Dataset Distillation of 3D Point Clouds via Distribution Matching

Jae-Young Yim[†], **Dongwook Kim**[†], **and Jae-Young Sim**[⋆]
Ulsan National Institute of Science and Technology
{yimjae0, donguk071, jysim}@unist.ac.kr

[†] Equal contribution [⋆] Corresponding author

Abstract

Large-scale datasets are usually required to train deep neural networks; however, they increase computational complexity, hindering practical applications. Recently, dataset distillation for images and texts has attracted considerable attention, as it reduces the original dataset to a small synthetic one to alleviate the computational burden of training while preserving essential task-relevant information. However, dataset distillation for 3D point clouds remains largely unexplored, as point clouds exhibit fundamentally different characteristics from those of images, making this task more challenging. In this paper, we propose a distribution-matching-based distillation framework for 3D point clouds that jointly optimizes the geometric structures and orientations of synthetic 3D objects. To address the semantic misalignment caused by the unordered nature of point clouds, we introduce a Semantically Aligned Distribution Matching (SADM) loss, which is computed on the sorted features within each channel. Moreover, to handle rotational variations, we jointly learn optimal rotation angles while updating the synthetic dataset to better align with the original feature distribution. Extensive experiments on widely used benchmark datasets demonstrate that the proposed method consistently outperforms existing dataset distillation approaches, achieving higher accuracy and strong cross-architecture generalization.

1 Introduction

With the increasing demand for large-scale training datasets, the high cost of retraining models from scratch has become a major challenge, motivating research on dataset distillation [22]. The objective of dataset distillation is to produce a significantly smaller synthetic dataset from a large original dataset, that preserves the essential task-relevant information contained in the original dataset. Hence the models trained on the reduced synthetic dataset are encouraged to achieve comparable performance to those trained on the original dataset. Existing dataset distillation methods [3, 6, 8, 11, 30, 29, 31, 18, 27, 5, 9] can be broadly classified into gradient matching, trajectory matching, and distribution matching approaches. The gradient matching and trajectory matching methods aim to ensure that the synthetic dataset produces similar optimization dynamics to the original dataset. To this end, the gradient matching method [30] minimizes the difference between the gradients computed on the synthetic and original datasets. However, it potentially overlooks long-term dependencies in the training process. Rather than comparing individual gradients, the trajectory matching methods [3, 6, 8, 11] encourage the models trained on the synthetic dataset follow similar optimization trajectories to those trained on the original dataset. Note that these methods train the networks while optimizing the synthetic dataset, and significantly increase the computational complexity. To alleviate the computational burden in dataset distillation, the distribution matching techniques [29, 31, 18, 27, 5] have been introduced. The networks are randomly initialized and used

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

without training to extract features, after which the feature distributions of the original and synthetic datasets are compared.

While dataset distillation has been extensively studied for structured data such as images [30, 3, 29] and texts [10, 12, 13], its application to 3D point clouds remains almost unexplored. PCC [28] simply applied an existing distillation method for images [30] to the 3D point clouds without considering inherent characteristics of 3D point clouds, however it still suffers from high computational complexity. Unlike the structured data, 3D point clouds consist of unordered and irregularly distributed points in 3D space. Therefore, semantically similar regions across different 3D models are often associated with inconsistent orders (i.e., point indices), which makes direct feature comparison incorrect, thereby worsening the performance of distribution-matching-based dataset distillation. We refer to this issue as *semantic misalignment*. Moreover, the datasets of 3D point clouds also suffers from *rotational variation*. 3D point clouds are often captured or synthesized under arbitrary poses due to the absence of canonical orientations. Such rotational differences cause objects of the same class to produce different features, significantly increasing intra-class variability. As a result, it becomes difficult to construct a representative synthetic dataset that faithfully follows the feature distribution of the original dataset.

In this paper, we propose an optimization framework combining Semantically Aligned Distribution Matching (SADM) and orientation optimization for dataset distillation of 3D point clouds. A major challenge in this setting is achieving effective feature alignment despite the unordered nature of point clouds and their arbitrary orientations. The SADM loss addresses the issue of misaligned semantic structures by sorting the point-wise feature values within each channel before computing the distance between two feature distributions. In parallel, we employ learnable rotation parameters to address the orientation variation by estimating the optimal poses of 3D models while generating the synthetic dataset. By simultaneously enforcing semantic consistency and orientation alignment, the proposed method achieves superior performance of dataset distillation compared with the existing methods, as demonstrated by extensive experiments on standard 3D point cloud classification benchmarks.

The key contributions of this paper are summarized as follows:

- To the best of our knowledge, we are the first to propose a distribution-matching-based dataset distillation method of 3D point clouds, that jointly optimizes the shapes and orientations of synthetic dataset.
- We devised the SADM loss, computed on the sorted features within each channel, to preserve semantic alignment between the compared 3D objects.
- We validated the proposed method through extensive experiments on the four benchmark datasets widely used for 3D point clouds classification, and showed the superiority of the proposed method over the existing dataset distillation techniques.

2 Related Works

2.1 3D Point Data Analysis

3D point clouds are generally unordered exhibiting irregular characteristics, that makes it difficult to apply the convolution operations in deep neural networks commonly used for images. To process such data, early studies [26, 33] converted point clouds into structured representations, such as multi-view images or voxel grids, enabling the use of standard convolutional neural networks (CNN) architectures. However, these conversions introduce quantization errors and increase memory overhead. PointNet [15] addressed this limitation by directly learning the features from unordered points, which serves as the backbone in our dataset distillation framework. Subsequent methods, such as PointNet++ [16], PointConv [24], and DGCNN [23], extended PointNet by capturing local geometric relationships through hierarchical architecture. Additionally, attempts have been also made to apply the transformer [21] to 3D point clouds processing, where Point Transformer [32] utilizes the attention mechanism to capture the long-range dependencies. We adopt these architectures as evaluation networks to assess the cross-architecture generalization capability of our distilled datasets.

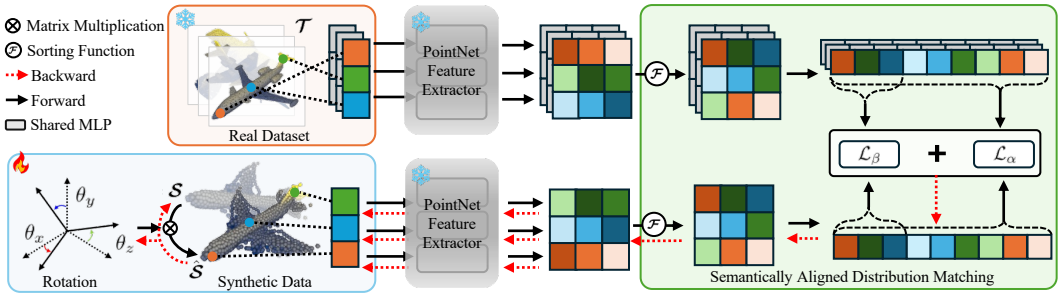


Figure 1: The overall framework of the proposed dataset distillation method for 3D point clouds.

2.2 Coreset Selection

Coreset selection is a technique designed to select representative samples from a given dataset, while maintaining the model's performance even with the sampled data. The random selection method [17] randomly chooses a subset of data samples from the whole dataset. It is simple but suffers from the robustness due to the lack of informative criteria for sampling. The K-center method [19] selects data samples iteratively that maximize the minimum distance to the set of already selected ones, taking into account the data distribution. The Herding method [2, 1] iteratively takes data points that minimize the discrepancy between the mean embeddings of the selected subset and the entire dataset in the feature space. The coreset selection operates within the space of existing samples, and hence lacks the flexibility to synthesize informative patterns. This limitation naturally leads to the emergence of dataset distillation, which generates synthetic datasets capturing task-specific patterns beyond the original samples.

2.3 Dataset Distillation

Dataset distillation [22] methods can be largely categorized into the gradient matching [30], trajectory matching [3, 6, 8, 11], and distribution matching [29, 31, 18, 27, 5] approaches. The gradient matching, first introduced by DC [30], minimizes the difference between the gradients computed from the original and synthetic datasets, respectively, guiding the synthetic dataset to follow the training direction of the original dataset. The trajectory matching methods [3, 6, 8, 11], initially proposed by MTT [3], encourage the models trained on the synthetic dataset to follow the optimization trajectories similar to those trained on the original dataset rather than comparing individual gradients. ATT [11] automatically adjusts the length of the training trajectories between the synthetic and original datasets, enabling more effective and precise matching. The distribution matching, introduced by DM [29], focuses on minimizing the distance between the feature distributions of the original and synthetic datasets. DataDAM [18] enhances the distribution matching by aligning the feature maps using attention mechanism, achieving unbiased feature representation with low computational overhead. Furthermore, M3D [27] employs the Gaussian kernel function in the distribution matching loss, enabling the alignment across higher-order statistical characteristics of the feature distributions. Recently, a gradient matching-based point clouds distillation method [28] has been introduced, however it simply applied the existing image-based method without considering the unique characteristics of 3D point clouds.

3 Methodology

We propose a novel dataset distillation method for 3D point clouds that first addresses the key challenges of semantic misalignment and rotational variation. We perform SADM that effectively aligns the features of semantically consistent structures across different 3D models. To handle the rotational variation, we also estimate optimal orientations while generating the synthetic dataset. Figure 1 shows the overall framework of the proposed method.

