UPCDD: UNSUPERVISED POINT CLOUD DATASET DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Dataset distillation (DD) aims to condense a large real dataset into a compact synthetic set that preserves its training utility while reducing storage and repeated training costs. While recent approaches have extended DD to 3D point clouds, they predominantly rely on ground-truth semantic annotations, which are difficult and expensive to acquire in the 3D domain. This dependence prevents unlabeled point-cloud data from being directly distilled and is particularly restrictive for tasks requiring costly fine-grained annotations, such as part segmentation. To address this limitation, we propose **UPCDD** (Unsupervised Point Cloud Dataset Distillation). To the best of our knowledge, UPCDD is the **first** framework to enable label-free dataset distillation for point clouds. UPCDD leverages self-supervised representation learning and adaptive feature-space clustering to discover latent semantic partitions, effectively substituting manual labels with proxy supervision. UPCDD can be instantiated by adapting existing label-dependent 3D distillation methods, while a novel augmented objective is designed to stabilize label-free distillation, facilitating effective adaptation and transfer. Experiments on 3D shape classification and part segmentation demonstrate that UPCDD retains useful semantic information under severe compression and transfers across distillation methods, architectures, and datasets. Moreover, the distilled dataset can be reused for downstream training without repeatedly accessing the full unlabeled collection, providing a compact and architecture-flexible alternative for exploiting large unlabeled point-cloud datasets.

1 INTRODUCTION

As the demand for point-cloud data in 3D perception continues to grow, the increasing dataset scale imposes substantial storage requirements and training costs. **Dataset distillation (DD)** addresses this challenge by compressing a large dataset into a significantly smaller set that preserves its training utility while reducing storage requirements and training costs (Wang et al., 2018; Zhao et al., 2021). Recent studies have extended DD to 3D point clouds (Zhang et al., 2024; Bo & Wang, 2025; Yim et al., 2025; Zhang et al., 2026; Kim et al., 2026), progressively addressing point-specific challenges in geometry, orientation, permutation, resolution, and compact parameterization.

However, existing point-cloud distillation methods fundamentally rely on *ground-truth semantic annotations* to define the class-wise grouping, conditioning, or matching structure, preventing them from directly distilling unlabeled point-cloud data into compact training sets. This dependence becomes particularly restrictive when finer-grained supervision is required, as point-wise annotation for part segmentation is burdensome (Yi et al., 2016; Bo & Wang, 2025).

Although self-supervised learning (SSL) can learn transferable 3D representations from unlabeled point clouds through contrastive learning, completion, and masked modeling (Xie et al., 2020; Wang et al., 2021; Yu et al., 2022; Pang et al., 2022; Zhang et al., 2022), it primarily compresses knowledge at the *model level*, where information is retained in pretrained network parameters rather than in a compact standalone training set. Dataset distillation instead performs compression at the *data level*, materializing training utility in a small synthetic dataset.

A distilled dataset is a compact and reusable training asset that can be stored, transferred, and repeatedly used to train different architectures without accessing the original data (Wang et al., 2018; Bo & Wang, 2025; Yim et al., 2025). By shifting the computational burden to a one-time

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.

representation-learning and distillation phase, the resulting synthetic set makes downstream training highly efficient.

To this end, we propose **UPCDD** (*Unsupervised Point Cloud Dataset Distillation*). To the best of our knowledge, UPCDD is the **first framework for label-free dataset distillation of point clouds**. UPCDD first learns geometry-aware representations from unlabeled point clouds and discovers latent partitions with automatically selected granularity. These partitions, together with confidence estimates, act as confidence-aware proxy supervision, making existing class-conditioned 3D distillation objectives well-defined without labels. We further introduce feature alignment and dispersion regularization to stabilize distillation under noisy proxy supervision. The resulting formulation provides a general interface for adapting label-dependent 3D DD objectives to the label-free setting, which we instantiate on two structurally different paradigms: gradient-matching DD3D (Bo & Wang, 2025) and distribution-matching SADM (Yim et al., 2025).

Our main contributions are:

- We introduce, to the best of our knowledge, the **first label-free point-cloud dataset distillation framework**, enabling unlabeled point-cloud collections to be condensed into compact and reusable training sets without ground-truth annotations.
- We develop **adaptive proxy supervision and augmented distillation**, combining automatic partition-granularity selection, confidence-aware denoising, feature alignment, and dispersion regularization to construct robust proxy supervision from unlabeled point clouds and regularize distillation under noisy pseudo-partitions.
- We evaluate UPCDD on **object-level classification and part segmentation**, and further demonstrate the utility and reusability of the distilled point clouds through cross-method, cross-architecture, and cross-dataset experiments.

2 Related Works

General dataset distillation and condensation. Dataset distillation aims to *learn* a compact synthetic training set that preserves the training utility of a large dataset, allowing learning under tight storage and compute budgets (Wang et al., 2018). Early work formulates DD as a bi-level optimization problem that directly optimizes synthetic samples (Wang et al., 2018). Subsequent methods introduce efficient surrogates from different perspectives: *gradient matching* aligns optimization signals between real and synthetic data (Zhao et al., 2021; Zhao & Bilen, 2021); *trajectory matching* captures multi-step learning effects (Cazenavette et al., 2022; Du et al., 2023; Guo et al., 2024); and *feature or distribution matching* aligns feature distributions in latent spaces (Wang et al., 2022; Zhao & Bilen, 2023; Li et al., 2024; Deng et al., 2024). Other work studies complementary properties such as the diversity and realism of distilled samples (Sun et al., 2024). Most DD work is developed in 2D vision and commonly relies on labels for class-conditioned synthesis and alignment.

Dataset distillation for 3D point clouds. Point cloud distillation is complicated by unstructured point sets, resolution variability, and geometric diversity. PCC (Zhang et al., 2024) introduces point-cloud condensation through parameter matching, while DD3D (Bo & Wang, 2025) develops generative synthesis with orientation and resolution handling. SADM (Yim et al., 2025) further introduces permutation-robust feature statistics (and orientation-aware mechanisms), and CoMPR (Zhang et al., 2026)combines gradient and feature-distribution matching with a point-recycling strategy. Recent work addresses class imbalance in point-cloud distillation through adaptive class-wise synthetic budgeting and long-tail-aware distribution matching (You et al., 2026). Kim et al. (2026) further improve storage-efficient point-cloud distillation through learnable morphing of low-resolution anchor shapes and uniformity-aware matching. Despite strong performance, these 3D methods are commonly developed under supervised assumptions (or require explicit category structure) to define class-wise alignment and evaluation.

Self-supervised learning. Self-supervised learning (SSL) learns transferable 3D representations from unlabeled point clouds, e.g., contrastive pretraining (PointContrast) (Xie et al., 2020) and masked representation learning (Wang et al., 2021; Yu et al., 2022; Pang et al., 2022; Zhang et al., 2022). More broadly, clustering-based representation learning uses cluster or prototype assignments derived from learned features as label-free training signals (Caron et al., 2018; 2020).

Label-free and self-supervised dataset distillation. Prior work has explored dataset distillation from unlabeled images for efficient self-supervised pretraining. Lee et al. (2024) learn synthetic images together with continuous target representations, while Yu et al. (2025) improve this paradigm through compact parameterization and augmentation-aware representation modeling. Joshi et al. (2025) instead distill synthetic data by matching student knowledge-distillation trajectories derived from an SSL teacher. Cazenavette et al. (2025) study dataset distillation for training linear probes on pretrained self-supervised vision encoders. These methods primarily compress unlabeled images for efficient representation pretraining, rather than addressing the label-dependent structure in 3D point-cloud DD.

This gap motivates our study of how the supervision structure required by existing 3D point-cloud DD can be reconstructed from unlabeled data, forming the basis of the label-free formulation introduced next.

3 METHOD

3.1 OVERVIEW

We introduce **UPCDD** (*Unsupervised Point Cloud Dataset Distillation*), a general framework for label-free 3D dataset distillation. UPCDD reconstructs the supervision structure required by existing label-dependent methods from unlabeled point clouds. As illustrated in Figure 1, self-supervised representations are organized into adaptive latent partitions with confidence-aware weighting, providing the grouping and conditioning signals required by different 3D distillation objectives. UPCDD can therefore be instantiated by adapting existing label-dependent methods (Bo & Wang, 2025; Yim et al., 2025). The distillation is further regularized by feature alignment and dispersion to stabilize optimization under noisy proxy supervision. The resulting framework produces compact and reusable synthetic point-cloud datasets without ground-truth annotations.

3.2 PROBLEM FORMULATION: FROM SUPERVISED TO LABEL-FREE 3D DISTILLATION

Given unlabeled point clouds $\mathcal{D} = \{x_i\}_{i=1}^{N}$, with $x_i \in \mathbb{R}^{N_p \times 3}$, we aim to construct a synthetic set $\mathcal{S} = \{(\tilde{x}_{k,j}, k)\}_{k=1,j=1}^{K,m}$ of size $M = Km \ll N$ that preserves the training utility. Here, m denotes the number of synthetic point clouds per partition (CPP), j indexes the synthetic samples within a partition, and K denotes the number of latent partitions. In supervised 3D distillation, semantic labels define real-data groups, synthetic conditions, and class-wise matching targets. UPCDD reconstructs this structure through $(\hat{y}_i, w_i)$, where $\hat{y}_i \in \{1, \ldots, K\}$ is a partition index and $w_i \in [0, 1]$ is its confidence. This yields a partitioned pseudo-labeled real dataset $\hat{\mathcal{D}}_k = \{(x_i, \hat{y}_i, w_i) \mid \hat{y}_i = k\}$ and a corresponding synthetic set $\mathcal{S}_k = \{\tilde{x}_{k,j}\}_{j=1}^{m}$. With this proxy supervision, the resulting matching objective is

$$\mathcal{L}_{\text{match}} = \sum_{k=1}^{K} d_{\text{match}}\Big(\mathcal{A}_\phi(\mathcal{S}_k, k), \mathcal{A}_\phi(\hat{\mathcal{D}}_k, k; w)\Big), \tag{1}$$

where $\mathcal{A}_\phi$ extracts the method-specific distillation signals, such as gradients or feature statistics, from a distillation network f_ϕ, and d_{match} measures the discrepancy between real and synthetic signals. The partitions provide proxy supervision, while the confidence weights w modulate the contribution of real-data targets.

