Point Cloud Quantization Through Multimodal Prompting for 3D Understanding

Hongxuan Li[1], **Wencheng Zhu**[1,2*], **Huiying Xu**[3], **Xinzhong Zhu**[3], **Pengfei Zhu**[1]

[1]College of Intelligence and Computing, Tianjin University
[2]Haihe Laboratory of Information Technology Application Innovation
[3]School of Computer Science and Technology, Zhejiang Normal University
{lihongxuan, wenchengzhu, zhupengfei}@tju.edu.cn, {xhy, zxz}@zjnu.edu.cn

Abstract

Vector quantization has emerged as a powerful tool in large-scale multimodal models, unifying heterogeneous representations through discrete token encoding. However, its effectiveness hinges on robust codebook design. Current prototype-based approaches relying on trainable vectors or clustered centroids fall short in representativeness and interpretability, even as multimodal alignment demonstrates its promise in vision-language models. To address these limitations, we propose a simple multimodal prompting-driven quantization framework for point cloud analysis. Our methodology is built upon two core insights: 1) Text embeddings from pre-trained models inherently encode visual semantics through many-to-one contrastive alignment, naturally serving as robust prototype priors; and 2) Multimodal prompts enable adaptive refinement of these prototypes, effectively mitigating vision-language semantic gaps. The framework introduces a dual-constrained quantization space, enforced by compactness and separation regularization, which seamlessly integrates visual and prototype features, resulting in hybrid representations that jointly encode geometric and semantic information. Furthermore, we employ Gumbel-Softmax relaxation to achieve differentiable discretization while maintaining quantization sparsity. Extensive experiments on the ModelNet40 and ScanObjectNN datasets clearly demonstrate the superior effectiveness of the proposed method.

Code — https://github.com/li-hongxuan/PCQ

Introduction

Human cognition organizes concepts not by rigid definitions, but through prototypes, where their semantic meaning emerges from graded similarity to exemplary instances (Hampton 2006). Linguistic studies characterize prototypes by *vagueness* (ambiguous boundaries), *typicality* (graded membership), *genericity* (class-wide applicability), and *opacity* (non-transparent categorization) (Geeraerts 2006). Interestingly, textual descriptions inherently exhibit these prototype characteristics. Language shapes concepts through hierarchical abstraction while maintaining fuzzy boundaries (Xu et al. 2025). This inherent semantic isomorphism raises a critical question for multimodal learning: *Given that text embeddings possess an intrinsically prototype-like structure, can they serve as a bridge between visual perception and conceptual understanding?*

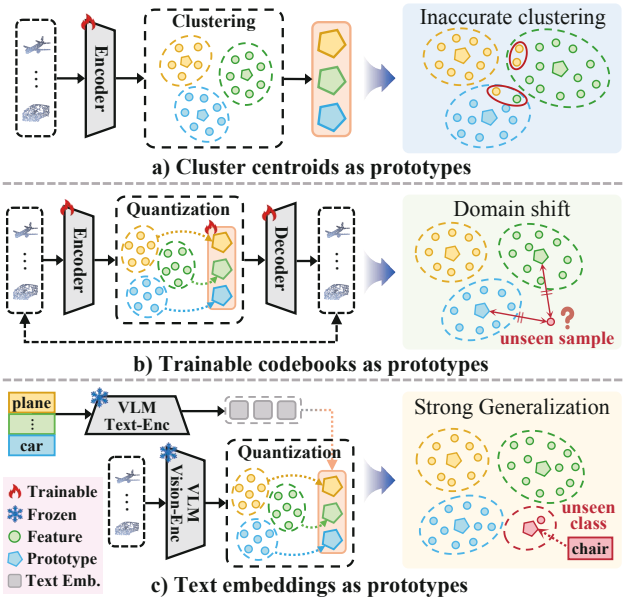

Figure 1: a) Cluster centroids as prototypes and b) Trainable codebooks as prototypes suffer from inaccurate clustering and domain shift, which reduces their representativeness and generalization. c) Our method leverages a pre-trained vision-language model to derive text-driven semantic prototypes, refined during fine-tuning to enhance representativeness, interpretability, and generalization for 3D understanding.

Recent advancements in large multimodal models, such as CLIP (Radford et al. 2021), ULIP (Xue et al. 2023), and Uni3D (Zhou et al. 2024) have revolutionized 2D/3D understanding (Cho, Kim, and Kim 2023) by aligning heterogeneous modalities into a unified embedding space through contrastive learning (Li et al. 2022; Zeng et al. 2023). These models achieve notable performance across various tasks, from image classification (Chen et al. 2023b) to point cloud understanding (Zhang, Luo, and Lei 2024). Despite such progress, a fundamental misalignment is inherent: while visual encoders excel in feature extraction (Ning et al. 2024;

*Corresponding author

Guo et al. 2023; Srivastava and Sharma 2024), text encoders primarily focus on modeling high-level semantic hierarchies (Gan et al. 2022; Palanisamy et al. 2024). This limited alignment capability of multimodal models often results in degraded performance in downstream tasks.

To address semantic gaps in cross-modal understanding, recent research has turned to prototype learning (Wei et al. 2023; Yang et al. 2022). Existing methods fall into cluster-based and codebook-based categories. Cluster-based approaches employ cluster centroids derived from training data as representative prototypes (Li et al. 2021). While intuitive, such centroids are constrained by data distribution and initialization, failing to capture intra-class diversity and thus limiting representativeness and generalization (Shu et al. 2022). In contrast, codebook-based methods learn a set of prototypes by optimizing task-specific objectives (Van Den Oord, Vinyals et al. 2017). Although flexible, these methods often suffer from domain shift and unstable convergence, while offering limited interpretability. Crucially, neither category effectively leverages cross-modal knowledge to enable robust and generalizable prototype learning.

In this paper, we introduce a method to effectively utilize text-guided prototypes derived from pre-trained vision-language models. Our approach stems from two key observations: 1) Vision-language models achieve point cloud-text alignment through many-to-one contrastive learning, where diverse 3D instances of the same class are aligned with a single text embedding, e.g., "a 3D shape of a chair". This directly mirrors prototype characteristics of vagueness (tolerance for intra-class variance) and genericity (class-wide applicability). 2) Text embeddings inherently possess prototype-like properties such as typicality (graded similarity to class exemplars) and opacity (implicit categorization), making them well-suited as semantic prototypes for visual representation learning. By reformulating text embeddings as trainable prototypes, our framework enables dynamic integration of visual features with language-derived conceptual knowledge, effectively bridging the vision-language domain gap. To further enhance vision-language alignment, we refine these prototypes using learnable multimodal prompts, optimized under compactness and separation constraints. We then quantize point cloud features into discrete prototypes via Gumbel-Softmax to generate text-like features. Finally, a cross-modal feature fusion module combines fine-grained visual details with high-level semantic concepts.

