COHERENCE-PRESERVING DISTRIBUTION MATCHING FOR 3D POINT CLOUD DATASET DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Dataset distillation compresses a large training set into a compact synthetic set while preserving its training utility. For 3D point clouds, distribution matching must additionally account for arbitrary point indexing. Existing Semantically Aligned Distribution Matching (SADM) addresses this issue by independently sorting pointwise feature responses within each channel. However, such channel-wise sorting discards the cross-channel associations among responses originating from the same point, introducing an unnecessary invariance to independent channel permutations. We propose Coherence-Preserving Distribution Matching (CPDM), which complements channel-wise matching with a shared point ordering derived from aggregated normalized feature responses. Applying this shared ordering to complete pointwise feature vectors preserves their cross-channel coherence while remaining invariant to point re-indexing. CPDM further employs trained expert checkpoints for feature extraction and auxiliary classification supervision. Theoretically, we show that, under distinct point scores, the full shared-order representation is invariant to shared point permutations and identifiable up to such permutations, avoiding the additional ambiguity induced by independent channel-wise sorting. Experiments on five 3D classification benchmarks demonstrate consistent improvements across a wide range of compression budgets. In the non-parameterized setting, CPDM achieves the highest mean accuracy among the compared methods in 12 of 15 dataset–budget configurations, while its parameterized variant outperforms 3DDP in 14 of 15 configurations. Cross-architecture evaluations and ablation studies further support the transferability and effectiveness of coherence-preserving matching.

1 INTRODUCTION

Dataset distillation aims to replace a large training set with a compact set of optimized synthetic samples while preserving its utility for training new models (Wang et al., 2018; Kim et al., 2022; Guo et al., 2024). Figure 1 provides a conceptual overview of this process for 3D point clouds, where only a few synthetic objects per class are optimized to represent the original training distribution (Zhang et al., 2024c; Yim et al., 2025; Kim et al., 2026). Among existing approaches, distribution matching is particularly efficient because it directly aligns feature distributions of real and synthetic data without differentiating through an unrolled training process (Zhao & Bilen, 2023; Wang et al., 2022; Zhang et al., 2024a). Its effectiveness, however, critically depends on whether the matching representation preserves the information relevant to the underlying data structure.

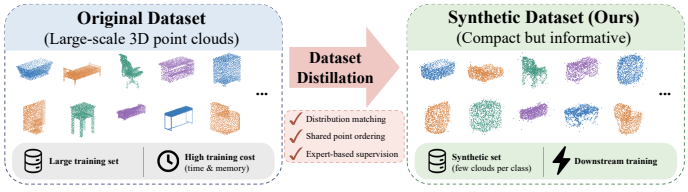


Figure 1: Conceptual overview of 3D point cloud dataset distillation via distribution matching.

Point clouds have no intrinsic point order. Consequently, directly comparing pointwise feature arrays by index can penalize different indexings of the same object. Semantically Aligned Distribution Matching (SADM) (Yim et al., 2025) addresses this issue by independently sorting pointwise feature responses within each channel before distribution matching. Although this removes dependence on point

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.

indexing, it introduces another ambiguity. After independent channel-wise sorting, responses occupying the same rank in different channels need not originate from the same point. The representation therefore preserves the response distribution of each individual channel while discarding which channel responses originally co-occurred at a physical point. We refer to this cross-channel association as *pointwise feature coherence*.

More formally, independent channel-wise sorting is invariant not only to a shared permutation of points, which is desirable for unordered point clouds, but also to independent point permutations within different feature channels. Consequently, feature maps containing different complete pointwise feature vectors can become identical after sorting. This reveals a mismatch between the invariance required by point clouds and the larger invariance induced by channel-wise sorting: a desirable representation should ignore *how points are indexed* without ignoring *which feature responses belong to the same point*.

To address this issue, we propose **Coherence-Preserving Distribution Matching (CPDM)**. CPDM derives a single importance score for each point by aggregating normalized responses across feature channels and uses the resulting shared ordering for all channels. Thus, every reordered feature vector still corresponds to one complete pointwise representation, preserving cross-channel coherence during distribution matching. The resulting coherence-preserving objective complements the channel-wise matching objective of SADM. We further use trained expert checkpoints for feature extraction and an auxiliary classification loss to provide task-relevant supervision. Theoretically, we show that, under distinct point scores, the resulting full representation is invariant to shared point permutations and identifiable up to such permutations, avoiding the additional independent-channel permutation ambiguity of channel-wise sorting.

We evaluate CPDM on five point cloud classification benchmarks under both direct and parameterized distillation settings. With PointNet evaluation, CPDM achieves the highest mean accuracy among the compared non-parameterized methods in 12 of 15 dataset–budget configurations. When combined with the shape-morphing parameterization of 3DDP (Kim et al., 2026), it improves over 3DDP in 14 of 15 configurations. Cross-architecture evaluations, component ablations, and representation diagnostics further examine the transferability of the distilled data and the contribution of coherence-preserving matching.

Our contributions are summarized as follows:

- We identify an overlooked limitation of channel-wise feature sorting for point cloud distribution matching: although it removes dependence on point indexing, it also introduces invariance to independent channel permutations, thereby discarding the cross-channel associations that define complete pointwise features.
- We propose Coherence-Preserving Distribution Matching (CPDM), which introduces a shared point ordering to preserve pointwise feature coherence while retaining permutation invariance. We further characterize its invariance and identifiability properties theoretically.
- We demonstrate the effectiveness and transferability of CPDM through extensive evaluations on five 3D classification benchmarks, covering multiple compression budgets, cross-architecture generalization, challenging ScanObjectNN variants, and both direct and parameterized distillation settings.

2 Related Work

Dataset Distillation and Distribution Matching. Dataset distillation compresses large training sets into compact synthetic sets while preserving training utility (Wang et al., 2018). Representative approaches match network gradients (Zhao et al., 2021a), training trajectories (Cazenavette et al., 2022), or feature distributions (Zhao & Bilen, 2023). Recent distribution-matching methods improve feature-extractor sampling (Zhao et al., 2023), incorporate attention alignment (Sajedi et al., 2023), or capture higher-order statistics through kernel-based matching (Zhang et al., 2024b). TGDD leverages pretrained expert trajectories for stage-wise feature matching and introduces an expert-guided distribution constraint (Ran et al., 2026). CPDM adopts related task-aware supervision, but focuses on preserving the pointwise structure of features during distribution matching.

Point Cloud Dataset Distillation. PCC adapts gradient matching to optimize synthetic point clouds (Zhang et al., 2024c), whereas SADM introduces channel-wise feature sorting and joint orientation optimization to accommodate the unordered nature of point clouds (Yim et al., 2025). DD3D addresses orientation and resolution variation using a rotator and point-wise generator (Bo & Wang, 2025), while recent work also studies long-tailed point cloud distillation (You et al., 2026). Complementarily, 3DDP improves storage efficiency through learnable shape morphing and uniformity-aware matching (Kim et al., 2026). Unlike these approaches, CPDM specifically addresses the loss of cross-channel pointwise associations induced by independent feature sorting, and is applicable to both direct point-cloud optimization and the parameterized setting of 3DDP.

3 METHODOLOGY

3.1 PROBLEM FORMULATION

Point Cloud Dataset Distillation. Let $\mathcal{D} = \{(X_i, y_i)\}_{i=1}^{M}$ denote the real training set, where $X_i \in \mathbb{R}^{N\times 3}$ is a point cloud with N points and $y_i \in \{1, \dots, C\}$ is its class label. Given a budget of m point clouds per class (PPC), dataset distillation aims to optimize a compact synthetic set $\mathcal{S} = \{(\widetilde{X}_j, \widetilde{y}_j)\}_{j=1}^{Cm}$, with $Cm \ll M$, such that a model trained on $\mathcal{S}$ generalizes similarly to one trained on $\mathcal{D}$. In the direct setting, each $\widetilde{X}_j \in \mathbb{R}^{N\times 3}$ is optimized through its point coordinates while its label remains fixed.

Distribution Matching. Let $F_\theta(X) \in \mathbb{R}^{d\times N}$ denote the pointwise features extracted by a network with parameters θ, and let $z_\theta(X)$ be a vector representation constructed from $F_\theta(X)$. Following distribution-matching-based point cloud distillation (Yim et al., 2025; Kim et al., 2026), we optimize synthetic data by matching class-conditional representation distributions:

$$\mathcal{L}_{\mathrm{DM}}(\mathcal{D}, \mathcal{S}; z) = \mathbb{E}_{\theta\sim p_\Theta}\left[\sum_{c=1}^{C} \mathrm{MMD}_k^2\big(z_\theta(\mathcal{D}_c), z_\theta(\mathcal{S}_c)\big)\right], \tag{1}$$

where $\mathcal{D}_c$ and $\mathcal{S}_c$ denote the real and synthetic samples of class c, respectively, p_Θ specifies the feature extractor distribution, and MMD_k is the empirical maximum mean discrepancy under kernel k (Gretton et al., 2012). The construction of z_θ is particularly important for point clouds because it must be insensitive to arbitrary point indexing.

SADM as the Base Objective. SADM (Yim et al., 2025) addresses point-order ambiguity by sorting feature responses independently within each channel. Let sort_N denote descending sorting along the point dimension. Its full and top-response representations are

$$z_\theta^{\mathrm{sort}}(X) = \mathrm{vec}(\mathrm{sort}_N(F_\theta(X))), \qquad z_\theta^{\max}(X) = \Big[\max_n F_\theta(X)_{a,n}\Big]_{a=1}^{d}. \tag{2}$$

The retained SADM matching loss is therefore

$$\mathcal{L}_{\mathrm{SADM}} = \alpha\, \mathcal{L}_{\mathrm{DM}}(\mathcal{D}, \mathcal{S}; z^{\mathrm{sort}}) + \beta\, \mathcal{L}_{\mathrm{DM}}(\mathcal{D}, \mathcal{S}; z^{\max}), \tag{3}$$

where $\alpha, \beta \geq 0$ weight the two terms.

Although channel-wise sorting removes dependence on point indexing, each channel is reordered independently. Consequently, responses appearing at the same rank across channels need not originate from the same input point, so the resulting representation can discard pointwise cross-channel associations. CPDM retains $\mathcal{L}_{\mathrm{SADM}}$ while introducing a shared-order matching term that preserves these associations.

3.2 COHERENCE-PRESERVING DISTRIBUTION MATCHING

We propose CPDM to complement the channel-wise order-statistic matching of SADM. Here, *pointwise feature coherence* refers to the association among feature responses originating from the same point. CPDM preserves this association by constructing a shared point order for all feature channels and matching the resulting coherent representations.