3.3 THE PROPOSED FRAMEWORK

To bridge the gap between self-supervised representation learning and generative dataset distillation, we propose a three-phase framework. This pipeline transforms unlabeled point clouds into a structured feature space, mines latent semantics to generate proxy supervision, and leverages pseudo-labels to drive the distillation process.

Phase 1: Self-supervised representation learning. We first train an encoder f_θ on the unlabeled collection $\mathcal{D}$ to construct a geometry-aware feature space for subsequent latent partitioning. This phase can be instantiated with different self-supervised learning methods according to the scale and characteristics of the dataset. Our implementation adopts masked point-cloud reconstruction.

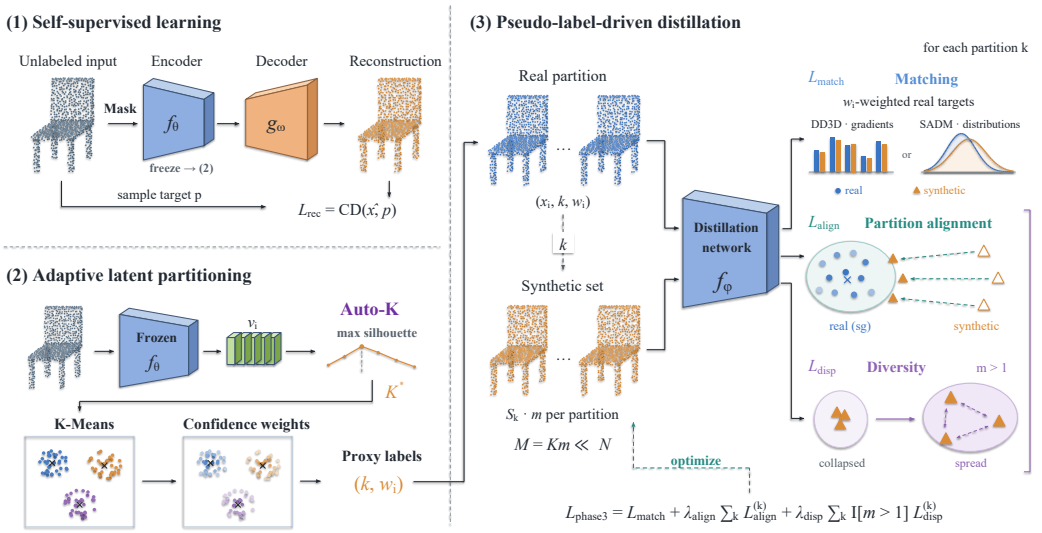


Figure 1: **Overview of the proposed UPCDD framework.** UPCDD converts an unlabeled point-cloud collection into a compact synthetic training set through three phases. **(1) Self-supervised learning:** an encoder f_θ is trained by masked point-cloud reconstruction and then frozen to provide geometry-aware representations. **(2) Adaptive latent partitioning:** frozen features are clustered by K-Means, with Auto-K selecting the partition granularity using the silhouette score; each sample receives a proxy partition index k and a confidence weight w_i to down-weight unreliable assignments. **(3) Pseudo-label-driven distillation:** these proxy partitions replace ground-truth classes as grouping and conditioning signals. For each partition, m synthetic point clouds are optimized, yielding a compact synthetic set of size $M = Km \ll N$. An augmented distillation objective (feature alignment and dispersion) further regularizes optimization under noisy proxy supervision, facilitating the adaptation of label-dependent 3D distillation methods to the label-free setting.

Specifically, a fraction r of each point cloud is retained as $x_i^{\text{vis}} = \text{Mask}(x_i; r)$, and a lightweight decoder g_ω reconstructs a target point set $p_i = \text{Sample}(x_i, T)$ from the encoded partial input. The encoder and decoder are optimized using:

$$\mathcal{L}_{\text{rec}} = \text{CD}\big(g_\omega(f_\theta(x_i^{\text{vis}})), p_i\big)\,, \tag{2}$$

where T is the target point count and CD denotes the bidirectional Chamfer distance. The encoder f_θ is frozen to extract features for Phase 2.

Phase 2: Adaptive latent semantic partitioning. We extract global features v_i with the frozen encoder for all unlabeled point clouds. To determine the partition granularity, we search over a candidate set $\mathcal{K}$ by maximizing the silhouette score. Auto-K (automatic K-selection) selects the granularity of the latent partitioning in the learned feature space rather than estimating the semantic class count. After selecting $K^\star$ by the silhouette score, each sample is assigned to its nearest K-Means centroid:

$$K^\star = \arg\max_{K' \in \mathcal{K}} \text{Sil}\big(\text{KMeans}(\{v_i\}_{i=1}^N, K')\big)\,, \qquad \hat{y}_i = \arg\min_{k \in \{1,\dots,K^\star\}} \|v_i - \mu_k\|_2^2. \tag{3}$$

where μ_k denotes the k-th K-Means centroid. To down-weight unreliable assignments, let $d_i = \|v_i - \mu_{\hat{y}_i}\|_2$ and $\bar{d} = \text{median}_i\, d_i$. We define

$$w_i = \text{clip}_{[\epsilon,1]}\left[\exp\left(-\left(\frac{d_i}{\alpha \bar{d} + \epsilon}\right)^2\right)\right], \tag{4}$$

where α controls the weighting scale and $\epsilon > 0$ provides numerical stability. This yields $\hat{\mathcal{D}} = \{(x_i, \hat{y}_i, w_i)\}_{i=1}^N$: partition indices provide proxy grouping and conditioning signals, while confidence weights reduce the contribution of unreliable real samples.

Phase 3: Pseudo-label-driven augmented distillation. The partition assignments obtained in Phase 2 provide the grouping and conditioning signals required by label-dependent 3D distillation,

while the confidence weights modulate the contribution of real samples when constructing the method-specific matching targets. Using the proxy supervision, we optimize the synthetic set $\mathcal{S} = \{(\tilde{x}_{k,j}, k)\}_{k=1,j=1}^{K,\ m}$ with

$$\mathcal{L}_{\text{phase3}} = \mathcal{L}_{\text{match}} + \lambda_{\text{align}} \sum_{k=1}^{K} \mathcal{L}_{\text{align}}^{(k)} + \lambda_{\text{disp}} \sum_{k=1}^{K} \mathbb{I}[m > 1] \mathcal{L}_{\text{disp}}^{(k)}. \tag{5}$$

The matching term matches partition-wise training signals such as gradients, feature statistics, or other method-specific surrogates, while the additional terms regularize partition-consistent feature alignment and encourage feature diversity.

Let $u_\phi(x) \in \mathbb{R}^D$ denote the global feature extracted by the surrogate network f_ϕ, and define $\bar{u}_\phi(x) = u_\phi(x)/\max(\|u_\phi(x)\|_2, \epsilon)$. For a real minibatch of point clouds $\mathcal{B}_k$ sampled from the k-th real partition, the feature alignment term is

$$\mathcal{L}_{\text{align}}^{(k)} = \sum_{j=1}^{m} \left[1 - \frac{1}{|\mathcal{B}_k|} \sum_{x \in \mathcal{B}_k} \langle \bar{u}_\phi(\tilde{x}_{k,j}), \text{sg}(\bar{u}_\phi(x)) \rangle \right], \tag{6}$$

where sg stops gradients through real features. This term aligns each synthetic cloud with real features from its assigned partition. When $m > 1$, we encourage intra-partition diversity through a dispersion regularizer:

$$\mathcal{L}_{\text{disp}}^{(k)} = \log \left[\frac{1}{m^2} \sum_{a=1}^{m} \sum_{b=1}^{m} \exp\left(-\frac{\|u_\phi(\tilde{x}_{k,a}) - u_\phi(\tilde{x}_{k,b})\|_2^2}{D} \right) \right]. \tag{7}$$

Together, these terms encourage synthetic clouds to remain consistent with their partition while retaining intra-partition diversity. Overall, Eq. 5 augments the primary confidence-weighted matching objective with partition-consistent feature alignment and dispersion-aware regularization, aiming to yield a more stable label-free distillation procedure under proxy supervision.

3.4 Transferable Instantiation

UPCDD can be instantiated by adapting existing label-dependent 3D distillation methods in Phase 3.

Instantiation on generative gradient matching. For generative gradient-matching methods such as DD3D (Bo & Wang, 2025), the latent partition index k serves as both the generator condition and the real-data grouping index. Writing $\tilde{x}_{k,j} = G_\psi(z_{k,j}, k)$, with $z_{k,j}$ denoting the generator input, we compute

$$g_k^{\text{real}} = \nabla_\phi \frac{\sum_{x_i \in \mathcal{B}_k} w_i\, \ell(f_\phi(x_i), k)}{\sum_{x_i \in \mathcal{B}_k} w_i}, \qquad g_k^{\text{syn}} = \nabla_\phi \frac{1}{m} \sum_{j=1}^{m} \ell(f_\phi(\tilde{x}_{k,j}), k). \tag{8}$$

where ℓ is cross-entropy. In this instantiation, the method-specific partition-wise matching term in Eq. 1 is $D_{\text{grad}}(g_k^{\text{syn}}, g_k^{\text{real}})$, using the DD3D gradient-distance metric. We retain DD3D's generator parameterization and outer/inner optimization structure, while Eq. 5 supplies the shared alignment and dispersion regularizers. Details are in Alg 1.

Instantiation on generative distribution matching. UPCDD can also be instantiated with distribution-matching objectives such as SADM (Yim et al., 2025).Following SADM, we use channel-wise sorted point features to account for the unordered structure of point clouds, while replacing its class-wise grouping with confidence-aware latent partitions.