Our main contributions are summarized as follows:

- We propose a text-driven 3D quantization framework that unifies vision-language alignment through point cloud feature discretization.
- We reformulate text features as trainable prototypes and model their quantization via Gumbel distribution for end-to-end gradient estimation, while preserving discrete semantics for cross-modal generalization.
- Extensive experiments on the ModelNet40 and ScanObjectNN datasets demonstrate superior performance compared to state-of-the-art methods while maintaining competitive parameter efficiency.

Related Work

3D Multi-Modality Models. Extending large multimodal models to point cloud analysis presents significant challenges (Qian et al. 2022; Zhou et al. 2023). Recent approaches can be broadly categorized into two paradigms: projection-based (Ma et al. 2022) and unified representation learning methods (Huang et al. 2024a; Qi et al. 2024). Projection-based approaches, such as PointCLIP (Zhang et al. 2022b) and PointCLIP V2 (Zhu et al. 2023), transform 3D point clouds into multi-view 2D images and leverage pre-trained vision-language models. CLIP2Point (Huang et al. 2023) extends this pipeline by fusing depth and RGB features across views. Duoduo CLIP (Lee, Zhang, and Chang 2025) incorporates multi-view projections with spatial context. However, these approaches inherently suffer from information loss during the 2D projection process (Hegde, Valanarasu, and Patel 2023; Hess et al. 2024). In contrast, ULIP (Xue et al. 2023, 2024), Openshape (Liu et al. 2023a), and Uni3D (Zhou et al. 2024) achieve cross-modal alignment by training 3D encoders on triplets of point clouds, images, and text. PPT (Sun et al. 2024) employs parameter-efficient tuning of pre-trained 3D encoders. Despite effectiveness, these methods face a semantic gap that visual encoders prioritize object perception (Huang et al. 2024b; Zhang, Dong, and Ma 2023), while text encoders capture abstract concepts (Xu, Zhu, and Clifton 2023). Recent work explores prototype-based methods. ProtoCLIP (Chen et al. 2023a) learns visual prototypes through contrastive language guidance, while Partslip (Liu et al. 2023b) aligns part-level 3D features with textual descriptions.

Vector Quantization. Vector quantization is a prominent topic in representation learning. Existing VQ methods can be categorized into deterministic quantization and stochastic quantization. Deterministic quantization (Zhan et al. 2022), such as VQ-VAE (Van Den Oord, Vinyals et al. 2017), selects codebook entries via argmax/min operations and has seen extensions like multiscale codebooks in VQ-VAE2 (Razavi, Van den Oord, and Vinyals 2019), improved training stability in DVAE (Vahdat, Andriyash, and Macready 2018), and integration of adversarial losses in VQ-GAN (Esser, Rombach, and Ommer 2021). VIM (Yu et al. 2021) optimizes codebook learning efficiency and reconstruction fidelity. Stochastic quantization samples tokens from learned probability distributions (Maddison, Tarlow, and Minka 2014), thus requiring gradient estimation techniques for non-differentiable problems. VQ-Wave2Vec (Baevski, Schneider, and Auli 2019) employs Gumbel-Softmax reparameterization (Jang, Gu, and Poole 2016) for differentiable sampling. DALL-E (Ramesh et al. 2021) leverages stochastic quantization for text-to-image generation. Theoretical grounding for soft assignment has also been provided (Roy et al. 2018). However, deterministic approaches often suffer from codebook collapse and limited semantic expressiveness, while stochastic variants can struggle with training instability (Zhang et al. 2024). Crucially, neither category learns codebooks that sufficiently represent the rich semantics of language. To address these limitations, our framework redefines prototypes through text-guided quantization.

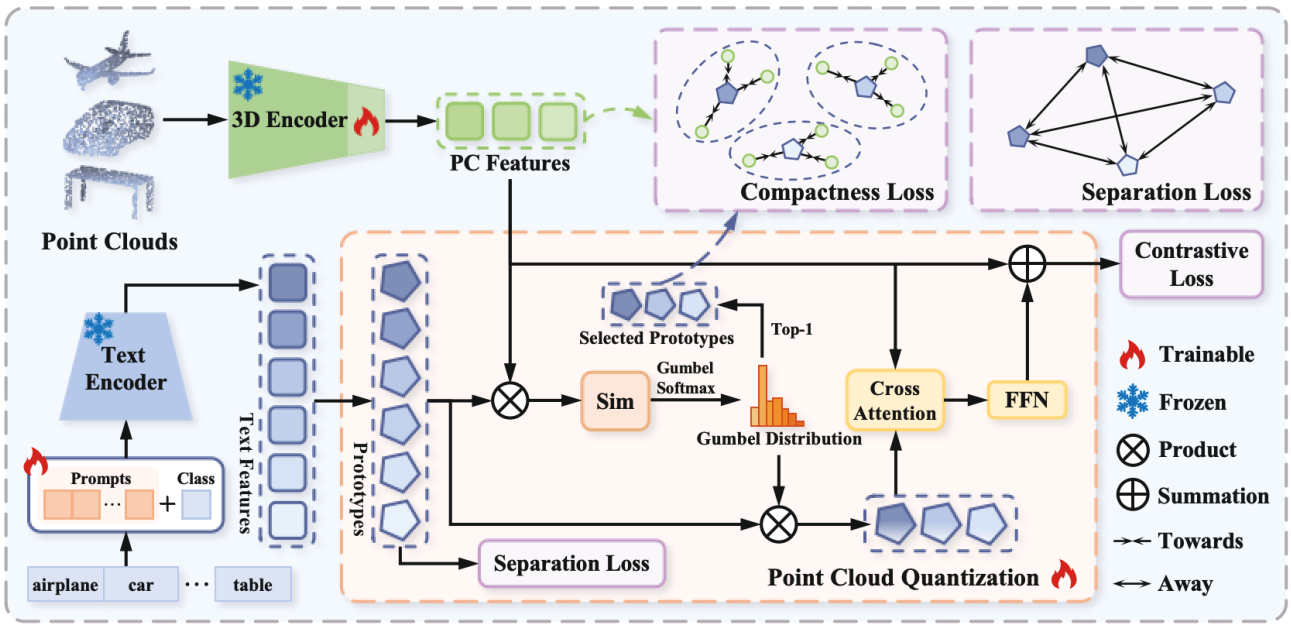

Figure 2: Framework of the proposed approach. Our method comprises feature extraction and point cloud quantization modules. The feature extraction module uses ULIP-2 text encoder and 3D point cloud encoder to extract text and point cloud features. The quantization module then takes these text features as prototypes and quantizes point cloud features into prototype features. To enable differentiable sampling, discrete features are modeled through a Gumbel distribution, and Gumbel-Softmax reparameterization is adopted to represent point cloud features with prototype features. Finally, point cloud features are combined with prototype features via cross-modal feature fusion to produce the final hybrid representation. Notably, parameter-efficient fine-tuning is employed to optimize both prototype and point cloud features, constrained by compactness and separation losses.

