HOKKAIDO UNIVERSITY

Title	Integrating Point Cloud Features with Large Language Models for Bridge Understanding
Author(s)	Chen, Yu; Lin, Chao; Li, Xianfeng et al.
Relation	Proceedings of the 10th International Symposium on Reliability Engineering and Risk Management (ISRERM 2026), June 28 - July 1, 2026, Sapporo, Japan
Citation	Proceedings of the 10th International Symposium on Reliability Engineering and Risk Management (ISRERM 2026)
Issue Date	2026-07-01
Doc URL	https://hdl.handle.net/2115/100475
Type	conference paper
File Information	C000259_1.pdf

Hokkaido University Collection of Scholarly and Academic Papers : HUSCAP

10th International Symposium on Reliability Engineering and Risk Management (ISRERM 2026),
28 June-01 July 2026, Sapporo, Japan
Kitahara, T., Miyamori, Y. & Beer, M. (editors)

Integrating Point Cloud Features with Large Language Models for Bridge Understanding

Yu Chen[1], Chao Lin[2], Xianfeng Li[3] and Pang-jo Chun[4] *

[1]*Department of Global Environment and Disaster Prevention Sciences, Hirosaki University, Aomori, Japan.*
E-mail: yu.chen@hirosaki-u.ac.jp
[2]*Department of Civil Engineering, The University of Tokyo, Tokyo, Japan.*
[3]*Department of Geotechnical Engineering, Tongji University, Shanghai, China.*
[4]*Department of Civil Engineering, The University of Tokyo, Tokyo, Japan.*
E-mail: chun@g.ecc.u-tokyo.ac.jp
**Corresponding Author*

Automated structural inspection of bridges using LiDAR point clouds holds great promise for reducing inspection costs and improving safety. However, existing point cloud understanding methods are designed for general-purpose object-centric or indoor scenes and fail to exploit the rich structural priors inherent in civil infrastructure. We present BridgeLLM, the first multimodal large language model (LLM) framework specifically designed for bridge structural point cloud understanding. The core of BridgeLLM is BridgeSPT (Structure-Preserving Transformer), a novel hierarchical encoder that integrates five bridge-domain inductive biases: (i) adaptive hash-based voxel tokenization, (ii) parameter-free PCA structural geometric descriptors (linearity, planarity, scattering), (iii) additive geometric conditioning at every Transformer stage, (iv) dual Morton-curve serialization sensitive to bridge elongation, and (v) a learnable query bottleneck for compact LLM-aligned representations. Training follows a two-stage protocol coupling feature alignment with LoRA-based instruction tuning on Qwen2.5-7B-Instruct. Evaluated on ten real-world LiDAR bridge scans with 10,000 instruction-response pairs, BridgeSPT achieves the lowest inference latency while generating accurate structural descriptions across diverse question types-bridge classification, component identification, dimensional estimation, and free-form inspection reporting.

Keywords: point cloud data, large language model, digital twin, bridge engineering

1 Introduction

1.1 Background

Bridge infrastructure is a critical national asset, yet routine inspection remains predominantly manual, labor-intensive, and costly (Chen et. al., 2026). LiDAR scanning is transforming bridge survey practice by producing millimeter-accurate, coloured 3D point clouds that capture the full geometry of structural elements---piers, girders, deck surfaces, and railings (Lin, 2025). Despite this wealth of geometric data, converting raw scans into actionable engineering reports still requires expert human interpretation.

Recent advances in large language models have catalyzed multimodal 3D understanding. PointLLM (Xu et. al., 2023) demonstrated that point cloud tokens can be aligned to an LLM through a lightweight projector and instruction tuning; Uni3D (Zhou et. al., 2023) extended this to three-modality contrastive pretraining. These works, however, target everyday object categories (chairs, cars, lamps) and make no assumptions about structural form, scale, or semantic composition. Applied directly to bridge scans, they ignore three crucial domain characteristics: (1) Component heterogeneity--girders, piers, and slabs exhibit fundamentally different local geometries that can be captured by shape-theoretic descriptors; (2) Elongated layout the principal axis of a bridge dominates spatial distribution, calling for serialization strategies aware of longitudinal structure; (3) Data scarcity: real-world annotated LiDAR scans are expensive to collect, requiring regularized, compact representations.