Let $\tilde{h}_\phi(x) = \text{Sort}_{\text{ch}}(h_\phi(x))$ denote the channel-wise sorted point-feature representation and $h_\phi^{\text{gf}}(x)$ the max-pooled global feature. For each partition, we compute a confidence-weighted real statistic and its synthetic counterpart as

$$\hat{\mu}_{k,\text{real}}^{\text{sort}} = \frac{\sum_{(x_i, w_i) \in \hat{\mathcal{D}}_k} w_i\, \tilde{h}_\phi(x_i)}{\sum_{(x_i, w_i) \in \hat{\mathcal{D}}_k} w_i}, \qquad \mu_{k,\text{syn}}^{\text{sort}} = \frac{1}{m} \sum_{j=1}^{m} \tilde{h}_\phi(s_{k,j}^{\vartheta}), \tag{9}$$

where $s_{k,j}^{\vartheta} = R_{\vartheta_{k,j}}(\tilde{x}_{k,j})$ denotes the rotated synthetic cloud when rotation optimization is enabled. The partition-wise matching objective is

$$\mathcal{L}_{\text{match}}^{\text{SADM}} = \sum_{k=1}^{K} m \left[\lambda_{\text{sort}} \left\| \hat{\mu}_{k,\text{real}}^{\text{sort}} - \mu_{k,\text{syn}}^{\text{sort}} \right\|_2^2 + \lambda_{\text{mmd}} \mathcal{M}_\kappa^w \left(h_\phi^{\text{gf}}; \hat{\mathcal{D}}_k, \mathcal{S}_k^{\vartheta} \right) \right], \tag{10}$$

where $\mathcal{M}_\kappa^w$ denotes the confidence-aware Gaussian-kernel MMD on global features. Synthetic coordinates and rotation parameters are jointly optimized as in SADM, while Eq. 5 supplies the shared alignment and dispersion regularizers.

Extension to part segmentation. Within each shape partition, we cluster frozen point-wise features to construct local pseudo-parts, assign globally distinct pseudo-part indices, and combine point-level and shape-level confidence. The confidence-weighted part proportions determine intervals of the generator's latent coordinate, assigning each generated point the corresponding pseudo-part label. These labels define point-wise cross-entropy targets for real and synthetic clouds. Gradient matching alternates between all-point and part-specific masked supervision, extending the same proxy-supervision construction to fine-grained point-level training.

For part segmentation, we extend proxy supervision from shapes to points. Within each shape partition k, frozen point-wise features $h_{i,p}$ are clustered into P_k local pseudo-parts, with P_k selected by the silhouette criterion. Each point is assigned

$$q_{i,p} = \arg\min_{q\in\{1,\dots,P_k\}} \|h_{i,p} - \nu_{k,q}\|_2^2, \qquad \hat{r}_{i,p} = kP_{\max} + q_{i,p}, \qquad \omega_{i,p} = w_i\, w_{i,p}^{\text{part}}, \tag{11}$$

where $\nu_{k,q}$ is the local centroid and $\omega_{i,p}$ combines shape- and point-level confidence.

We estimate the confidence-weighted pseudo-part proportion

$$\pi_{k,r} = \frac{\sum_{i:\hat{y}_i=k} \sum_p \omega_{i,p} \mathbb{I}[\hat{r}_{i,p} = r]}{\sum_{i:\hat{y}_i=k} \sum_p \omega_{i,p}}. \tag{12}$$

The cumulative proportions $\{\pi_{k,r}\}$ partition the generator coordinate $z \in [0, 1]$, thereby assigning each generated point a pseudo-part label. Using these point-wise labels, gradient matching alternates between confidence-weighted supervision over all points and part-specific masked supervision, while retaining the alignment and dispersion regularizers of Phase 3.

4 Experiments

We evaluate the proposed UPCDD framework on two 3D point-cloud tasks, *shape classification* and *part segmentation*. Our experiments are organized into (i) the **unsupervised** setting, where we assess the direct semantic alignment of distilled data, its downstream training utility, and its cross-dataset transferability, and (ii) the **supervised** setting, where we conduct a **supervised instantiation** to demonstrate the effect of the proposed Phase-3 augmented alignment objective within existing 3D distillation frameworks. We further provide part segmentation, ablations and cross-architecture evaluation.

4.1 Experimental Setups

Datasets and Pre-processing. Following DD3D (Bo & Wang, 2025), we evaluate shape classification on ScanObjectNN (OBJ_BG) (Uy et al., 2019), ModelNet40 (Wu et al., 2015), and MVPNet100 (Yu et al., 2023), and part segmentation on ShapeNet-Part (Yi et al., 2016). We use the aligned setting, with $N_p = 1024$ points per object for classification and $N_p = 2048$ for segmentation. Real point clouds and the final synthesized clouds are centered and scaled to the unit sphere. Synthetic budgets are $m \in \{1, 5, 10\}$ clouds per partition, giving $M = Km$ in the label-free setting and $M = Cm$ under ground-truth class conditioning. Additional implementation details are provided in Appendix B.

Evaluation protocols. We evaluate label-free distilled sets through direct unsupervised evaluation and frozen linear probing. Direct evaluation resolves the pseudo-label permutation using a Hungarian mapping fitted only on the real training split (Appendix B.2). For downstream and cross-dataset evaluation, a fresh consumer is trained on the fixed synthetic set, after which its backbone is frozen and a linear classifier is fitted using 10% labeled real data B.2. Cross-architecture and segmentation settings follow their task-specific protocols detailed in Appendix B.2–B.3.

4.2 Main Experiments

Direct unsupervised evaluation. We instantiate UPCDD on DD3D and SADM to compare with pseudo-labeled real-data baselines, including PL-Full and the best pseudo-label core-set results among

Table 1: **Label-free in-domain shape classification under direct unsupervised evaluation.** We report $\mathrm{ACC}_{\mathrm{dir}}$ (%). Auto-K selects the latent partition granularity without using the ground-truth class count, while Default-K sets K to the number of ground-truth categories only as a reference for comparison.

Dataset	CPP	UPCDD-DD3D		UPCDD-SADM		Pseudo-labeled Real Data	
		Auto-K	Default-K	Auto-K	Default-K	PL-Full	Best PL-Core
ScanObjectNN	1	26.12±1.18	**26.87±2.04**	18.76±1.48	**21.17±0.94**	30.41	21.34
	5	**30.90±0.44**	28.49±1.21	19.28±2.34	**22.38±1.21**		22.55
	10	**31.30±0.52**	29.47±1.29	23.58±2.42	**24.27±2.13**		24.44
ModelNet40	1	**45.91±0.35**	44.27±1.10	**27.71±1.32**	27.07±1.52	52.88	41.73
	5	**47.35±0.66**	46.42±1.38	**30.39±2.70**	28.28±2.47		44.21
	10	**48.39±0.41**	47.07±1.29	**31.81±1.46**	31.69±1.26		44.77
MVPNet100	1	15.09±0.35	**15.38±0.40**	**13.18±1.22**	11.74±1.15	19.96	17.14
	5	**18.48±0.21**	18.40±0.12	**14.71±1.69**	14.59±2.36		18.70
	10	**19.14±0.25**	18.97±0.18	15.46±1.84	**15.57±1.78**		18.87

Table 2: **Downstream linear-probe accuracy (LP@10%).** LP@10% Adam@50: test OA at fixed epoch 50 of an Adam linear probe trained with 10% labeled real data. UPCDD-DD3D and UPCDD-SADM select K independently under Auto-K.

Dataset	m	Random Init	PL-Full	Best PL-Core		UPCDD-DD3D		UPCDD-SADM	
				K	Acc	K	Acc	K	Acc
ScanObjectNN	1	20.60±3.91	42.57±1.60	5	37.87±0.49	7	**38.21±1.35**	5	36.83±0.37
	5		42.86±2.57	5	39.39±0.32	7	39.16±0.61	5	**39.97±0.91**
	10		43.49±1.81	5	39.90±0.87	7	41.22±1.59	5	**42.99±0.51**
ModelNet40	1	38.14±3.48	78.17±0.77	22	76.05±1.48	22	**76.52±0.74**	31	76.20±0.63
	5		78.51±0.62	22	77.07±0.69	22	76.45±0.75	31	**78.33±0.27**
	10		78.09±0.74	22	77.82±0.66	22	76.83±0.19	31	**78.86±0.69**
MVPNet100	1	19.97±0.29	43.86±0.25	51	42.71±1.07	60	**42.98±0.42**	105	41.83±0.88
	5		43.77±0.37	51	44.48±0.79	60	43.05±0.91	105	**45.37±0.29**
	10		43.73±0.45	51	45.08±0.77	60	43.64±0.30	105	**46.29±0.35**

random selection, Herding, K-Center. PL-Full trains on all pseudo-labeled real point clouds, while the core-set baseline selects a subset under the same budget. We measure the **direct unsupervised accuracy $\mathbf{ACC}_{\mathrm{dir}}$** (see Appendix B.2). Both instantiations achieve competitive accuracy without any supervised adaptation, showing that our label-free distilled point clouds enable the backbone to learn meaningful semantic structure. Auto-K also makes the latent grouping operational without specifying the true class count. This validates the quality of the unsupervised distilled dataset and the effectiveness of UPCDD.

Downstream utility of the distilled sets. Table 2 evaluates the representations learned from the synthetic sets through frozen linear probing (Sautier et al., 2022; Liu et al., 2023), together with random-initialization and pseudo-labeled real-data references. The distilled sets remain competitive with strong real-data coresets and benefit from increased synthetic capacity. Since the backbone is frozen during labeled readout, these results demonstrate the label-free distilled sets provide useful training signals for learning effective *geometry-aware semantic* representations. For reference, Appendix A.5 reports the linear-probe performance of the Phase-1 masked SSL encoder. Producer-side construction costs and subsequent training-time measurements are provided in Appendix A.6.