3.1 Preliminaries

Problem Definition. The objective of dataset distillation is to compress the task-relevant information in the original dataset $\boldsymbol{\mathcal{T}} = \{\mathbf{t}_i\}_{i=1}^{|\mathcal{T}|}$ and generate a much smaller synthetic dataset $\boldsymbol{\mathcal{S}} = \{\mathbf{s}_i\}_{i=1}^{|\mathcal{S}|}$, where $|\boldsymbol{\mathcal{S}}| \ll |\boldsymbol{\mathcal{T}}|$, such that a model trained on $\boldsymbol{\mathcal{S}}$ achieves close performance to that trained on $\boldsymbol{\mathcal{T}}$. Given a 3D point cloud sample $\mathbf{p}$ following a real data distribution with the corresponding class label l, the optimal synthetic dataset $\boldsymbol{\mathcal{S}}^\star$ can be obtained via

$$\boldsymbol{\mathcal{S}}^\star = \arg\min_{\boldsymbol{\mathcal{S}}} \mathbb{E}_{\mathbf{p}} \left[||\mathcal{L}(\phi_{\boldsymbol{\mathcal{T}}}(\mathbf{p}), l) - \mathcal{L}(\phi_{\boldsymbol{\mathcal{S}}}(\mathbf{p}), l)||^2 \right], \tag{1}$$

where $\mathcal{L}$ denotes a task-specific loss function, such as the cross-entropy loss, and $\phi_{\boldsymbol{\mathcal{T}}}$ and $\phi_{\boldsymbol{\mathcal{S}}}$ represent the models to estimate the class label which are trained on $\boldsymbol{\mathcal{T}}$ and $\boldsymbol{\mathcal{S}}$, respectively.

Distribution Matching. The distribution matching (DM) strategy focuses on aligning the feature distributions derived from the original and synthetic datasets, respectively, via

$$\boldsymbol{\mathcal{S}}^\star = \arg\min_{\boldsymbol{\mathcal{S}}} D(\phi(\boldsymbol{\mathcal{T}}), \phi(\boldsymbol{\mathcal{S}})), \tag{2}$$

where ϕ is the feature extractor and D denotes a distance function. We employ a randomly initialized, untrained network, which has been demonstrated to sufficiently capture the structural information for distribution alignment [29]. Also, the Maximum Mean Discrepancy (MMD) [7] loss $\mathcal{L}_{\text{MMD}}$ is often used as D, given by

$$\mathcal{L}_{\text{MMD}}(\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{S}}) = K(\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{T}}) + K(\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{S}}) - 2K(\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{S}}), \tag{3}$$

where $K(\cdot, \cdot)$ is a kernel function. We used the Gaussian kernel in this work as

$$K(\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{S}}) = \frac{1}{|\boldsymbol{\mathcal{T}}| \cdot |\boldsymbol{\mathcal{S}}|} \sum_{\mathbf{t} \in \boldsymbol{\mathcal{T}}} \sum_{\mathbf{s} \in \boldsymbol{\mathcal{S}}} \exp\left(-\frac{\|\phi(\mathbf{t}) - \phi(\mathbf{s})\|^2}{2\sigma} \right). \tag{4}$$

3.2 Semantically Aligned Distribution Matching

Whereas the pixels of image exhibit well structured spatial relationships with one another and consistently indexed across different images, the points in 3D point clouds are unordered with different indices across different models. Therefore, the conventional distribution matching methods cannot be directly applied to 3D point clouds, since the features of semantically similar structures are not aligned between two compared models. Motivated by the importance of preserving semantic correspondence, we investigate the relationship between the feature values and semantic significance in 3D point clouds. We first extracted point-wise features using a randomly initialized network ϕ, and sorted the feature values for each of 1024 channels according to their size. Figure 2 visualizes the points corresponding to the largest (top), 200th largest (middle), and 500th largest (bottom) features, respectively. We observe that, even without model training, the points associated with the largest features consistently capture semantically meaningful regions, such as edges and corners, across different classes. In contrast, the points yielding low-ranked features are distributed around less informative regions. This suggests that the features reflect the relative importance of points in characterizing the structures of 3D objects, and the points with similar orders of sorted features tend to capture semantically related regions across different 3D models.

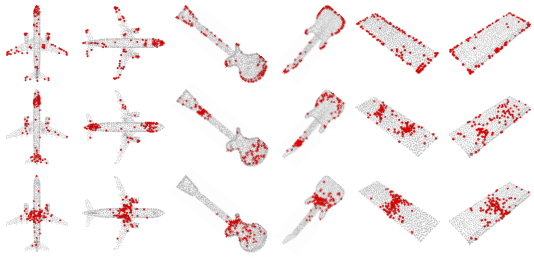


Figure 2: Visualization of the points corresponding to the largest (top), 200th largest (middle), and 500th largest (bottom) features in each channel.

In order to preserve the semantic correspondence across different 3D models, we propose a Semantically Aligned Distribution Matching (SADM) loss. Specifically, given a 3D point cloud object $\mathbf{p}$, we extract the features using ϕ via

$$\phi(\mathbf{p}) = \begin{bmatrix} f_{1,1} & f_{2,1} & \cdots & f_{C,1} \\ f_{1,2} & f_{2,2} & \cdots & f_{C,2} \\ \vdots & \vdots & \ddots & \vdots \\ f_{1,N} & f_{2,N} & \cdots & f_{C,N} \end{bmatrix} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_C], \quad \mathbf{f}_i \in \mathbb{R}^N, \tag{5}$$

where N denotes the number of points in $\mathbf{p}$, C is the number of feature channels, and $\mathbf{f}_i = \{f_{i,1}, f_{i,2}, ..., f_{i,N}\}$ represents the feature vector of the i-th channel where $f_{i,j}$ is the feature value of the j-th point. We then perform channel-wise sorting to the extracted features of $f_{i,j}$'s in the descending order of their values, and obtain the sorted feature vector $\tilde{\mathbf{f}}_i = \left\{\tilde{f}_{i,1}, \tilde{f}_{i,2}, ..., \tilde{f}_{i,N}\right\}$ such that $\tilde{f}_{i,j} \geq \tilde{f}_{i,j+1}$. Then we have the set of the sorted feature vectors as

$$\tilde{\phi}(\mathbf{p}) = \left[\tilde{\mathbf{f}}_1, \tilde{\mathbf{f}}_2, \ldots, \tilde{\mathbf{f}}_C\right], \quad \tilde{\mathbf{f}}_i \in \mathbb{R}^N. \tag{6}$$

Note that $\mathbf{t}$ in the original dataset $\boldsymbol{\mathcal{T}}$ and $\mathbf{s}$ in the synthetic dataset $\boldsymbol{\mathcal{S}}$ exhibit different orders of points in general, and therefore the features in $\phi(\mathbf{t})$ and $\phi(\mathbf{s})$ are inconsistently aligned with each other worsening the desired behavior of comparison in (3). On the contrary, the sorted features $\tilde{\phi}(\mathbf{t})$ and $\tilde{\phi}(\mathbf{s})$ exhibit consistent ordering of features in each channel reflecting their relative semantic importance, and thus facilitates reliable feature comparison across different 3D objects. To measure the discrepancy between the sorted features, we redefine the Gaussian kernel as

$$\tilde{K}(\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{S}}) = \frac{1}{|\boldsymbol{\mathcal{T}}| \cdot |\boldsymbol{\mathcal{S}}|} \sum_{\mathbf{t} \in \boldsymbol{\mathcal{T}}} \sum_{\mathbf{s} \in \boldsymbol{\mathcal{S}}} \exp\left(-\frac{||\tilde{\phi}(\mathbf{t}) - \tilde{\phi}(\mathbf{s})||^2}{2\sigma}\right). \tag{7}$$

Then we devise the loss $\mathcal{L}_\alpha$ from $\mathcal{L}_{\text{MMD}}$ (3), given by

$$\mathcal{L}_\alpha(\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{S}}) = \tilde{K}(\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{T}}) + \tilde{K}(\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{S}}) - 2\tilde{K}(\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{S}}). \tag{8}$$

We additionally employ $\mathcal{L}_\beta$ to boost the role of the most significant feature in each channel, defined as

$$\mathcal{L}_\beta(\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{S}}) = \tilde{K}_{\text{top}}(\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{T}}) + \tilde{K}_{\text{top}}(\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{S}}) - 2\tilde{K}_{\text{top}}(\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{S}}), \tag{9}$$

where $\tilde{K}_{\text{top}}(\cdot, \cdot)$ denotes the Gaussian kernel computed with only the largest feature in each channel. The SADM loss is then formulated as a weighted sum of the two losses, given by

$$\mathcal{L}_{\text{SADM}}(\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{S}}) = \lambda_1 \mathcal{L}_\alpha(\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{S}}) + \lambda_2 \mathcal{L}_\beta(\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{S}}), \tag{10}$$

where λ_1, λ_2 are the weighting parameters. By using $\mathcal{L}_{\text{SADM}}$ instead of $\mathcal{L}_{\text{MMD}}$, we facilitate reliable matching of feature distributions considering semantic structures of 3D point clouds, thereby improving the performance dataset distillation.