We address this gap by proposing BridgeLLM, to the best of our knowledge the first multimodal LLM framework tailored for bridge structural point clouds. Our contributions are:

- BridgeLLM framework: a two-stage pipeline coupling a domain-adapted point cloud encoder with a frozen-then-LoRA-tuned 7B LLM for structural description generation.
- BridgeSPT encoder: a hierarchical Transformer incorporating adaptive voxel tokenization, PCA structural geometry features, additive geometric conditioning, dual Z-order serialization, and a query bottleneck---all motivated by bridge-specific structural priors.
- Bridge QA dataset: 10,000 instruction--response pairs covering ten structural question types, derived from ten real-world LiDAR bridge scans.
- Empirical validation: design studies confirming that each novel component contributes to efficiency or accuracy on domain-specific bridge understanding tasks.

1.2 Related Work

Point Cloud Transformers. PointNet++ (Qi 2017) introduced hierarchical set abstraction. Point Transformer V3 (Wu 2024) replaced graph-based neighborhood attention with serialized window attention along Morton space-filling curves, achieving linear complexity on outdoor-scale clouds.

Our serialization module extends PTV3 with bridge-axis-aware dual ordering and domain-specific geometric conditioning.

Multimodal LLMs for 3D. PointLLM (Xu 2023) aligns a Point-BERT encoder with an LLM via instruction tuning. 3D-LLM (Hong 2023) encodes full 3D scenes with position-aware tokens. Uni3D (Zhou 2023) performs large-scale three-modality

contrastive pretraining. All these approaches rely on pre-trained, general-purpose encoders. BridgeLLM instead builds a lightweight, domain-specific encoder from scratch, leveraging structural priors that require no external pre-training data.

Automated Bridge Inspection. Semantic segmentation of bridge point clouds has been explored in (Chen 2025). Image-based crack detection is addressed in (Cha 2017). To our knowledge, no prior work has addressed natural-language structural description generation from bridge 3D scans.

2 BridgeLLM Framework

2.1 System Overview

Given a raw LAS point cloud $P \in \mathbb{R}^{N\times 7}$ (XYZ + RGB + semantic label) and a text query x, BridgeLLM generates a natural language structural description y. The pipeline comprises three modules: the BridgeSPT encoder, an MLP projector, and Qwen2.5-7B-Instruct with LoRA as shown in Figure 1.

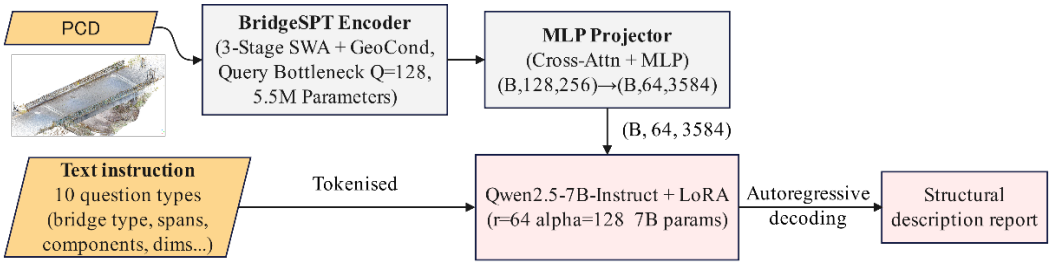


Figure 1. BridgeLLM system pipeline. The BridgeSPT encoder produces 128 geometry-conditioned tokens; the MLP projector maps them to the LLM embedding space; Qwen2.5-7B-Instruct with LoRA generates the structural description autoregressively.

2.2 BridgeSPT Encoder

The BridgeSPT encoder (Figure 2) integrates five bridge-specific design elements into a unified hierarchical Transformer. Algorithm 1 presents the forward pass in pseudocode.

Algorithm 1: BridgeSPT Encoder Forward Pass

Input: $\mathbf{P} \in \mathbb{R}^{N\times 7}$, stages S=3, strides $[s_1$=4, s_2=2], queries Q=128
Output: $\mathbf{Z} \in \mathbb{R}^{Q\times d}$