Cross-dataset transfer. We transfer each source-trained backbone to the other two classification datasets and fit a new linear classifier using 10% of the target training labels. Table 3 covers all six directed dataset pairs and all three budgets, evaluating every target category on target test sets. These results show that representations learned from the distilled sets contain transferable semantic structure and remain useful beyond the source dataset.

Supervised distillation: effect of our Phase-3 optimization. We evaluate in a supervised setting to isolate the effect of the proposed augmented distillation objective from proxy-supervision construction.

Table 3: **Cross-dataset transfer over the full target category set.** $\mathrm{ACC}_{\mathrm{dir}}$: Hungarian-mapped test OA (%) on the full target label set. LP: frozen backbone, target LP@10% Adam@50. DD3D and SADM choose K independently. SO = ScanObjectNN, MN = ModelNet40, MV = MVPNet100.

Direction	Metric	UPCDD-DD3D			UPCDD-SADM		
		m=1	m=5	m=10	m=1	m=5	m=10
SO→MN	$\mathrm{ACC}_{\mathrm{dir}}$	16.23 ± 0.61	15.41 ± 0.32	14.72 ± 0.54	25.80 ± 1.30	15.64 ± 0.57	12.62 ± 0.35
SO→MN	LP	75.68 ± 1.72	74.66 ± 1.15	75.14 ± 1.64	74.35 ± 1.46	75.28 ± 1.72	76.20 ± 1.44
SO→MV	$\mathrm{ACC}_{\mathrm{dir}}$	7.70 ± 0.60	8.34 ± 0.38	8.26 ± 0.30	9.65 ± 0.64	6.99 ± 0.28	6.48 ± 0.28
SO→MV	LP	40.92 ± 0.90	40.65 ± 0.85	41.34 ± 0.55	39.85 ± 0.20	41.70 ± 0.62	42.73 ± 0.77
MN→SO	$\mathrm{ACC}_{\mathrm{dir}}$	27.32 ± 1.83	26.66 ± 1.68	26.76 ± 1.15	30.92 ± 0.92	25.93 ± 2.09	25.06 ± 1.25
MN→SO	LP	38.96 ± 1.34	39.82 ± 1.15	40.62 ± 1.07	36.03 ± 1.90	43.26 ± 1.67	44.58 ± 1.47
MN→MV	$\mathrm{ACC}_{\mathrm{dir}}$	12.09 ± 0.57	10.88 ± 0.26	10.94 ± 0.35	11.29 ± 0.37	10.01 ± 0.58	9.70 ± 0.66
MN→MV	LP	41.72 ± 0.57	41.79 ± 0.93	41.57 ± 0.93	40.00 ± 1.27	44.35 ± 0.69	44.66 ± 0.69
MV→SO	$\mathrm{ACC}_{\mathrm{dir}}$	24.66 ± 2.37	28.29 ± 0.68	25.82 ± 1.54	27.17 ± 2.24	27.39 ± 0.99	28.66 ± 1.09
MV→SO	LP	37.52 ± 1.55	39.36 ± 1.39	41.19 ± 1.72	35.63 ± 0.91	42.17 ± 0.62	44.41 ± 1.24
MV→MN	$\mathrm{ACC}_{\mathrm{dir}}$	25.49 ± 1.26	23.78 ± 2.04	28.14 ± 1.64	31.77 ± 2.49	25.87 ± 1.90	32.08 ± 1.38
MV→MN	LP	74.66 ± 0.97	75.23 ± 0.74	76.39 ± 0.54	74.82 ± 1.31	77.49 ± 0.51	78.44 ± 0.48

Table 4: **Supervised distillation (%).** Test Overall Accuracy (OA) under different CPC (point Cloud Per Class) budgets. Coreset baselines (Random/Herding/K-Center) are computed in our setting. † denotes re-implementation or evaluation under the same protocol. Ours-DD3D and Ours-SADM instantiate the proposed Phase-3 augmented alignment objective within DD3D- and SADM-style supervised distillation pipelines, respectively.

		Coreset-based			Distillation-based			
Dataset	**CPC**	**Random**	**Herding**	**K-Center**	**DD3D†**	**SADM†**	**Ours-DD3D**	**Ours-SADM**
ScanObjectNN	1	16.78 ± 0.62	23.06 ± 0.74	23.55 ± 0.77	22.10 ± 0.84	19.67 ± 0.70	$\mathbf{31.00 \pm 0.94}$	18.36 ± 1.46
	5	28.93 ± 0.97	34.97 ± 1.04	22.07 ± 1.22	41.64 ± 1.24	37.14 ± 1.23	$\mathbf{42.43 \pm 0.88}$	38.77 ± 4.95
	10	34.22 ± 0.97	41.81 ± 1.03	29.16 ± 0.75	43.85 ± 1.20	42.84 ± 1.04	48.26 ± 0.83	$\mathbf{52.19 \pm 1.11}$
ModelNet40	1	37.95 ± 1.13	44.47 ± 1.38	44.65 ± 1.29	59.53 ± 0.66	55.50 ± 1.48	$\mathbf{63.31 \pm 0.41}$	57.93 ± 2.93
	5	66.99 ± 0.44	67.53 ± 0.53	61.32 ± 0.65	75.75 ± 0.47	76.10 ± 0.86	76.15 ± 0.52	$\mathbf{77.52 \pm 0.88}$
	10	72.76 ± 0.19	73.02 ± 0.42	70.23 ± 0.66	78.65 ± 0.21	80.09 ± 1.42	79.66 ± 0.35	$\mathbf{81.26 \pm 0.41}$
MVPNet100	1	9.84 ± 0.25	8.00 ± 0.24	8.24 ± 0.22	9.39 ± 0.58	13.06 ± 0.64	$\mathbf{19.82 \pm 0.53}$	13.36 ± 1.27
	5	17.80 ± 0.23	19.97 ± 0.24	10.86 ± 0.34	25.06 ± 0.49	24.23 ± 0.98	25.92 ± 0.39	$\mathbf{27.61 \pm 0.80}$
	10	23.74 ± 0.26	24.83 ± 0.34	14.00 ± 0.39	31.39 ± 0.30	31.38 ± 1.15	31.59 ± 0.41	$\mathbf{34.30 \pm 0.39}$

Table 4 compares coreset selection and distillation-based baselines under the same CPP budget. As shown in Table 4, using the augmented alignment objective in our Phase3 design yields improvements on all datasets. On both distillation instantiations, it strengthens distillation, demonstrating its robustness and transferability. These results further verify that the objective can also benefit supervised point-cloud distillation.

4.2.1 Extension to Part Segmentation

We extend proxy supervision to point-level pseudo-parts and evaluate it on ShapeNet-Part. Table 5 reports both *direct unsupervised evaluation* via Hungarian mapping (part-level) and *downstream performance* (Linear / Fine-tune / Scratch) (Appendix B.3). The direct evaluation indicates that the distilled sets preserve meaningful fine-grained structure without ground-truth supervision, while the downstream results demonstrate effective point-level representation learning and low-label transfer. Table 9 further evaluates the Phase-3 objective in supervised settings. Together, these results show that UPCDD extends beyond shape-level classification to the more challenging setting of fine-grained point-cloud segmentation.

Table 5: **Label-free ShapeNet-Part segmentation.** Each entry reports In.IoU / point-wise OA (%).

CPP	$\mathrm{ACC}_{\mathrm{dir}}$	Linear	Fine-tune	Scratch
1	40.23 / 46.15	73.65 / 86.47	76.62 / 88.41	
5	41.84 / 48.33	73.49 / 86.27	76.65 / 87.83	75.92 / 87.90
10	44.32 / 49.27	74.49 / 86.61	77.31 / 88.64	

4.3 Ablation Studies

Cross-architecture Generalization. We further evaluate whether the distilled dataset produced by UPCDD-DD3D using a PointNet backbone can transfer to other architectures. We keep the distillation unchanged and evaluate on a suite of advanced backbones (DGCNN, PointMLP, PointNet++, PCT, and PointNeXt). The cross-architecture evaluation in Table 6 uses the 400-epoch pseudo-label consumer protocol, followed by frozen linear probing on the locked 10% labelled bank. Overall, competitive performance across diverse evaluation architectures indicates that our unsupervised distilled dataset is not tightly coupled to the distillation backbone, and can provide effective *geometry-aware semantic structure* that benefits stronger, more modern models.

Table 6: **Cross-architecture generalization (OA, %).** Frozen LP@10% results using PointNet-produced distilled sets under CPP$= \{1, 5, 10\}$. SO = ScanObjectNN, MN = ModelNet40, and MV = MVPNet100. Entries are mean$\pm$std.

Model	ModelNet40 (MN)			MVPNet100 (MV)			ScanObjectNN (SO)		
	CPP=1	CPP=5	CPP=10	CPP=1	CPP=5	CPP=10	CPP=1	CPP=5	CPP=10
DGCNN	72.93 ± 0.57	74.66 ± 0.79	75.28 ± 0.07	36.74 ± 0.25	39.10 ± 0.25	39.11 ± 0.26	29.26 ± 1.98	32.47 ± 1.56	34.83 ± 1.12
PointMLP	37.30 ± 4.27	60.28 ± 2.21	66.83 ± 0.37	15.66 ± 1.26	29.66 ± 1.33	34.60 ± 0.51	15.43 ± 1.64	17.33 ± 2.84	29.37 ± 1.70
PointNet++	73.26 ± 0.21	74.89 ± 0.47	75.53 ± 0.46	42.86 ± 0.22	42.52 ± 0.44	41.96 ± 0.86	34.71 ± 1.56	32.93 ± 1.64	36.66 ± 2.54
PCT	75.18 ± 0.97	72.35 ± 0.17	71.96 ± 1.25	40.76 ± 0.38	40.69 ± 0.28	40.31 ± 0.81	31.67 ± 1.37	32.87 ± 1.47	34.42 ± 2.67
PointNeXt	66.77 ± 1.41	69.49 ± 0.72	71.14 ± 0.34	32.91 ± 1.41	38.94 ± 0.73	42.57 ± 0.94	30.98 ± 2.16	33.28 ± 0.98	32.42 ± 1.84

Sensitivity to clustering granularity. Table 7 investigates the effect of latent partition granularity k under direct unsupervised evaluation (CPP$= 1$, UPCDD-DD3D). Here, k controls the granularity of latent partitions in SSL feature space, while "default" denotes setting k to the number of ground-truth categories as a reference.

