Learning 3D Representations from Procedural 3D Programs

Anonymous CVPR submission

Paper ID 0000

Abstract

Self-supervised learning has emerged as a promising approach for acquiring transferable 3D representations from unlabeled 3D point clouds. Unlike 2D images, which are widely accessible, acquiring 3D assets requires specialized expertise or professional 3D scanning equipment, making it difficult to scale and raising copyright concerns. To address these challenges, we propose learning 3D representations from procedural 3D programs that automatically generate 3D shapes using simple primitives and augmentations.

Remarkably, despite lacking semantic content, the 3D representations learned from the procedurally generated 3D shapes perform on par with state-of-the-art representations learned from semantically recognizable 3D models (e.g., airplanes) across various downstream 3D tasks, including shape classification, part segmentation, and masked point cloud completion. We provide a detailed analysis on factors that make a good 3D procedural programs. Extensive experiments further suggest that current self-supervised learning methods on point clouds do not rely on semantics of 3D shapes, shedding light on the nature of 3D representations learned.

1. Introduction

Self-supervised learning (SSL) aims at learning representations from unlabeled data that can transfer effectively to various downstream tasks. Inspired by the success of SSL in language [10] and image representation learning [14, 15], SSL for 3D point cloud understanding has gained considerable interest [26, 37, 55]. Recently, Point-MAE [26] and its follow-ups [41, 59, 61] exploit the masked autoencoding scheme for 3D point cloud representation learning, showing substantial improvements in various 3D shape understanding tasks (*e.g.*, shape classification, part segmentation, and scene instance segmentation).

However, unlike language and image data, which are abundantly available, 3D assets are less accessible due to the expertise required for creating 3D shapes using specialized software (*e.g.*, Blender) or professional 3D scanners (*e.g.* LiDAR sensors). This scarcity of 3D shapes, known as data desert [11], limits the scalability of existing representation learning methods. Recent efforts have sought to expand 3D point cloud datasets at both the object level [8, 9] and scene level [3, 12, 53]. However, challenges unique to 3D data collection — such as copyright issues, format diversity, and scalability — remain unresolved.

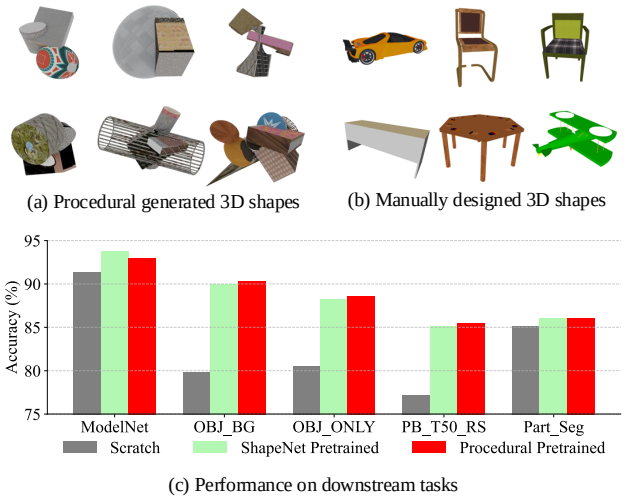


Figure 1. Self-supervised learning from **(a)** procedurally generated 3D shapes [17, 45, 49, 50] performs comparably to learning from **(b)** ShapeNet models that are semantically meaningful [4] across various downstream 3D understanding tasks. Both outperforms training from scratch significantly. In **(c)**, the x-axis represents various tasks and benchmarks: ModelNet40 [22] and three variants of ScanObjectNN [34] for shape classification, and ShapeNet-Part [54] for part segmentation.

To address these challenges, we explore learning point cloud representations from *solely synthetic data* generated via procedural 3D programs [17, 45, 49, 50] — examples are shown in Fig. 1a. Our data generation pipeline begins with sampling shapes from a set of simple 3D primitives (*e.g.*, cubes, cylinders, and spheres). These primitives undergo affine transformations (*e.g.*, scaling, translation, and rotation) and are combined to create diverse geometries. We then augment the composed shapes using predefined operations (*e.g.*, Boolean operations) to further enhance topological diversity, and uniformly sample 3D surface points

from these shapes for point cloud representation learning. We generate 150K synthetic 3D point clouds in 600 CPU hours, with a scalable pipeline capable of producing unlimited 3D shapes free of copyright concerns. Additionally, we generate 2K synthetic scenes, including room layouts and objects following the described steps, requiring 500 CPU hours. This leads us to two key questions:

- *Can existing self-supervised learning methods effectively capture rich and meaningful 3D representations from procedurally generated shapes alone?*
- *How do these representations compare to those learned from human-crafted, semantically rich 3D models?*