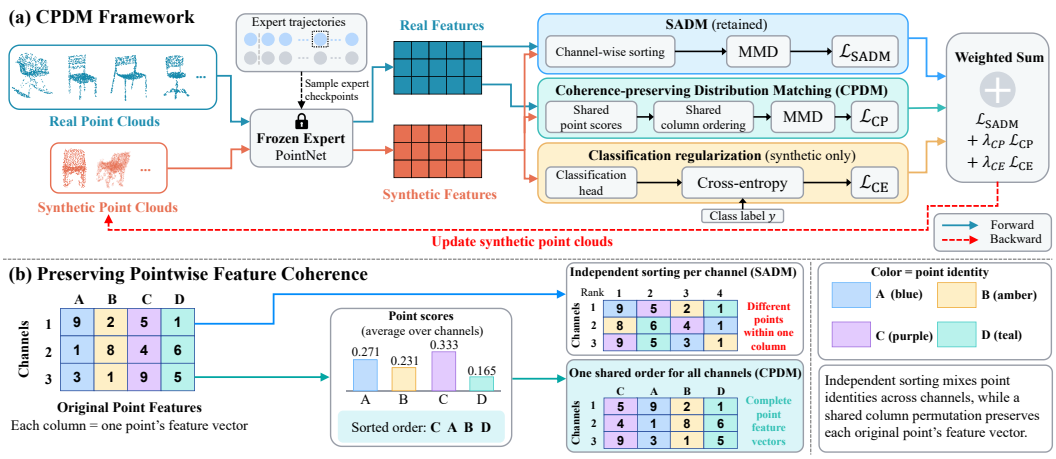


Figure 2: Overview of CPDM. A sampled expert provides features for channel-wise distribution matching and coherence-preserving matching with a shared point ordering. The synthetic data are optimized using both matching terms and an auxiliary classification loss.

Expert Feature Extraction. Instead of randomly initialized networks, we sample feature extractors from checkpoints collected along supervised expert trajectories, following trajectory-guided distribution matching (Ran et al., 2026). Let $\mathcal{E}$ denote the checkpoint buffer. At each distillation iteration, we uniformly sample $\theta \sim \mathcal{E}$ and freeze its parameters during the synthetic update. For a point cloud X, the extractor produces pointwise features $F_\theta(X) = F \in \mathbb{R}^{d\times N}$, where $F_{:,n}$ is the complete d-dimensional feature vector of point n.

Shared Point Ordering. To obtain a channel-balanced importance score, we standardize each feature channel over its points:

$$\overline{F}_{a,n} = \frac{F_{a,n} - \mu_a}{\sqrt{v_a + \varepsilon}}, \tag{4}$$

where μ_a and v_a are the mean and variance of channel a. We then normalize responses over points within each channel and average them across channels:

$$s_n = \frac{1}{d}\sum_{a=1}^{d} \frac{\exp(\overline{F}_{a,n}/\tau)}{\sum_{j=1}^{N}\exp(\overline{F}_{a,j}/\tau)}, \qquad \tau > 0. \tag{5}$$

Thus, s_n measures the average relative emphasis placed on point n across feature channels.

Let π be the descending order of $\{s_n\}_{n=1}^{N}$. Unlike SADM, which sorts each channel independently, CPDM applies this single permutation to the complete feature matrix:

$$\widehat{F}_\theta(X) = F_\theta(X)_{:,\pi}, \qquad z_\theta^{\text{cp}}(X) = \text{vec}\Big(\widehat{F}_\theta(X)\Big), \qquad z_\theta^{\text{cp},K}(X) = \text{vec}\Big(\widehat{F}_\theta(X)_{:,1:K}\Big). \tag{6}$$

Because the same order is shared by all channels, every column of $\widehat{F}_\theta(X)$ remains the complete feature vector of one point. The ranking is computed independently within each point cloud and does not assume that equal ranks across different objects correspond to the same semantic part.

Coherence-Preserving Matching. Using the class-conditional distribution-matching objective in Eq. 1, we define

$$\mathcal{L}_{\text{CP}} = \alpha\,\mathcal{L}_{\text{DM}}(\mathcal{D}, \mathcal{S}; z^{\text{cp}}) + \beta\,\mathcal{L}_{\text{DM}}(\mathcal{D}, \mathcal{S}; z^{\text{cp},K}), \tag{7}$$

where the full term preserves coherence over all points, while the top-K term further emphasizes points receiving high shared scores. We use the same α and β as in $\mathcal{L}_{\text{SADM}}$. During optimization, the discrete ordering is treated as fixed, while gradients propagate through the reordered synthetic features to the synthetic point coordinates.

3.3 Overall Optimization Objective

In addition to distribution matching, we use the sampled expert to provide direct class supervision for the synthetic point clouds. Let $g_\theta(X)$ denote its classification logits. We define

$$\mathcal{L}_{\mathrm{CE}} = \mathbb{E}_{\theta\sim\mathrm{Unif}(\mathcal{E})}\left[\sum_{c=1}^{C}\frac{1}{|\mathcal{S}_c|}\sum_{\widetilde{X}\in\mathcal{S}_c}\ell_{\mathrm{CE}}\big(g_\theta(\widetilde{X}),c\big)\right], \tag{8}$$

where the same sampled expert is used for feature matching and classification supervision.

The complete objective is

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{SADM}} + \lambda_{\mathrm{CP}}\mathcal{L}_{\mathrm{CP}} + \lambda_{\mathrm{CE}}\mathcal{L}_{\mathrm{CE}}, \tag{9}$$

where λ_{CP} and λ_{CE} control the contributions of coherence-preserving matching and classification supervision, respectively. For direct distillation, gradients of $\mathcal{L}_{\mathrm{total}}$ are used to optimize the synthetic point coordinates, while the expert parameters remain unoptimized.

Parameterized Setting. We further integrate CPDM with the learnable shape-morphing parameterization of 3DDP (Kim et al., 2026). Let $\mathcal{A}$ and $\mathcal{W}$ denote its coarse anchors and learnable mixing logits, respectively, and let $\mathcal{S}(\mathcal{A},\mathcal{W})$ denote the decoded synthetic set. After partitioning real point clouds to the corresponding resolution, we apply the same objective in Eq. 9 and optimize both $\mathcal{A}$ and $\mathcal{W}$. We adopt the shape-morphing parameterization of 3DDP but replace its original uniformity-aware matching objective with CPDM.

3.4 Theoretical Analysis

We characterize the invariances induced by channel-wise and shared-order representations. Let $F\in\mathbb{R}^{d\times N}$ be a pointwise feature map and $\mathcal{P}_N$ the group of point permutations. For a permutation-equivariant extractor, re-indexing a point cloud transforms its features as $F\mapsto FP$ for a single $P\in\mathcal{P}_N$ shared by all channels.

Independent-channel ambiguity. Let $\mathcal{T}_{\mathrm{sort}}$ denote the channel-wise sorting used by SADM. For arbitrary $P_1,\dots,P_d\in\mathcal{P}_N$,

$$F'_{a,:} = F_{a,:}P_a \quad\Longrightarrow\quad \mathcal{T}_{\mathrm{sort}}(F') = \mathcal{T}_{\mathrm{sort}}(F). \tag{10}$$

Thus, SADM is invariant to the product group $\mathcal{P}_N^d$, whereas valid point re-indexings correspond only to the diagonal subgroup

$$\Delta(\mathcal{P}_N) = \{(P,\dots,P) : P\in\mathcal{P}_N\} \subsetneq \mathcal{P}_N^d$$

for $d, N > 1$. This enlarged invariance discards the coupling among channel responses belonging to the same point, so feature maps with different pointwise feature vectors can become indistinguishable after channel-wise sorting.

Shared-order representation. Let P_F be the permutation induced by the shared scores in Eq. 5, and define the full CPDM representation as

$$\mathcal{C}(F) = FP_F. \tag{11}$$

When the point scores are distinct, the score function is permutation equivariant and the induced ordering satisfies $P_{FP} = P^{-1}P_F$. Consequently,

$$\mathcal{C}(FP) = \mathcal{C}(F), \qquad \mathcal{C}(F) = \mathcal{C}(G) \Longleftrightarrow G = FP \text{ for some } P\in\mathcal{P}_N. \tag{12}$$

Hence, the full CPDM representation removes the ambiguity due to point re-indexing without introducing the additional independent-channel ambiguity of SADM.

These results concern representation-level coherence rather than semantic correspondence between equally ranked points, and the identifiability claim applies to the full representation rather than the auxiliary top-K term. Detailed proofs, the treatment of tied scores, a counterexample, and the implication for kernel distribution matching are provided in Appendix A.

Table 1: PointNet classification accuracy (%) on five point cloud datasets. Baseline and Whole results are taken from 3DDP (Kim et al., 2026). PPC denotes point clouds per class. Whole denotes training on the full original training set. **OOM** denotes out of memory during distillation. Bold numbers indicate the highest mean accuracy within each DD group for each dataset–PPC configuration. CPDM† denotes CPDM equipped with the shape-morphing parameterization of 3DDP.