3.3 Estimation of Optimal Rotations

3D objects usually exhibit different orientations from each other. Therefore, while generating optimal 3D objects in the synthetic dataset in terms of their geometric shapes, we also estimate their optimal rotations best representing various orientations of 3D objects in the original dataset. In practice, we introduce three rotation angles, θ_x, θ_y, and θ_z, corresponding to rotations around the x-, y-, and z-axes, respectively. These angles are treated as learnable parameters, allowing the orientation of each synthetic 3D object in $\boldsymbol{\mathcal{S}}$ to be adaptively adjusted during dataset distillation. Therefore, instead of minimizing $\mathcal{L}_{\text{SADM}}(\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{S}})$ in (10), we minimize $\mathcal{L}_{\text{SADM}}(\boldsymbol{\mathcal{T}}, \mathcal{R}_{\boldsymbol{\theta}}(\boldsymbol{\mathcal{S}}))$, where $\mathcal{R}_{\boldsymbol{\theta}}$ denotes the rotation operator according to the rotation parameters $\boldsymbol{\theta} = (\theta_x, \theta_y, \theta_z)$. The overall objective is formally defined as

$$\{\boldsymbol{\mathcal{S}}^\star, \boldsymbol{\theta}^\star\} = \arg\min_{\{\boldsymbol{\mathcal{S}}, \boldsymbol{\theta}\}} \mathcal{L}_{\text{SADM}}(\boldsymbol{\mathcal{T}}, \mathcal{R}_{\boldsymbol{\theta}}(\boldsymbol{\mathcal{S}})). \tag{11}$$

Note that we optimize the synthetic 3D objects as well as their rotation parameters simultaneously, during the dataset distillation process. Specifically, at each iteration, we randomly initialize the network parameters and construct the synthetic dataset $\boldsymbol{\mathcal{S}}$ by randomly selecting samples from the original dataset. We then form $\mathcal{R}_{\boldsymbol{\theta}}(\boldsymbol{\mathcal{S}})$ by rotating the objects in $\boldsymbol{\mathcal{S}}$ according to the angles θ_x, θ_y, and θ_z, initially set to zero. For each class, a mini-batch is sampled from $\boldsymbol{\mathcal{T}}$, with a batch size of 8 per class. The corresponding synthetic mini-batch is sampled from $\mathcal{R}_{\boldsymbol{\theta}}(\boldsymbol{\mathcal{S}})$, with the batch size determined by the number of point cloud objects per class (PPC). The joint optimization process iteratively updates both the geometric structure of the synthetic dataset and their rotation parameters by minimizing $\mathcal{L}_{\text{SADM}}(\boldsymbol{\mathcal{T}}, \mathcal{R}_{\boldsymbol{\theta}}(\boldsymbol{\mathcal{S}}))$. This ensures that the synthetic dataset preserves the geometric characteristics of the original dataset while aligning their orientations more effectively. The benefit of this joint optimization is justified in the following proposition.

Table 1: Comparison of quantitative performance. DC, DM, and the proposed method were initialized with random selection [17] for fair comparison. 'Whole' refers to the classification accuracy obtained by training the network on the entire original dataset without any distillation. 'Ratio' represents the percentage of the size of the distilled dataset compared to that of the original dataset. The best and the second best scores are highlighted in bold and underlined, respectively.

Datasets	**ModelNet10** [25]			**ModelNet40** [25]			**ShapeNet** [4]			**ScanObjectNN** [20]		
PPC	1	3	10	1	3	10	1	3	10	1	3	10
Ratio (%)	0.25	0.75	2.5	0.4	1.2	4.0	0.15	0.45	1.5	0.15	0.45	1.5
Whole	91.41			87.84			82.49			63.84		
Random	35.5±4.7	75.2±1.7	85.3±1.1	34.6±1.8	60.0±1.32	74.1±0.4	34.0±3.0	54.8±1.5	63.1±1.0	13.9±1.4	20.4±1.3	34.8±1.1
Herding	40.1 ±5.2	78.0 ±1.3	86.9±0.6	54.4 ±2.0	68.5 ±1.0	78.8 ±0.4	49.5 ±2.3	59.8 ±0.9	66.9 ±0.5	15.7 ±1.7	27.7 ±1.4	38.7 ±1.7
K-center	40.1 ±5.2	77.6±1.9	83.2±1.4	54.4±2.0	63.0 ±2.7	65.3 ±1.1	49.5 ±2.3	51.4 ±1.7	47.8 ±0.7	15.7 ±1.7	19.8 ±0.8	24.0 ±1.0
DM	31.9±4.3	77.6±1.6	86.1±0.8	32.3±4.8	62.0±1.9	75.2±0.5	27.8±4.0	56.1±1.2	64.3±0.7	14.9±2.0	23.3±1.5	35.6±1.4
DC	43.9±5.5	75.0±2.4	86.1±1.1	52.0±1.7	66.6±1.3	75.6±0.5	49.8±1.3	60.1±1.4	64.6±0.7	**18.6±1.6**	24.3±2.5	35.3±1.2
Ours	**44.7 ±6.1**	**84.4 ±1.2**	**87.8 ±1.0**	**55.8 ±1.5**	**72.1 ±0.8**	**80.1 ±0.4**	**50.2 ±1.4**	**63.7 ±0.7**	**68.4 ±0.5**	17.4 ±1.5	**31.6 ±1.0**	**43.9 ±1.9**

Table 2: Comparison of cross-architecture generalization performance evaluated on PointNet++ [16], DGCNN [23], PointConv [24], and PT [32], respectively.

Datasets	**ModelNet10** [25]			**ModelNet40** [25]			**ShapeNet** [4]			**ScanObjectNN** [20]		
Method	**DM**	**DC**	**Ours**	**DM**	**DC**	**Ours**	**DM**	**DC**	**Ours**	**DM**	**DC**	**Ours**
PointNet++	35.0±8.3	51.5±7.8	**82.9±1.4**	14.1±2.7	46.9±6.0	**68.5±0.5**	20.3±3.7	34.3±3.4	**53.2±1.5**	15.5±3.6	11.6±2.2	**22.0±3.9**
DGCNN	56.6±5.3	62.6±2.2	**79.0±2.2**	34.8±3.5	52.3±2.7	**66.9±1.1**	13.8±2.2	38.0±2.7	**52.8±1.6**	17.7±2.4	12.5±1.3	**19.0±1.5**
PointConv	33.4±6.2	40.0±10.5	**56.6±4.9**	20.0±5.7	37.9±3.9	**51.3±6.2**	17.4±3.0	34.0±4.8	**47.4±2.3**	15.2±1.7	14.2±1.8	**16.8±1.5**
PT	48.5±6.6	61.8±1.6	**77.6±1.6**	26.7±5.0	47.5±2.7	**61.6±1.1**	34.2±5.3	42.6±2.5	**55.5±0.5**	17.1±0.8	14.9±0.5	**21.9±2.7**

Proposition 1. *Jointly optimizing the synthetic dataset $\boldsymbol{\mathcal{S}}$ and the rotation parameters $\boldsymbol{\theta}$ guarantees a lower or equal loss to that of optimizing $\boldsymbol{\mathcal{S}}$ alone.*

$$\min_{\{\boldsymbol{\mathcal{S}},\boldsymbol{\theta}\}} \mathcal{L}_{\text{SADM}}(\boldsymbol{\mathcal{T}}, \mathcal{R}_{\boldsymbol{\theta}}(\boldsymbol{\mathcal{S}})) \leq \min_{\boldsymbol{\mathcal{S}}} \mathcal{L}_{\text{SADM}}(\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{S}}), \tag{12}$$

where $\mathcal{R}_{\boldsymbol{\theta}}$ denotes the rotation operator according to $\boldsymbol{\theta}$.

Proof Sketch. The proof is provided in Appendix A. Jointly optimizing over $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\theta}$ enlarges the feasible set, as the original optimization over $\boldsymbol{\mathcal{S}}$ alone is a special case of the joint optimization with fixed $(\theta_x, \theta_y, \theta_z) = (0, 0, 0)$. □

4 Experimental Results

4.1 Experimental Setup

The proposed method was evaluated on the ModelNet10 [25], ModelNet40 [25], ShapeNet [4], and ScanObjectNN [20] datasets. The ModelNet10, ModelNet40, and ShapeNet are synthetic datasets generated from CAD models, containing 10, 40, and 55 classes, respectively. The ScanObjectNN consists of real-world scanned objects from 15 classes. Following previous methods [30, 29], we measure the classification accuracy trained on the distilled synthetic dataset. We ensure the fairness by training the network 10 times and report the mean and standard deviation of accuracy. We evaluate the performance across different PPC values of 1, 3, and 10. Each point cloud object contains 1,024 points, which is a common standard used in the 3D point clouds classification tasks [15, 16]. The dataset distillation process for the synthetic dataset was optimized for 1,500 iterations, where the performance was evaluated at every 250 iterations by training the PointNet on the synthetic dataset and testing it on the original test dataset.

4.2 Performance Comparison

4.2.1 Dataset Distillation

We compare the performance of the proposed method with that of 1) the three most representative coreset selection methods: random selection [17], Herding [2, 1], and K-Center [19], 2) two existing image dataset distillation methods of DC [30] and DM [29]. Table 1 compares the quantitative performance of the proposed method with that of the existing methods. DM and DC were initialized with random selection. The synthetic datasets were optimized using PointNet [15], and classification

Table 3: Comparison of cross-dataset generalization performance at the PPC value of 3, where the models are trained on a subset of ModelNet40 and evaluated on a subset of ShapeNet, and vice versa.

Setting	DC	DM	PCC	Ours
SN → MN40	54.43	49.66	57.43	**61.54**
MN40 → SN	53.75	47.35	49.42	**54.40**

Table 4: Performance comparison across different initialization strategies. The accuracies for PPC values of 1, 3, and 10 are averaged. **MN10**: ModelNet10. **MN40**: ModelNet40. **SN**: ShapeNet. **SONN**: ScanObjectNN.

Init	Method	MN10	MN40	SN	SONN
Noise	**DC**	18.77	6.05	5.24	12.73
	DM	26.53	12.04	12.09	13.55
	Ours	**70.69**	**65.25**	**58.44**	**29.03**
Random	**DC**	68.32	64.71	58.15	26.07
	DM	65.17	56.50	49.39	24.57
	Ours	**72.48**	**69.31**	**60.76**	**31.01**
K-center	**DC**	69.57	61.85	42.88	21.65
	DM	63.04	49.93	44.82	21.59
	Ours	**72.72**	**68.20**	**60.70**	**28.25**
Herding	**DC**†	71.50	66.79	59.23	26.98
	DM	64.76	59.16	52.61	27.58
	Ours	**72.93**	**69.67**	**62.25**	**31.08**

† DC [30] with Herding initialization corresponds to PCC [28]

Table 5: Comparison of training times (in hours) required for dataset distillation at the PPC value of 3.