Methodology

As illustrated in Figure 2, our framework introduces a dual-stream architecture that connects visual perception and semantic abstraction through a text-guided prototype learning paradigm. The key innovation lies in redefining text embeddings from pre-trained vision-language models as trainable visual prototypes, leveraging the semantic hierarchy learned through contrastive alignment pretraining.

Our Point Cloud Quantization (PCQ) framework converts continuous features into discrete semantic tokens. In our experiments, we adopt ULIP-2 as the pre-trained multimodal backbone because it achieves comparable performance to Uni3D-Ti while using fewer parameters, making it suitable for memory-limited scenarios. To enable adaptive prompt tuning, we introduce learnable prompts optimized with two complementary regularizers: compactness loss to encourage intra-prototype coherence and separation loss to promote inter-prototype diversity. Together, these objectives drive the prototypes toward representations that reduce modality misalignment. Next, we take the textual embeddings as visual prototypes and quantize the visual features into discrete tokens via Gumbel-Softmax reparameterization. Finally, we fuse original point-cloud features with their corresponding prototypes to produce the downstream task representation.

Preliminaries

ULIP-2 constructs a large-scale dataset of aligned text-image-point cloud triplets derived from Objaverse (Deitke et al. 2023) to facilitate joint pre-training of a unified representation. Each triplet $U_i = (I_i, T_i, P_i)$ comprises multi-view rendered images I_i of the 3D object, textual descriptions T_i of images, and sampled point cloud P_i. ULIP-2 employs three modality-specific encoders: a vision encoder $\mathcal{F}_{\mathcal{I}}(\cdot)$ and a text encoder $\mathcal{F}_{\mathcal{T}}(\cdot)$, both initialized from SLIP (Mu et al. 2022), and a point cloud encoder $\mathcal{F}_{\mathcal{P}}$ based on the trainable PointBERT (Yu et al. 2022) backbone. The weights of $\mathcal{F}_{\mathcal{I}}$ and $\mathcal{F}_{\mathcal{T}}$ are initialized and kept frozen to preserve semantic alignment, while the 3D encoder $\mathcal{F}_{\mathcal{P}}$ is trainable. The embedding U_i is formalized as,

$$\mathbf{h}_i^I, \mathbf{h}_i^T, \mathbf{h}_i^P = \mathcal{F}_{\mathcal{I}}(I_i), \mathcal{F}_{\mathcal{T}}(T_i), \mathcal{F}_{\mathcal{P}}(P_i). \tag{1}$$

During pretraining, ULIP-2 optimizes cross-modal contrastive losses between modalities m_1 and m_2 as,

$$\mathcal{L}_{m_1,m_2} = -\frac{1}{2}\sum_{(i,j)}\left[\log\frac{\exp(f(\mathbf{h}_i^{m_1}, \mathbf{h}_j^{m_2}))}{\sum_k \exp(f(\mathbf{h}_i^{m_1}, \mathbf{h}_k^{m_2}))} + \log\frac{\exp(f(\mathbf{h}_i^{m_1}, \mathbf{h}_j^{m_2}))}{\sum_k \exp(f(\mathbf{h}_k^{m_1}, \mathbf{h}_j^{m_2}))}\right], \tag{2}$$

where (i, j) represents positive pairs in each training batch, and $f(\mathbf{a}, \mathbf{b}) = \mathbf{a}^\top \mathbf{b}$ computes cosine similarity between two input features. The final loss combines all modality pairs through weighted summation,

$$\mathcal{L}_{\text{Final}} = \alpha\mathcal{L}_{(I,T)} + \beta\mathcal{L}_{(I,P)} + \gamma\mathcal{L}_{(P,T)}, \tag{3}$$

with α, β, and $\gamma \in [0, 1]$ controlling the relative importance of image-text, image-point, and point-text alignment.

Point Cloud Quantization

Adaptive Prompt Tuning. Our framework leverages a pretrained text encoder $\mathcal{F}_\mathcal{T}$ and a 3D point cloud encoder $\mathcal{F}_\mathcal{P}$ from ULIP-2 for feature extraction. To preserve semantic knowledge from large-scale pretraining while addressing vision-language semantic misalignment in downstream datasets, we freeze the text encoder and introduce m learnable prompt tokens for parameter-efficient adaptation. The textual prototype $\mathbf{h}_k^T$ for the k-th class is then computed as,

$$\mathbf{h}_k^T = \mathcal{F}_\mathcal{T}(\mathbf{T}_k), \text{ where } \mathbf{T}_k = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m, \mathbf{c}_k]. \quad (4)$$

$\mathbf{T}_k$ denotes the k-th input to the text encoder containing m learnable prompt vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m$, where $k \in [1, K]$ and K is the total number of classes. Here, $\mathbf{c}_k$ represents the text token of the k-th class name, e.g., "plane" or "car". By augmenting the fixed class embedding with learnable prompts, this design enriches the semantic representation with adaptable knowledge, facilitating more effective and flexible prototype generation.

To maintain the generalization capabilities of the pretrained 3D encoder $\mathcal{F}_\mathcal{P}$, we freeze all its layers except the final Transformer block, which is fine-tuned to adapt to task-specific features. The feature extraction is formalized as,

$$\mathbf{h}_i^P = \mathcal{F}_{\mathcal{P}'}(P_i), \ \ i \in [1, N], \quad (5)$$

where $\mathcal{F}_{\mathcal{P}'}(\cdot)$ denotes the partially fine-tuned 3D encoder, with only the final Transformer block for training. P_i is the i-th point cloud instance and N is total number of instances. This parameter-efficient fine-tuning strategy optimizes the alignment between text and point cloud features.

Prototype-Guided Differentiable Quantization. A fundamental challenge in multimodal alignment lies in the discrete-continuous gap: text encodes structured semantics through discrete and interpretable tokens, while visual features are inherently continuous, enabling generalization at the cost of inter-class ambiguity. To address this gap, we propose using text-derived embeddings as semantic prototypes and quantizing continuous visual features into this textual prototype space. This strategy enhances interpretability by grounding visual representations in human-readable semantic structures while reducing inter-class feature overlap through well-defined and discrete decision boundaries. However, hard quantization is non-differentiable, hindering end-to-end training. We address this using a differentiable Gumbel-Softmax relaxation that enables soft and probabilistic assignments during training.