Dataset	PPC	Coreset Selection			DD w/o Param.						DD w/ Param.		Whole
		Random	Herding	K-Center	DM	DC	MTT	PCC	SADM	**CPDM**	3DDP	**CPDM†**	
ModelNet10	1	$28.1_{\pm 4.2}$	$34.0_{\pm 6.5}$	$34.0_{\pm 6.5}$	$25.8_{\pm 6.9}$	$32.8_{\pm 8.5}$	$27.8_{\pm 5.8}$	$33.0_{\pm 8.0}$	$35.9_{\pm 8.2}$	$\mathbf{37.3_{\pm 7.2}}$	$87.7_{\pm 0.7}$	$\mathbf{87.8_{\pm 0.5}}$	92.18
	3	$74.5_{\pm 1.8}$	$76.9_{\pm 1.2}$	$75.9_{\pm 1.8}$	$77.4_{\pm 1.2}$	$74.5_{\pm 2.6}$	$73.6_{\pm 1.7}$	$70.7_{\pm 1.6}$	$\mathbf{83.5_{\pm 0.7}}$	$79.9_{\pm 1.2}$	$89.8_{\pm 0.5}$	$\mathbf{91.9_{\pm 0.7}}$	
	10	$84.7_{\pm 0.7}$	$86.1_{\pm 0.7}$	$82.2_{\pm 1.5}$	$85.0_{\pm 0.7}$	$84.6_{\pm 0.6}$	$85.3_{\pm 1.2}$	$86.3_{\pm 1.1}$	$87.4_{\pm 1.1}$	$\mathbf{88.8_{\pm 0.9}}$	$92.2_{\pm 0.5}$	$\mathbf{92.6_{\pm 0.8}}$	
ModelNet40	1	$34.0_{\pm 2.1}$	$54.1_{\pm 2.1}$	$54.1_{\pm 2.1}$	$31.1_{\pm 4.7}$	$50.3_{\pm 2.0}$	$33.4_{\pm 2.1}$	$55.3_{\pm 1.4}$	$54.8_{\pm 1.3}$	$\mathbf{56.6_{\pm 1.4}}$	$73.2_{\pm 1.1}$	$\mathbf{79.7_{\pm 1.2}}$	88.78
	3	$59.9_{\pm 1.6}$	$69.1_{\pm 1.0}$	$62.1_{\pm 2.7}$	$61.5_{\pm 2.1}$	$66.0_{\pm 1.1}$	$59.5_{\pm 0.6}$	$66.2_{\pm 1.6}$	$71.3_{\pm 0.7}$	$\mathbf{74.1_{\pm 0.9}}$	$80.3_{\pm 0.5}$	$\mathbf{84.1_{\pm 0.7}}$	
	10	$73.3_{\pm 0.9}$	$77.6_{\pm 0.6}$	$64.3_{\pm 1.3}$	$74.9_{\pm 0.8}$	$74.3_{\pm 0.9}$	$73.4_{\pm 0.5}$	$77.9_{\pm 0.9}$	$79.6_{\pm 0.6}$	$\mathbf{82.7_{\pm 0.5}}$	$82.5_{\pm 0.6}$	$\mathbf{86.3_{\pm 0.6}}$	
ShapeNet	1	$33.5_{\pm 2.5}$	$49.1_{\pm 2.4}$	$49.1_{\pm 2.4}$	$26.3_{\pm 3.6}$	$48.7_{\pm 1.6}$	$32.4_{\pm 2.6}$	$50.9_{\pm 3.5}$	$51.1_{\pm 2.3}$	$\mathbf{54.0_{\pm 2.7}}$	$60.5_{\pm 1.1}$	$\mathbf{72.2_{\pm 0.5}}$	82.49
	3	$53.4_{\pm 1.4}$	$58.8_{\pm 1.0}$	$50.6_{\pm 1.6}$	$52.5_{\pm 1.6}$	$56.6_{\pm 1.1}$	$53.5_{\pm 2.0}$	$58.9_{\pm 1.7}$	$62.2_{\pm 1.6}$	$\mathbf{67.4_{\pm 1.5}}$	$65.9_{\pm 0.6}$	$\mathbf{76.0_{\pm 0.7}}$	
	10	$62.4_{\pm 0.9}$	$66.3_{\pm 0.4}$	$46.9_{\pm 0.7}$	$63.1_{\pm 0.8}$	$63.7_{\pm 0.8}$	$62.3_{\pm 1.1}$	$65.4_{\pm 0.8}$	$68.0_{\pm 0.5}$	$\mathbf{74.1_{\pm 0.7}}$	$68.9_{\pm 0.6}$	$\mathbf{77.4_{\pm 0.5}}$	
ScanObjectNN	1	$13.5_{\pm 1.8}$	$15.1_{\pm 1.7}$	$15.1_{\pm 1.7}$	$13.7_{\pm 1.8}$	$15.2_{\pm 2.0}$	$14.3_{\pm 2.5}$	$16.0_{\pm 2.4}$	$17.6_{\pm 1.5}$	$\mathbf{19.9_{\pm 2.1}}$	$32.6_{\pm 1.6}$	$\mathbf{39.9_{\pm 1.9}}$	63.43
	3	$19.7_{\pm 0.7}$	$26.9_{\pm 1.4}$	$18.8_{\pm 1.1}$	$26.4_{\pm 2.4}$	$24.6_{\pm 2.2}$	$20.1_{\pm 1.3}$	$25.5_{\pm 2.2}$	$\mathbf{32.6_{\pm 1.6}}$	$30.3_{\pm 1.1}$	$41.3_{\pm 1.1}$	$\mathbf{53.6_{\pm 0.9}}$	
	10	$34.1_{\pm 1.6}$	$38.3_{\pm 1.6}$	$23.5_{\pm 0.9}$	$37.4_{\pm 1.2}$	$38.5_{\pm 1.6}$	$37.1_{\pm 2.0}$	$34.6_{\pm 1.4}$	$43.7_{\pm 2.0}$	$\mathbf{49.6_{\pm 1.6}}$	$49.8_{\pm 0.7}$	$\mathbf{61.9_{\pm 0.8}}$	
OmniObject3D	1	$24.0_{\pm 0.6}$	$30.5_{\pm 1.0}$	$30.5_{\pm 1.0}$	$15.1_{\pm 0.9}$	$31.1_{\pm 0.5}$	$25.8_{\pm 1.3}$	$35.8_{\pm 0.6}$	$33.2_{\pm 0.4}$	$\mathbf{39.3_{\pm 0.8}}$	$41.9_{\pm 1.3}$	$\mathbf{62.7_{\pm 1.1}}$	74.98
	3	$40.9_{\pm 1.2}$	$42.3_{\pm 1.4}$	$35.9_{\pm 1.5}$	$40.3_{\pm 1.3}$	$44.3_{\pm 1.0}$	$44.8_{\pm 1.1}$	$46.2_{\pm 0.7}$	$51.6_{\pm 0.4}$	$\mathbf{57.2_{\pm 0.5}}$	$58.4_{\pm 1.5}$	$\mathbf{65.0_{\pm 0.9}}$	
	10	$59.5_{\pm 2.1}$	$59.3_{\pm 1.7}$	$58.3_{\pm 1.6}$	$60.0_{\pm 1.6}$	$56.5_{\pm 0.8}$	**OOM**	$57.0_{\pm 0.6}$	$\mathbf{65.2_{\pm 1.7}}$	$63.3_{\pm 1.3}$	$\mathbf{68.4_{\pm 1.2}}$	$66.5_{\pm 0.9}$	

4 Experiments

4.1 Experimental Setup

Datasets and Baselines. We evaluate CPDM on five point cloud classification benchmarks: ModelNet10 and ModelNet40 (Wu et al., 2015), ShapeNet (Chang et al., 2015), ScanObjectNN (Uy et al., 2019), and OmniObject3D (Wu et al., 2023). ModelNet10, ModelNet40, and ShapeNet contain 10, 40, and 55 object categories, respectively, while ScanObjectNN contains 15 categories of real-world scanned objects. We use PB_T50_RS as the default ScanObjectNN variant. Following the protocol of 3DDP (Kim et al., 2026), we retain 156 OmniObject3D categories and use an 80%/20% train–test split. All point clouds contain 1,024 points, and we evaluate PPC $\in \{1, 3, 10\}$.

We compare against three coreset methods: Random (Rebuffi et al., 2017), Herding (Castro et al., 2018), and K-Center (Sener & Savarese, 2018); three general dataset-distillation methods, DM (Zhao & Bilen, 2023), DC (Zhao et al., 2021a), and MTT (Cazenavette et al., 2022); and the point-cloud-specific methods PCC (Zhang et al., 2024c) and SADM (Yim et al., 2025). We additionally compare with 3DDP and evaluate CPDM combined with its shape-morphing parameterization. Baseline and full-dataset results in the main comparison are taken from 3DDP (Kim et al., 2026). Unless otherwise specified, PointNet is used for both distillation and evaluation.

Implementation and Evaluation. We optimize the synthetic data for 2,000 iterations using SGD with momentum 0.5 and initialize them from real point clouds. At each iteration, we sample a checkpoint from a buffer of trained PointNet experts and use its intermediate pointwise features for both $\mathcal{L}_{\mathrm{SADM}}$ and $\mathcal{L}_{\mathrm{CP}}$. We use the feature map before global max pooling, temperature $\tau = 1$, top-K value $K = 128$, and loss weights $\lambda_{\mathrm{CP}} = 1$ and $\lambda_{\mathrm{CE}} = 10^{-3}$. Both matching objectives use $\alpha = 0.002$ and $\beta = 0.001$. No data augmentation is used. For evaluation, each synthetic dataset is used to train 10 independently initialized networks from scratch, and we report the mean and standard deviation of test accuracy. Following 3DDP, evaluation networks are trained for 500 epochs with SGD using learning rate 0.01, momentum 0.9, weight decay 5×10^{-4}, and batch size 8. All CPDM experiments are conducted on a single NVIDIA GeForce RTX 4090 GPU. Additional implementation details are provided in Appendix B.

4.2 Performance Comparison

Evaluation on PointNet. Table 1 compares classification performance on five datasets. Without parameterization, CPDM achieves the best mean accuracy in 12 of the 15 dataset–PPC configura-

Table 2: Cross-architecture generalization performance (%) at PPC$=1$ in the non-parameterized setting. Distillation uses PointNet, and evaluation is performed on PointNet++ (PN++), PointConv (PC), and Point Transformer (PT).

Dataset	Arch.	Random	DM	DC	MTT	PCC	SADM	CPDM
ModelNet10	PN++	$22.4_{\pm 6.9}$	$12.1_{\pm 2.9}$	$15.3_{\pm 6.5}$	$20.4_{\pm 6.9}$	$20.7_{\pm 6.1}$	$\mathbf{25.9_{\pm 7.1}}$	$23.8_{\pm 7.9}$
	PC	$17.7_{\pm 10.1}$	$10.8_{\pm 3.6}$	$14.2_{\pm 4.7}$	$21.3_{\pm 7.6}$	$16.0_{\pm 8.7}$	$20.5_{\pm 13.1}$	$\mathbf{28.4_{\pm 9.1}}$
	PT	$44.1_{\pm 6.3}$	$22.4_{\pm 9.1}$	$26.7_{\pm 6.9}$	$39.3_{\pm 7.4}$	$45.9_{\pm 7.5}$	$49.0_{\pm 7.7}$	$\mathbf{49.6_{\pm 8.4}}$
ModelNet40	PN++	$36.8_{\pm 2.2}$	$1.5_{\pm 1.3}$	$8.6_{\pm 3.0}$	$37.2_{\pm 1.8}$	$13.5_{\pm 3.3}$	$40.0_{\pm 2.8}$	$\mathbf{44.7_{\pm 2.6}}$
	PC	$23.1_{\pm 3.8}$	$3.9_{\pm 1.9}$	$11.1_{\pm 3.4}$	$24.2_{\pm 4.6}$	$14.7_{\pm 3.8}$	$29.1_{\pm 3.0}$	$\mathbf{31.1_{\pm 3.9}}$
	PT	$28.9_{\pm 1.2}$	$6.2_{\pm 4.4}$	$14.8_{\pm 3.1}$	$29.0_{\pm 1.2}$	$40.2_{\pm 2.2}$	$44.5_{\pm 1.3}$	$\mathbf{46.3_{\pm 1.7}}$
ShapeNet	PN++	$25.3_{\pm 2.5}$	$2.0_{\pm 1.4}$	$7.2_{\pm 2.0}$	$24.6_{\pm 2.5}$	$21.8_{\pm 3.2}$	$35.1_{\pm 1.3}$	$\mathbf{36.5_{\pm 2.7}}$
	PC	$19.0_{\pm 3.1}$	$3.8_{\pm 1.0}$	$10.3_{\pm 3.4}$	$19.4_{\pm 3.9}$	$16.5_{\pm 4.4}$	$20.3_{\pm 5.0}$	$\mathbf{26.8_{\pm 4.7}}$
	PT	$26.3_{\pm 1.3}$	$7.4_{\pm 2.9}$	$19.1_{\pm 4.9}$	$26.3_{\pm 1.7}$	$38.3_{\pm 1.6}$	$36.3_{\pm 2.6}$	$\mathbf{43.0_{\pm 1.9}}$
ScanObjectNN	PN++	$18.0_{\pm 1.4}$	$14.5_{\pm 4.7}$	$15.8_{\pm 3.7}$	$\mathbf{18.8_{\pm 2.6}}$	$13.0_{\pm 3.6}$	$9.3_{\pm 2.1}$	$17.2_{\pm 3.1}$
	PC	$12.4_{\pm 2.2}$	$10.0_{\pm 2.1}$	$9.9_{\pm 1.8}$	$13.1_{\pm 1.8}$	$12.0_{\pm 2.5}$	$10.9_{\pm 3.4}$	$\mathbf{14.6_{\pm 2.9}}$
	PT	$12.5_{\pm 0.9}$	$10.2_{\pm 2.2}$	$12.3_{\pm 1.8}$	$12.2_{\pm 1.6}$	$\mathbf{16.2_{\pm 1.5}}$	$15.8_{\pm 2.0}$	$16.1_{\pm 1.8}$

tions. The gains are particularly clear on ModelNet40 and ShapeNet, where CPDM consistently improves over SADM across all PPC values. For example, at PPC$=10$, accuracy increases from 68.0% to 74.1% on ShapeNet and from 43.7% to 49.6% on ScanObjectNN. The three exceptions occur on ModelNet10 and ScanObjectNN at PPC$=3$, and OmniObject3D at PPC$=10$. When combined with the shape-morphing parameterization of 3DDP, CPDM† achieves higher mean accuracy than 3DDP in 14 of 15 configurations. The largest improvement is observed on OmniObject3D at PPC$=1$, where accuracy increases from 41.9% to 62.7%. Improvements are also obtained at every tested PPC on ShapeNet and ScanObjectNN. The only exception is OmniObject3D at PPC$=10$, where CPDM† achieves 66.5% compared with 68.4% for 3DDP. These results show that coherence-preserving matching benefits both direct and parameterized synthetic representations.