Method	MN10	MN40	SN	SONN
DC	1.52	5.43	7.57	2.19
DM	**0.04**	**0.11**	**0.15**	**0.05**
Ours	0.08	0.26	0.35	0.11

performance was evaluated on PointNet to demonstrate their validity. Among the coreset selection methods, Herding achieves the best overall performance. When PPC is set to 1, Herding and K-Center yield identical results since K-Center selects the first data point using the same algorithm to Herding. As PPC increases, Herding consistently outperforms other coreset selection methods, showing its effectiveness. However, because coreset selection methods rely solely on sample selection without further optimization, they struggle to capture complex feature distributions. The features used in DM are extracted from the layer before the classifier, and they discard substantial information through the global max pooling process of PointNet. This information loss weakens the effectiveness of the distribution matching, preventing DM from accurately capturing the feature distribution of the original dataset. DC, on the other hand, aims to match the gradients of the network. However, PointNet is significantly larger than typical networks in image-based distillation tasks, making it difficult for DC to precisely align the gradients. As a result, it consistently underperforms compared to the proposed method. The proposed method outperforms both coreset selection and existing distillation techniques. When the PPC is small, accurately capturing the original distribution is challenging, leading less noticeable improvement. As the PPC increases, the proposed method more effectively matches the feature distributions significantly improving the performance.

4.2.2 Cross-Architecture Generalization

We also present the cross-architecture generalization performance in Table 2, where the synthetic dataset is optimized using PointNet [15] and evaluated on different backbone networks including PointNet++[16], DGCNN[23], PointConv [24], and PT [32], respectively. Unlike PointNet, the architectures used for generalization performance comparison follow hierarchical structure similar to CNN-based networks in image processing, exhibiting significantly different characteristics from those of PointNet. This difference makes it essential for the distilled dataset to retain rich and transferable features that generalize well toward different architectures. Due to significant information loss caused by the pooling operation, DM primarily captures the features specific to the PointNet, failing to preserve the original feature distribution. Consequently, the synthetic dataset lacks diverse structural details, resulting in poor generalization performance, particularly on hierarchical models. DC, while performing better than DM, also suffers from key limitations. It matches the gradients obtained from a trained PointNet, the synthetic dataset becomes overfitted to that specific architecture and fails to generalize toward other networks. Consequently, the generalization ability of DC degrades when applied to other architectures. In contrary, the proposed method addresses these limitations by directly matching the feature distributions extracted from randomly initialized network. By comparing feature maps containing richer and less architecture-biased information, the synthetic dataset captures the features relevant to the PointNet while following the original data distribution more closely. Wherease DC suffers from the overfitting and DM loses critical information, the proposed method yields higher adaptability showing strong generalization performance across various backbone networks.

Table 6: Performance comparison between DM and the proposed method at the PPC value of 3, where the proposed method was trained during the same time as DM.

Method	MN10	MN40	SN	SONN
DM	77.58	62.04	56.14	23.33
Ours	**83.64**	**70.56**	**63.34**	**33.06**

Table 7: Ablation study of the proposed semantic alignment (**SA**) method. Experiments were performed at the PPC value of 3.

Method	MN10	MN40	SN	SONN
Random	75.17	59.96	54.84	20.42
w/o **SA**	75.01	59.96	53.91	20.20
w/ **SA**	**84.42**	**72.08**	**63.74**	**31.84**

4.2.3 Cross-Dataset Generalization

To further evaluate the generalization capability of the distilled datasets across different domains, we provide an experimental result of cross-dataset experiment in Table 3. Specifically, we constructed the subsets of ShapeNet and ModelNet40, respectively, by selecting the 16 object classes shared between them. Then we performed the experiments in which the dataset was distilled on the subset of ShapeNet and tested on ModelNet40 (**SN** $\rightarrow$ **MN40**), and vice versa (**MN40** $\rightarrow$ **SN**). As shown in Table 3, the proposed method consistently achieves the best performance in both settings of the cross-dataset evaluation, indicating that our distilled datasets effectively capture geometric patterns that are useful regardless of the datasets.

4.2.4 Initialization Strategies

Table 4 presents the classification performance of DC [30], DM [29], and PCC [28] under different initialization strategies, including uniform noise, random selection [17], Herding [2, 1], and K-Center [19]. Note that the results for PCC are obtained using our implementation, where PCC is equivalent to performing DC with Herding initialization. When initialized with noise, DC fails to effectively optimize the synthetic dataset because the network has far more parameters than typical networks used for image dataset distillation, making it difficult to align the gradients. Therefore, the original dataset cannot properly guide the training process, leading to poor convergence. In contrast, the structured initializations such as random selection, Herding, or K-Center improve the performance, where the Herding performs the best by selecting the most representative samples. Unlike DC, DM directly matches feature representations rather than relying on gradients, making it less sensitive to initialization strategies. However, due to information loss, DM consistently underperforms DC, even when structured initializations are used. In contrast, the proposed method achieves favorable performance even when initialized from uniform noise. Furthermore, it significantly outperforms other approaches under heuristic initialization strategies, such as random selection. These results demonstrate that the proposed method is robust across diverse initialization schemes and effectively addresses the instability issues inherent in existing dataset distillation methods.

4.2.5 Training Time

Table 5 compares the training times between the proposed method and the existing methods. DM [29] achieves the fastest training time as it uses only a small subset of the overall features, however it often ignores many useful features leading to performance degradation. In contrast, DC [30] incurs a significantly higher computational cost as it aligns the gradients across all the parameters. Notably, for the largest dataset of ShapeNet [4], DC requires over 7 hours for training 50 times slower that DM that completes the training in just 9 minutes (0.15 hours). Since the proposed method is based on the distribution matching approach, its training time is marginally increased compared to DM, yet achieves significant improvement in dataset distillation performance. To further justify the computational overhead, we conducted an experiment where DM was trained with its default setting, whereas the proposed method was trained during the same training time as DM. As shown in Table 6, even under the same training time condition, our method consistently outperforms DM.

4.2.6 Qualitative Comparison of Synthetic Datasets

Figure 3 compares the resulting synthetic datasets distilled by using the proposed method and the existing methods, respectively. The first row shows the original datasets and the second row shows initialized point cloud objects. DC [30] fails to deviate significantly from the initialized objects and causes noise. Similarly, DM [29] also maintains the original shape of the initialized objects while

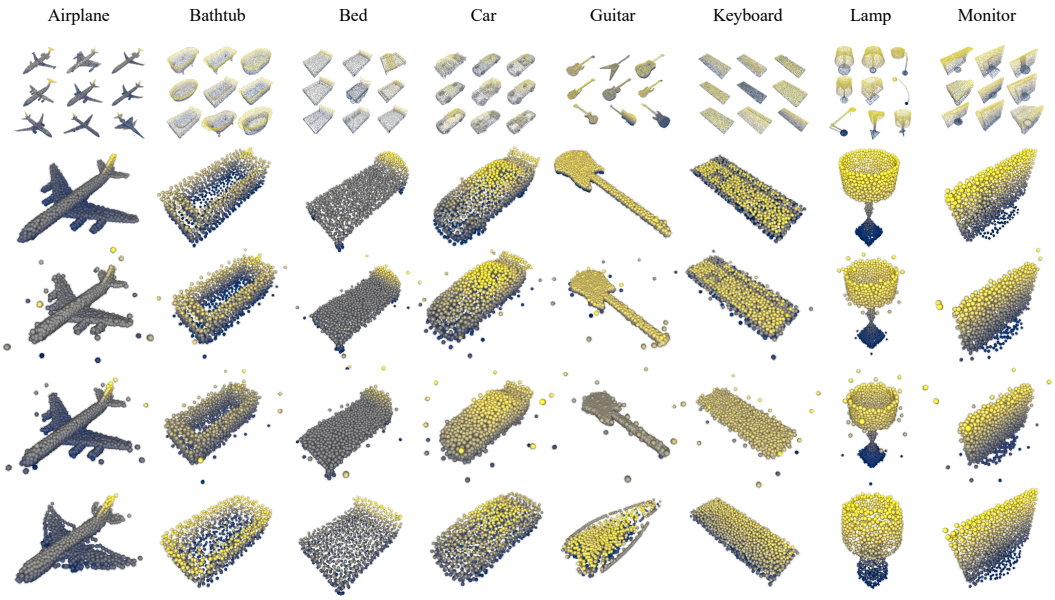


Figure 3: Comparison of synthetic datasets distilled from the ModelNet40 [25] dataset. Point clouds were colorized according to the y-coordinates. From top to bottom, the orginal dataset (first), initialized point cloud models (second), and the distilled synthetic datasets by using DC [30] (third), DM [29] (fourth), and the proposed method (fifth).

shifting certain points only, that hinders to capture meaningful structural changes. In contrast, as shown in the bottom row, the proposed method successfully preserves the overall semantic structures of 3D objects in each class, while selectively learning essential features. For example, in the airplane class, we observe significant changes in the edges and corners, that are critical for class discrimination, especially around the wings. Also, in the guitar class, the proposed method jointly optimizes the shape and orientation of the synthesized 3D objects.

4.3 Ablation Study

4.3.1 Effect of SADM Loss

Table 7 presents an ablation study on the impact of the proposed SADM loss on classification performance using PointNet [15]. Without preserving semantic alignment, the unordered nature of point clouds restricts effective feature matching, resulting in performance nearly identical to that of random selection [17]. This indicates that the model has not been properly optimized. In contrast, when semantic alignment is applied, accuracy improves across all datasets, as the aligned features provide a consistent and structured representation that facilitates more effective feature matching during optimization. These results confirm that preserving inherent semantic alignment via feature sorting enables the network to exploit well-aligned corresponding features, significantly enhancing classification performance.