We investigate learning 3D representations exclusively from synthetic shapes generated via procedural programs, without relying on human-crafted data. To validate this approach, we conduct extensive benchmarks across representative self-supervised learning methods, including Point-MAE [26], Point-M2AE [59], PCP-MAE [61], and Masked Scene Contrast (MSC) [41]. Specifically, we evaluate 3D object SSL methods on tasks such as shape classification, part segmentation, and masked point cloud reconstruction, while 3D scene SSL methods are evaluated on semantic and instance segmentation tasks. Our main findings are:

- Despite lacking semantics, SSLs from solely synthetic data achieve performance comparable to their ShapeNet-pretrained counterparts. We also validate the effectiveness of procedurally generated data in scene-level 3D understanding tasks (Fig. 1c, Tab. 1 - Tab. 5).
- We provide detailed insights into the factors that influence the quality of procedurally generated 3D datasets. We observe that learning performance improves notably with greater geometric diversity and increased dataset size (Tab. 4 and Fig. 5).
- Our in-depth analysis reveals structural similarities between 3D shape representations from models pretrained on synthetic 3D shapes and semantically meaningful 3D shapes, providing insights into the nature of the learned representations (t-SNE visualization in Fig. 7).

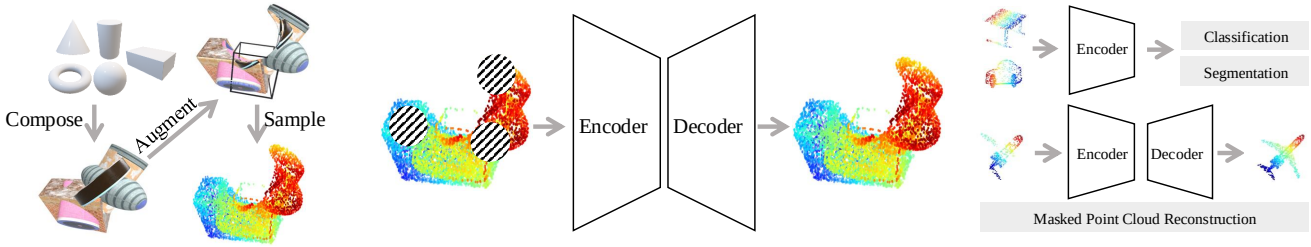
To our best knowledge, this is the first systematic large-scale study on SSLs from procedural 3D programs. Our work is inspired by and builds upon a recent study that successfully trained large 3D reconstruction models exclusively on procedurally generated shapes [17, 45]. Our exploration is also closely related to prior efforts that learn image representations from procedural programs [1, 2]. Concurrent to our work, Yu *et al*. [56] demonstrated that procedurally generated videos can perform as effectively as natural videos for self-supervised video representation learning. Furthermore, our study is orthogonal yet complementary to recent efforts in scaling up 3D shape datasets [8, 9, 53].

2. Related Work

3D Object Datasets. Large-scale datasets are essential in advancing 3D deep learning. Unlike 2D images or videos, building and annotating 3D models requires expertise with professional 3D software or scanning equipment, making the process more costly and time-intensive. Despite these challenges, significant efforts have been made to curate extensive 3D shape datasets [4, 6, 8, 12, 25, 30, 32, 39, 43]. For example, ShapeNet provides 3 million CAD models, with 51K of them being clean, high-quality models, which serves as the standard benchmark for training and evaluating models in the 3D object recognition community. A more significant limitation of ShapeNet is its strong bias towards rigid, man-made artifacts, reflecting the inherent bias of its source 3D model repositories. More recently, Objaverse-XL [9] expanded the 3D dataset to 10.2 million models, though scaling up has introduced challenges, including increased noise, format diversity, and unresolved copyright and legal issues. In contrast, we explore procedural 3D programs that generate 3D shapes from simple primitives. This synthetic approach can theoretically produce an unlimited number of 3D shapes without licensing concerns. Moreover, procedural 3D programs provide a principled approach to mitigating biases inherently present in manually curated datasets [1].

Supervised Learning for 3D Point Clouds. Unlike 2D images, 3D point clouds are inherently unordered and have an irregular structure, leading to extensive research on specialized neural network architectures [19, 27, 28, 38, 40, 42, 48, 62]. For instance, PointNet [27] introduces permutation-invariant operators and pooling layers feature aggregation across 3D points. PointNet++ [28] builds on this by incorporating a hierarchical spatial structure to capture more localized geometric information. DGCNN [38] adopts a graph-based approach, constructing a graph from input point clouds and applying a graph CNN to aggregate features from unordered 3D points. More recent models, such as the Point Transformer [40, 42, 62], utilize a modified transformer architecture tailored specifically for 3D point cloud processing. In our work, we employ a standard transformer architecture, aligning with recent trends in self-supervised learning [15, 26, 55].