Cross-Architecture Generalization. We further evaluate non-parameterized synthetic datasets distilled with PointNet (Qi et al., 2017a) on PointNet++ (Qi et al., 2017b), PointConv (Wu et al., 2019), and Point Transformer (Zhao et al., 2021b) at PPC$=1$. As shown in Table 2, CPDM obtains the best mean accuracy in 9 of 12 dataset–architecture configurations and exceeds SADM in 11 of them. It ranks first across all three evaluation architectures on both ModelNet40 and ShapeNet; for example, on ShapeNet it improves the strongest competing result from 20.3% to 26.8% with PointConv and from 38.3% to 43.0% with Point Transformer. The consistent gains across multiple target architectures on these two datasets suggest that the distilled data capture transferable information rather than benefiting only the PointNet architecture used during distillation. The gains are less uniform on ScanObjectNN, where MTT and PCC remain stronger with PointNet++ and Point Transformer, respectively, while SADM slightly outperforms CPDM on ModelNet10 with PointNet++. Overall, CPDM transfers favorably beyond the PointNet architecture used for distillation, although the gain depends on the target architecture and dataset.

Evaluation on ScanObjectNN Variants. Table 3 further evaluates robustness across the four ScanObjectNN variants. CPDM achieves the best mean accuracy on three variants and improves over SADM by up to 3.2 percentage points, with the largest gain on PB_T50_R (19.9% vs. 16.7%). On PB_T25_R, CPDM obtains 17.8%, below SADM's 18.8%. Evaluating multiple variants provides a stricter test than relying only on the default PB_T50_RS setting and helps assess whether the observed improvement is specific to a particular ScanObjectNN configuration. The results show that the advantage generally extends across variants, although it is not universal.

Table 3: Comparison of classification accuracy (%) across four ScanObjectNN variants at PPC=1.

Variant	Random	Herding	K-Center	DM	SADM	CPDM
PB_T25	12.6	14.3	14.3	12.9	19.4	**21.1**
PB_T25_R	10.2	15.0	15.0	12.8	**18.8**	17.8
PB_T50_R	9.6	14.4	14.4	11.5	16.7	**19.9**
PB_T50_RS	13.5	15.1	15.1	13.7	17.6	**19.9**

Table 4: Component-wise ablation study of CPDM on ModelNet40 and ShapeNet. "Expert" denotes the use of trained expert checkpoints instead of randomly initialized feature extractors.

Method	Expert	$\mathcal{L}_{\text{SADM}}$	$\mathcal{L}_{\text{CP}}$	$\mathcal{L}_{\text{CE}}$	ModelNet40 PPC=1	ModelNet40 PPC=3	ModelNet40 PPC=10	ShapeNet PPC=1	ShapeNet PPC=3	ShapeNet PPC=10
SADM	×	✓	×	×	$54.8_{\pm 1.3}$	$71.3_{\pm 0.7}$	$79.6_{\pm 0.6}$	$51.1_{\pm 2.3}$	$62.2_{\pm 1.6}$	$68.0_{\pm 0.5}$
Expert-SADM	✓	✓	×	×	$55.5_{\pm 1.6}$	$70.0_{\pm 1.1}$	$80.2_{\pm 0.7}$	$52.2_{\pm 2.1}$	$65.5_{\pm 1.4}$	$70.0_{\pm 0.6}$
Expert-SADM + CP	✓	✓	✓	×	$56.1_{\pm 1.1}$	$72.7_{\pm 0.6}$	$81.1_{\pm 0.6}$	$53.0_{\pm 2.0}$	$66.8_{\pm 1.6}$	$72.6_{\pm 0.6}$
Expert-SADM + CE	✓	✓	×	✓	$\mathbf{57.1_{\pm 1.4}}$	$71.2_{\pm 0.8}$	$81.4_{\pm 0.4}$	$52.7_{\pm 2.3}$	$66.9_{\pm 1.4}$	$71.4_{\pm 0.7}$
CPDM	✓	✓	✓	✓	$56.6_{\pm 1.4}$	$\mathbf{74.1_{\pm 0.9}}$	$\mathbf{82.7_{\pm 0.5}}$	$\mathbf{54.0_{\pm 2.7}}$	$\mathbf{67.4_{\pm 1.5}}$	$\mathbf{74.1_{\pm 0.7}}$

4.3 Ablation Study

Component-wise Ablation. Table 4 disentangles the contributions of expert feature extraction, coherence-preserving matching, and classification supervision on ModelNet40 and ShapeNet, with $\mathcal{L}_{\text{SADM}}$ retained in all configurations. Replacing random feature extractors with trained experts improves the SADM reference in five of six settings. Adding $\mathcal{L}_{\text{CP}}$ to Expert-SADM further improves all six configurations by 0.6–2.7 percentage points, demonstrating a benefit from coherence-preserving matching alone. More importantly, CPDM also exceeds Expert-SADM + CE in five of six settings, showing that $\mathcal{L}_{\text{CP}}$ contributes beyond expert features and classification supervision. Conversely, adding $\mathcal{L}_{\text{CE}}$ to Expert-SADM + CP improves all six settings, and the full CPDM achieves the highest mean accuracy in five of them. Together, these results support the complementary contributions of coherence-preserving matching and classification supervision.

Sensitivity to Cross-Channel Incoherence. We directly test whether the matching representations detect disruptions of pointwise cross-channel associations. Using features extracted by a fixed trained PointNet expert on ModelNet40, we independently permute point indices within an increasing fraction $r \in \{0.1, 0.2, \ldots, 1.0\}$ of feature channels while preserving the values within every channel, and measure the normalized Euclidean discrepancy between the original and perturbed representations. As shown in Figure 3, SADM yields exactly zero discrepancy because independent channel-wise sorting eliminates such permutations. In contrast, the shared-order representation of CPDM produces positive discrepancy at every tested corruption level, which increases as more channels are perturbed. This diagnostic empirically confirms that CPDM retains cross-channel associations that are invisible to channel-wise sorting, consistent with the theoretical analysis in Section 3.4. Additional hyperparameter analyses are provided in Appendix C.1.

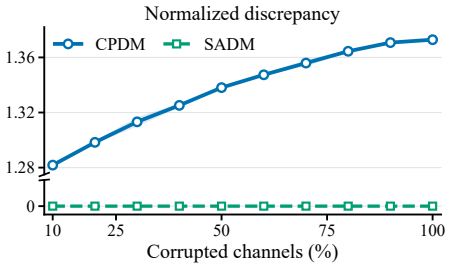


Figure 3: Relative representation discrepancy under independent within-channel point permutations on ModelNet40.

Effect of Distillation Backbones. Table 5 examines whether the effectiveness of CPDM depends on the backbone used for distillation. On ModelNet10 at PPC$=1$, we perform distillation using Point Transformer, PointConv, PointNet++, DGCNN (Wang et al., 2019), and PointNet, and evaluate the resulting synthetic datasets on four architectures. Across the 20 distillation–evaluation combinations, CPDM achieves higher mean accuracy than SADM in 16 cases. In particular, when Point Transformer or PointConv is used for distillation, CPDM improves over SADM across all four evaluation architectures, with the largest gain observed for PointConv-to-Point Transformer (21.6% to 47.9%). Among the CPDM configurations, PointNet yields the highest mean accuracy for three of the four evaluation architectures. These results indi-

Table 5: Classification accuracy (%) on ModelNet10 at PPC$=1$ with different distillation (D) and evaluation (E) backbones. Each pair reports SADM (unshaded) and CPDM (shaded).

$D \backslash E$	PT		PC		PN++		PN	
PT	35.6	**47.7**	17.1	**24.4**	12.4	**23.7**	24.7	**31.6**
PC	21.6	**47.9**	11.1	**13.4**	15.5	**21.4**	16.8	**28.5**
PN++	44.2	**45.1**	**21.8**	18.6	11.5	**21.4**	33.8	**34.7**
DGCNN	**50.7**	50.4	20.9	**22.7**	17.2	**21.4**	**38.5**	35.3
PN	49.0	**49.6**	20.5	**28.4**	**25.9**	23.8	35.9	**37.3**

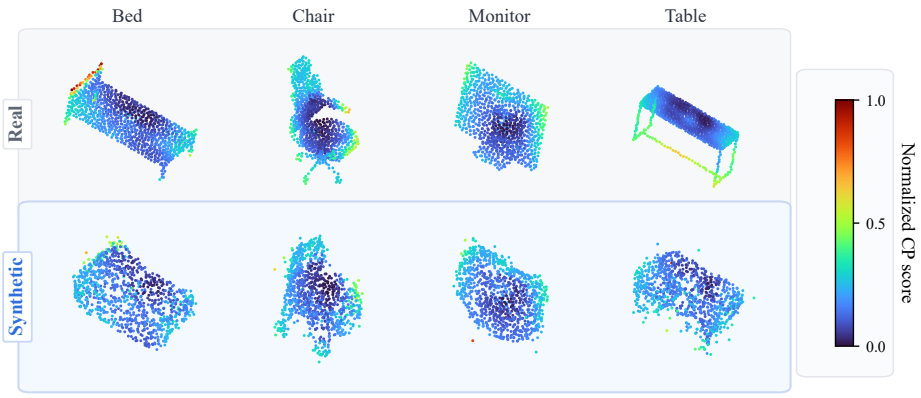


Figure 4: Visualization of shared point scores on ModelNet10. The top and bottom rows show real and synthetic samples, respectively. Scores are computed by aggregating normalized feature responses across channels according to Eq. 5. For visualization only, the scores are min-max normalized within each point cloud.

cate that the benefit of coherence-preserving matching is not restricted to PointNet-based distillation, although its magnitude depends on the distillation and evaluation architectures.

4.4 Visualization of Shared Point Scores

Figure 4 visualizes the shared point scores defined in Eq. 5 for real and CPDM-distilled samples from four ModelNet10 categories. The scores exhibit clear spatial variation, with relatively high responses frequently appearing around selected boundaries or structural components. Importantly, CPDM does not assume that equally ranked points from different objects correspond to the same semantic part. Instead, the scores define a single ordering within each point cloud that is shared by all feature channels, thereby preserving complete pointwise feature vectors during matching. Additional visualizations of distilled point clouds and feature distributions are provided in Appendix C.2.