Table 7: **Sensitivity to latent partition granularity.** ACC_{dir} (%) at CPP$= 1$. "def." uses the ground-truth class count only as a diagnostic reference.

SO		MN		MV	
k	ACC_{dir}	k	ACC_{dir}	k	ACC_{dir}
5	22.42	15	29.38	30	11.28
10	24.65	**30**	**44.54**	**80**	**15.53**
15 (def.)	**26.87**	40 (def.)	44.27	100 (def.)	15.38
20	22.50	60	39.55	150	13.30
25	18.24	80	41.17	200	12.62

Table 8: **Component ablation of UPCDD-DD3D.** ACC_{dir} (%) under Auto-K at CPP$= 5$. Components are added cumulatively from pseudo-label-driven gradient matching.

Configuration	SO	MN
Pseudo-label gradient matching	28.40	46.31
+ Confidence weighting	29.09	46.86
+ Feature alignment	30.12	46.92
+ Dispersion regularization (Full)	**30.90**	**47.35**

Component Ablation. Table 8 evaluates the contributions of the proposed components in UPCDD-DD3D under Auto-K at CPP$= 5$. Starting from pseudo-label-driven gradient matching, confidence weighting improves ACC_{dir} on both datasets. Feature alignment and dispersion regularization provide further gains.

5 Conclusion

This paper introduces UPCDD, to the best of our knowledge, the first framework to enable dataset distillation for 3D point clouds in unsupervised settings. By leveraging pseudo-partitions derived from self-supervised representations as proxy supervision for distillation, UPCDD effectively captures geometry-aware representations without relying on human annotations. We further introduce an augmented alignment objective that incorporates feature-space regularization to enforce semantic consistency and prevent mode collapse. Experiments on shape classification and part segmentation validate that the distilled dataset preserves intrinsic semantic structures and exhibits cross-architecture generalizability and Cross-dataset transferability. Coupled with competitive downstream performance, our framework establishes a solid baseline for label-free 3D data distillation.

AI USE STATEMENT

Generative AI tools were used to assist with writing and language polishing, literature retrieval and discovery, and aspects of research ideation and execution, including feedback on methodology and experiments and assistance with implementation and result interpretation. All AI-assisted research suggestions, technical content, and literature-related claims were reviewed and verified by the authors. The authors take full responsibility for the final content of this work, including all text, claims, and artifacts produced with AI assistance.

ETHICS STATEMENT

This work uses publicly available 3D point-cloud benchmark datasets and does not involve human subjects, personally identifiable information, or sensitive personal data. We are not aware of any specific ethical risks beyond those generally associated with machine learning research.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the UPCDD framework and its optimization procedure in Section 3, with additional algorithmic and implementation details in Appendix B. Dataset sources, pre-processing, evaluation protocols, training schedules, reporting procedures, and hyperparameters are documented in the appendix. In particular, Appendix B.2–B.3 specifies the evaluation and statistical reporting protocols used throughout the experiments.

REFERENCES

Deyu Bo and Xinchao Wang. Point cloud dataset distillation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 4588–4601. PMLR, 2025. URL https://proceedings.mlr.press/v267/bo25a.html.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Computer Vision – ECCV 2018*, volume 11218 of *Lecture Notes in Computer Science*, pp. 139–156. Springer, 2018. doi: 10.1007/978-3-030-01264-9_9. URL https://doi.org/10.1007/978-3-030-01264-9_9.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/70feb62b69f16e0238f741fab228fec2-Abstract.html.

George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. doi: 10.1109/CVPR52688.2022.01045. URL https://openaccess.thecvf.com/content/CVPR2022/html/Cazenavette_Dataset_Distillation_by_Matching_Training_Trajectories_CVPR_2022_paper.html.

George Cazenavette, Antonio Torralba, and Vincent Sitzmann. Dataset distillation for pre-trained self-supervised vision models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2019.

Wenxiao Deng, Wenbin Li, Tianyu Ding, Lei Wang, Hongguang Zhang, Kuihua Huang, Jing Huo, and Yang Gao. Exploiting inter-sample and inter-feature relations in dataset distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17057–17066, June 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Deng_Exploiting_Inter-sample_and_Inter-feature_Relations_in_Dataset_Distillation_CVPR_2024_paper.html.

Jiawei Du, Yidi Jiang, Vincent Y. F. Tan, Joey Tianyi Zhou, and Haizhou Li. Minimizing the accumulated trajectory error to improve dataset distillation. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3749–3758, June 2023. doi: 10.1109/CVPR52729.2023.00365. URL https://openaccess.thecvf.com/content/CVPR2023/html/Du_Minimizing_the_Accumulated_Trajectory_Error_To_Improve_Dataset_Distillation_CVPR_2023_paper.html.

Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. URL https://openaccess.thecvf.com/content_iccv_2017/html/Dizaji_Deep_Clustering_via_ICCV_2017_paper.html.

Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. PCT: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, 2021. doi: 10.1007/s41095-021-0229-5. URL https://link.springer.com/article/10.1007/s41095-021-0229-5.

Ziyao Guo, Kai Wang, George Cazenavette, Hui Li, Kaipeng Zhang, and Yang You. Towards lossless dataset distillation via difficulty-aligned trajectory matching. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=rTBL8OhdhH.

Siddharth Joshi, Jiayi Ni, and Baharan Mirzasoleiman. Dataset distillation via knowledge distillation: Towards efficient self-supervised pre-training of deep networks. In *International Conference on Learning Representations*, 2025.

Dongwook Kim, Jae-Young Yim, and Jae-Young Sim. Parameterization-based dataset distillation of 3D point clouds through learnable shape morphing. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Qe7dKZOtWM.

Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, 1955. doi: 10.1002/nav.3800020109. URL https://doi.org/10.1002/nav.3800020109.

Dong Bok Lee, Seanie Lee, Joonho Ko, Kenji Kawaguchi, Juho Lee, and Sung Ju Hwang. Self-supervised dataset distillation for transfer learning. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/fe42607d1ededbda8b629333b4a639f7-Abstract-Conference.html.

Hongcheng Li, Yucan Zhou, Xiaoyan Gu, Bo Li, and Weiping Wang. Diversified semantic distribution matching for dataset distillation. In *Proceedings of the 32nd ACM International Conference on Multimedia (MM '24)*, pp. 7542–7550, 2024. doi: 10.1145/3664647.3680900. URL https://doi.org/10.1145/3664647.3680900.

Youquan Liu, Lingdong Kong, Jun Cen, Runnan Chen, Wenwei Zhang, Liang Pan, Kai Chen, and Ziwei Liu. Segment any point cloud sequences by distilling vision foundation models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 37193–37229, 2023. doi: 10.52202/075280-1617.

Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=3Pbra-_u76D.

Yatian Pang, Wenxiao Wang, Francis E. H. Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *Computer Vision – ECCV 2022*, volume 13662 of *Lecture Notes in Computer Science*, pp. 604–621. Springer, 2022. doi: 10.1007/978-3-031-20086-1_35. URL https://link.springer.com/chapter/10.1007/978-3-031-20086-1_35.

Charles R. 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 (NeurIPS)*, 2017. URL https://arxiv.org/abs/1706.02413.

Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Abed Al Kader Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. PointNeXt: Revisiting PointNet++ with improved training and scaling strategies. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/9318763d049edf9a1f2779b2a59911d3-Abstract-Conference.html.

Corentin Sautier, Gilles Puy, Spyros Gidaris, Alexandre Boulch, Andrei Bursuc, and Renaud Marlet. Image-to-lidar self-supervised distillation for autonomous driving data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9891–9901, 2022. doi: 10.1109/CVPR52688.2022.00966.

Peng Sun, Bei Shi, Daiwei Yu, and Tao Lin. On the diversity and realism of distilled dataset: An efficient dataset distillation paradigm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9390–9399, 2024.

Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Duc 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 (ICCV)*, 2019. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Uy_Revisiting_Point_Cloud_Classification_A_New_Benchmark_Dataset_and_Classification_ICCV_2019_paper.html.

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. SCAN: Learning to classify images without labels. In *European Conference on Computer Vision (ECCV)*, 2020. doi: 10.1007/978-3-030-58607-2_16. URL https://arxiv.org/abs/2005.12320.

Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matt J. Kusner. Unsupervised point cloud pre-training via occlusion completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9782–9792, 2021.

Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Guan Huang, Hakan Bilen, Xinchao Wang, and Yang You. CAFE: Learning to condense dataset by aligning features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12196–12205, 2022.

Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018. URL https://arxiv.org/abs/1811.10959.

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):146:1–146:12, 2019. doi: 10.1145/3326362. URL https://doi.org/10.1145/3326362.

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 (CVPR)*, 2015. doi: 10.1109/CVPR.2015.7298801. URL https://openaccess.thecvf.com/content_cvpr_2015/html/Wu_3D_ShapeNets_A_2015_CVPR_paper.html.

Saining Xie, Jiatao Gu, Demi Guo, Charles R. Qi, Leonidas J. Guibas, and Or Litany. PointContrast: Unsupervised pre-training for 3D point cloud understanding. In *Computer Vision – ECCV 2020*, volume 12348 of *Lecture Notes in Computer Science*, pp. 574–591. Springer, 2020. doi: 10.1007/978-3-030-58580-8_34. URL https://link.springer.com/chapter/10.1007/978-3-030-58580-8_34.

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):210:1–210:12, 2016. doi: 10.1145/2980179.2980238. URL https://doi.org/10.1145/2980179.2980238.