For each point cloud instance P_i, we compute the pairwise cosine similarity between its feature $\mathbf{h}_i^P$ and all text-derived prototypes $\mathbf{h}_k^T$, s_{ik} represents the semantic affinity between P_i and the k-th class prototype guiding the quantization of point cloud features into prototype-aligned representations. We then model the assignment process using a Gumbel distribution and perform differentiable optimization via the Gumbel-Softmax reparameterization technique (Xu et al. 2022). Formally, the prototype assignment probability is computed as $q_{ik} = \frac{\exp(s_{ik})}{\sum_{j=1}^K \exp(s_{ij})}$, where q_{ik} represents the probability of the k-th prototype being assigned to the i-th point cloud feature. Gumbel-Softmax injects stochasticity via additive Gumbel noise as,

$$y_{ik} = \frac{\exp\left(\frac{\log q_{ik} - \log(-\log \epsilon_k)}{\tau}\right)}{\sum_{j=1}^K \exp\left(\frac{\log q_{ij} - \log(-\log \epsilon_j)}{\tau}\right)}. \quad (6)$$

ϵ_k represents noise sampled from a uniform distribution $U[0, 1]$, τ is a temperature parameter that controls the sharpness of the distribution. In our experiments, we adopt a default value of $\tau = 1$. The quantized probability weight y_{ik} approximates a one-hot prototype selection while retaining differentiability. The quantized feature $\mathbf{v}_i$ is derived as,

$$\mathbf{v}_i = \sum_{k=1}^K y_{ik} \mathbf{h}_k^T. \quad (7)$$

Cross-Modal Feature Fusion. We fuse the original point cloud feature $\mathbf{h}_i^P$ with its quantized feature $\mathbf{v}_i$ to integrate geometric details from 3D data with high-level semantic abstractions from language. The resulting hybrid feature $\mathbf{f}_i$ preserves spatial structure while being enriched with semantic guidance as $\mathbf{f}_i = \text{FFN}\left(\text{CrossAttention}\left(\mathbf{h}_i^P, \mathbf{v}_i\right)\right) + \mathbf{h}_i^P$. The *CrossAttention* layer uses $\mathbf{h}_i^P$ as queries, and $\mathbf{v}_i$ as keys/values to attend to semantically relevant prototypes. A residual connection ensures that geometric information is preserved, while selectively enhancing it with language semantics.

Objective Function

While multimodal alignment exhibits strong discriminative capabilities, it often suffers from high intra-class variance and insufficient inter-class separation in the embedding space. We introduce a dual regularization strategy that simultaneously enforces prototype-wise compactness and separation. As illustrated in Figure 2, our framework employs a three-component constraint combining contrastive, compactness, and separation losses.

Contrastive Loss. This forms the cornerstone of adaptive prompt tuning, designed to align the hybrid feature $\mathbf{f}_i$ with its corresponding text feature $\mathbf{h}_{y_i}^T$ in a shared embedding space. Formally, the contrastive loss is defined as,

$$\mathcal{L}_{\text{Align}} = -\frac{1}{N} \sum_{i=1}^N \log \frac{\exp\left(\cos(\mathbf{f}_i, \mathbf{h}_{y_i}^T)\right)}{\sum_{j=1}^K \exp\left(\cos(\mathbf{f}_i, \mathbf{h}_j^T)\right)}. \quad (8)$$

Compactness Loss. This loss aims to minimize intra-class variance by encouraging point cloud features to cluster tightly around assigned prototypes. The loss is defined as,

$$\mathcal{L}_{\text{Comp}} = \|\mathbf{H}^P - \mathbf{Q}\mathbf{H}^T\|^2, \quad (9)$$

where $\mathbf{H}^P = [\mathbf{h}_1^P; \mathbf{h}_2^P; \ldots; \mathbf{h}_N^P] \in \mathbb{R}^{N \times d}$ and $\mathbf{H}^T = [\mathbf{h}_1^T; \mathbf{h}_2^T; \ldots; \mathbf{h}_K^T] \in \mathbb{R}^{K \times d}$ denotes the matrix of point cloud features and textual prototype embeddings. $\mathbf{Q} = [\mathbf{q}_1; \mathbf{q}_2; \ldots; \mathbf{q}_N] \in \mathbb{R}^{N \times K}$ is the assignment matrix where $\mathbf{q}_i$ is a one-hot vector with $q_{ik} = 1$ if $k = \arg\max_k y_{ik}$, i.e., the prototype with the highest similarity to the i-th sample, and 0 otherwise. The compactness loss encourages point cloud features to align closely with their assigned text-guided prototypes, thereby reducing intra-class variance.

Dataset	Supervised Training					Unsupervised Pre-Training + Full Fine-Tuning								PEFT	
	PointNet (Qi et al. 2017a)	PointNet++ (Qi et al. 2017b)	PointCNN (Li et al. 2018)	DGCNN (Wang et al. 2019)	MVTN (Hamdi et al. 2021)	PointBERT (Yu et al. 2022)	MaskPoint (Liu et al. 2022)	PointMAE (Pang et al. 2022)	PointCMT (Yan et al. 2022)	PointM2AE (Zhang et al. 2022a)	ACT (Dong et al. 2022)	ULIP (Xue et al. 2023)	ULIP-2 (Xue et al. 2024)	PPT* (Sun et al. 2024)	**PCQ** (Ours)
MN40 (Wu et al. 2015)	89.2	90.7	92.2	92.9	93.5	93.2	93.8	93.8	93.5	93.4	93.7	<u>94.1</u>	–	93.6	**94.1**
OBJ (Uy et al. 2019)	79.2	84.3	85.5	86.2	92.3	88.1	89.7	88.3	92.3	88.8	91.9	–	–	<u>93.1</u>	**93.5**
BG (Uy et al. 2019)	73.3	82.3	86.1	82.8	92.6	87.4	89.3	90.0	92.6	91.2	93.3	–	–	<u>95.4</u>	**95.5**
PB (Uy et al. 2019)	68.0	77.9	78.5	78.1	82.8	83.1	84.6	85.2	86.4	86.4	88.2	86.4	**89.7**	88.9	<u>89.0</u>

Table 1: Accuracy (%) comparison on the ModelNet40 and ScanObjectNN datasets. Results marked with * represent our reproduced implementations using official codebases. **Bold** and <u>underline</u> indicate the best and second-best results, respectively.