5 Conclusion and Discussion

We presented Coherence-Preserving Distribution Matching (CPDM) for 3D point cloud dataset distillation. CPDM complements channel-wise distribution matching with a shared point ordering that preserves the cross-channel associations within each point's feature vector. Trained expert checkpoints provide feature representations and auxiliary classification supervision. Our theoretical analysis shows that, when point scores are distinct, the full shared-order representation is invariant to shared point permutations and identifiable up to such permutations, avoiding the additional ambiguity introduced by independent channel-wise sorting. Experiments on five classification benchmarks demonstrate improved accuracy in most tested configurations, both with direct synthetic representations and when combined with the shape-morphing parameterization of 3DDP. Cross-architecture evaluations and component ablations further provide evidence of transferability and the contribution of coherence-preserving matching beyond expert features and classification supervision alone.

Limitations and Future Work. Shared point ordering preserves within-point feature associations but does not guarantee semantic correspondence between equally ranked points from different objects. The theoretical results concern the full feature representation, rather than guaranteeing downstream classification performance. In addition, CPDM requires offline expert training and checkpoint storage, whose costs are separate from the storage budget of the final distilled dataset. Our evaluation is currently limited to object-level classification; its effectiveness for point-level tasks such as part segmentation remains to be investigated. Reducing expert-buffer preparation costs and extending coherence-preserving matching to dense prediction are directions for future work.

AI USE STATEMENT

Generative AI tools were used to assist with research ideation and execution, including brainstorming and refining methodological and experimental choices. They were also used to draft portions of the manuscript and to improve the clarity, grammar, and presentation of the writing. All AI-assisted research ideas, technical suggestions, and written content were critically reviewed, revised, and verified by the authors. The authors take full responsibility for the methodology, experimental results, claims, and final content of this work.

REPRODUCIBILITY STATEMENT

The formulation of CPDM and its optimization objectives are presented in Sections 3.1–3.3, with the assumptions and proofs supporting the theoretical claims provided in Appendix A. Algorithm 1 summarizes the distillation procedure. Section 4.1 and Appendix B describe the datasets, synthetic-data initialization, expert-buffer construction and checkpoint sampling, optimization settings, shape-morphing parameterization, and evaluation protocol. The default hyperparameters are summarized in Table 6. As detailed in Appendix B.5, each distilled dataset is evaluated by training ten independently initialized networks from scratch, and the mean and standard deviation of test accuracy are reported. Baseline and full-dataset results in Table 1 are taken from 3DDP (Kim et al., 2026).

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, 2025.

Francisco M. Castro, Manuel J. Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European Conference on Computer Vision*, pp. 233–248, 2018.

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*, pp. 10718–10727, 2022. doi: 10.1109/CVPR52688.2022.01045.

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An information-rich 3D model repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

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*, 2024.

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.

Jang-Hyun Kim, Jinuk Kim, Seong Joon Oh, Sangdoo Yun, Hwanjun Song, Joonhyun Jeong, Jung-Woo Ha, and Hyun Oh Song. Dataset condensation via efficient synthetic-data parameterization. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 11102–11118, 2022.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660, 2017a. doi: 10.1109/CVPR.2017.16.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5099–5108, 2017b.

Fengli Ran, Xiao Pu, Bo Liu, Xiuli Bi, and Bin Xiao. TGDD: Trajectory guided dataset distillation with balanced distribution. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(18): 15662–15670, 2026. doi: 10.1609/aaai.v40i18.38596.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017. doi: 10.1109/CVPR.2017.587.

Ahmad Sajedi, Samir Khaki, Ehsan Amjadian, Lucy Z. Liu, Yuri A. Lawryshyn, and Konstantinos N. Plataniotis. DataDAM: Efficient dataset distillation with attention matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17097–17107, 2023.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

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*, pp. 1588–1597, 2019.

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 (CVPR)*, pp. 12196–12205, 2022.

Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.

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.

Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, Dahua Lin, and Ziwei Liu. OmniObject3D: Large-vocabulary 3D object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 803–814, 2023. doi: 10.1109/CVPR52729.2023.00084.

Wenxuan Wu, Zhongang Qi, and Fuxin Li. PointConv: Deep convolutional networks on 3D point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9621–9630, 2019.

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

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*, volume 38, 2025. doi: 10.52202/085713-2477.

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

Hansong Zhang, Shikun Li, Fanzhao Lin, Weiping Wang, Zhenxing Qian, and Shiming Ge. DANCE: Dual-view distribution alignment for dataset condensation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2024a.

Hansong Zhang, Shikun Li, Pengju Wang, Dan Zeng, and Shiming Ge. M3D: Dataset condensation by minimizing maximum mean discrepancy. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(8):9314–9322, 2024b. doi: 10.1609/aaai.v38i8.28784.

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*, pp. 6384–6393, 2024c. doi: 10.1145/3664647.3680767.

Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 6514–6523, 2023.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *International Conference on Learning Representations*, 2021a.

Ganlong Zhao, Guanbin Li, Yipeng Qin, and Yizhou Yu. Improved distribution matching for dataset condensation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7856–7865, 2023. doi: 10.1109/CVPR52729.2023.00759.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H. S. Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021b. doi: 10.1109/ICCV48922.2021.01595.

A Theoretical Analysis of Feature Coherence

This appendix provides a detailed theoretical analysis of why channel-wise feature sorting is insufficient for preserving pointwise feature coherence and how the full CPDM representation resolves the resulting ambiguity. The key observation is that the unordered nature of a point cloud requires invariance to a *shared* permutation of points, whereas channel-wise sorting is invariant to a substantially larger class of transformations that independently permute points in different feature channels.

A.1 Independent-Channel Permutation Ambiguity of SADM

Let

$$F = [\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_N] \in \mathbb{R}^{d\times N}, \tag{13}$$

denote a pointwise feature map, where $\boldsymbol{f}_n \in \mathbb{R}^d$ is the complete feature vector associated with the n-th point. We consider pointwise features extracted before permutation-invariant global aggregation. For a permutation-equivariant pointwise feature extractor, re-indexing the input points induces the same permutation on the point dimension of the feature map.

Let $\mathcal{P}_N$ denote the group of $N \times N$ permutation matrices. For any $P \in \mathcal{P}_N$, a valid re-indexing of the point cloud therefore acts on the feature map as

$$F \mapsto FP. \tag{14}$$

Importantly, the same permutation P is shared by all feature channels, since all channel responses belonging to one point constitute a single pointwise feature vector. Hence, the intrinsic unorderedness of point clouds requires invariance only to the shared permutation group $\mathcal{P}_N$.

SADM instead sorts the point responses independently within every feature channel. Let $\mathcal{T}_{\mathrm{sort}}(\cdot)$ denote this channel-wise sorting operator:

$$[\mathcal{T}_{\mathrm{sort}}(F)]_{a,:} = \mathrm{sort}_{\downarrow}(F_{a,:}), \qquad a = 1, \ldots, d. \tag{15}$$

Consider arbitrary permutation matrices $P_1, \ldots, P_d \in \mathcal{P}_N$, and construct another feature map F' as

$$F'_{a,:} = F_{a,:}P_a, \qquad a = 1, \ldots, d. \tag{16}$$

Since sorting eliminates the original ordering within each channel,

$$\mathcal{T}_{\mathrm{sort}}(F') = \mathcal{T}_{\mathrm{sort}}(F) \qquad \forall\, (P_1, \ldots, P_d) \in \mathcal{P}_N^d. \tag{17}$$

Therefore, SADM is invariant not only to the desired shared point permutation, but also to the product group $\mathcal{P}_N^d$ of independent channel-wise permutations. Shared permutations correspond to the diagonal subgroup $\Delta(\mathcal{P}_N)$, which is isomorphic to $\mathcal{P}_N$. For $d > 1$ and $N > 1$,

$$\Delta(\mathcal{P}_N) = \{(P, \ldots, P) : P \in \mathcal{P}_N\} \subsetneq \mathcal{P}_N^d. \tag{18}$$

This enlarged invariance induces an information loss. In particular, channel-wise sorting preserves the empirical marginal distribution of each feature channel,

$$\widehat{\mu}_F^{(a)} = \frac{1}{N}\sum_{n=1}^{N} \delta_{F_{a,n}}, \qquad a = 1, \ldots, d, \tag{19}$$

but discards their cross-channel coupling within individual points. The complete pointwise feature distribution is instead

$$\widehat{\mu}_F = \frac{1}{N}\sum_{n=1}^{N} \delta_{\boldsymbol{f}_n}, \qquad \boldsymbol{f}_n \in \mathbb{R}^d, \tag{20}$$

which explicitly records which channel responses co-occur at the same point. Consequently, identical channel-wise marginals do not imply identical pointwise feature distributions.

A simple counterexample illustrates this ambiguity. Consider

$$F = \begin{bmatrix} 3 & 1 & 0 \\ 3 & 2 & 0 \end{bmatrix}, \qquad F' = \begin{bmatrix} 3 & 1 & 0 \\ 0 & 3 & 2 \end{bmatrix}. \tag{21}$$

The pointwise feature vectors of F are

$$\{(3,3),(1,2),(0,0)\}, \tag{22}$$

whereas those of F' are

$$\{(3,0),(1,3),(0,2)\}. \tag{23}$$

Thus, there exists no shared permutation $P \in \mathcal{P}_3$ such that $F' = FP$. Nevertheless,

$$\mathcal{T}_{\text{sort}}(F) = \mathcal{T}_{\text{sort}}(F') = \begin{bmatrix} 3 & 1 & 0 \\ 3 & 2 & 0 \end{bmatrix}. \tag{24}$$

Hence, the two feature maps are indistinguishable after channel-wise sorting, despite having different pointwise feature compositions.

A.2 Coherence-Preserving Shared Point Ordering

CPDM avoids the above ambiguity by deriving a single importance score for each point from all feature channels and applying the resulting ordering jointly to the complete pointwise feature vectors.

For completeness, we restate the shared point-score construction. For each channel a, the feature responses are first normalized across the point dimension:

$$\overline{F}_{a,n} = \frac{F_{a,n} - \mu_a}{\sqrt{v_a + \varepsilon}}, \tag{25}$$

where

$$\mu_a = \frac{1}{N}\sum_{n=1}^{N} F_{a,n}, \qquad v_a = \frac{1}{N}\sum_{n=1}^{N}(F_{a,n} - \mu_a)^2. \tag{26}$$

We then normalize the responses within each channel using a temperature $\tau > 0$:

$$A_{a,n} = \frac{\exp(\overline{F}_{a,n}/\tau)}{\sum_{m=1}^{N}\exp(\overline{F}_{a,m}/\tau)}. \tag{27}$$

The shared importance score of the n-th point is defined by aggregating its responses across all channels:

$$s_n(F) = \frac{1}{d}\sum_{a=1}^{d} A_{a,n}. \tag{28}$$

Let

$$\boldsymbol{s}(F) = [s_1(F), \ldots, s_N(F)]^\top \in \mathbb{R}^N. \tag{29}$$

Let $P_F \in \mathcal{P}_N$ denote the permutation matrix induced by sorting $\boldsymbol{s}(F)$ in descending order. Equivalently, $P_F^\top \boldsymbol{s}(F)$ is monotonically non-increasing. We define the coherence-preserving representation as

$$\mathcal{C}(F) = FP_F. \tag{30}$$

Unlike channel-wise sorting, this operation never separates channel responses belonging to the same point. Each complete feature vector $\boldsymbol{f}_n$ is moved as a single unit.