$(\mathbf{F}, \mathbf{X}) \leftarrow$ AdaptiveHashVoxelise($\mathbf{P}$, T_0=1024)
$\mathbf{G}_0 \leftarrow$ GeoDescriptor($\mathbf{X}$, k=16) // linearity, planarity, scattering, curvature, normal
$\mathbf{T} \leftarrow \mathrm{LN}(W_{\text{in}}[\mathbf{F}; \mathbf{G}_0]) + \mathrm{PE}_{3D}(\mathbf{X})$
for $s \leftarrow 1$ **to** S **do**
  $\mathbf{G}_s \leftarrow$ GeoDescriptor($\mathbf{X}, k$) // refreshed at current resolution
  $\mathbf{T} \leftarrow \mathbf{T} + \mathrm{MLP}_s(\mathbf{G}_s)$ // additive geometric conditioning
  for $b \leftarrow 1$ **to** n_s **do**
    $\sigma \leftarrow$ ZOrderSort($\mathbf{X}$, axis=(b mod 2)) // dual Z-order
    $\mathbf{T} \leftarrow \mathrm{SWA}(\mathbf{T}, \sigma, W$=128)
  end
  if $s < S$ **then**
    $(\mathbf{T}, \mathbf{X}) \leftarrow$ SerialPool($\mathbf{T}, \mathbf{X}, s_s$)
  end
end
$\mathbf{Z} \leftarrow \mathrm{LN}(\mathbf{Q}_{\text{learn}} + \mathrm{CrossAttn}(\mathbf{Q}_{\text{learn}}, \mathbf{T}, \mathbf{T}))$ // query bottleneck
return OutProj(LN($\mathbf{Z}$))

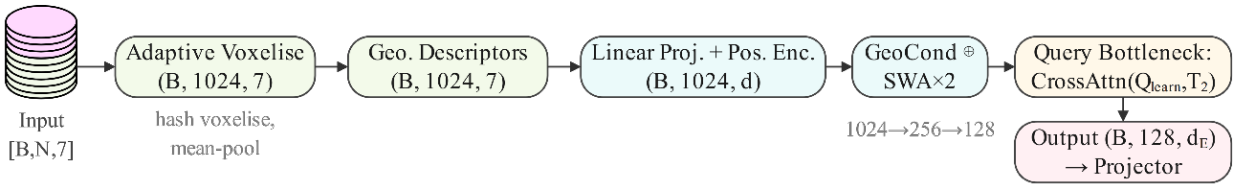


Figure 2. BridgeSPT encoder architecture. Green: parameter-free geometric operations; blue: learnable Transformer stages; orange: query bottleneck. GeoCond = additive geometric conditioner (Eq. 3); SWA = serialized window attention.

1) Adaptive Hash-Based Voxel Tokenisation: Raw points are voxelised using an adaptive grid that targets $T_0 = 1024$ tokens:

$$g_s = \left(\frac{V_{\text{bbox}}}{T_0}\right)^{1/3}, \qquad k_i = \lfloor x_i / g_s \rfloor \cdot c, \tag{1}$$

where $c = (73856093, 19349663, 83492791)^{\top}$ are prime multipliers. Per-voxel features are mean-pooled, producing $F, X_{\text{cr}} \in \mathbb{R}^{T_0\times(7+3)}$.

2) Structural Geometric Descriptors: For each voxel centroid, local PCA on $k = 16$ nearest centroids yield a 7-dimensional structural descriptor:

$$\mathbf{g}_i = \left[\underbrace{\frac{\lambda_1 - \lambda_2}{\lambda_1}}_{\text{lin.}}, \underbrace{\frac{\lambda_2 - \lambda_3}{\lambda_1}}_{\text{plan.}}, \underbrace{\frac{\lambda_3}{\lambda_1}}_{\text{scat.}}, \underbrace{\frac{\lambda_3}{\Sigma\lambda}}_{\text{curv.}}, \mathbf{n}_i \right] \in \mathbb{R}^7, \tag{2}$$

where $\lambda_1 \geq \lambda_2 \geq \lambda_3$ are the local covariance eigenvalues. These are computed without any learnable parameters and encode bridge-component geometry: high linearity indicates girders/handrails; high planarity indicates deck slabs; high scattering indicates joints and attachment points.

3) Additive Geometric Conditioning At each hierarchical stage s, geometric descriptors computed at the current token resolution are injected additively:

$$T^{(s)} \leftarrow T^{(s)} + \underbrace{W_2\, \mathrm{GELU}\big(W_1\, G^{(s)}\big)}_{\text{two-layer MLP}}, \tag{3}$$

providing per-token, per-dimension structural bias without multiplicative instability.