Method	ModelNet40					ScanObjectNN				
	1-shot	2-shot	4-shot	8-shot	16-shot	1-shot	2-shot	4-shot	8-shot	16-shot
PointNet (Qi et al. 2017a)	27.4	32.3	55.1	64.8	72.3	18.8	26.2	26.6	35.0	35.2
SimpleView (Goyal et al. 2021)	27.5	36.1	58.8	68.3	78.4	23.0	23.6	29.6	32.0	37.4
CurveNet (Xiang et al. 2021)	40.0	55.5	70.0	75.0	80.4	25.4	26.4	26.6	30.0	35.2
PointNet++ (Qi et al. 2017b)	40.1	55.5	72.0	78.2	80.2	27.1	32.3	41.0	47.5	54.8
PointCLIP (Zhang et al. 2022b)	52.1	67.5	75.6	80.5	85.4	30.0	42.1	46.7	50.0	54.9
PointCLIP V2 (Zhu et al. 2023)	<u>60.5</u>	71.2	76.9	80.5	85.4	34.0	43.2	49.1	52.2	54.9
PPT (Sun et al. 2024)	59.9	<u>73.8</u>	<u>81.0</u>	<u>86.1</u>	<u>89.1</u>	<u>35.2</u>	<u>49.4</u>	<u>57.7</u>	<u>65.2</u>	<u>73.9</u>
PCQ (Ours)	**61.1**	**76.3**	**81.5**	**87.5**	**90.8**	**41.3**	**52.7**	**59.7**	**71.0**	**76.5**
Δ	+0.6	+2.5	+0.5	+1.4	+1.7	+6.1	+3.3	+2.0	+5.8	+2.6

Table 2: Few-shot accuracy (%) comparison on ModelNet40 and ScanObjectNN (PB). **Bold** and <u>underline</u> indicate the best and second-best results, respectively. Δ indicates the absolute improvement of our method over the second-best result.

Method	10%	20%	100%
PointNet (Qi et al. 2017a)	72.7	73.5	80.4
PointNet++ (Qi et al. 2017b)	74.8	76.8	81.9
PointCNN (Li et al. 2018)	60.4	64.1	84.6
PointBERT (Yu et al. 2022)	76.4	79.6	84.1
PPT (Sun et al. 2024)	80.8	84.0	86.4
PCQ (Ours)	**82.6**	**84.9**	**86.6**

Table 3: Mean class-wise IoU ($\mathbf{mIoU}_C$) for part segmentation on ShapeNetPart under different training data ratios.

Separation Loss. To maximize prototype discriminability, we enforce uniform geometric spacing among class prototypes in the embedding space. This distributional regularization encourages prototypes to spread out evenly on the hypersphere, reducing inter-class confusion and improving generalization to unseen samples. The constraint is formulated as minimizing the Kullback-Leibler divergence,

$$\mathcal{D}_{\mathrm{KL}}\left(P_{\text{proto}} \parallel \mathcal{U}\right) = \frac{1}{K}\sum_{k=1}^{K}\sum_{j\neq k} p_{kj}\log\frac{p_{kj}}{u_{kj}}, \tag{10}$$

where P_{proto} denotes the probability distribution over prototypes and $\mathcal{U}$ is the uniform distribution. Each element is defined as,

$$p_{ij} = \frac{\exp\left(-\|\mathbf{h}_i^T - \mathbf{h}_j^T\|^2\right)}{\sum_{k\neq i}\exp\left(-\|\mathbf{h}_i^T - \mathbf{h}_k^T\|^2\right)}. \tag{11}$$

Minimizing $\mathcal{D}_{\mathrm{KL}}$ drives the learned prototype distribution p_{ij} toward uniformity, encouraging approximately equal pairwise distances. When pairwise distances are balanced, the total inter-prototype dispersion is maximized, leading to enhanced class separability. We thus derive the separation loss as,

$$\mathcal{L}_{\text{Sep}} = \sum_{i\neq j}\exp\left(-\|\mathbf{h}_i^T - \mathbf{h}_j^T\|^2\right). \tag{12}$$

We introduce hyperparameters λ_1 and λ_2 to balance three complementary components of our objective function. The overall loss is formulated as,

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{Align}} + \lambda_1\mathcal{L}_{\text{Comp}} + \lambda_2\mathcal{L}_{\text{Sep}}. \tag{13}$$

The multi-loss framework has three objectives: the contrastive loss reduces intra-class variance by pulling together positive pairs while pushing apart negative samples across classes. The compactness loss enforces tight clustering of point cloud features, strengthening local feature aggregation, and the separation loss maximizes prototype distances.

$\mathcal{L}_A$	$\mathcal{L}_S$	$\mathcal{L}_C$	Acc (%)
✔			69.95
✔	✔		69.19
✔		✔	70.01
✔	✔	✔	**71.03**

(a) Ablation on loss function components.

Network Component	Acc (%)
w/o PC adapter	56.73
w/o Learnable prompt	67.66
w/o PC quantization	67.59
Ours	**71.03**

(b) Ablation on framework components. *w/o* stands for without.

Prompt Design	Acc (%)
{class}	67.66
"A 3D shape of a " + {class}	69.19
Learnable prompt + {class}	**71.03**

(c) Comparison of prompt design strategies. {class} indicates the class name.

Prototype Strategy	Acc (%)
Cluster centroids	69.60
Trainable codebooks	70.06
Text embeddings	**71.03**

(d) Comparison of different prototype strategies.

Table 4: Ablation study of our model on the ScanObjectNN-PB 8-shot setting. **Bold** indicates the best performance.

Method	BG	PB	MN40
PPT	83.2	70.6	18.8
PCQ	**86.9**	**72.8**	**21.5**
Δ	+3.7	+2.2	+2.7

Table 5: Cross-dataset generalization with models trained on OBJ and evaluated on BG, PB, and ModelNet40.

Method	1-shot	2-shot	4-shot	8-shot	16-shot
Uni3D (PEFT)	45.07	53.68	56.49	65.58	68.56
Uni3D (PCQ)	**50.17**	**56.42**	**57.11**	**65.93**	**71.20**
Δ	+5.10	+2.74	+0.62	+0.35	+2.64

Table 6: Few-shot accuracy (%) of PEFT and PCQ on Uni3D backbone. Δ indicates the absolute accuracy improvement.

Experiments

Main Results

Point Cloud Recognition. As shown in Table 1, our approach achieves strong performance across all benchmark datasets while maintaining high parameter efficiency. Despite using fewer trainable parameters under the Parameter-Efficient Fine-Tuning (PEFT) paradigm, our method delivers competitive results compared to both fully supervised and fully fine-tuned approaches. Notably, on the OBJ variant, our method surpasses the current state-of-the-art PEFT approach by +0.4%. For the PB variant, while our accuracy is lower than that of fully fine-tuned ULIP-2, it uses only 8.0% of ULIP-2's trainable parameters, highlighting a good trade-off between performance and model complexity.

Few-Shot Recognition. Following prior protocols (Zhu et al. 2023), we evaluate our method under 1-shot to 16-shot settings, sampling 1–16 instances per class for training and testing on the full test set. As listed in Table 2, our approach achieves state-of-the-art performance across all settings and exhibits strong gains in extreme data scarcity, e.g., +6.1% improvement on 1-shot ScanObjectNN. The performance advantage is most pronounced in low-shot scenarios, indicating strong generalization and sample efficiency under limited supervision. Moreover, consistent improvements on both synthetic and real-world datasets demonstrate the robustness and broad applicability of our framework.