Jae-Young Yim, Dongwook Kim, and Jae-Young Sim. Dataset distillation of 3D point clouds via distribution matching. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://neurips.cc/virtual/2025/poster/117394. Poster. OpenReview: https://openreview.net/forum?id=YtmfBtIOPQ.

Jiahao You, Xu Han, Jinfeng Xu, and Xianzhi Li. Long-tailed 3D point cloud dataset distillation. *arXiv preprint arXiv:2607.26763*, 2026.

Sheng-Feng Yu, Jia-Jiun Yao, and Wei-Chen Chiu. Boost self-supervised dataset distillation via parameterization, predefined augmentation, and approximation. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=2RfWRKwxYh.

Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Chenming Zhu, Zhangyang Xiong, Tianyou Liang, Guanying Chen, Shuguang Cui, and Xiaoguang Han. MVImgNet: A large-scale dataset of multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9150–9161, 2023. doi: 10.1109/CVPR52729.2023.00883. URL https://openaccess.thecvf.com/content/CVPR2023/html/Yu_MVImgNet_A_Large-Scale_Dataset_of_Multi-View_Images_CVPR_2023_paper.html.

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.

Hongliang Zhang, Xiaoqi An, Jiawei Lian, Lei Luo, and Jian Yang. CoMPR: Efficient point cloud dataset condensation via bidirectional matching and point recycling. *Pattern Recognition*, 172: 112494, 2026. doi: 10.1016/j.patcog.2025.112494. URL https://doi.org/10.1016/j.patcog.2025.112494.

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, 2022.

Wenxiao Zhang, Ziqi Wang, Li Xu, Xun Yang, and Jun Liu. Informative point cloud dataset extraction for classification via gradient-based points moving. In *Proceedings of the 32nd ACM International Conference on Multimedia (MM '24)*, pp. 6384–6393, 2024. doi: 10.1145/3664647.3680767. URL https://doi.org/10.1145/3664647.3680767.

Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12674–12685. PMLR, 2021.

Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 6503–6512, 2023. doi: 10.1109/WACV56688.2023.00645. URL https://openaccess.thecvf.com/content/WACV2023/html/Zhao_Dataset_Condensation_With_Distribution_Matching_WACV_2023_paper.html.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=mSAKhLYLSsl.

A Results

A.1 Controlled supervised ShapeNet-Part segmentation

Table 9 further evaluates the Phase-3 objective under ground-truth shape and part supervision. Ours-DD3D replaces the original DD3D objective and improves DD3D† from 62.70% to 64.00% Instance IoU at CPP$=1$ and provides comparable performance at larger budgets. These results suggest that the augmented objective can also benefit supervised segmentation distillation.

Table 9: **Controlled supervised ShapeNet-Part segmentation.** Test Instance IoU (In.IoU, %) and point-wise OA (%). DD3D† denotes our re-implementation under the same protocol.

CPP	DD3D†		Coreset		Ours-DD3D	
	In.IoU	OA	In.IoU	OA	In.IoU	OA
1	62.70	72.29	56.43	66.06	**64.00**	**74.41**
5	62.72	74.74	62.58	73.89	**62.84**	**75.31**
10	60.39	73.33	**65.04**	**77.39**	62.97	75.57

A.2 Critical Protocol Ablation

Since direct unsupervised evaluation resolves the permutation between predicted pseudo-labels and semantic labels through a Hungarian mapping, we additionally verify that the reported accuracy does not arise merely from the permutation-matching procedure. We compare UPCDD-DD3D and UPCDD-SADM with an untrained PointNet evaluated under exactly the same Hungarian-mapping protocol.

Table 10: **Critical protocol ablation.** $\mathrm{ACC}_{\mathrm{dir}}$ (%) for UPCDD-DD3D and UPCDD-SADM under Auto-K. Random Hung uses an untrained PointNet under the same Hungarian-mapping protocol and is independent of CPP.

Data	CPP	DD3D	SADM	Random Hung
SO	1	26.12 ± 1.18	18.76 ± 1.48	14.09 ± 0.95
	5	30.90 ± 0.44	19.28 ± 2.34	
	10	31.30 ± 0.52	23.58 ± 2.42	
MN	1	45.91 ± 0.35	27.71 ± 1.32	6.50 ± 2.51
	5	47.35 ± 0.66	30.39 ± 2.70	
	10	48.39 ± 0.41	31.81 ± 1.46	
MV	1	15.09 ± 0.35	13.18 ± 1.22	4.30 ± 0.61
	5	18.48 ± 0.21	14.71 ± 1.69	
	10	19.14 ± 0.25	15.46 ± 1.84	

The substantial gap between UPCDD-DD3D/SADM and Random Hung across all three datasets shows that the direct unsupervised accuracy cannot be explained by label-permutation matching alone.

A.3 Rotation Compatibility

We additionally evaluate UPCDD-DD3D on ScanObjectNN with fixed random $\mathrm{SO}(3)$ rotations and DD3D's rotator. As shown in Table 11, the moderate performance decrease supports that existing orientation modules can be combined with UPCDD and deal with different orientation.

Table 11: **Rotation compatibility on ScanObjectNN.** $\mathrm{ACC}_{\mathrm{dir}}$ (%) over eight evaluation seeds.

K setting	CPP	Aligned	Random SO(3), No Rotator	Random $\mathrm{SO}(3)$ + Rotator
Default-K	1	26.87 ± 2.04	8.07 ± 0.89	23.17 ± 2.10
	5	28.49 ± 1.21	8.71 ± 1.00	25.69 ± 1.46
Auto-K	1	26.12 ± 1.18	10.33 ± 1.11	23.43 ± 2.80
	5	30.90 ± 0.44	10.59 ± 0.50	28.25 ± 0.88

A.4 Full-Data Supervised Reference

Table 12 reports models trained from scratch on the complete labeled training set. These results provide a supervised full-data reference for interpreting the performance gap of the highly compressed label-free distilled sets.

Table 12: **Full-data supervised reference (OA, %).**

Protocol	Architecture	ScanObjectNN	ModelNet40	MVPNet100
DD3D	PointNet	65.22	87.76	71.29
	DGCNN	74.73	89.91	82.32
	PointMLP	78.83	91.17	89.89
	PointNet++	83.30	91.21	89.54
	PCT	74.53	89.75	87.04
	PointNeXt	72.12	89.71	88.11
SADM	PointNet	63.27	84.70	67.89

A.5 Phase-1 SSL Reference

We additionally evaluate the Phase-1 encoder trained by masked point-cloud reconstruction. The encoder is frozen, and a linear classifier is trained with Adam. Table 13 reports the linear-probe accuracy at epoch 50, providing a reference for the representations learned before distillation. In contrast, Table 2 evaluates representations learned by fresh consumers trained on the compact distilled sets.

Table 13: **Phase-1 masked SSL reference.** Frozen linear-probe test OA (%) at epoch 50 using Adam. The pretrained encoder is evaluated directly, without synthetic-set training.

Dataset	LP (Adam@50)
ScanObjectNN	44.18 ± 1.60
ModelNet40	79.57 ± 0.94
MVPNet100	42.00 ± 0.14

A.6 Efficiency Analysis

We further report the wall-clock efficiency of the DD3D-based UPCDD instantiation on one NVIDIA GeForce RTX 5090. Table 14 reports the producer-side distillation time, including the additional SSL pretraining and clustering cost introduced by UPCDD. Table 15 reports the downstream training time on the pseudo-labeled full real set and the distilled synthetic set. Although UPCDD introduces an additional one-time cost for constructing pseudo partitions and optimizing the distilled set, the resulting synthetic set is substantially cheaper to reuse for downstream training.

Table 14: **Producer-side distillation time of the DD3D-based UPCDD instantiation.** Wall-clock time is measured on one NVIDIA GeForce RTX 5090. UPCDD Total includes SSL pretraining, clustering, and label-free distillation.

Dataset	CPP	DD3D (min)	SSL + Clustering (min)	UPCDD Distill. (min)	UPCDD Total (min)
ScanObjectNN	1	5.81	2.36	7.54	9.90
	5	10.51	2.42	12.16	14.58
	10	15.63	2.43	32.89	35.32
ModelNet40	1	19.70	9.68	29.34	39.02
	5	53.32	9.51	49.96	59.47
	10	73.34	9.61	80.51	90.12
MVPNet100	1	93.49	28.27	129.97	158.24
	5	220.28	28.41	185.99	214.40
	10	259.71	29.08	320.70	349.78

Table 15: **Downstream training time comparison.** Wall-clock time is measured on one NVIDIA GeForce RTX 5090. PL-Full denotes training on the full pseudo-labeled real set, while Distilled denotes training on the distilled synthetic set. Speedup is computed as PL-Full training time divided by distilled-set training time.

Dataset	CPP	SSL Full Set (s)	PL-Full Train (s)	Distilled Train (s)	Speedup
ScanObjectNN	1	141.56	28.11	0.56	50.2×
	5	144.69	28.18	0.90	31.3×
	10	145.81	27.98	1.74	16.1×
ModelNet40	1	569.21	113.99	0.79	144.3×
	5	569.21	113.70	2.71	42.0×
	10	569.21	114.03	4.73	24.1×
MVPNet100	1	1694.47	339.81	2.95	115.2×
	5	1703.40	338.90	6.68	50.7×
	10	1743.17	337.95	12.76	26.5×

Table 15 highlights the reuse advantage of the distilled set. Training on the distilled set is substantially faster than training on the full pseudo-labeled real set across all datasets and budgets. The speedup ranges from 16.1× to 144.3×. This confirms the practical motivation of UPCDD: the method trades a one-time producer-side cost for substantially cheaper repeated downstream training on compact distilled point clouds.

Together, these measurements characterize the construction cost and subsequent training efficiency of the distilled data.