4.3.2 Effect of Optimal Rotation Estimation

Table 8 presents the ablation study to evaluate the impact of joint optimization of synthetic dataset and rotation angles. We categorized the test datsets into the aligned, mixed, and rotated groups to analyze the effectiveness of the proposed rotation estimation. The aligned group consists of the datasets where the objects maintain consistent orientations, including ModelNet10 and ShapeNet [4]. The mixed group includes ModelNet40 [25] where only certain classes exhibit rotation variations. The rotated group is ScanObjectNN [20] where the objects exhibit arbitrary rotations across all classes. The baseline (first row) shows the performance without optimization which is identical to that of the random selection. In the second row, we present the results where only the orientation of the synthetic data is optimized using the proposed SADM loss, without updating the synthetic dataset. This leads to performance improvements across all the groups compared to random selection. The improvement

Table 8: Ablation study of the proposed optimal rotation estimation. Experiments were performed at the PPC value of 3.

$\mathcal{S}$	θ	Aligned	Mixed	Rotated
_	_	65.01	59.96	20.42
_	✓	66.22	62.17	24.89
✓	_	73.96	71.51	29.72
✓	✓	**74.35**	**72.08**	**31.84**

Table 9: Performance comparison of different point sorting schemes at the PPC value of 3. For fair comparison, the rotation parameter optimization was not applied.

Method	MN10	MN40	SN	SONN
Unsorted	75.21	59.96	53.91	20.20
Axis-Aligned	76.44	60.85	53.97	21.80
Z-order [14]	76.55	61.47	55.17	21.76
SADM	**83.17**	**67.52**	**60.71**	**27.66**

becomes more significant as the classes with rotation variation become more dominant. Optimizing the synthetic dataset further boosts the performance, with the best results achieved when both the synthetic data and rotation angles are optimized simultaneously. In particular, the accuracy on the rotated dataset improves significantly, demonstrating that the proposed joint optimization enables the model to be resilient to rotation variations.

4.3.3 Effect of Point Sorting

We compared the performance of the proposed SADM against two representative point sorting schemes. The axis-aligned sorting method simply orders the points in the ascending order of their z-coordinates, without considering any structural or semantic relationships. The Z-order sorting method [14] assigns an index to each point based on its 3D coordinates simultaneously, ensuring that points close in space receive nearby indices. As shown in Table 9, the existing sorting methods yield only marginal improvements over the baseline that does not use point sorting. Moreover, they fail to provide consistent semantic alignment across different samples. In contrast, the proposed SADM significantly outperforms these methods by achieving semantically consistent matching, thereby enhancing the effectiveness of dataset distillation for 3D point clouds.

4.4 Limitation

While the proposed SADM loss enhances the semantic consistency by sorting the features, it does not completely align semantically meaningful regions across different point cloud objects. Estimating optimal rotations also increases the complexity, particularly when the datasets are already well-aligned to the canonical axes.

5 Conclusion

We proposed a semantically aligned and orientation-aware dataset distillation framework for 3D point clouds. To address the inconsistency in point ordering between compared 3D objects, we devised a Semantically Aligned Distribution Matching (SADM) loss that compares sorted features within each channel. Additionally, we introduced learnable rotation angle parameters to estimate the optimal orientation of synthetic objects. The geometric structures and orientations of the synthetic objects are jointly optimized during dataset distillation. Experimental results on four widely used benchmark datasets—ModelNet10 [25], ModelNet40 [25], ShapeNet [4], and ScanObjectNN [20]—demonstrate that the proposed method outperforms existing distillation approaches while maintaining strong cross-architecture generalization. Furthermore, the effectiveness of each component is validated through extensive ablation studies.

Acknowledgement

This work was supported in part by the National Research Foundation of Korea (NRF) Grant funded by Korea Government [Ministry of Science and ICT (MSIT)] under Grant RS-2024-00392536; in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant funded by Korea Government (MSIT) (Leading Generative AI Human Resources Development) under Grant IITP-2025-RS-2024-00360227; in part by the Artificial Intelligence Gradate School Program, Ulsan National Institute of Science of Technology, under Grant RS-2020-II201336; and in part by the Artificial Intelligence Innovation Hub under Grant RS-2021-II212068.

References

[1] Eden Belouadah and Adrian Popescu. Scail: Classifier weights scaling for class incremental learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020.

[2] 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*, 2018.

[3] 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*, 2022.

[4] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. 2015.

[5] 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*, 2024.

[6] Jiawei Du, Yidi Jiang, Vincent YF 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*, 2023.

[7] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[8] Ziyao Guo, Kai Wang, George Cazenavette, HUI LI, Kaipeng Zhang, and Yang You. Towards lossless dataset distillation via difficulty-aligned trajectory matching. In *In International Conference on Learning Representations*, 2024.

[9] Hongcheng Li, Yucan Zhou, Xiaoyan Gu, Bo Li, and Weiping Wang. Diversified semantic distribution matching for dataset distillation. In *Proceedings of the ACM International Conference on Multimedia*, 2024.

[10] Yongqi Li and Wenjie Li. Data distillation for text classification. *arXiv preprint arXiv:2104.08448*, 2021.

[11] Dai Liu, Jindong Gu, Hu Cao, Carsten Trinitis, and Martin Schulz. Dataset distillation by automatic training trajectories. In *Proceedings of the European Conference on Computer Vision*. Springer, 2024.

[12] Aru Maekawa, Naoki Kobayashi, Kotaro Funakoshi, and Manabu Okumura. Dataset distillation with attention labels for fine-tuning bert. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 119–127, 2023.

[13] Aru Maekawa, Satoshi Kosugi, Kotaro Funakoshi, and Manabu Okumura. Dilm: Distilling dataset into language model for text-level dataset distillation. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 3138–3153, 2024.

[14] Guy M Morton. A computer oriented geodetic data base and a new technique in file sequencing. 1966.

[15] 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/CVF Conference on Computer Vision and Pattern Recognition*, 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 *Proceedings of the Advances in Neural Information Processing Systems*, 2017.

[17] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[18] 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*, 2023.

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

[20] 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*, 2019.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems*, 2017.

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

[23] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *In TRANSACTIONS ON GRAPHICS*, 2019.

[24] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[25] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2015.

[26] Tan Yu, Jingjing Meng, and Junsong Yuan. Multi-view harmonized bilinear network for 3d object recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[27] Hansong Zhang, Shikun Li, Pengju Wang, Dan Zeng, and Shiming Ge. M3d: Dataset condensation by minimizing maximum mean discrepancy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 2024.

[28] 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 ACM International Conference on Multimedia*, 2024.

[29] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023.

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

[31] 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*, 2023.

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

[33] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract and introduction are consistent with the main contributions of the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in the Section 4.4 of the main paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The paper includes a proposition, for which the assumption is clearly stated, correct proof is provided in the Supplementary material.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The experimental setup is described in detail in Section 4.1 and the supplementary material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The paper includes all necessary details to reproduce the main experimental results, and we are planning to release the official code upon publication.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All experimental details, including data splits, training settings, and hyperparameter choices, are provided in Section 4.1 and the supplementary material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports standard deviation as error bars to indicate variability.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Details on computational resources are provided in the supplementary material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We follow the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The paper have no negative societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper has no particular risks for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the datasets and models are cited properly.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We plan to release the implementation code upon publication. The released code will include the full implementation of proposed framework.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This work does not involve any crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This work does not involve any research with human subjects, so IRB approval is not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: No large language models (LLMs) were used in the development of the core methods in this research.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Appendix

A Proof of Proposition 1

Proposition 1. *Jointly optimizing the synthetic dataset $\boldsymbol{\mathcal{S}}$ and the rotation parameters $\boldsymbol{\theta} = (\theta_x, \theta_y, \theta_z)$ guarantees a lower or equal loss to that of optimizing $\boldsymbol{\mathcal{S}}$ alone.*

$$\min_{\{\boldsymbol{\mathcal{S}}, \boldsymbol{\theta}\}} \mathcal{L}_{\text{SADM}}(\boldsymbol{\mathcal{T}}, \mathcal{R}_{\boldsymbol{\theta}}(\boldsymbol{\mathcal{S}})) \leq \min_{\boldsymbol{\mathcal{S}}} \mathcal{L}_{\text{SADM}}(\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{S}}), \tag{13}$$

where $\mathcal{R}_{\boldsymbol{\theta}}$ denotes the rotation operator according to $\boldsymbol{\theta}$.

Proof. Let us consider two optimization objectives. The first is the baseline optimization, which optimizes only the synthetic dataset without applying any rotation

$$L := \min_{\boldsymbol{\mathcal{S}}} \mathcal{L}_{\text{SADM}}\left(\boldsymbol{\mathcal{T}},\, \boldsymbol{\mathcal{S}}\right). \tag{14}$$

The second is the proposed joint optimization over both the synthetic dataset $\boldsymbol{\mathcal{S}}$ and the rotation parameters $\boldsymbol{\theta} = (\theta_x, \theta_y, \theta_z)$, given

$$L^{\star} := \min_{\{\boldsymbol{\mathcal{S}},\, \boldsymbol{\theta}\}} \mathcal{L}_{\text{SADM}}\left(\boldsymbol{\mathcal{T}},\, \mathcal{R}_{\boldsymbol{\theta}}(\boldsymbol{\mathcal{S}})\right). \tag{15}$$

Consider a special case of the proposed joint optimization framework where no rotation is applied, i.e., the rotation parameters are fixed to $\boldsymbol{\theta} = (0, 0, 0)$. In such a case, $\mathcal{R}_{\boldsymbol{\theta}}(\boldsymbol{\mathcal{S}}) = \boldsymbol{\mathcal{S}}$, and the baseline optimization can be regarded as a special case of the proposed joint optimization. This implies that the feasible set of the baseline optimization is a subset of that of the joint optimization such that

$$\{(\boldsymbol{\mathcal{S}},\, \boldsymbol{\theta}) \,|\, \boldsymbol{\theta} = (0, 0, 0)\} \subseteq \{(\boldsymbol{\mathcal{S}},\, \boldsymbol{\theta})\}. \tag{16}$$

The joint optimization is performed over a larger feasible set than that of the baseline optimization, and therefore its optimal solution must not be higher than that of the baseline. Therefore,

$$L^{\star} \leq L. \tag{17}$$

B Expeimental Details

Datasets. Table 10 summarizes the number of training and testing samples for ModelNet10 [25], ModelNet40 [25], ShapeNet [4], and ScanObjectNN [20].