Shape Part Segmentation. We evaluate 3D part segmentation performance on the ShapeNetPart dataset under varying levels of supervision. To adapt our framework for segmentation, we attach a part segmentation head to the 3D encoder. The model is trained using different proportions of the labeled training data, and we report the mean class-wise IoU ($\mathbf{mIoU}_C$) as the evaluation metric. As shown in Table 3, our method achieves the highest performance across all supervision settings, attaining $\mathbf{mIoU}_C$ scores of 82.6%, 84.9%, and 86.6% when trained with 10%, 20%, and 100% of the data, respectively. These results demonstrate not only the strong segmentation capability of our learned representations but also their effectiveness in low-data regimes.

Ablation Studies

We conduct ablations on ScanObjectNN-PB with an 8-shot setting to balance data scarcity and evaluation stability.

Loss Function Analysis. As listed in Table 4a, we evaluate the contribution of each loss component. Using only the alignment loss $\mathcal{L}_{\text{Align}}$ as the baseline achieves 69.95% accuracy. Incorporating the compactness loss $\mathcal{L}_{\text{Comp}}$ brings a marginal improvement of 0.06%, suggesting that optimizing intra-class compactness alone has a limited effect on overall discriminability. In contrast, adding the separation loss $\mathcal{L}_{\text{Sep}}$ leads to a 0.76% decrease, indicating that optimizing inter-class separation in isolation can disrupt intra-class coherence and harm generalization. When all three losses are combined, the model achieves an accuracy of 71.03%, highlighting the importance of dual regularization.

Component Analysis. As shown in Table 4b, the learnable text prompts enable adaptive refinement of textual representations, while the point cloud adapter fine-tunes the visual encoder to improve cross-modal alignment. The point cloud quantization module discretizes continuous visual features into text-guided prototypes, and we integrate these features with textual prototypes. The removal of any component results in a significant performance drop.

Textual Prompt Analysis. Table 4c presents an evaluation of different textual prompt designs, including the bare class name {class}, the handcrafted template "A 3D shape of a " + {class}, and Learnable prompts + {class}. The fixed template yields a 1.53% accuracy improvement over the baseline, demonstrating the necessity of structural prompts. Notably, learnable prompts achieve a significant performance gain of 1.84% over fixed templates, showing that prompts optimized via backpropagation possess stronger task adaptability and cross-modal alignment capabilities.

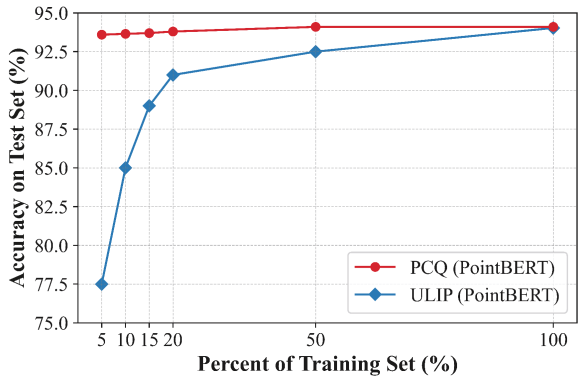

Figure 3: Data efficiency comparison. Models are trained on varying percentages of data and evaluated on the full test set.

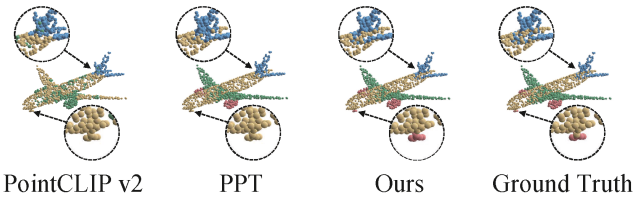

Figure 4: Part segmentation visualization on ShapeNetPart.

Prototype Strategy Analysis. As shown in Table 4d, we evaluate three prototype learning strategies: cluster centroids, trainable codebooks, and text embeddings. Using cluster centroids introduces dependency on the training data, leading to a 1.43% accuracy drop under distribution shifts. Trainable codebooks optimize prototypes through gradient descent but show unstable convergence, achieving only 70.06% accuracy. In contrast, our approach addresses these limitations, reaching 71.03% accuracy with the large-scale pretraining for multimodal alignment.

More Analyses

Data Efficiency Analysis. Large-scale pre-trained models show the potential to reduce dependence on labeled data for downstream adaptation. We evaluate the data efficiency of our parameter-efficient tuning framework through comparisons with the ULIP baseline. We conduct experiments on ModelNet40 by progressively enlarging training subsets (5%, 10%, 15%, 20%, 50%, 100%) while keeping the full test set fixed for all experiments. As illustrated in Figure 3, our approach outperforms the baseline in low-data scenarios, achieving 93.6% accuracy with merely 5% training samples.

Cross-Dataset Generalization Analysis. As shown in Table 5, we train our model on the OBJ variant of ScanObjectNN, and evaluate its performance on the BG and PB variants and the ModelNet40 dataset. Our method consistently outperforms the baseline, with accuracy gains of +3.7% on BG, +2.2% on PB, and +2.7% on ModelNet40. These results prove that our approach is robust across dataset variants and generalizes well to out-of-domain data.

Backbone Analysis. To assess the generality of our approach across different architectures, we integrate PCQ into the Uni3D-Ti (Zhou et al. 2024) backbone and evaluate it on ScanObjectNN (PB) under few-shot settings. As shown in Table 6, PCQ consistently outperforms standard PEFT with adapters across all shots, proving its architecture-agnostic nature and ability to deliver stable performance gains.

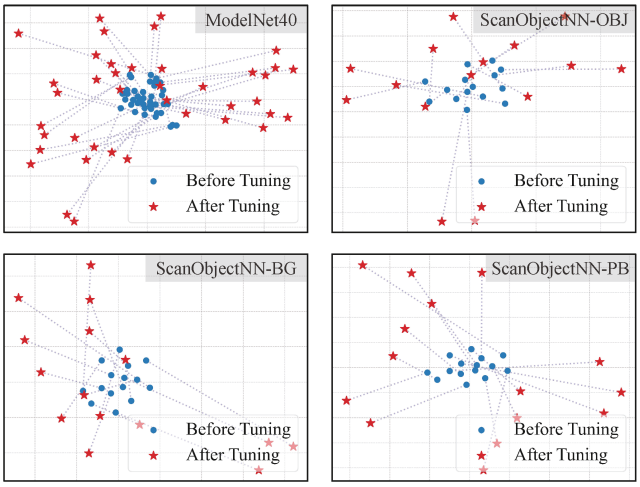

Figure 5: The t-SNE visualization before (●) and after (★) model fine-tuning across four datasets. Dashed lines connect corresponding prototypes across training phases.

Part Segmentation Visualizations. As depicted in Figure 4, our method surpasses both PointCLIP v2 and PPT in ShapeNetPart part segmentation, with particularly notable improvements in fine-grained regions such as the tail and landing gear. The predicted boundaries are sharper, evidencing a superior ability to capture fine-grained details.