B Additional Implementation Details

B.1 Architectures

For stable and efficient bi-level distillation, we use a lightweight point backbone PointNet as the surrogate network f_ϕ, consistent with prior 3D dataset distillation works (Bo & Wang, 2025). We evaluate the same distilled set under multiple architectures, including DGCNN Wang et al. (2019), PointMLP Ma et al. (2022), PointNet++ Qi et al. (2017), PCT Guo et al. (2021), and PointNeXt Qian et al. (2022) to test cross-architecture generalization. The exact configurations are provided in the appendix.

B.2 Evaluation Protocols

(i) **Unsupervised setting (UPCDD).** After distillation, we instantiate the final distilled dataset $\mathcal{S}$ by uniformly querying the trained generator G_ψ over pseudo-label indices $k \in \{1, \dots, K\}$ with CPP samples per pseudo-class using fresh latent coordinate, followed by the same normalization used throughout training. We then train a new backbone on $\mathcal{S}$ with a cross-entropy objective on pseudo-labels $\hat{y}$ to obtain a synthetic-pretrained model f_{syn}. Since f_{syn} is trained solely on $\mathcal{S}$, its performance provides a direct proxy for the quality of the distilled set.

Direct unsupervised evaluation. To assess intrinsic semantic alignment without supervised adaptation, we evaluate f_{syn} on real data and align its predicted pseudo-classes with ground-truth labels using a mapping fitted only on the training split $\mathcal{D}_{\text{tr}}$. The resulting mapping is then fixed and evaluated on the test split $\mathcal{D}_{\text{te}}$:

$$\pi^* = \arg \max_{\pi \in \Pi_{K,C}} \sum_{i \in \mathcal{D}_{\text{tr}}} \mathbb{I}(y_i = \pi(\hat{y}_i)) \,, \tag{13}$$

$$\text{ACC}_{\text{dir}} = \frac{1}{|\mathcal{D}_{\text{te}}|} \sum_{j \in \mathcal{D}_{\text{te}}} \mathbb{I}(y_j = \pi^*(\hat{y}_j)) \,. \tag{14}$$

Here, $\hat{y}$ denotes the pseudo-class predicted by f_{syn}, K is the number of pseudo-classes, and C is the number of ground-truth classes. When $K \leq C$, $\Pi_{K,C}$ is the set of injective mappings from pseudo-classes to ground-truth classes, and Eq. 13 is solved by the Hungarian algorithm (Kuhn, 1955;

Van Gansbeke et al., 2020; Ghasedi Dizaji et al., 2017). When $K > C$, $\Pi_{K,C}$ allows many-to-one mappings; equivalently, each pseudo-class is independently assigned to its majority ground-truth class on $\mathcal{D}_{\mathrm{tr}}$. Pseudo-classes with no training support are left unmatched and are counted as incorrect at test time. We report the resulting direct unsupervised accuracy $\mathbf{ACC}_{\mathrm{dir}}$.

Downstream performance. For the PointNet downstream evaluation, we train a fresh consumer on the fixed distilled set for 200 epochs using Adam and cross-entropy on the pseudo-labels. We then freeze its backbone and train only a linear classifier using 10% of the labeled real training split, corresponding to 230, 984, and 2,929 objects for ScanObjectNN, ModelNet40, and MVPNet100, respectively. The linear classifier is trained with Adam for 50 epochs, consistent with established frozen linear-probing protocols in 3D point-cloud representation learning (Sautier et al., 2022; Liu et al., 2023). We report the fixed epoch-50 test result as mean$\pm$sample standard deviation over three evaluation seeds.

For comparison with the distilled sets, we additionally evaluate two real-data references under the same pseudo-supervision and linear-probing protocol. PL-Full trains the consumer on the full pseudo-labeled real training set. PL-Core instead trains the consumer on a compact subset of pseudo-labeled real samples, selected using Random, Herding, or K-Center; Best PL-Core reports the strongest of these selection methods for each configuration. Random Init follows the same frozen linear-probing protocol but uses a randomly initialized backbone without synthetic-set pretraining. For cross-dataset transfer, the source backbone learned from the distilled set is frozen and only a new linear classifier is trained using 10% of the target training labels. The same fixed 50-epoch Adam readout and three-seed reporting protocol is used.

(ii) **Supervised setting (Phase-3 design evaluation).** Phase-3 design is further adapted to supervised setting, where we instantiate the same Phase-3 distillation framework, but set the conditioning index k (and the labels used in cross-entropy) to the ground-truth class label (or shape category and part label for segmentation) Bo & Wang (2025). We compare our supervised results against representative 3D distillation baselines (DD3D, SADM). We additionally include coreset baselines (Random/Herding/K-Center) under the same budget for reference. For part segmentation, since both global shape semantic information and fine-grained part details are required, many dataset distillation methods struggle with this task. We follow the practice of reporting a *Random* coreset Bo & Wang (2025).

B.3 Common Settings and Reporting

(i) **Synthetic budget and compression ratio.** We use $\mathrm{CPP} \in \{1, 5, 10\}$, where CPP denotes the number of synthetic point Clouds Per Partition. In the label-free setting, the distilled-set size is $|\mathcal{S}| = K \times \mathrm{CPP}$, where K is the number of latent partitions. In the supervised setting, $|\mathcal{S}| = C \times \mathrm{CPP}$, where C is the number of ground-truth classes. The corresponding compression ratio $|\mathcal{S}|/|\mathcal{D}_{\mathrm{train}}|$ falls in the range of approximately 0.1%–6.5% across the evaluated settings. (ii) **Statistics.** We run distillation once to obtain a final synthetic set, then report mean ±std over 10 evaluation runs with different random seeds in the unsupervised evaluation. For the PointNet downstream and cross-dataset experiments, we use a fixed distilled set and report mean$\pm$std over three evaluation seeds. (iii) **Downstream training schedule.** For the PointNet classification experiments, a fresh consumer is trained on the fixed distilled set for 200 epochs using pseudo-label supervision. Its backbone is then frozen, and a linear classifier is trained with Adam for 50 epochs using 10% of the labeled real training set. We report the fixed epoch-50 result rather than selecting the best epoch. For part segmentation, distillation is more challenging than classification, especially in unsupervised settings due to its fine-grained, point-level supervision needs. We therefore evaluate multiple generator checkpoints, each with a fixed 50-epoch downstream budget, and report the best-achieved score over all checkpoints and epochs, using the same rule for all methods on segmentation task. This should be interpreted as an *upper-envelope* performance under a fixed search budget, measured in this strict low-budget regime with limited training epochs. (iv) **Metrics.** For shape classification, we report the **Overall Accuracy (OA)**, which is the ratio of correctly classified test shapes. For part segmentation, we report the **Instance Intersection-over-Union (Instance IoU)** and **point-wise OA**. $\mathbf{ACC}_{\mathrm{dir}}$ is direct unsupervised overall accuracy defined in Eq. 14. Specifically, the Instance IoU is computed by averaging the IoU scores of all part categories appearing in a shape instance, and then averaging these

scores across all test shapes. The point-wise OA is the ratio of points that are correctly classified into their part categories over the test set.

B.4 Datasets and Sampling Protocol

Dataset statistics. We evaluate on four standard benchmarks covering both classification and part segmentation: ScanObjectNN, ModelNet40, MVPNet, and ShapeNetPart. For MVPNet, we use the MVPNet100 (mvp100full) subset. Table 16 summarizes the datasets used in our experiments.

Data sources. We evaluate on four widely-used 3D point cloud benchmarks, covering both classification and part segmentation:

- **ScanObjectNN:** `https://hkust-vgd.github.io/scanobjectnn/`.
- **ModelNet40:** `https://github.com/lmb-freiburg/orion`.
- **MVPNet (MVImgNet):** `https://github.com/GAP-LAB-CUHK-SZ/MVImgNet`.
- **ShapeNetPart:** `https://github.com/antao97/PointCloudDatasets`.

Table 16: Dataset statistics used in our experiments.

Dataset	Task	#Classes	Part Classes	#Points	Train	Test
ScanObjectNN	Cls	15	–	1024	2309	581
MVPNet100	Cls	100	–	1024	29288	12553
ModelNet40	Cls	40	–	1024	9843	2468
ShapeNetPart	Seg	16 (shape)	50	2048	12137	4744

C Additional Reference: Reported Results of CoMPR

For completeness, we also include the results reported by a concurrent work CoMPR (Zhang et al., 2026) in their original paper. These numbers are provided as an external reference rather than a strictly controlled comparison, since the experimental protocol, budget choices, and evaluated datasets are not fully aligned with our main setting. In particular, CoMPR reports results at CPC$=30$, while our main experiments focus on the more compact CPC$=1/5/10$ budgets.

Table 17: **Reported results of CoMPR.** We reproduce the results reported in the CoMPR paper as an external reference. These numbers are not directly comparable to our main results because the protocol and CPC budgets are not fully aligned.

Dataset	**CPC=1**	**CPC=5**	**CPC=30**
ScanObjectNN	39.29±0.01	59.67±0.01	64.58±0.01
ModelNet40	61.09±0.01	79.71±0.01	84.60±0.00
ModelNet40-C	57.29±0.01	72.55±0.01	77.45±0.01

D t-SNE visualization of pseudo partitions

We visualize the pseudo partitions produced by feature-space K-Means (in Fig. 2). We extract PointNet features for real point clouds and project them into 2D using t-SNE. As a comparison, we also visualize random partitions with the same number of groups ($K = 15$).

D.1 Pseudo-label Quality Analysis

We further evaluate the quality of the pseudo-labels produced by SSL feature clustering. Although UPCDD does not use ground-truth labels during distillation, we compute quantitative cluster-label agreement metrics after clustering for analysis only. We use NMI (cluster–label mutual agreement) and purity (majority-label consistency within each cluster). These results indicate that the pseudo-labels retain meaningful semantic structure rather than collapsing completely, supporting the overall effectiveness of our label-free distillation pipeline.