Self-supervised Learning for Point Clouds. Self-supervised learning (SSL) for 3D representations aims to learn features that transfer well across diverse point cloud tasks, such as shape classification, object detection, and part segmentation. Recent SSL approaches for point clouds generally fall into two categories: contrastive learning [5, 14, 46] and masked autoencoding [11, 13, 26, 29, 37, 55, 57, 58, 60]. PointContrast [46] and DepthContrast [5] use an instance discrimination task [14] to learn 3D representations. OcCo [37] learns point cloud representations by reconstructing the original point cloud from occluded views,



(a) Procedural 3D generation (b) Learning 3D representations with masked autoencoding (c) Downstream tasks

Figure 2. **Learning from procedural 3D programs.** **(a)** Synthetic 3D point clouds are generated by sampling, compositing, and augmenting simple primitives using procedural 3D programs [45]. **(b)** We experiment with multiple state-of-the-art self-supervised learning frameworks for learning 3D representations from synthetic data. Here, we illustrate the pretraining pipeline using Point-MAE [55], naming this variant *Point-MAE-Zero*, where "Zero" emphasizes the absence of any human-made 3D shapes. **(c)** We evaluate the pretrained models across various 3D shape understanding tasks.

while IAE [51] trains an autoencoder to recover implicit features from point cloud inputs. Inspired by the success of masked autoencoding in vision [15] and language representation learning [10], Point-BERT [55] and Point-MAE [26] learn representations by predicting masked regions, using transformers as the underlying architecture. More works have been built upon Point-MAE: Point-M2AE [59] introduces a pyramid architecture for both the encoder and decoder to incorporate multi-scale and hierarchical information in 3D shapes. More recently, PCP-MAE [61] highlights the issue of centroid leakage, which trivializes the pretext task in the PointMAE framework, and proposes predicting centroids as an additional objective to strengthen masked point reconstruction. 3D scene SSLs directly takes in scene-centric point clouds is an exciting research direction [16, 21, 36, 41, 47]. In this work, we adopt Point-MAE, Point-M2AE, PCP-MAE and MSC, as our primary self-supervised learning framework due to its state-of-the-art performance across various 3D understanding tasks. For ablation studies, we use Point-MAE as the baseline, as it serves as the foundational approach in this research direction and provides the most representative evaluation.

Learning from Synthetic Data Synthetic data has become popular in computer vision, especially in scenarios where ground-truth annotations are difficult to obtain or where privacy and copyright issues arise. State-of-the-art performance in mid-level or 3D vision tasks is often achieved through training on synthetic data, including tasks like optical flow [33], depth estimation [52], dense tracking [18], relighting [49], novel view synthesis [50], and material estimation [20]. Procedurally generated synthetic data has also been explored for self-supervised representation learning in images [1, 2] and videos [56], and more recently for multi-view feed-forward 3D reconstruction [45]. Concurrent with our study, Yu *et al*. [56] investigates self-supervised video representation learning from procedurally generated images and videos. In this work, we explore self-supervised representation learning for point clouds, using synthetic 3D shapes generated by procedural 3D programs.

3. Learning from Procedural 3D Programs

We first introduce the procedural 3D programs [45, 49, 50] for generating unlimited number of synthetic 3D shapes using composition of simple primitive shapes (*e.g*., cylinders) and shape augmentation (Sec. 3.1). We then describe the masked autoencoding scheme [26, 37, 55] for learning 3D representations from synthetic 3D datasets (Sec. 3.2).

3.1. Procedural 3D programs

There is a line of work synthesizing procedural 3D shapes for vision tasks such as novel view synthesis [50], relighting [49], and material estimation [20]. Following recent methods [45] that use procedural 3D shapes for sparse-view reconstruction, we address self-supervised 3D representation learning from purely synthetic datasets. Fig. 2a illustrates our data pipeline:

(1) Randomly sample $\mathbf{K}$ primitive shapes (cubes, spheres, cylinders, cones, tori) and apply affine transformations to combine them;

(2) Apply geometric augmentations (e.g., boolean differences, wireframe conversions) to enrich shape diversity (see [45] for details);

(3) Uniformly sample $\mathbf{N}$ surface points per synthesized shape as inputs for representation learning (Fig. 2b).

We experiment with various shape-generation configurations, such as changing the number of sampled primitives and applying augmentations. By default, each dataset consists of 150K shapes with $\mathbf{N} = 8192$ points each. Sec. 4 further analyzes the effects of dataset size and shape complexity on learned representations.

In order to generate procedural 3D scenes, we follow MegaSynth [17] to procedurally generate 2K synthetic 3D scenes. Specifically, we first generate a floor plan and generate procedural 3D shapes with the above pipeline and place procedural 3D shapes in the scene based on the generated floor plan. We include details on how we generate 2K procedural 3D scenes in the supplementary material.