We first establish the permutation equivariance of the proposed score function. Since the mean and variance in Eq. 25 are invariant to point re-indexing, channel-wise normalization commutes with any shared point permutation. Softmax over the point dimension is permutation equivariant, and averaging across channels commutes with the same point permutation. Therefore,

$$\boldsymbol{s}(FP) = P^\top \boldsymbol{s}(F), \qquad \forall P \in \mathcal{P}_N. \tag{31}$$

Proposition 1 (Permutation invariance)**.** *Assume that the entries of $\boldsymbol{s}(F)$ are distinct. Then the coherence-preserving representation is invariant to arbitrary re-indexing of points:*

$$\mathcal{C}(FP) = \mathcal{C}(F), \qquad \forall P \in \mathcal{P}_N. \tag{32}$$

Proof. From Eq. 31, applying P to the point dimension only permutes the entries of $\boldsymbol{s}(F)$. Since the scores are distinct, their descending order is unique. Therefore, the sorting permutation associated with FP satisfies

$$P_{FP} = P^{-1}P_F. \tag{33}$$

It follows that

$$\mathcal{C}(FP) = FPP_{FP} = FPP^{-1}P_F = FP_F = \mathcal{C}(F). \tag{34}$$

□

Remark on tied scores. The distinct-score assumption ensures that the shared point ordering defines a unique canonical permutation. When ties occur, several orderings within a tied group are equally compatible with the score vector, and strict canonical invariance depends on the tie-breaking rule. The invariance statements therefore assume distinct scores. Importantly, the central distinction from SADM remains unchanged: CPDM always applies one shared point permutation to complete feature vectors, rather than allowing independent permutations across feature channels.

We next show that the shared ordering eliminates exactly the intrinsic point permutation ambiguity without introducing the excessive channel-wise ambiguity of SADM.

Proposition 2 (Identifiability up to point permutation)**.** *Assume that the point scores of F and G contain no ties. Then two feature maps have identical coherence-preserving representations if and only if they differ by a valid shared point permutation:*

$$\mathcal{C}(F) = \mathcal{C}(G) \quad \Longleftrightarrow \quad G = FP \quad \textit{for some } P \in \mathcal{P}_N. \tag{35}$$

Proof. We first prove the reverse direction. If $G = FP$ for some $P \in \mathcal{P}_N$, Proposition 1 immediately gives

$$\mathcal{C}(G) = \mathcal{C}(FP) = \mathcal{C}(F). \tag{36}$$

For the forward direction, suppose

$$\mathcal{C}(F) = \mathcal{C}(G). \tag{37}$$

By the definition in Eq. 30,

$$FP_F = GP_G. \tag{38}$$

Since P_F and P_G are permutation matrices,

$$G = FP_FP_G^{-1}. \tag{39}$$

The product $P_FP_G^{-1}$ is itself a permutation matrix. Defining

$$P = P_FP_G^{-1} \in \mathcal{P}_N \tag{40}$$

therefore yields

$$G = FP. \tag{41}$$

□

Proposition 2 highlights the fundamental difference between SADM and CPDM. SADM identifies feature maps related by independent channel-wise permutations in $\mathcal{P}_N^d$, whereas CPDM identifies only feature maps related by the intrinsic shared point permutation in $\mathcal{P}_N$. In this sense, CPDM removes the ambiguity caused by arbitrary point indexing while retaining the cross-channel coupling that constitutes complete pointwise feature vectors.

A.3 Implication for Distribution Matching

We examine how the representation difference affects distribution matching. Let k be a Gaussian kernel on vector representations:

$$k(\boldsymbol{u}, \boldsymbol{v}) = \exp\left(-\gamma\|\boldsymbol{u} - \boldsymbol{v}\|_2^2\right), \qquad \gamma > 0. \tag{42}$$

For a characteristic kernel, zero population MMD implies equality of the distributions in its input representation space (Gretton et al., 2012). This property also applies when comparing two empirical probability measures using their exact kernel MMD. However, it cannot recover information already discarded by the representation mapping.

For the counterexample in Eq. 21, define $\boldsymbol{v} = \mathrm{vec}(\mathcal{T}_{\mathrm{sort}}(F))$ and $\boldsymbol{v}' = \mathrm{vec}(\mathcal{T}_{\mathrm{sort}}(F'))$. Equation 24 gives $\boldsymbol{v} = \boldsymbol{v}'$, although F and F' are not related by a shared point permutation. The corresponding singleton empirical distributions satisfy

$$\mathrm{MMD}_k^2(\delta_{\boldsymbol{v}}, \delta_{\boldsymbol{v}'}) = 0. \tag{43}$$

No kernel operating solely on these sorted representations can distinguish the pair, because the representation mapping has already identified them.

In contrast, equality of the full shared-order representations would imply $FP_F = F'P_{F'}$, or $F' = FP_FP_{F'}^{-1}$. The product $P_FP_{F'}^{-1}$ is a shared point permutation, contradicting the construction of the counterexample. Hence $\mathcal{C}(F) \neq \mathcal{C}(F')$. This implication does not require distinct scores: for any shared sorting permutations, equality of the reordered matrices implies equality up to a shared permutation. Distinct scores are needed for the invariance and reverse implication in the preceding propositions.

Define $\boldsymbol{u} = \mathrm{vec}(\mathcal{C}(F))$ and $\boldsymbol{u}' = \mathrm{vec}(\mathcal{C}(F'))$. Their squared Euclidean distance is

$$\|\boldsymbol{u} - \boldsymbol{u}'\|_2^2 = \|\mathcal{C}(F) - \mathcal{C}(F')\|_F^2, \tag{44}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Since $k(\boldsymbol{u}, \boldsymbol{u}) = k(\boldsymbol{u}', \boldsymbol{u}') = 1$, the Gaussian-kernel MMD between the singleton empirical distributions is

$$\mathrm{MMD}_k^2(\delta_{\boldsymbol{u}}, \delta_{\boldsymbol{u}'}) = 2\left[1 - \exp\left(-\gamma\|\boldsymbol{u} - \boldsymbol{u}'\|_2^2\right)\right] > 0. \tag{45}$$

Thus, these feature maps remain distinguishable in the full CPDM representation space.

More generally, matching distributions of channel-wise sorted feature maps constrains the distributions of their order statistics, but cannot recover the cross-channel association within individual points. The full CPDM representation retains these associations under a shared ordering and, under distinct scores, is identifiable up to point re-indexing. Consequently, it can distinguish a class of feature-map pairs that channel-wise sorting necessarily maps to identical representations.

The analysis concerns feature-map representations, not the invertibility of the feature extractor or exact semantic correspondence between equally ranked points from different objects. It also does not establish that discrepancy must increase monotonically with the amount of an independent-channel permutation perturbation. Such quantitative behavior must be assessed empirically.

Finally, the identifiability result applies to the full representation in Eq. 30, not to its top-K subset alone. The top-K component is auxiliary: it emphasizes high-importance points while preserving their complete feature vectors, and is combined with rather than substituted for the full CPDM term.

B Additional Experimental Details

B.1 Pseudocode of CPDM

Algorithm 1 summarizes the complete optimization procedure of CPDM. At each iteration, a trained expert checkpoint is sampled and used to extract pointwise features for both the retained SADM objective and the proposed coherence-preserving matching objective, while also providing auxiliary classification supervision for the synthetic point clouds. The expert parameters remain frozen throughout distillation, and only the synthetic data are updated according to the combined objective in Eq. 9.

B.2 Implementation Details

Table 6 summarizes the default hyperparameters used throughout our experiments. The same CPDM-specific settings are shared across datasets and compression budgets unless otherwise stated, while dataset-specific settings for the 3DDP parameterization are provided in Section B.4.

Unless otherwise specified, PointNet is used as the distillation backbone with full-resolution point clouds of 1,024 points. Synthetic point clouds are initialized from randomly sampled real training examples and optimized for 2,000 iterations using SGD with a learning rate of 10 and momentum of 0.5. At each iteration, we sample eight real point clouds per class, while all synthetic samples of that class participate in the matching objective. No data augmentation is used during distillation.

Algorithm 1 Coherence-Preserving Distribution Matching (CPDM)

Require: Real dataset $\mathcal{D}$, expert checkpoint buffer $\mathcal{E}$, PPC m, iterations T
Require: λ_{CP}, λ_{CE}, temperature τ, coherent top-K
Ensure: Distilled synthetic dataset $\mathcal{S}$

Initialize $\mathcal{S}$ from randomly sampled real point clouds → synthetic initialization
for $t = 1, \ldots, T$ **do**
  Sample a trained checkpoint $\theta \sim \mathrm{Unif}(\mathcal{E})$ and freeze θ → expert feature extractor
  if f_θ contains BatchNorm layers **then**
    Calibrate BatchNorm statistics using a class-balanced real batch, then fix them
  end if
  $\mathcal{L} \leftarrow 0$
  for each class $c = 1, \ldots, C$ **do**
    Sample real batch $\mathcal{B}_c^r \subset \mathcal{D}_c$ and obtain synthetic batch $\mathcal{B}_c^s = \mathcal{S}_c$
    Extract pointwise features $F^r \in \mathbb{R}^{B_r \times d \times N}$, $F^s \in \mathbb{R}^{B_s \times d \times N}$ using f_θ → shared expert features

    Channel-wise distribution matching
    $\bar{F}^r \leftarrow \mathrm{sort}_N(F^r)$, $\bar{F}^s \leftarrow \mathrm{sort}_N(F^s)$ → independent channel sorting
    Compute $\mathcal{L}_{\mathrm{SADM}}$ from the full sorted features and their top responses using Gaussian-kernel MMD → retained SADM objective

    Coherence-preserving distribution matching
    for $F \in \{F^r, F^s\}$ **do**
      Standardize each channel over the point dimension

$$\widetilde{F}_{a,n} = \frac{F_{a,n} - \mu_a}{\sqrt{v_a + \epsilon}}$$

      Compute shared point scores

$$s_n = \frac{1}{d} \sum_{a=1}^{d} \frac{\exp(\widetilde{F}_{a,n}/\tau)}{\sum_{j=1}^{N} \exp(\widetilde{F}_{a,j}/\tau)}$$

      $\pi \leftarrow \mathrm{argsort}(s, \mathrm{descending})$ → one shared point order
      $\widehat{F} \leftarrow F_{:,\pi}$ → preserve complete point features
    end for
    Compute $\mathcal{L}_{\mathrm{CP}}$ using MMD on both $\widehat{F}$ and $\widehat{F}_{:,1:K}$ → full + top-K coherent matching
    Compute $\mathcal{L}_{\mathrm{CE}} = \ell_{\mathrm{CE}}(g_\theta(\mathcal{B}_c^s), c)$ → task-aware supervision
    $\mathcal{L} \leftarrow \mathcal{L} + \mathcal{L}_{\mathrm{SADM}} + \lambda_{\mathrm{CP}}\mathcal{L}_{\mathrm{CP}} + \lambda_{\mathrm{CE}}\mathcal{L}_{\mathrm{CE}}$
  end for
  Update $\mathcal{S}$ by back-propagating $\nabla_{\mathcal{S}}\mathcal{L}$ while keeping θ frozen → optimize synthetic point clouds
end for
return $\mathcal{S}$

Pointwise features are extracted from the intermediate PointNet feature map before global max pooling. For the channel-wise SADM term, we use the full sorted representation together with the highest-ranked response of each channel, with $\alpha = 0.002$, $\beta = 0.001$, and top-1 matching. For coherence-preserving matching, shared point scores are computed with temperature $\tau = 1$, and the auxiliary truncated representation retains the top $K = 128$ points. We set

$$\lambda_{\mathrm{CP}} = 1, \qquad \lambda_{\mathrm{CE}} = 10^{-3}. \tag{46}$$

Both $\mathcal{L}_{\mathrm{SADM}}$ and $\mathcal{L}_{\mathrm{CP}}$ use the same Gaussian-kernel MMD implementation. Before backpropagation, the class-summed objective is scaled by the PPC value, with the same factor applied to all loss components.