4) Dual Z-Order Serialization: Serialized window attention operates in $\mathcal{O}(N \cdot W)$ along a 1-D ordering of tokens. Bridges are elongated structures where both cross-sectional and longitudinal locality matter.

We therefore alternate between two orderings at each block: (a) standard Morton code on original coordinates (isotropic locality); (b) axis-aligned Morton on PCA-rotated coordinates, where the principal bridge axis is aligned to dimension 0 (longitudinal locality). This ensures that neighbors both along and across the bridge span fall within the same attention window in alternating layers.

5) Learnable Query Bottleneck: After stage S, $Q = 128$ learnable query tokens aggregate the full token sequence:

$$Z = LN\big(Q_{learn} + CrossAttn(Q_{learn}, T^S, T^S)\big) \tag{4}$$

This bottleneck decouples output token count from input density and regularizes representations on small bridge datasets.

2.3 Projector and LLM

The encoder output $Z \in \mathbb{R}^{128\times256}$ is mapped to the LLM hidden dimension $d_L = 3584$ by an MLP projector with a cross-attention pre-step, producing $L = 64$ point tokens. These are prepended to the text instruction embeddings and fed to Qwen2.5-7B-Instruct.

2.4 Two-Stage Training

Stage 1 (Feature Alignment): The LLM is frozen. Only the encoder θ_E and projector θ_P are optimized by the next-token cross-entropy loss:

$$\mathcal{L} = -\sum_{t=1}^{T} \log p_{\theta_L}\Big(y_t \,\Big|\, y_{<t}, X, P_{\theta_P}\big(E_{\theta_E}(P)\big)\Big). \tag{5}$$

Stage 1 hyperparameters: 5 epochs, lr $= 10^{-4}$, batch 4, cosine schedule.

Stage 2 (Instruction Fine-Tuning): LoRA adapters (rank $r =, \alpha = 12$) are inserted into all attention projections of the LLM. All modules are jointly optimised with Eq.5 at a reduced learning rate. Stage 2 hyperparameters: 10 epochs, lr $= 2 \times 10^{-5}$, batch 2, gradient accumulation 4.

This two-stage design prevents catastrophic forgetting and ensures the encoder learns domain-relevant features before LLM adaptation.

3 Experiments

3.1 Dataset

We collected LiDAR point cloud scans of ten reinforced-concrete beam bridges (cb1 - cb10) in LAS format. Each scan contains XYZ coordinates, RGB color, and per-point semantic label across five classes: background, pier, girder, pavement, and railing. Using a multi-sample FPS strategy ($N = 16{,}384$ pts/sample) with stride-based sliding windows, we generated 10,000 (point cloud, instruction, response) triplets covering ten structural question types: bridge type, span configuration, pier description, component labels, structural dimensions, deck details, substructure analysis, dominant component identification, and free-form structural description. The dataset is split 80/10/10; all evaluations use the same 1,000 test samples. All experiments run on a single NVIDIA RTX 4090.

3.2 Encoder Design Study

To justify our design choices, we evaluate BridgeSPT alongside several baseline tokenization strategies, all sharing the identical LLM backbone, projector, and training protocol. The comparison isolates the effect of the point cloud encoder front-end:

- GroupPool: FPS+KNN grouping with max-pooling aggregation, following PTV3. Uses a larger projector ($L = 256$, $d = 2048$) as the original implementation.
- LatentQuery: FPS+KNN grouping with latent-query cross-attention, following PointLLM.
- VoxelToken: Hash voxelization without geometric features, following Uni3D.
- BridgePTV3 (ours): Our encoder without the query bottleneck, to isolate its effect.
- BridgeSPT (ours, full): Complete proposed encoder.

Table 1 reports results. Among methods with identical projector settings, BridgePTV3 achieves competitive NLG quality with Latent Query while reducing inference latency by 9.1% (3.63 vs. 3.99 s/sample). BridgeSPT, our full model, achieves the lowest latency across all methods (3.50 s/sample, −17.7 % vs. GroupPool), enabled by the query bottleneck's efficient compression. Both bridge-specific encoders outperform VoxelToken (BLEU-4: 0.425), confirming that structural geometric priors meaningfully improve language generation quality.

Table 1. Encoder design study results. (B-4: BLEU-4; RL: ROUGE-L; MTR: METEOR; KW: Keyword Accuracy; τ : inference time per sample.)