- **ModelNet10** consists of 10 categories of aligned 3D CAD models. There is no rotational variation.
- **ModelNet40** includes 40 categories from the same source to ModelNet10, with partially misaligned instances introducing moderate rotational variation.
- **ShapeNet** contains 55 categories of manually aligned 3D models without rotational variation.
- **ScanObjectNN** consists of 15 categories from real-world RGB-D scans with background clutter, occlusion, and sensor noise. Objects are unaligned, and rotation variation is inherent. We use the **PB_T50_RS** split, which includes perturbed background, translation jitter, rotation, and scaling.

Table 10: Train/test statistics for ModelNet10, ModelNet40, ShapeNet and ScanObjectNN.

	ModelNet10	ModelNet40	ShapeNet	ScanObjectNN
# of training samples	3991	9843	35708	11416
# of test samples	908	2468	10261	2882
# of classes	10	40	55	15
# of points for each sample	1024	1024	1024	1024

Network Architecture. We evaluate our method on five representative point-based architectures: PointNet [15], PointNet++ [16], DGCNN [23], PointConv [24], and Point Transformer [32].

- **PointNet** is a widely used neural network designed for processing 3D point clouds. We use PointNet as the backbone in dataset distillation model. While the original PointNet includes two transformation modules for aligning input and features, we only use the input transformation module. The rest of the architecture follows the standard PointNet design.
- **PointNet++** is used for cross-architecture evaluation. To reduce computational cost and accelerate training, we decrease the width of all MLP layers by half, while keeping the overall structure unchanged. We use two set abstraction modules with multi-scale grouping and one with single-scale grouping to extract point features, followed by two linear layers and a final classifier for prediction.
- **DGCNN** employs dynamic graph construction and EdgeConv layers to capture local geometric relationships by recomputing a k-NN graph ($k = 20$) at each layer in the feature space. We use the default architecture, which consists of four EdgeConv layers followed by two linear layers and a final classifier.
- **PointConv** extends convolution to point clouds by accounting for non-uniform point densities during feature learning. The architecture consists of three density-aware set abstraction layers, followed by two linear layers and a final classifier.
- **Point Transformer** applies self-attention mechanisms to point clouds to capture both local and global dependencies. We use the default configuration with four transformer blocks, followed by two linear layers and a final classifier.

Implementation Details. We optimized the synthetic dataset $\boldsymbol{\mathcal{S}}$ using stochastic gradient descent (SGD) with a learning rate of 0.01, a momentum of 0.5, a weight decay of 0, and a batch size of 8 per class sampled from the original dataset $\boldsymbol{\mathcal{T}}$, while the batch size of the synthetic dataset was set equal to the number of synthetic samples per class (PPC). The synthetic dataset optimization was performed for 1,500 iterations, and the corresponding configuration is summarized in Table 11a. To preserve fine-grained geometric details, the SADM loss was computed using the feature maps before the max pooling layer. Additionally, a secondary loss term was applied to the top-1 sorted feature values in each channel to emphasize semantically dominant regions. The loss weights λ_1 and λ_2 were determined based on PPC, and set to 0.002, 0.006, and 0.02 for λ_1, and 0.001, 0.003, and 0.01 for λ_2 when PPC was 1, 3, and 10, respectively.

The rotation parameters $\boldsymbol{\theta} = (\theta_x, \theta_y, \theta_z)$ were jointly optimized with the synthetic dataset using SGD with a momentum of 0.5 and a weight decay of 0. The learning rates were set to 0.5 for θ_x and θ_z, and 5.0 for θ_y, reflecting that the samples in datasets are vertically aligned. A step decay scheduler was used with a step size of 100 and a decay factor of 0.5. This setup is detailed in Table 11b.

For evaluation, we trained all backbone networks including PointNet, PointNet++, DGCNN, PointConv, and Point Transformer using SGD with a learning rate of 0.01, a momentum of 0.9, and a weight decay of 0.0005. A step decay scheduler was applied with a step size of 250 and a decay factor of 0.1. Each model was trained for 500 epochs with a batch size of 8. These test-time settings are listed in Table 11c.

All experiments were conducted on a single NVIDIA GeForce RTX 3090 GPU.

Table 11: Hyperparameter settings used for (a) optimizing the synthetic dataset, (b) optimizing the rotation parameters, and (c) evaluation network.

Hyperparameters	PPC1	PPC3	PPC10
Optimizer	SGD	SGD	SGD
Momentum	0.5	0.5	0.5
Weight Decay	0.0	0.0	0.0
Learning Rate	0.01	0.01	0.01
Iteration	1500	1500	1500
λ_1	0.002	0.006	0.02
λ_2	0.001	0.003	0.01
Batch Size ($\boldsymbol{\mathcal{T}}$)	8	8	8

(a)

Hyperparameters	
Optimizer	SGD
Momentum	0.5
Weight Decay	0.0
Learning Rate (θ_x)	0.5
Learning Rate (θ_y)	5.0
Learning Rate (θ_z)	0.5
Scheduler	StepLR
Step Size	100
Gamma	0.5

(b)

Hyperparameters	
Optimizer	SGD
Momentum	0.9
Weight Decay	0.0005
Learning Rate	0.01
Epochs	500
Batch Size	8
Scheduler	StepLR
Step Size	250
Gamma	0.1

(c)

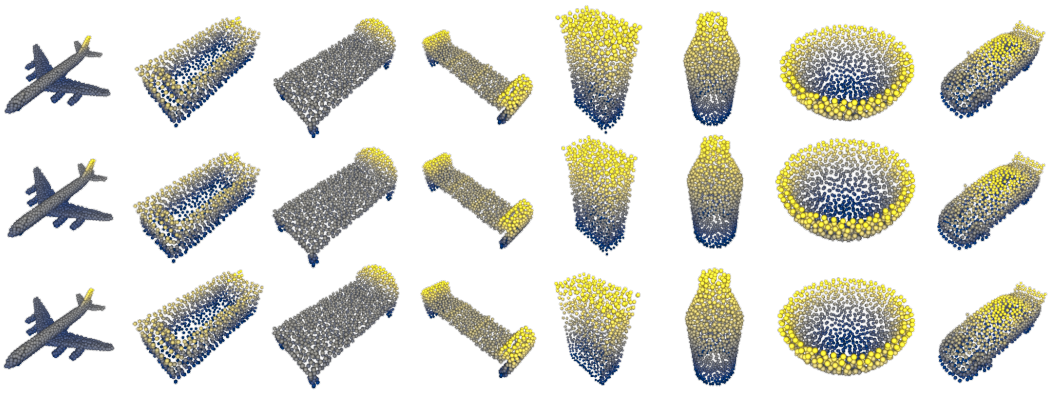


Figure 4: Visualization of synthetic samples of ModelNet40 [25] at PPC 1, obtained by using the random selection [17](top), DataDAM [18](middle), and MTT [3](bottom).

Baselines. All baseline methods were implemented on 3D point clouds using PointNet as the feature extractor, following their original designs. All methods were trained for 1,500 iterations using a batch size of 8 per class from the original dataset $\mathcal{T}$, and the training configurations are summarized in Table 12. Details for each method are as follows

- **DC** [30]: We followed the original framework and performed gradient matching. The number of inner and outer loop steps were set to 1/1/10 and 1/5/3 for PPC values of 1, 3, and 10, respectively
- **DM** [29]: Following the original setup, we used the feature vector obtained before the final classifier to compute the matching loss between the original and synthetic datasets.
- **MTT** [3]: We followed the original framework for trajectory matching, with modifications only in the hyperparameter settings.
- **DataDAM** [18]: We followed the original framework for feature matching, making only minor adjustments to hyperparameters.

Table 12: Hyperparameter settings of the baselines.

	DC	DM	MTT	DataDAM
Mode	PointNet	PointNet	PointNet	PointNet
Learning Rate	0.0001	1	0.0001	0.0001
Batch Size ($\mathcal{T}$)	8	8	8	8
Iteration	1500	1500	1500	1500
Inner Loop	1 / 1 / 10	–	–	–
Outer Loop	1 / 5 / 3	–	–	–

C Comparison with Additional Baselines

Table 13 presents the additional comparison results of the classification accuracy with DataDAM [18] and MTT [3]. For DataDAM and MTT, we used a possible learning rate to prevent divergence on the synthetic datasets. The performance of both DataDAM and MTT is almost similar to that of random selection [17], as shown in the second and third rows in Figure 4 where most points remain nearly unchanged from their random initialization in the first row.

D Additional Ablation Studies

To further support the effectiveness of the proposed rotation optimization method, we compared widely used rotation augmentation that applies random rotations to the training data to address the

Table 13: Classification accuracy of the proposed method compared with DataDAM [18] and MTT [3] initialized with randomly selected samples.

Datasets	**ModelNet10**			**ModelNet40**			**ShapeNet**			**ScanObjectNN**		
PPC	1	3	10	1	3	10	1	3	10	1	3	10
MTT	28.95	76.73	85.19	34.42	59.81	73.88	33.90	52.92	62.73	13.94	20.28	34.07
DataDAM	40.51	76.60	86.56	34.54	60.37	74.94	35.70	54.56	63.64	16.29	20.51	35.45
Ours	**44.70**	**84.96**	**87.79**	**55.80**	**72.08**	**80.07**	**50.20**	**63.74**	**68.35**	**17.29**	**31.84**	**43.91**

rotation variation of 3D point clouds. While this strategy is effective when dealing with datasets showing severe rotational variations, such as ScanObjectNN, it does not achieve good performance on datasets where the 3D objects are fully or partially aligned, such as ModelNet10, ShapeNet, and ModelNet40. As shown in Table 14, the proposed learnable rotation estimation method consistently improves the performance across all datasets, however, the random rotation augmentation introduces unnecessary variation in already aligned datasets, degrading the performance. This highlights the benefit of explicitly modeling the orientation via optimization over the naive augmentation.

