaoshui Huang, Guofeng Mei, Yuenan Hou, Zhaoyang Lyu, Bo Dai, Wanli Ouyang, and Yongshun Gong. Point cloud pre-training with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22935–22945, 2024.

Zhiming Zhou, Yong He, Qiaoyun Wu, Chaoxu Mu, and Ajmal Mian. 3dsmt: A hybrid spiking mamba-transformer for point cloud analysis. In *International Conference on Learning Representations*, volume 2026, pp. 23202–23222, 2026.