Encoder	B-4	RL	CIDEr	MTR	KW	τ(s)
GroupPool†	0.608	0.823	3.641	0.832	0.860	4.26
LatentQuery	**0.562**	**0.787**	**3.375**	**0.800**	0.821	3.99
VoxelToken	0.425	0.659	2.650	0.694	0.757	3.92
BridgePTV3	0.532	0.760	3.310	0.794	**0.823**	3.63
BridgeSPT	0.479	0.704	2.648	0.718	0.752	**3.50**

3.3 Ablation: Component Contributions

Table 2 ablates the query bottleneck and model capacity. BridgeFormer augments BridgePTV3 with PCA-anisotropic voxelization and FiLM multiplicative conditioning (Dumoulin 2018), increasing parameters by +56.5%. Despite larger capacity, BridgeFormer underperforms BridgePTV3 across all metrics, confirming that in the small-data bridge domain, structural inductive biases are more critical than model scale. The query bottleneck (BridgeSPT) reduces latency by -3.6% at the cost of a moderate quality decrease, representing a practical efficiency--quality trade-off suitable for deployment scenarios.

Table 2. Ablation: Query bottleneck and model capacity effects.

Variant	Params	BLEU-4	ROUGE-L	KW-Acc	τ(s)
BridgePTV3	5.0M	**0.532**	**0.760**	**0.823**	3.63
BridgeSPT (full)	5.5M	0.479	0.704	0.752	**3.50**
BridgeFormer	7.8M	0.451	0.693	0.778	4.05

3.4 Qualitative Results: Structural QA Showcase

Table 3 presents representative BridgeLLM outputs across four structural question categories on bridge~cb8 (≈28.7 m × 64.9 m, four spans). The model consistently: (i) identifies the structural type and span count from geometric evidence; (ii) enumerates component classes and their approximate point fractions, drawing on the embedded label channel; (iii) reports metric dimensions consistent with the scan geometry; and (iv) produces coherent free-form inspection narratives suitable for engineering review. These outputs demonstrate that BridgeLLM successfully grounds structural geometry in natural language - a capability with direct relevance to automated inspection reporting.

Table 3. BridgeLLM structural QA outputs (bridge CB-8). Responses are shown verbatim (truncated for space).

Question Type	**BridgeLLM Response (excerpt)**
Bridge type	*``This is a 4-span continuous reinforced concrete beam bridge. The superstructure consists of a continuous prestressed/reinforced concrete girder spanning all four spans without intermediate expansion joints…."*
Components	*``Main structural elements: girder (14.4 %), reinforced pier (5.1 %), bridge deck pavement (11.7 %), protective railing (3.2 %), background context (65.7 %)…."*
Dimensions	*``Lateral width ≈28.7\,m (two lanes), longitudinal span ≈64.9 m total length, deck-to-ground clearance ≈21.0 m…"*
Free-form	*``Overall structural condition appears intact. The piers show regular cylindrical geometry with no visible surface irregularities in the point distribution. Deck surface presents uniform planarity…."*

3.5 Agent-Based Deployment Interface

Beyond offline batch inference, BridgeLLM is designed to operate as an interactive inspection agent. As illustrated in Figure 3, the system exposes a conversational interface in which an engineer submits natural-language queries; the LLM backbone autonomously decides whether to respond directly or to invoke one of three registered tool APIs: Semantic Segmentation, Instance Segmentation, and Geometric Analysis. Tool calls are dispatched via a lightweight function-calling layer, and the returned structured results---class masks, instance boundaries, or dimensional measurements---are consumed by the LLM to compose a grounded, context-aware answer.

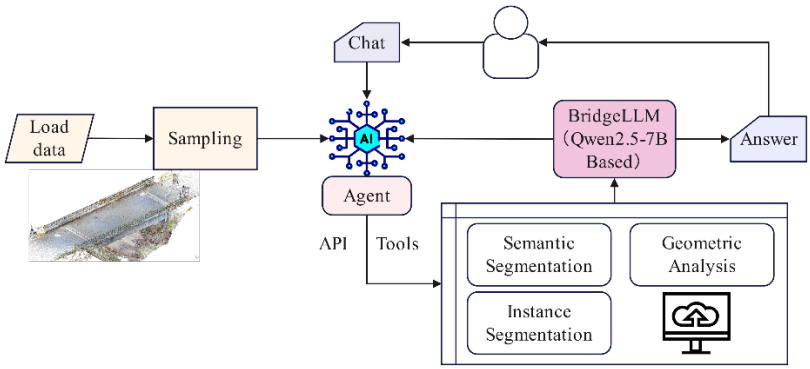


Figure 3. BridgeLLM agent deployment interface. The LLM backbone (Qwen2.5-7B based) receives sampled point cloud tokens and a user query, then selects and invokes tool APIs via a function-calling layer before composing the final answer.