Algorithm 1 UPCDD-DD3D: Pseudo-Partition Guided Gradient Alignment

Require: Unlabeled dataset $\mathcal{D} = \{x_i\}_{i=1}^N$; SSL encoder f_θ; candidate partition set $\mathcal{K}$; generator G_ψ; surrogate network f_ϕ; CPP m; distillation iterations I; outer steps T; inner steps S
Ensure: Distilled generator parameters ψ and synthetic set $\mathcal{S}$
Phase 1: Self-supervised representation learning
Train f_θ on $\mathcal{D}$ with the self-supervised objective, e.g., masked reconstruction in Eq. 2.
Discard the auxiliary pretraining head and freeze f_θ.
Extract global features $v_i = f_\theta(x_i)$ for all $x_i \in \mathcal{D}$.
Phase 2: Adaptive latent semantic partitioning
for each candidate $K \in \mathcal{K}$ **do**
  Run K-Means on $\{v_i\}_{i=1}^N$ with K clusters.
  Compute the silhouette score $\mathrm{Sil}(K)$.
end for
Select $K^\star = \arg\max_{K \in \mathcal{K}} \mathrm{Sil}(K)$ as in Eq. 3.
Run K-Means with $K^\star$ clusters and assign pseudo-labels $\hat{y}_i$ by Eq. 3.
Compute $d_i = \|v_i - \mu_{\hat{y}_i}\|_2$ and confidence weights w_i by Eq. 4.
Form the partition-annotated dataset $\hat{\mathcal{D}} = \{(x_i, \hat{y}_i, w_i)\}_{i=1}^N$ and subsets $\hat{\mathcal{D}}_k = \{x_i \mid \hat{y}_i = k\}$.
Phase 3: DD3D-style bi-level generative distillation
for distillation iteration $t = 1, \ldots, I$ **do**
  Initialize the surrogate network f_ϕ with random weights.
  for outer step $o = 1, \ldots, T$ **do**
    $\mathcal{L} \leftarrow 0$.
    for latent partition $k = 1, \ldots, K^\star$ **do**
      Sample a real mini-batch $B_k \subset \hat{\mathcal{D}}_k$ together with confidence weights $\{w_i\}_{x_i \in B_k}$.
      Sample noise codes $\{z_{k,j}\}_{j=1}^m$ and synthesize $\mathcal{S}_k = \{\tilde{x}_{k,j} = G_\psi(z_{k,j}, k)\}_{j=1}^m$.
      Compute confidence-weighted real gradient g_k^{real} and synthetic gradient g_k^{syn} following Eq. 8.
      Compute feature-level alignment $\mathcal{L}_{\mathrm{align}}^{(k)}$ by Eq. 6.
      if $m > 1$ **then**
        Compute dispersion regularization $\mathcal{L}_{\mathrm{disp}}^{(k)}$ by Eq. 7.
      else
        Set $\mathcal{L}_{\mathrm{disp}}^{(k)} = 0$.
      end if
      Accumulate

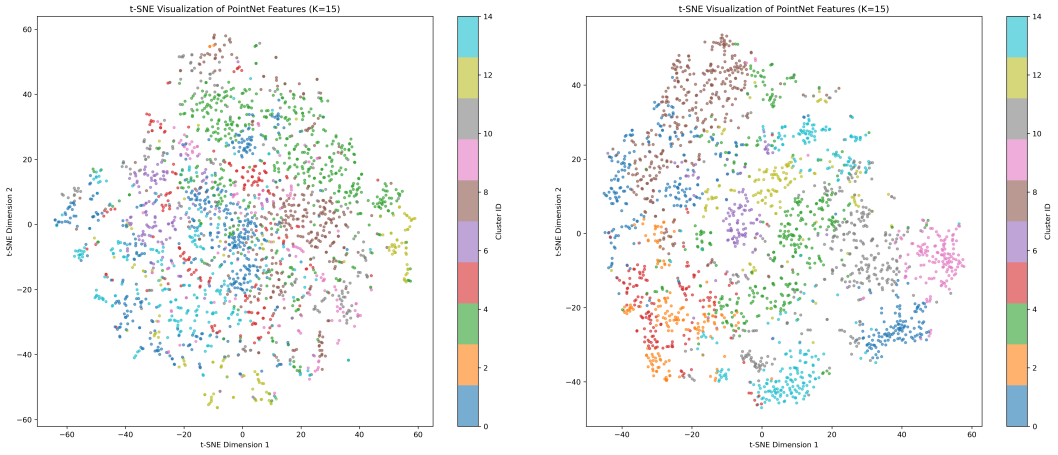
$$\mathcal{L} \leftarrow \mathcal{L} + D_{\mathrm{grad}}(g_k^{\mathrm{syn}}, g_k^{\mathrm{real}}) + \lambda_{\mathrm{align}} \mathcal{L}_{\mathrm{align}}^{(k)} + \lambda_{\mathrm{disp}} \mathcal{L}_{\mathrm{disp}}^{(k)}.$$

    end for
    Update generator parameters $\psi \leftarrow \psi - \eta_\psi \nabla_\psi \mathcal{L}$.
    Update f_ϕ for S inner steps using the DD3D inner-loop objective on the current synthetic set $\mathcal{S} = \bigcup_{k=1}^{K^\star} \mathcal{S}_k$.
  end for
end for

Table 18: **Quantitative pseudo-label quality.** NMI measures cluster–label mutual agreement, while purity measures majority-label consistency within each cluster. Ground-truth labels are used only for post-hoc analysis and are not used by UPCDD.

Dataset	NMI	Purity
ModelNet40	0.6419	0.6103
MVPNet100	0.3633	0.3041
ScanObjectNN	0.2801	0.3478

E Other Details



(a) Random partitions ($K = 15$). (b) K-Means pseudo partitions ($K = 15$).

Figure 2: **t-SNE visualization of PointNet features under different partitions.** Colors indicate partition IDs. Compared to random partitions, K-Means pseudo partitions exhibit more coherent structure in the learned feature space, supporting their use as grouping signals for label-free distillation.

Table 19: Hyperparameters used for Data Synthesis in unsupervised settings.

	ScanObjectNN	ModelNet40	MVPNet100
Phase-1 Pretrain epochs	400	400	400
Pretrain Optimizer	Adam	Adam	Adam
Pretrain LR	0.001	0.001	0.001
Generator Optimizer	Adam	Adam	Adam
Generator LR	0.001	0.001	0.001
Generator Weight Decay	0.0005	0.0005	0.0005
Iterations	200	400	400
Scheduler (Generator)	StepLR	StepLR	StepLR
StepLR setting	step=100 iter, $\gamma = 0.1$	step=200 iter, $\gamma = 0.1$	step=200 iter, $\gamma = 0.1$
Distill real batch	32	32	128

Table 20: Hyperparameters for frozen LP@10% in unsupervised settings. The backbone is frozen. Only the linear head is trained on the locked 10% label bank.

	ScanObjectNN	ModelNet40	MVPNet100
Probe epochs	50	50	50
Probe Optimizer	Adam	Adam	Adam
Probe LR	0.001	0.001	0.001
Probe Weight Decay	0.0005	0.0005	0.0005
Probe Scheduler	CosineAnnealingLR ($\eta_{\min} = 10^{-4}$)		
Batch Size	32	32	32
Label fraction	0.1	0.1	0.1

Table 21: Hyperparameters used for test in unsupervised settings.

	ScanObjectNN	ModelNet40	MVPNet100
Eval-train epochs	200	200	200
Eval Optimizer	Adam	Adam	Adam
Eval LR	0.001	0.001	0.001
Eval Weight Decay	0.0005	0.0005	0.0005
Eval Scheduler	CosineAnnealingLR	CosineAnnealingLR	CosineAnnealingLR
Batch Size	32	32	32
Eval fractions	0.1	0.1	0.1

Table 22: Hyperparameters for Data Synthesis (including Pseudo-part construction) and Downstream Test on ShapeNetPart.

Data Synthesis and Downstream Test			
Parameter	Value	Parameter	Value
Synthesis (Generator)		**Pseudo-part Construction**	
Synthesis Optimizer	Adam	Pseudo-part Pretrain epochs	400
Synthesis LR	0.0005	Max parts	8
Synthesis Weight Decay	0.00001	Silhouette K range	$2 \sim 8$
Synthesis Scheduler	StepLR (step=200 iter, γ=0.1)	Sample ratio	0.1
Synthesis Iterations	400		
Synthesis Batch Size	32		
Points	2048		
Part classes	50		
Eval (Training on distilled data)		**Downstream Test**	
Eval-train epochs	400	Downstream Training epochs	50
Eval Optimizer	SGD	Downstream Optimizer	Adam
Eval LR	0.01	Downstream LR	0.001
Eval Weight Decay	0.01	Downstream Batch Size	32
Eval Scheduler	CosineAnnealingLR	Data Fraction	0.1
Eval Batch Size	32		

Table 23: Hyperparameters used for Data Synthesis in supervised settings.

	ScanObjectNN	ModelNet40	MVPNet100	ShapeNet (16-class)
Generator Optimizer	Adam	Adam	Adam	Adam
Generator LR	0.001	0.001	0.001	0.001
Generator Weight Decay	0.0005	0.0005	0.0005	0.0005
Iterations	200	400	400	400
Scheduler (Generator)	StepLR	StepLR	StepLR	StepLR
StepLR setting	step=100 iter, $\gamma = 0.1$	step=200 iter, $\gamma = 0.1$	step=200 iter, $\gamma = 0.1$	step=100 iter, $\gamma = 0.1$
Distill real batch	32	32	128	32

Table 24: Hyperparameters used for Test in supervised settings.

	ScanObjectNN	ModelNet40	MVPNet100	ShapeNet (16-class)
Optimizer	Adam	Adam	Adam	Adam
Initial LR	0.001	0.001	0.001	0.001
Weight Decay	0.0005	0.0005	0.0005	0.0005
Batch Size	8	8	32	8
Epochs	200	200	200	200
Scheduler	CosineAnnealingLR	CosineAnnealingLR	CosineAnnealingLR	CosineAnnealingLR