3.2. Procedural Pretraining

Pretraining. As depicted in Fig. 2b, we adopt Point-MAE [15], Point-M2AE [59], PCP-MAE [61] to train on procedurally generated 3D shapes. These methods rely on a masked autoencoding scheme [10, 15, 55], where the input point cloud is split into irregular patches and a large portion of them (60% by default) is randomly masked. A Transformer-based encoder–decoder network then attempts to reconstruct these masked patches, thereby learning 3D representations. Concretely, the reconstruction loss is computed as the L_2 Chamfer Distance between the predicted point patches P_{pre} and the ground-truth patches P_{gt}:

$$L = \frac{1}{|P_{\text{pre}}|} \sum_{a \in P_{\text{pre}}} \min_{b \in P_{\text{gt}}} \|a - b\|_2^2 + \frac{1}{|P_{\text{gt}}|} \sum_{b \in P_{\text{gt}}} \min_{a \in P_{\text{pre}}} \|a - b\|_2^2. \quad (1)$$

For scene-level SSLs, we adopt MSC [41], which combines masked auto-encoding and contrasive learning, to train on procedurally generated 3D scenes. While earlier self-supervised approaches [37, 55] often train on human-crafted 3D models (*e.g.*, ShapeNet [4]), here we focus on purely synthetic data. We thus call add suffix *-Zero* to emphasize that it is trained exclusively on procedural 3D shapes with *zero* human-designed content. For clarity, we use the suffix *-SN* to indicate baselines trained on ShapeNet.

Downstream Probing. We evaluate baselines on several 3D tasks, as summarized in Fig. 2c. For shape classification, we augment the pretrained Transformer encoder with a three-layer MLP classification head. For part segmentation, we aggregate features from the 4th, 8th, and final layers of the encoder, upsample them to all 2048 input points, and employ a segmentation head. For masked point cloud reconstruction, we use both the pretrained encoder and decoder with no architectural modifications. For scene-level methods, we use both instance segmentation and semantic segmentation finetuned from the pretrained SSLs with a linear prediction head. Detailed implementation settings are in the supplementary material.

4. Experiments

We present a comprehensive evaluation of 3D shape representations pretrained with procedural 3D programs across various downstream object-level and scene-level tasks, including object classification, part segmentation, and 3D scene understanding (Sec. 4.1 – 4.4). We further provide an in-depth analysis of model behavior and ablation studies (Sec. 4.5). For each downstream task, we report the performance of relevant existing methods as a reference and focus on comparisons with SSLs pretrained on manually-curated 3D datasets, as well as models trained from scratch. Specifically, for object-level 3D understanding tasks, we evaluate the following three pretraining strategies: (1) *Scratch*: All network parameters are randomly initialized, with no pretraining. (2) *ShapeNet Pretrained (SN)*: Pretrained on 41,952 models in the ShapeNet [4] training split, relying on the officially released weights. (3) *Procedural 3D Programs Pretrained (Zero)*: Pretrained on 150K procedurally generated 3D models, using no human-crafted shapes. For scene-level tasks, we compare SSL models pretrained on ScanNet [7] and procedurally generated 3D scenes.

Methods	ModelNet40	OBJ-BG	OBJ-ONLY	PB-T50-RS
PointNet [27]	89.2	73.3	79.2	68.0
SpiderCNN [48]	92.4	77.1	79.5	73.7
PointNet++ [28]	90.7	82.3	84.3	77.9
DGCNN [38]	92.9	86.1	85.5	78.5
PointCNN [19]	–	86.1	85.5	78.5
PTv1 [62]	93.7	–	–	–
PTv2 [40]	94.2	–	–	–
OcCo [37]	92.1	84.9	85.5	78.8
Point-BERT [55]	93.2	87.4	88.1	83.1
Point-MAE-Scratch [55]	91.4	79.9	80.6	77.2
Point-MAE-SN [26]	**93.8**	90.0	88.3	85.2
Point-MAE-Zero	93.0	**90.4**	**88.6**	**85.5**
Point-M2AE-Scratch	92.2	90.0	87.6	85.6
Point-M2AE-SN [59]	**94.0**	**91.2**	88.8	86.4
Point-M2AE-Zero	92.9	90.4	**89.8**	**87.0**
PCP-MAE-Scratch	91.5	88.8	88.5	83.8
PCP-MAE-SN [61]	**94.0**	**95.5**	**94.3**	90.4
PCP-MAE-Zero	92.4	94.0	92.3	**90.5**

Table 1. **Object Classification.** We evaluate the object classification performance on ModelNet40 and three variants of ScanObjectNN. Classification accuracy (%) is reported (*higher is better*). **Top**: Performance of existing methods with various neural network architectures and pretraining strategies. **Bottom**: Comparison with our baseline methods.

4.1. Object Classification

Benchmarks. We use ModelNet40 [44] and ScanObjectNN [34] as the benchmarks for the shape classification task. ModelNet40 contains 12,311 clean 3D CAD objects across 40 categories, with 9,843 samples for training and 2,468 for testing. Following Point-MAE, we apply random scaling and translation as data augmentation during training, and a voting strategy during testing [22]. Following prior works [26, 37, 55], we also evaluate the few-shot classification performance on ModelNet40. ScanObjectNN is a more complex real-world 3D dataset, consisting of approximately 15,000 objects across 15 categories, with items scanned from cluttered indoor scenes. We report results on three ScanObjectNN variants: OBJ-BG, OBJ-ONLY, and PB-T50-RS, the latter being the most challenging due to its additional noise and occlusions.