B.3 Expert Checkpoint Buffer

We construct an offline buffer of supervised PointNet expert trajectories for each dataset. The experts are independently initialized and trained on the original training set using cross-entropy supervision. We use 10 expert trajectories and retain the checkpoint after every training epoch. The initialization checkpoint is excluded during distillation, leaving the trained checkpoints eligible for feature extraction.

Table 6: Default hyperparameters used for CPDM distillation and evaluation.

Setting	Value
Distillation backbone	PointNet
Distillation iterations	2,000
Synthetic optimizer	SGD
Synthetic learning rate	10
Synthetic momentum	0.5
Real batch size / class	8
Feature layer	`x_m`
Expert trajectories	10
Point-score temperature τ	1
Coherent top-K	128
α, β	0.002, 0.001
λ_{CP}	1
λ_{CE}	10^{-3}
Data augmentation	None
Evaluation runs	10
Evaluation epochs	500
Evaluation batch size	8

At every distillation iteration, one checkpoint is sampled uniformly from the pooled set of eligible expert checkpoints. Its learnable parameters remain frozen throughout the corresponding synthetic-data update, and the same checkpoint is used to compute $\mathcal{L}_{\mathrm{SADM}}$, $\mathcal{L}_{\mathrm{CP}}$, and $\mathcal{L}_{\mathrm{CE}}$. The expert networks are trained with SGD using learning rate 0.01, momentum 0.9, weight decay 5×10^{-4}, and batch size 8, without data augmentation. The learning rate is reduced by a factor of 0.1 approximately halfway through expert training.

Because PointNet contains BatchNorm layers, directly loading checkpoints with stale running statistics can introduce an additional discrepancy between real and synthetic forward passes. After loading a checkpoint, we therefore perform a BatchNorm calibration forward pass using a class-balanced real-data batch containing 8 samples per class. The BatchNorm running statistics are updated during this pass and are then fixed for the real and synthetic forward passes used to compute the distillation losses. No network parameters are optimized during this procedure.

B.4 Parameterized Distillation

For the parameterized experiments, we adopt the learnable shape-morphing representation of 3DDP (Kim et al., 2026). Each stored group consists of four coarse anchor point clouds. Following the original 3DDP configuration, we use 252 points per anchor and 16 morphed samples per group on ModelNet10, and 255 points per anchor and 4 morphed samples per group on the remaining datasets. The anchor coordinates and morphing logits are jointly optimized with learning rate 10.

To match the resolution of the decoded synthetic samples, each sampled real point cloud is partitioned into corresponding low-resolution point clouds before feature extraction. We retain the shape-morphing parameterization and storage-budget convention of 3DDP, but replace its original uniformity-aware matching loss with the CPDM objective in Eq. 9. Thus, the parameterized experiments evaluate the compatibility of CPDM with the 3DDP representation rather than introducing an additional parameterization mechanism.

B.5 Evaluation Details

Each distilled dataset is evaluated by training 10 independently initialized networks from scratch and reporting the mean and standard deviation of test classification accuracy. Unless otherwise specified, the evaluation network is trained for 500 epochs using SGD with learning rate 0.01, momentum 0.9, weight decay 5×10^{-4}, and batch size 8. A StepLR scheduler reduces the learning rate by a factor of 0.1 after 250 epochs. No data augmentation is used during evaluation.

For the main results, both distillation and evaluation use PointNet. For cross-architecture evaluation, the same PointNet-distilled synthetic dataset is used to train PointNet++, PointConv, or Point

Table 7: Hyperparameter analysis of CPDM on ModelNet10 (MN10), ModelNet40 (MN40), and ScanObjectNN (SONN). Results are classification accuracy (%). The number of experts is varied at PPC$=3$, while λ_{CP} and λ_{CE} are examined at PPC$=10$.

# Experts	5	10	20	50	100
MN10	79.4	79.9	80.5	81.6	81.1
MN40	73.0	74.1	73.4	73.7	74.1
SONN	29.5	30.3	31.0	30.9	31.2

(a) PPC=3

λ_{CP}	0.5	1.0	2.0
MN10	87.7	88.8	88.1
MN40	82.8	82.7	82.7
SONN	48.3	49.6	48.7

(b) PPC=10

λ_{CE}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
MN10	86.8	88.8	85.3	83.5
MN40	82.7	82.7	79.5	76.3
SONN	46.3	49.6	43.1	41.7

(c) PPC=10

Transformer from scratch, without re-distilling the data for each evaluation architecture. All CPDM experiments are conducted on a single NVIDIA GeForce RTX 4090 GPU.

C Additional Results

C.1 Hyperparameter Analysis

Table 7 analyzes the sensitivity of CPDM to the number of expert trajectories and the loss weights λ_{CP} and λ_{CE} on ModelNet10, ModelNet40, and ScanObjectNN. At PPC$=3$, increasing the number of experts from 5 to 100 leads to relatively moderate variations in accuracy: 2.2, 1.1, and 1.7 percentage points on the three datasets, respectively. Performance does not improve monotonically with a larger expert pool; for example, ModelNet10 achieves its highest accuracy with 50 experts, whereas ModelNet40 obtains the same highest value with 10 and 100 experts. Given the relatively small performance variation, we use 10 trajectories by default to reduce expert-buffer preparation and storage costs while retaining competitive accuracy.

At PPC$=10$, CPDM is relatively insensitive to λ_{CP} over the tested range $[0.5, 2.0]$, with accuracy varying by at most 1.3 percentage points on each dataset. Setting $\lambda_{\mathrm{CP}} = 1$ achieves the highest accuracy on ModelNet10 and ScanObjectNN and remains within 0.1 percentage points of the best result on ModelNet40. The classification-loss weight has a more pronounced effect. $\lambda_{\mathrm{CE}} = 10^{-3}$ achieves the best or joint-best performance across all three datasets, whereas increasing it to 10^{-1} reduces accuracy by 5.3, 6.4, and 7.9 percentage points on ModelNet10, ModelNet40, and ScanObjectNN, respectively. These results indicate that CPDM is robust to moderate changes in the coherence-matching weight, while classification supervision should remain auxiliary rather than dominate the matching objective. Accordingly, we use $\lambda_{\mathrm{CP}} = 1$ and $\lambda_{\mathrm{CE}} = 10^{-3}$ in the main experiments.

C.2 Additional Visualizations

Qualitative Comparison with SADM. Figure 5 qualitatively compares the initialized point clouds with synthetic samples distilled by SADM and CPDM across eight ScanObjectNN categories. Both distillation methods substantially reorganize the initialized point coordinates, while retaining coarse class-dependent geometric structures and producing distinct local point configurations. The resulting CPDM samples are not intended to reconstruct individual training instances; rather, their coordinates are optimized to match feature distributions and satisfy classification supervision. This comparison provides a qualitative view of how CPDM transforms the initial point clouds relative to SADM under the same distillation setting.

Visualization of Feature Distributions. Figure 6 qualitatively compares the feature distributions of real and distilled point clouds on ModelNet10. We extract features for the real samples and the synthetic datasets produced by SADM and CPDM using the same converged PointNet, and fit a single t-SNE embedding jointly to all features. Consequently, the two panels share the same embedding coordinates and the same real-data reference. Synthetic features from both methods generally lie within or near class-dependent regions occupied by real samples, while differences remain in their local arrangements. We emphasize that t-SNE provides only a qualitative visualization of the real–synthetic feature relationship and is not used as a quantitative measure of distribution matching quality.

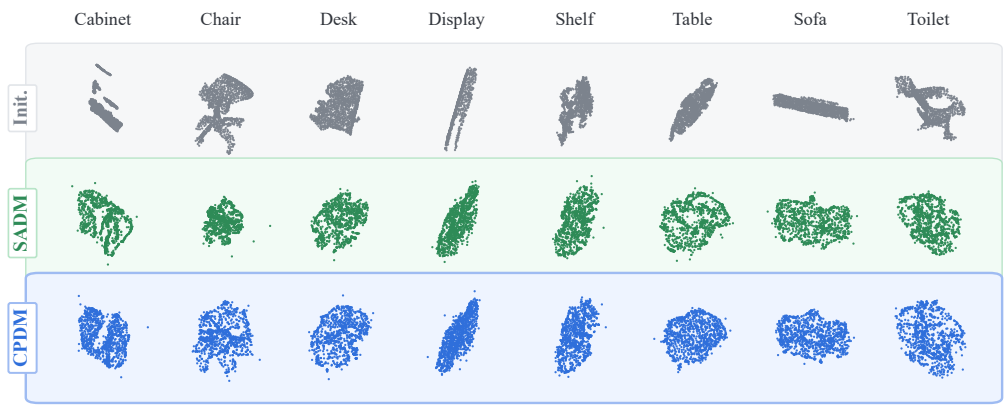
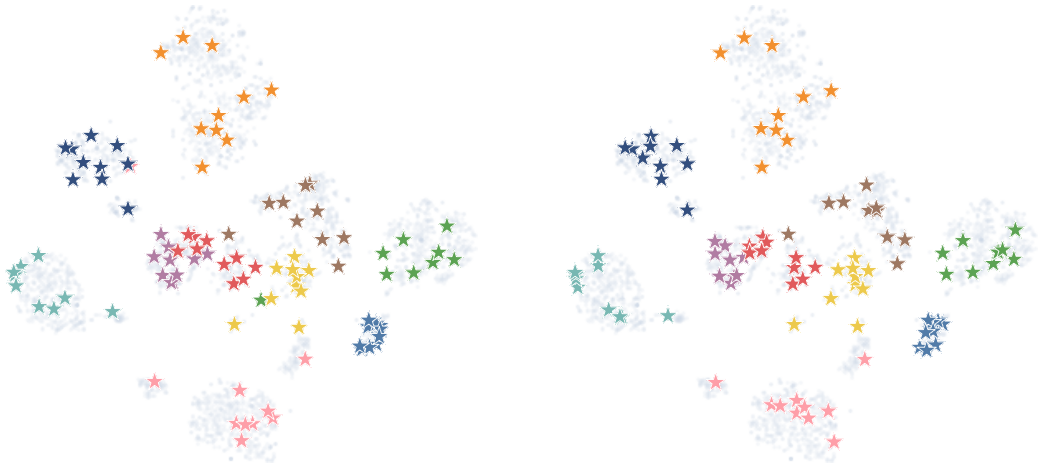


Figure 5: Qualitative comparison of initialized point clouds and samples distilled by SADM and CPDM on ScanObjectNN. The rows correspond to initialization (gray), SADM (green), and CPDM (blue), respectively.