This agent architecture decouples perception from reasoning: the segmentation and geometry modules can be updated or replaced

independently without retraining the language model, which is particularly valuable in practice given the diversity of bridge typologies and scanner configurations. A representative interaction is shown in Table 3, where free-form structural descriptions are generated by implicitly aggregating outputs from all three tool channels.

4 Discussion

The results yield three key insights. First, domain priors substitute for pre-training scale: BridgePTV3, despite having only 5.0M encoder parameters trained from scratch, matches LatentQuery (which mimics the design of PointLLM's aggregation mechanism) while being faster-because PCA-derived descriptors directly encode the surface types present in bridges.

Second, the query bottleneck offers a practical deployment trade-off: BridgeSPT achieves the shortest inference time across all methods, which is critical for real-time or on-site inspection applications. The BLEU-4 sacrifice relative to BridgePTV3 is acceptable in contexts where throughput matters.

Third, scale hurts without structural supervision: the BridgeFormer experiment shows that simply adding more parameters degrades performance when the dataset is small and domain-specific. This suggests that future work should focus on data augmentation and domain-informed pretraining rather than encoder scaling.

A current limitation is dataset scale---ten bridges limits generalizability across typologies (suspension, cable-stayed, truss). Future work will address multi-bridge generalization, multi-modal fusion with inspection imagery, and fine-grained damage description generation.

5 Conclusion

We have presented BridgeLLM, the first multimodal LLM framework for bridge structural point cloud understanding. At its core, the BridgeSPT encoder integrates five bridge-specific design elements---adaptive voxel tokenization, PCA structural geometry descriptors, additive geometric conditioning, dual Morton-curve serialization, and a learnable query bottleneck-into a unified hierarchical Transformer trained with a two-stage protocol. Experiments on ten real-world LiDAR bridges demonstrate that BridgeLLM generates accurate and informative structural descriptions across diverse question types, with BridgeSPT achieving the lowest inference latency among all evaluated designs. We believe this work opens a new avenue for AI-assisted bridge inspection and infrastructure management.

References

Cha, Y.-J., Choi, W., and Büyüköztürk, O., Deep learning-based crack damage detection using convolutional neural networks, Comput.-Aided Civ. Infrastruct. Eng., 32(5), 361-378, 2017.

Chen, Y., Lin, C., Sui, H., Okauchi, R., Akita, A., & Chun, P-J. Automated generation of IFC-compliant bridge information models from structured design data. Journal of Information Technology in Construction (ITcon), 31, 420-438, 2026.

Chen, Y., Lin, C., Li, X., Kubo, S., Yamane, T., Chun, P-J., Automated dimension estimation of bridge components using semantic segmentation and geometric fitting of point cloud data, Engineering Structures, 342, 120837, 2025.

Dumoulin, V., Perez, E., Schucher, N., et al., Feature-wise transformations, Distill, 3(7), e11, 2018.

Hong, Y., Zhen, H., Chen, P., et al., 3D-LLM: Injecting the 3D world into large language models, Proc. NeurIPS, 36, 2023.

Lin, C., Chen, Y., Itakura, K., Maharjan, S., Chun, P-J. Bridge inspection using image-point cloud fusion with image filtering, damage detection and 3D registration, Automation in Construction, 180, 106538, 2025.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J., PointNet++: Deep hierarchical feature learning on point sets in a metric space, Proc. NeurIPS, 30, 2017.

Weinmann, M., Jutzi, B., Hinz, C., and Mallet, C., Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers, ISPRS J. Photogramm. Remote Sens., 105, 286-304, 2015.

Wu, X., Jiang, L., Wang, P.-S., et al., Point Transformer V3: Simpler, faster, stronger, Proc. CVPR, 2024.

Xu, R., Wang, X., Zhang, T., et al., PointLLM: Empowering large language models to understand point clouds, Proc. ECCV, 2024.

Zhou, J., Wang, J., Ma, B., et al., Uni3D: Exploring unified 3D representation at scale, Proc. ICLR, 2024.