Table 14: Performance comparison between the proposed rotation optimization and the random rotation augmentation (Aug.). All experiments were conducted at PPC 3.

Prop.	**Aug.**	**Aligned**	**Mixed**	**Rotated**
_	_	65.01	59.96	20.42
_	✓	58.92	52.56	31.42
✓	_	**74.35**	**72.08**	**31.84**

E Additional Qualitative Results

Figure 5 shows randomly sampled 3D models from ModelNet40. Figures 6, 7, and 8 visualize the synthesized models at PPC 1 obtained by using DC, DM, and the proposed method, respectively. While DC and DM maintain the geometric shapes of the original data with only slight movement of some points, the proposed method generates noise-free samples with high visual quality across all classes.

Figure 9 compares the optimization process for DM [29] and the proposed method. DM fails to converge even after multiple training iterations, however the proposed method progressively refines the point cloud models into more structured shapes.

Figure 10 shows the results of the proposed method on ModelNet40 at different PPC values: 1, 3, and 10. As PPC increases, the distilled synthetic datasets exhibit more diverse shapes and orientations. In particular, as shown in the last row, the human class appears in diverse poses. Additional qualitative results of the proposed method on the ShapeNet and ScanObjectNN datasets can be found in Figures 11 and 12, respectively.

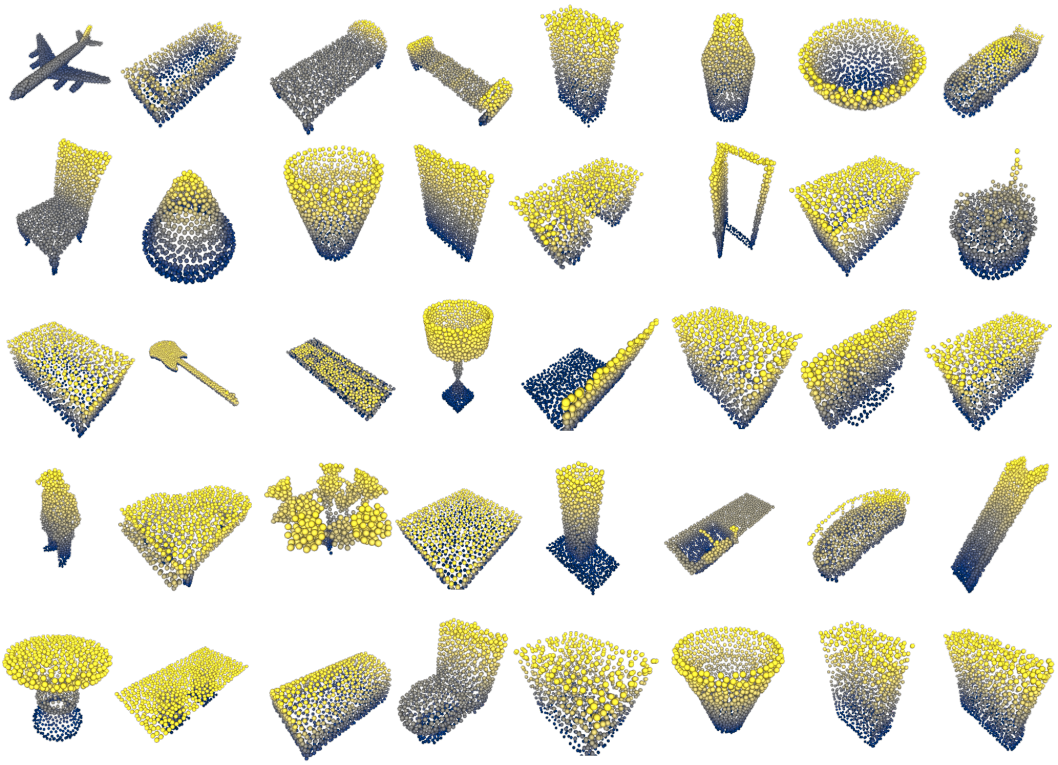


Figure 5: Randomly sampled 3D models of all classes from ModelNet40 [25].

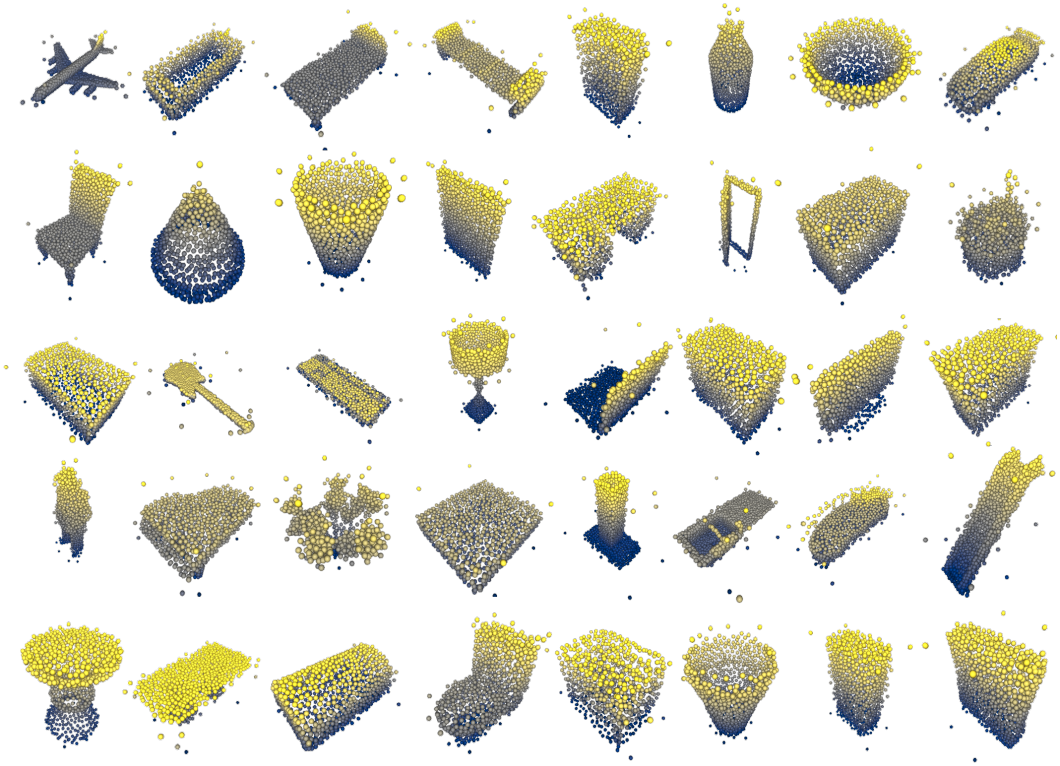


Figure 6: The synthetic dataset distilled from ModelNet40 by using DC at PPC 1.

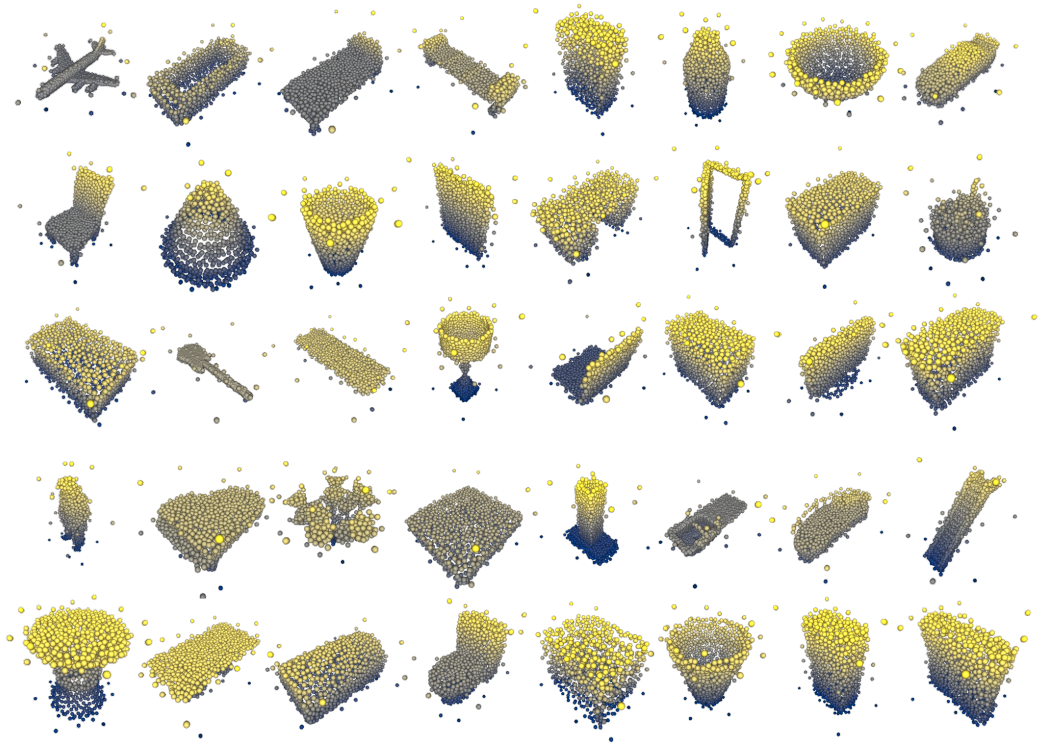


Figure 7: The synthetic dataset distilled from ModelNet40 by using DM at PPC 1.