Transfer Learning. Table 1 summarizes object classification results across several settings. On ModelNet40, the "-Zero" variants (*e.g.*, Point-MAE-Zero, Point-M2AE-Zero, PCP-MAE-Zero) generally fall slightly behind their ShapeNet-pretrained counterparts ("-SN"), reflecting the larger domain gap between synthetic shapes and the clean 3D models in ModelNet40. By contrast,

Methods	5w/10s	5w/20s	10w/10s	10w/20s
DGCNN-rand [38]	31.6±2.8	40.8±4.6	19.9±2.1	16.9±1.5
DGCNN-OcCo [38]	90.6±2.8	92.5±1.9	82.9±1.3	86.5±2.2
Transformer-OcCo [37]	94.0±3.6	95.9±2.3	89.4±5.1	92.4±4.6
Point-BERT [55]	94.6±3.1	96.3±2.7	91.0±5.4	92.7±5.1
Point-MAE-Scratch [55]	87.8±5.2	93.3±4.3	84.6±5.5	89.4±6.3
Point-MAE-SN [26]	**96.3±2.5**	**97.8±1.8**	**92.6±4.1**	**95.0±3.0**
Point-MAE-Zero	95.4±2.5	97.7±1.6	91.3±5.1	95.0±3.5
Point-M2AE-Scratch	87.5±2.6	90.0±5.5	86.4±3.2	89.6±4.3
Point-M2AE-SN* [26]	**93.4±3.1**	**96.2±1.5**	**91.8±4.5**	**92.9**±3.2
Point-M2AE-Zero	91.4±1.8	94.2±2.2	88.3±3.7	91.0±2.9
PCP-MAE-Scratch	86.4±2.6	85.0±6.0	88.9±4.1	90.7±4.2
PCP-MAE-SN [61]	**97.4±2.3**	**99.1±0.8**	**93.5±3.7**	**95.9±2.7**
PCP-MAE-Zero	95.3±3.4	98.4±1.4	91.5±4.4	94.7±3.0

Table 2. **Few-shot classification on ModelNet40.** We evaluate performance on four n-way, m-shot configurations. For example, 5w/10s denotes a 5-way, 10-shot classification task. The table reports the mean classification accuracy (%) and standard deviation across 10 independent runs for each configuration. **Top**: Results from existing methods for comparison. **Bottom**: Comparison with our baseline methods. Note that results for Point-M2AE-SN are reproduced using publicly available code with our own configuration, as the original configuration was not provided.

Methods	mIoU_I	aero	bag	cap	car	chair	earphone	guitar	knife
PointNet [27]	83.7	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9
PointNet++ [28]	85.1	82.4	79.0	87.7	77.3	90.8	71.8	91.0	85.9
DGCNN [38]	85.2	84.0	83.4	86.7	77.8	90.6	74.7	91.2	87.5
OcCo [37]	85.1	83.3	85.2	88.3	79.9	90.7	74.1	91.9	87.6
Point-BERT [55]	85.6	84.3	84.8	88.0	79.8	91.0	81.7	91.6	87.9
Point-MAE-Scratch [55]	85.1	82.9	**85.4**	87.7	78.8	90.5	**80.8**	91.1	**87.7**
Point-MAE-SN [26]	**86.1**	84.3	85.0	88.3	80.5	91.3	78.5	**92.1**	87.4
Point-MAE-Zero	**86.1**	**85.0**	84.2	**88.9**	**81.5**	**91.6**	76.9	**92.1**	87.6
Point-M2AE-Scratch	84.7	85.1	86.8	88.6	**81.1**	91.5	79.9	**92.1**	87.8
Point-M2AE-SN [59]	**85.0**	84.5	87.2	**89.3**	**81.1**	**91.8**	**80.1**	92.0	**89.2**
Point-M2AE-Zero	84.9	**85.3**	**87.3**	88.7	**81.1**	91.7	79.4	91.9	88.2
PCP-MAE-Scratch	83.8	84.3	83.1	88.7	80.3	91.2	77.1	92.0	**88.1**
PCP-MAE-SN [61]	84.3	**85.0**	84.0	**88.7**	81.0	**91.6**	77.6	91.8	87.6
PCP-MAE-Zero	**84.4**	84.6	**84.3**	88.5	**81.7**	91.5	**81.1**	**92.1**	87.0