Prototype Visualization. We use t-SNE to visualize prototype embeddings. Connection lines indicate prototype evolution trajectories, and distances between prototypes reflect the degree of class separation. As shown in Figure 5, prototypes are tightly clustered and poorly separated before fine-tuning. After fine-tuning, the prototypes exhibit improved inter-class separation, forming well-separated clusters. This increased separability reduces quantization mismatches.

Conclusion

In this paper, we propose a text-driven point cloud quantization framework that enhances 3D understanding by leveraging vision-language alignment. We introduce the Gumbel-Softmax relaxation to discretize continuous point cloud features into learnable textual prototypes, enabling differentiable quantization while promoting semantic interpretability. A hybrid mechanism is then designed to effectively integrate quantized prototypes with original features. We further introduce dual regularizations, in which the compactness loss minimizes intra-class variance, while the separation loss reduces inter-class overlaps. Experiments on benchmarks validate the effectiveness of the proposed approach. In the future, we plan to explore dynamic prototype generation for fine-grained part-level correspondence.

Acknowledgments

This work was supported by the National Key Research and Development Program of China under Grant 2025YFA0921700, the National Natural Science Foundation of China (No.s 62406221, 62222608, 62436002), the Tianjin Natural Science Funds for Distinguished YoungScholar (No.23JCIQIC00270), the Natural Science Foundation of Tianjin (No.25JCQNJC00770), and the 2024 Open Research Project of Intelligent Policing and National Security Risk Management Laboratory under Grant ZHKFYB2404.

References

Baevski, A.; Schneider, S.; and Auli, M. 2019. vq-wav2vec: Self-supervised learning of discrete speech representations. In *ICLR*.

Chen, D.; Wu, Z.; Liu, F.; Yang, Z.; Zheng, S.; Tan, Y.; and Zhou, E. 2023a. ProtoCLIP: Prototypical Contrastive Language Image Pretraining. *TNNLS*, 36(1): 610–624.

Chen, R.; Liu, Y.; Kong, L.; et al. 2023b. Clip2scene: Towards label-efficient 3d scene understanding by clip. In *CVPR*, 7020–7030.

Cho, E.; Kim, J.; and Kim, H. J. 2023. Distribution-aware prompt tuning for vision-language models. In *ICCV*, 22004–22013.

Deitke, M.; Schwenk, D.; Salvador, J.; et al. 2023. Objaverse: A universe of annotated 3d objects. In *CVPR*, 13142–13153.

Dong, R.; Qi, Z.; Zhang, L.; et al. 2022. Autoencoders as cross-modal teachers: Can pretrained 2d image transformers help 3d representation learning? In *ICLR*.

Esser, P.; Rombach, R.; and Ommer, B. 2021. Taming transformers for high-resolution image synthesis. In *CVPR*, 12873–12883.

Gan, Z.; Li, L.; Li, C.; Wang, L.; Liu, Z.; Gao, J.; et al. 2022. Vision-language pre-training: Basics, recent advances, and future trends. *Foundations and Trends in Computer Graphics and Vision*, 14(3–4): 163–352.

Geeraerts, D. 2006. Prototype theory. *Cognitive linguistics: Basic readings*, 34: 141–165.

Goyal, A.; Law, H.; Liu, B.; Newell, A.; and Deng, J. 2021. Revisiting point cloud shape classification with a simple and effective baseline. In *ICML*, 3809–3820.

Guo, Z.; Zhang, R.; Zhu, X.; et al. 2023. Point-bind & point-llm: Aligning point cloud with multi-modality for 3d understanding, generation, and instruction following. *arXiv preprint arXiv:2309.00615*.

Hampton, J. A. 2006. Concepts as prototypes. *Psychology of learning and motivation*, 46: 79–113.

Hegde, D.; Valanarasu, J. M. J.; and Patel, V. 2023. Clip goes 3d: Leveraging prompt tuning for language grounded 3d recognition. In *ICCV*, 2028–2038.

Hess, G.; Tonderski, A.; Petersson, C.; Åström, K.; and Svensson, L. 2024. Lidarclip or: How i learned to talk to point clouds. In *WACV*, 7438–7447.

Huang, R.; Pan, X.; Zheng, H.; Jiang, H.; Xie, Z.; Wu, C.; Song, S.; and Huang, G. 2024a. Joint representation learning for text and 3d point cloud. *PR*, 147: 110086.

Huang, T.; Dong, B.; Yang, Y.; et al. 2023. Clip2point: Transfer clip to point cloud classification with image-depth pre-training. In *ICCV*, 22157–22167.

Huang, X.; Huang, Z.; Li, S.; Qu, W.; He, T.; Hou, Y.; Zuo, Y.; and Ouyang, W. 2024b. Frozen clip transformer is an efficient point cloud encoder. In *AAAI*, 2382–2390.

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Lee, H.-H.; Zhang, Y.; and Chang, A. X. 2025. Duoduo CLIP: Efficient 3D Understanding with Multi-View Images. In *ICLR*.

Li, J.; Li, D.; Xiong, C.; and Hoi, S. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 12888–12900.

Li, J.; Zhou, P.; Xiong, C.; and Hoi, S. C. 2021. Prototypical contrastive learning of unsupervised representations. In *ICLR*.

Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; and Chen, B. 2018. Pointcnn: Convolution on x-transformed points. In *NeurIPS*, volume 31, 828–838.

Liu, M.; Shi, R.; Kuang, K.; Zhu, Y.; Li, X.; Han, S.; Cai, H.; Porikli, F.; and Su, H. 2023a. Openshape: Scaling up 3d shape representation towards open-world understanding. In *NeurIPS*, volume 36, 44860–44879.

Liu, M.; Zhu, Y.; Cai, H.; et al. 2023b. Partslip: Low-shot part segmentation for 3d point clouds via pretrained image-language models. In *CVPR*, 21736–21746.

Ma, X.; Qin, C.; You, H.; Ran, H.; and Fu, Y. 2022. Rethinking Network Design and Local Geometry in Point Cloud: A Simple Residual MLP Framework. In *ICLR*.

Maddison, C. J.; Tarlow, D.; and Minka, T. 2014. A* sampling. In *NeurIPS*, volume 27, 3086–3094.

Mu, N.; Kirillov, A.; Wagner, D.; and Xie, S. 2022. Slip: Self-supervision meets language-image pre-training. In *ECCV*, 529–544.

Ning, X.; Yu, Z.; Li, L.; Li, W.; and Tiwari, P. 2024. DILF: Differentiable rendering-based multi-view Image–Language Fusion for zero-shot 3D shape understanding. *Information Fusion*, 102: 102033.

Palanisamy, K.; Chao, Y.-W.; Du, X.; Xiang, Y.; et al. 2024. Proto-clip: Vision-language prototypical network for few-shot learning. In *IROS*, 2594–2601.