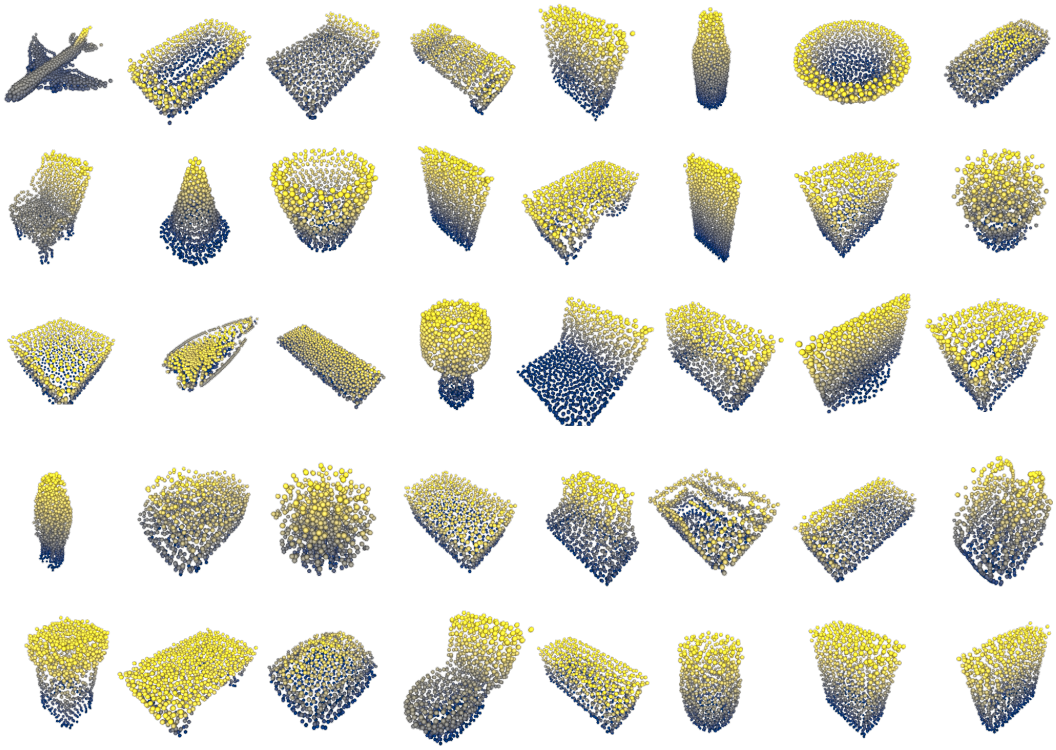
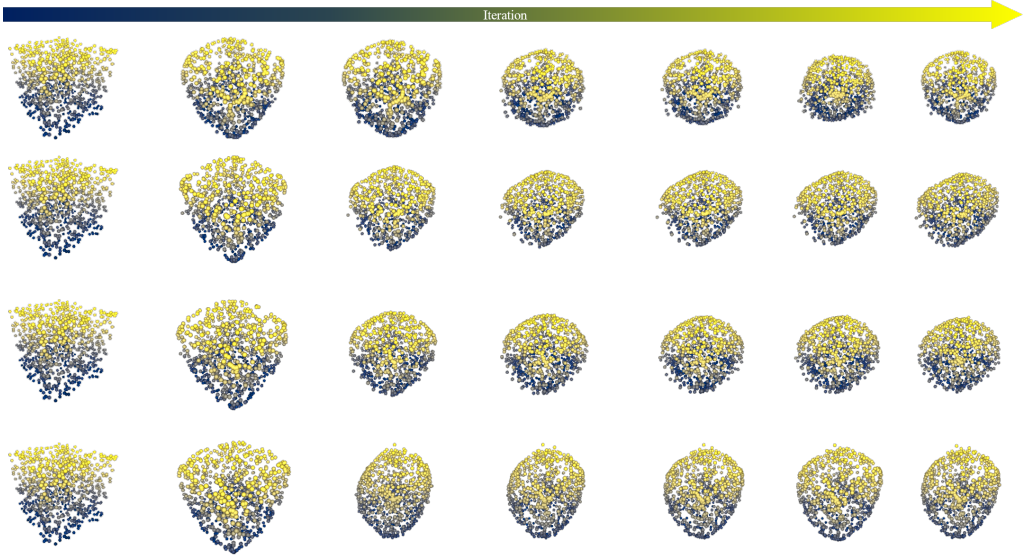
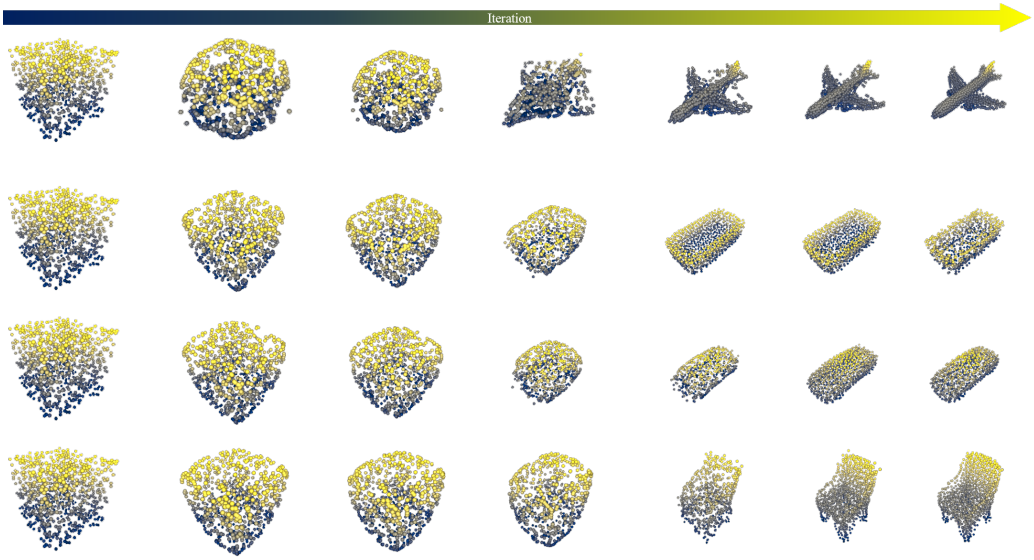


Figure 8: The synthetic dataset distilled from ModelNet40 by using the proposed method at PPC 1.

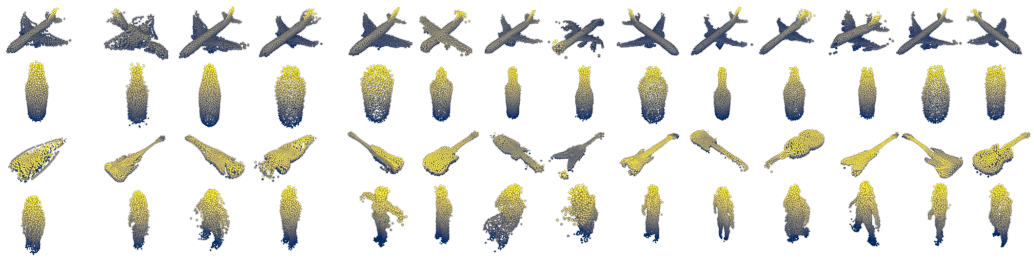


(a)



(b)

Figure 9: Optimization process in the ModelNet40 at PPC 1 initialized from uniform noise, distilled by using (a) DM [29] and (b) the proposed method.



(a) PPC1 (b) PPC3 (c) PPC10

Figure 10: The synthetic dataset distilled from ModelNet40 by using the proposed method at three PPC values of 1, 3, and 10, respectively.

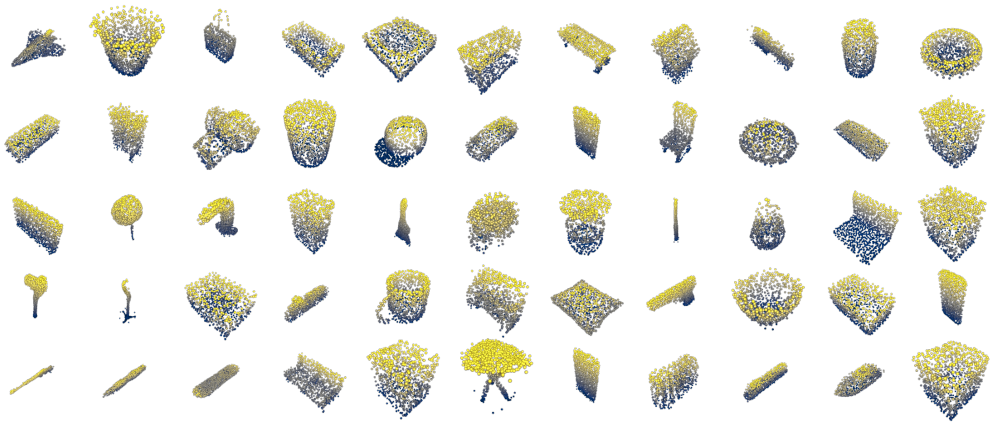


Figure 11: The synthetic dataset distilled from ShapeNet by using the proposed method at PPC 1.

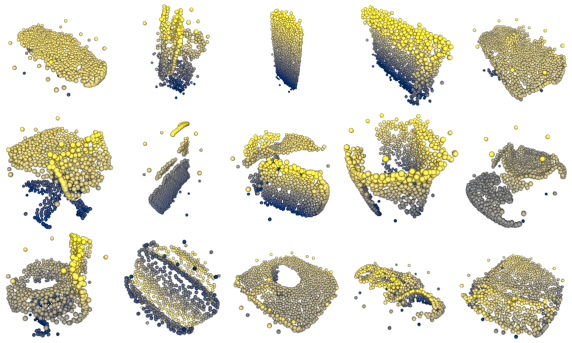


Figure 12: The synthetic dataset distilled from ScanObjectNN by using the proposed method at PPC 1.