Methods	lamp	laptop	motor	mug	pistol	rocket	skateboard	table
PointNet [27]	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
PointNet++ [28]	83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6
DGCNN [38]	82.8	95.7	66.3	94.9	81.1	63.5	74.5	82.6
OcCo [37]	84.7	95.4	75.5	94.4	84.1	63.1	75.7	80.8
Point-BERT [55]	85.2	95.6	75.6	94.7	84.3	63.4	76.3	81.5
Point-MAE-Scratch [55]	85.3	95.6	73.9	**94.9**	83.5	61.2	74.9	80.6
Point-MAE-SN [26]	**86.1**	**96.1**	75.2	94.6	84.7	63.5	77.1	**82.4**
Point-MAE-Zero	86.0	96.0	**77.8**	94.8	**85.3**	**64.7**	**77.3**	81.4
Point-M2AE-Scratch	85.7	96.0	76.4	**95.4**	**85.5**	63.8	76.3	82.4
Point-M2AE-SN [59]	**86.4**	95.8	**77.7**	95.3	85.2	**65.3**	**77.0**	82.2
Point-M2AE-Zero	85.8	**96.2**	76.6	94.9	84.8	64.4	76.8	**82.5**
PCP-MAE-Scratch	84.9	95.0	76.0	95.0	85.0	63.2	75.4	81.0
PCP-MAE-SN [61]	85.8	**96.4**	76.1	**95.2**	84.8	**64.0**	**77.4**	**81.4**
PCP-MAE-Zero	**86.0**	96.1	**76.6**	94.6	**85.1**	63.6	76.8	80.4

Table 3. **Part Segmentation Results.** We report the mean Intersection over Union (IoU) across all instances (mIoU_I) and the IoU (%) for each category on the ShapeNetPart benchmark (*higher values indicate better performance*).

on ScanObjectNN—which contains real-world scans with broader geometric variability—the "-Zero" models often match or exceed the performance of their "-SN" counterparts. For instance, PCP-MAE-Zero outperforms PCP-MAE-SN on the PB-T50-RS variant, and Point-M2AE-Zero closely matches or exceeds Point-M2AE-SN in several cases. These findings indicate that the diverse geometry in procedurally synthesized data can be advantageous for certain real-world tasks. Meanwhile, all pretrained models (including both "-SN" and "-Zero") surpass their respective from-scratch baselines and outperform existing self-supervised approaches (*e.g.*, OcCo [37] and Point-BERT [55]).

Few-shot Classification. We evaluate few-shot classification on ModelNet40 using standard n-way, m-shot protocols, where n denotes the number of randomly selected classes and m the number of examples per class. Each evaluation samples 20 unseen instances from each class. We repeat this procedure 10 times, reporting mean accuracy (%) and standard deviation. Table 2 presents results for $n = \{5, 10\}$ and $m = \{10, 20\}$. Similar to transfer learning experiments, Point-MAE-Zero performs slightly below Point-MAE-SN, likely due to the larger domain gap between procedural shapes and ModelNet40 data. Nevertheless, both methods significantly outperform scratch-trained models and previous approaches such as DGCNN [38] and Transformer-OcCo [37].

4.2. Part Segmentation

The 3D part segmentation task aims to assign a part label to each point in a shape. We evaluate our methods and baselines on ShapeNetPart [54], which contains 16,881 models across 16 object categories. Consistent with previous works [26, 27, 55], we sample 2,048 points from each shape, resulting in 128 patches in our masked autoencoding pipeline (see Sec. 3).

Table 3 presents the mean Intersection-over-Union (mIoU) across all instances, along with per-category IoU. Across various models, both Point-MAE-Zero and Point-MAE-SN deliver comparable performance, indicating that procedurally generated shapes can learn robust 3D representations without explicit semantic content. Similarly, Point-M2AE-Zero and PCP-MAE-Zero achieve results on par with their ShapeNet-pretrained counterparts, further highlighting the versatility of procedural data in self-supervised representation learning.

In line with our observations in Sec. 4.1, the "-Zero" and "-SN" models surpass scratch-trained baselines and earlier methods that use different architectures [27, 28, 38] or alternative pretraining strategies [37, 55]. Despite lacking high-level semantic cues, these procedurally trained autoencoders still capture sufficient geometric structure to achieve strong segmentation performance.

4.3. Masked Point Cloud Completion

The masked point cloud completion task aims to reconstruct masked regions of input 3D point clouds, serving as a self-supervised pretext for learning 3D representations [26] (see Fig. 2 and Sec. 3).