Pang, Y.; Wang, W.; Tay, F. E.; Liu, W.; Tian, Y.; and Yuan, L. 2022. Masked autoencoders for point cloud self-supervised learning. In *ECCV*, 604–621.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 652–660.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, volume 30, 5099–5108.

Qi, Z.; Fang, Y.; Sun, Z.; Wu, X.; Wu, T.; Wang, J.; Lin, D.; and Zhao, H. 2024. Gpt4point: A unified framework for point-language understanding and generation. In *CVPR*, 26417–26427.

Qian, G.; Li, Y.; Peng, H.; Mai, J.; Hammoud, H.; Elhoseiny, M.; and Ghanem, B. 2022. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. In *NeurIPS*, volume 35, 23192–23204.

Radford, A.; Kim, J. W.; Hallacy, C.; et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*, 8748–8763.

Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; and Sutskever, I. 2021. Zero-shot text-to-image generation. In *ICML*, 8821–8831.

Razavi, A.; Van den Oord, A.; and Vinyals, O. 2019. Generating diverse high-fidelity images with vq-vae-2. In *NeurIPS*, volume 32, 14837–14847.

Roy, A.; Vaswani, A.; Neelakantan, A.; and Parmar, N. 2018. Theory and experiments on vector quantized autoencoders. *arXiv preprint arXiv:1805.11063*.

Shu, M.; Nie, W.; Huang, D.-A.; Yu, Z.; Goldstein, T.; Anandkumar, A.; and Xiao, C. 2022. Test-time prompt tuning for zero-shot generalization in vision-language models. In *NeurIPS*, volume 35, 14274–14289.

Srivastava, S.; and Sharma, G. 2024. Omnivec: Learning robust representations with cross modal sharing. In *WACV*, 1236–1248.

Sun, H.; Wang, Y.; Chen, W.; Deng, H.; and Li, D. 2024. Parameter-efficient Prompt Learning for 3D Point Cloud Understanding. In *ICRA*, 9478–9486.

Uy, M. A.; Pham, Q.-H.; Hua, B.-S.; Nguyen, T.; and Yeung, S.-K. 2019. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *ICCV*, 1588–1597.

Vahdat, A.; Andriyash, E.; and Macready, W. 2018. Dvae#: Discrete variational autoencoders with relaxed boltzmann priors. In *NeurIPS*, volume 31, 1869–1878.

Van Den Oord, A.; Vinyals, O.; et al. 2017. Neural discrete representation learning. In *NeurIPS*, volume 30, 6306–6315.

Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019. Dynamic graph cnn for learning on point clouds. *TOG*, 38(5): 1–12.

Wei, Y.; Ye, J.; Huang, Z.; Zhang, J.; and Shan, H. 2023. Online prototype learning for online continual learning. In *ICCV*, 18764–18774.

Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, 1912–1920.

Xiang, T.; Zhang, C.; Song, Y.; Yu, J.; and Cai, W. 2021. Walk in the Cloud: Learning Curves for Point Clouds Shape Analysis. In *ICCV*, 895–904.

Xu, J.; De Mello, S.; Liu, S.; Byeon, W.; Breuel, T.; Kautz, J.; and Wang, X. 2022. Groupvit: Semantic segmentation emerges from text supervision. In *CVPR*, 18134–18144.

Xu, P.; Zhu, X.; and Clifton, D. A. 2023. Multimodal learning with transformers: A survey. *TPAMI*, 45(10): 12113–12132.

Xu, R.; Wang, X.; Wang, T.; Chen, Y.; Pang, J.; and Lin, D. 2025. Pointllm: Empowering large language models to understand point clouds. In *ECCV*, 131–147.

Xue, L.; Gao, M.; Xing, C.; et al. 2023. Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding. In *CVPR*, 1179–1189.

Xue, L.; Yu, N.; Zhang, S.; et al. 2024. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. In *CVPR*, 27091–27101.

Yan, X.; Zhan, H.; Zheng, C.; Gao, J.; Zhang, R.; Cui, S.; and Li, Z. 2022. Let images give you more: Point cloud cross-modal training for shape analysis. In *NeurIPS*, volume 35, 32398–32411.

Yang, T.; Wang, Y.; Lu, Y.; and Zheng, N. 2022. Visual concepts tokenization. In *NeurIPS*, volume 35, 31571–31582.

Yu, J.; Li, X.; Koh, J. Y.; et al. 2021. Vector-quantized image modeling with improved vqgan. In *ICLR*.

Yu, X.; Tang, L.; Rao, Y.; Huang, T.; Zhou, J.; and Lu, J. 2022. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *CVPR*, 19313–19322.

Zeng, Y.; Jiang, C.; Mao, J.; Han, J.; Ye, C.; Huang, Q.; Yeung, D.-Y.; Yang, Z.; Liang, X.; and Xu, H. 2023. CLIP2: Contrastive language-image-point pretraining from real-world point cloud data. In *CVPR*, 15244–15253.

Zhan, F.; Yu, Y.; Wu, R.; Zhang, J.; Cui, K.; Zhang, C.; and Lu, S. 2022. Auto-regressive image synthesis with integrated quantization. In *ECCV*, 110–127.

Zhang, J.; Dong, R.; and Ma, K. 2023. Clip-fo3d: Learning free open-world 3d scene representations from 2d dense clip. In *ICCV*, 2048–2059.

Zhang, R.; Guo, Z.; Gao, P.; Fang, R.; Zhao, B.; Wang, D.; Qiao, Y.; and Li, H. 2022a. Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. In *NeurIPS*, volume 35, 27061–27074.

Zhang, R.; Guo, Z.; Zhang, W.; et al. 2022b. Pointclip: Point cloud understanding by clip. In *CVPR*, 8552–8562.

Zhang, Y.; Luo, H.; and Lei, Y. 2024. Towards CLIP-driven Language-free 3D Visual Grounding via 2D-3D Relational Enhancement and Consistency. In *CVPR*, 13063–13072.

Zhang, Y.; Yu, K.; Wu, S.; and He, Z. 2024. Conceptual Codebook Learning for Vision-Language Models. In *ECCV*, 235–251.

Zhou, J.; Wang, J.; Ma, B.; Liu, Y.-S.; Huang, T.; and Wang, X. 2024. Uni3d: Exploring unified 3d representation at scale. In *ICLR*.

Zhou, Y.; Gu, J.; Li, X.; Liu, M.; Fang, Y.; and Su, H. 2023. PartSLIP++: Enhancing Low-Shot 3D Part Segmentation via Multi-View Instance Segmentation and Maximum Likelihood Estimation. *arXiv preprint arXiv:2312.03015*.

Zhu, X.; Zhang, R.; He, B.; et al. 2023. Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In *ICCV*, 2639–2650.