(a) SADM (b) CPDM

Figure 6: Joint t-SNE visualization of real and distilled point-cloud features on ModelNet10. Features are extracted using the same converged PointNet. Gray points denote the same real samples in both panels, while colored stars denote synthetic samples, with colors consistently representing classes across panels.

Visualization of Distilled Point Clouds. Figures 7–11 provide additional visualizations of point clouds distilled by CPDM at PPC$=$ 1 on ModelNet10, ModelNet40, ShapeNet, ScanObjectNN, and OmniObject3D, respectively. Despite the extremely limited compression budget, the distilled samples retain recognizable class-dependent geometric characteristics across both synthetic CAD datasets and real-world scanned objects. Meanwhile, their local point configurations can differ substantially from those of natural samples, since CPDM optimizes the point coordinates for downstream training utility rather than faithful reconstruction of individual training instances. These qualitative results illustrate the diverse task-oriented representations learned by CPDM across datasets with different scales and geometric characteristics.

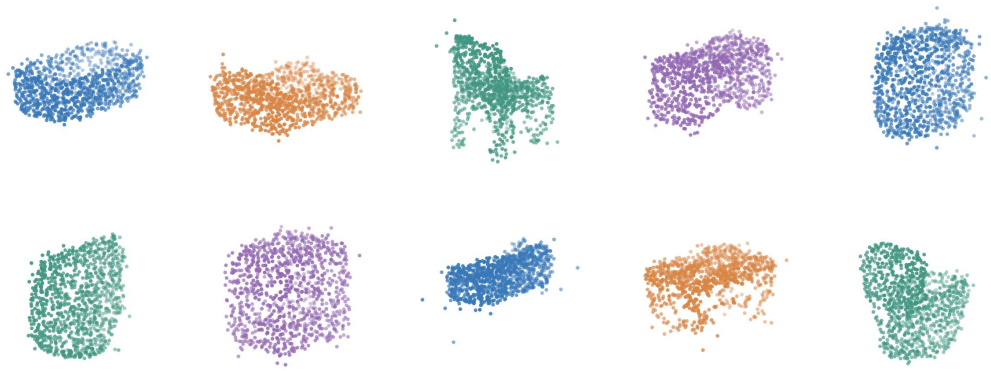


Figure 7: Visualization of point clouds distilled by CPDM on ModelNet10 at PPC= 1.

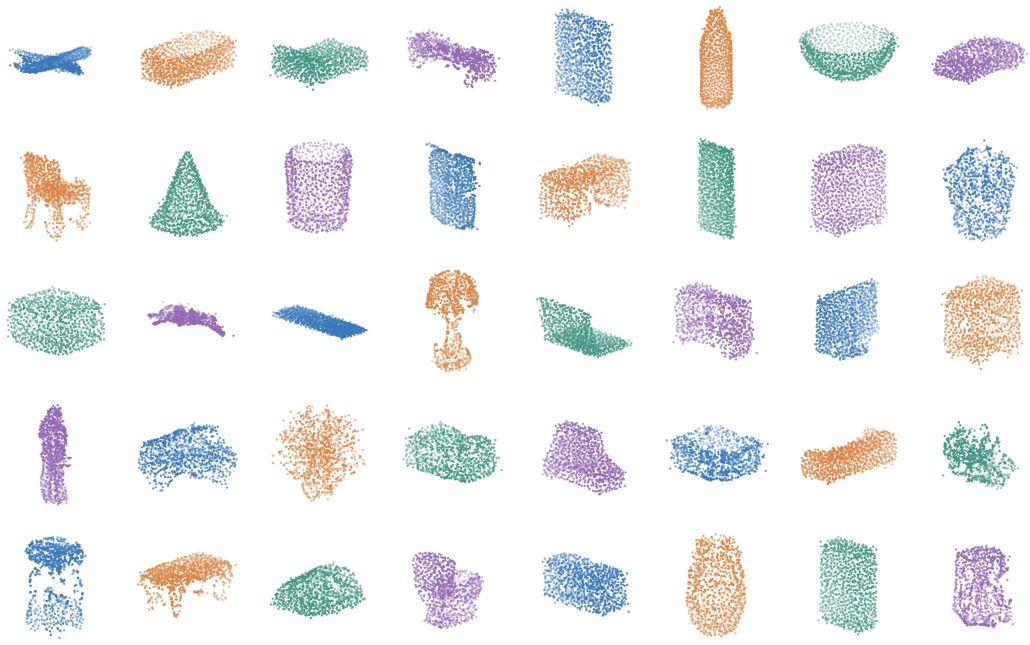


Figure 8: Visualization of point clouds distilled by CPDM on ModelNet40 at PPC= 1.



Figure 9: Visualization of point clouds distilled by CPDM on ShapeNet at PPC= 1.

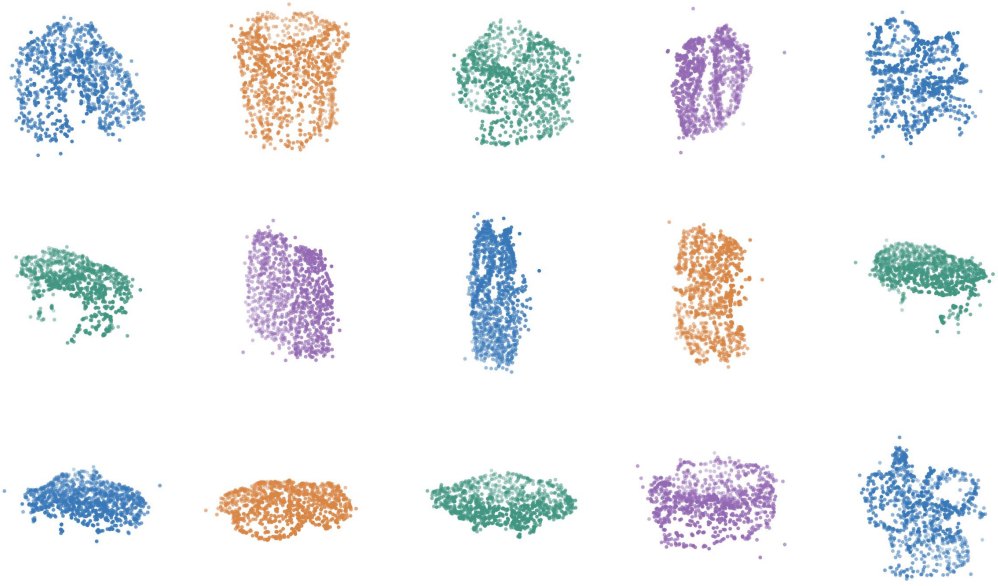


Figure 10: Visualization of point clouds distilled by CPDM on ScanObjectNN at PPC= 1.

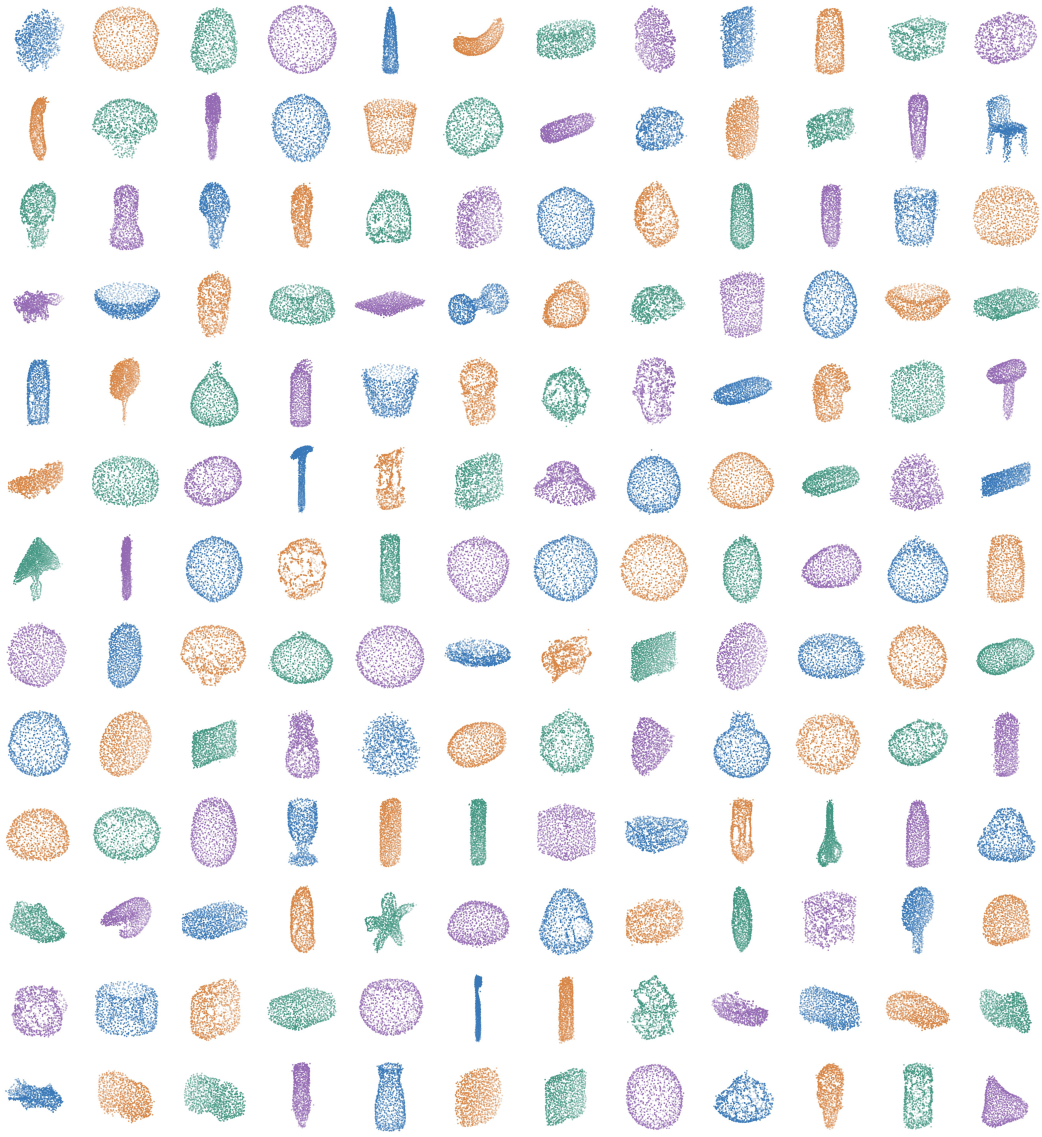


Figure 11: Visualization of point clouds distilled by CPDM on OmniObject3D at PPC= 1.