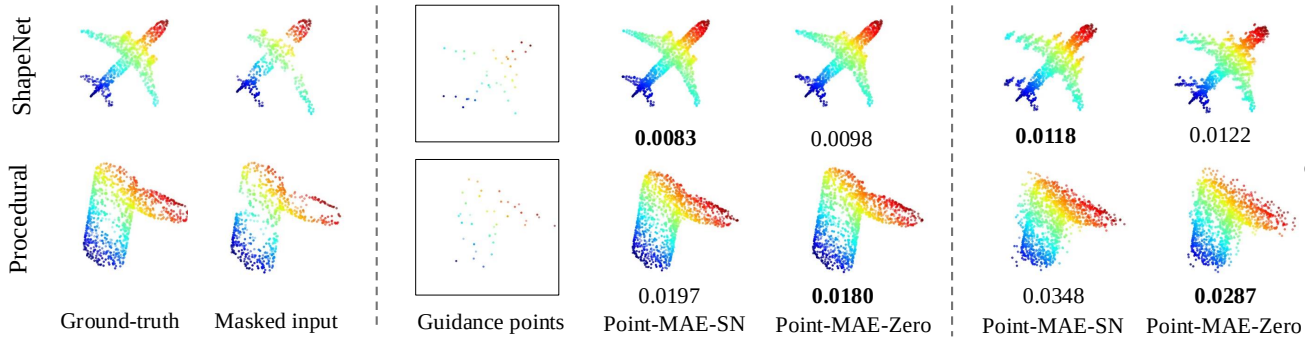


Figure 3. **Masked Point Cloud Completion.** This figure visualizes shape completion results with Point-MAE-SN and Point-MAE-Zero on the ShapeNet test split and procedurally synthesized 3D shapes. **Left**: Ground truth point clouds and masked inputs (60% mask ratio). **Middle**: Completions guided by masked input patch centers [26]. **Right**: Reconstructions without any guidance points. The L_2 Chamfer distance (*lower is better*) between the predicted 3D point clouds and the ground truth is displayed below each reconstruction.

	With Guidance		Without Guidance	
Methods	ShapeNet	Synthetic	ShapeNet	Synthetic
Point-MAE-SN [26]	**0.015**	**0.024**	**0.024**	0.039
Point-MAE-Zero	0.016	**0.024**	0.026	**0.037**
Point-M2AE-SN [59]	**0.002**	**0.005**	**0.007**	0.011
Point-M2AE-Zero	0.003	**0.005**	0.010	**0.009**
PCP-MAE-SN [61]	-	-	**0.016**	**0.028**
PCP-MAE-Zero	-	-	**0.016**	**0.028**

Table 4. **Masked Point Cloud Completion.** The table reports the L_2 Chamfer distance (*lower is better*) between predicted masked points and ground truth on the test set of ShapeNet and procedurally synthesized 3D shapes. *With Guidance*: center points of masked patches are added to mask tokens in the pretrained decoder, guiding masked point prediction during inference. *Without Guidance*: *Without Guidance*: no information from masked patches is available during inference.

	Semantic Segmentation			Instance Segmentation		
Methods	mIoU	mAcc	allAcc	mAP	AP50	AP25
MSC-scan [41]	73.85	81.80	90.49	39.75	60.51	**76.49**
MSC-Zero (1k)	72.69	80.80	90.13	39.03	58.57	75.24
MSC-Zero (2k)	**73.86**	**82.03**	**91.18**	**40.81**	**62.28**	76.26

Table 5. **Masked Scene Contrast Results.** Performance comparison between MSC-Scan and MSC-Zero with different amounts of pretraining data for semantic and instance segmentation tasks.

During pretraining, points are grouped into patches, with a subset of patches (60% by default) randomly masked. Only visible patches are encoded, while masked patch centers can optionally guide the decoder ("with guidance") or be omitted entirely ("without guidance"). After pretraining, models can reconstruct masked points even without such guidance. We quantitatively compare Point-MAE and Point-M2AE pretrained on ShapeNet ("-SN") and procedural shapes ("-Zero") in both guidance conditions, using the ShapeNet test split and 2,000 unseen synthetic shapes (see Tab. 4). All methods perform slightly better on their in-domain data. Removing guidance significantly decreases performance across all methods, highlighting its importance during masked reconstruction. Notably, Point-MAE-Zero and Point-M2AE-Zero closely match or even surpass their SN counterparts in reconstructing synthetic shapes, and remain competitive on ShapeNet shapes despite the lack of semantic training signals. PCP-MAE is a special case since it predicts centers before decoding point cloud and we find PCP-MAE-SN and PCP-MAE-Zero achieve similar performances both on seen and unseen domains.

Fig. 3 further illustrates that procedural-only models (*e.g.*, Point-MAE-Zero) effectively reconstruct familiar ShapeNet objects (*e.g.*, airplane wings, chair legs) without semantic supervision, likely by exploiting geometric symmetries. Similarly, SN-pretrained models generalize effectively to synthetic shapes not encountered during pretraining. Overall, these findings from Fig. 3 and Tab. 4 underscore that masked autoencoding primarily captures geometric rather than semantic information, enabling robust reconstruction across domains.

4.4. Scene-level 3D Understanding Tasks

Given the effectiveness of procedural 3D programs for pretraining self-supervised learning (SSL) models on 3D objects, a natural question arises: *Can procedural 3D programs similarly benefit SSL for 3D scenes?* We adopt Masked Scene Contrast (MSC) [41], a state-of-the-art SSL method for 3D scenes. We pretrain MSC on ScanNet [7] (1K scenes), commonly used for 3D scene SSL pretraining, and compare it against MSC pretrained on our procedurally generated scenes (denoted MSC-Zero). We conduct experiments with MSC-Zero using varying amounts of data (1K and 2K procedural scenes). MSC-Zero trained with 2K procedurally generated 3D scenes achieves outperforms MSC pretrained on ScanNet in both 3D semantic and instance segmentation tasks. This aligns with our observations on object-level 3D understanding tasks. We discuss the exact procedures of generating such data and implementation details in the supplementary materials.

Methods	Pre-train Loss	Downstream Accuracy
Scratch	–	77.24
Point-MAE-SN	2.62	85.18
Point-MAE-Zero		
(a) Single Primitive	3.17	83.93
(b) Multiple Primitives	4.10	84.52
(c) Complex Primitives	4.43	84.73
(d) Shape Augmentation	**5.28**	**85.46**

Figure 4. **Impact of 3D Shape Complexity on Performance. Left:** Examples of procedurally generated 3D shapes with increasing complexity, used for pretraining. Textures are shown for illustration purposes only; in practice, only the surface points are used. **Right:** Comparison of pretraining masked point reconstruction loss (Eqn. 1) [26] and downstream classification accuracy on the ScanObjectNN dataset [34]. Each row in Point-MAE-Zero represents an incrementally compounded effect of increasing shape complexity and augmentation, with the highest accuracy achieved using shape augmentation.

4.5. Analysis

Complexity of Synthetic 3D shapes. We examine how the geometric complexity of synthetic datasets impacts pretraining and downstream performance. We consider four progressively complex configurations: **(a) Single Primitive**: a single shape with affine transformations; **(b) Multiple Primitives ($\leq$3)**: up to three combined shapes; **(c) Complex Primitives ($\leq$9)**: up to nine combined shapes; **(d) Shape Augmentation**: further modified via boolean differences and wire-frame conversions.

Fig. 4 displays samples from each configuration alongside quantitative comparisons of pretraining performance and downstream classification accuracy on PB-T50-RS, the most challenging variant of ScanObjectNN [34]. As shape complexity increases, the pretraining task becomes more difficult, leading to higher reconstruction losses at the 300th training epoch. However, the downstream classification performance of Point-MAE-Zero improves. This underscores the importance of topological diversity in shapes for effective self-supervised point cloud representation learning.

We observe that the reconstruction loss on our dataset with single primitives (*i.e.*, 3.17) is higher than on ShapeNet (*i.e.*, 2.62) which consists of more diverse 3D shapes. We hypothesize that this is because ShapeNet is relatively smaller than our dataset (50K vs. 150K) and ShapeNet models are coordinate-aligned.

Dataset Size. Fig. 5 illustrates the effect of dataset size (*i.e.*, the number of procedurally generated 3D shapes) on Point-MAE-Zero's performance in the shape classification task on the PB-T50-RS benchmark.

Our experiments show that performance improves as dataset size increases, despite the dataset being procedurally generated. However, simply enlarging the dataset appears to yield diminishing returns, which may be due to intrinsic limitations of a purely synthetic 3D dataset or the representation learning bottleneck within Point-MAE.

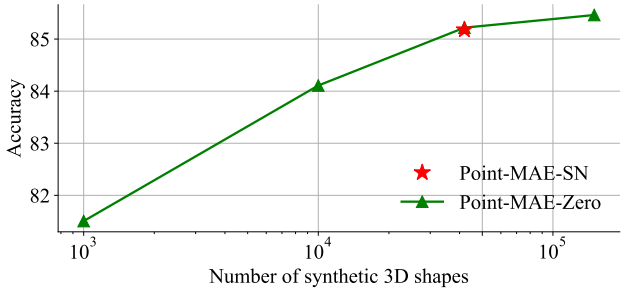


Figure 5. **Impact of pretraining dataset size.** We report the classification accuracy (%) on the PB-T50-RS subset of ScanObjectNN [34] as a function of the pretraining dataset size.

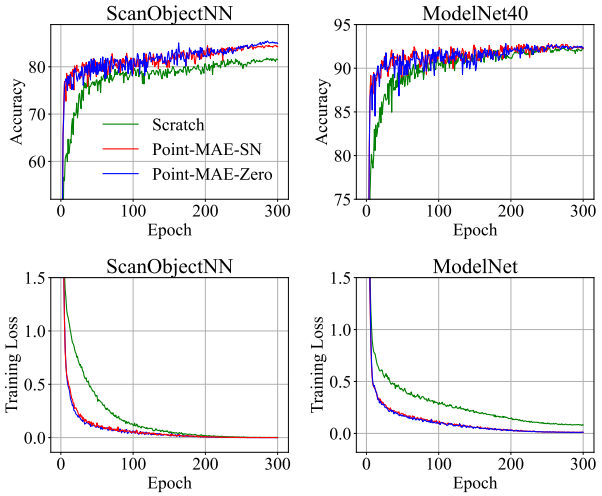
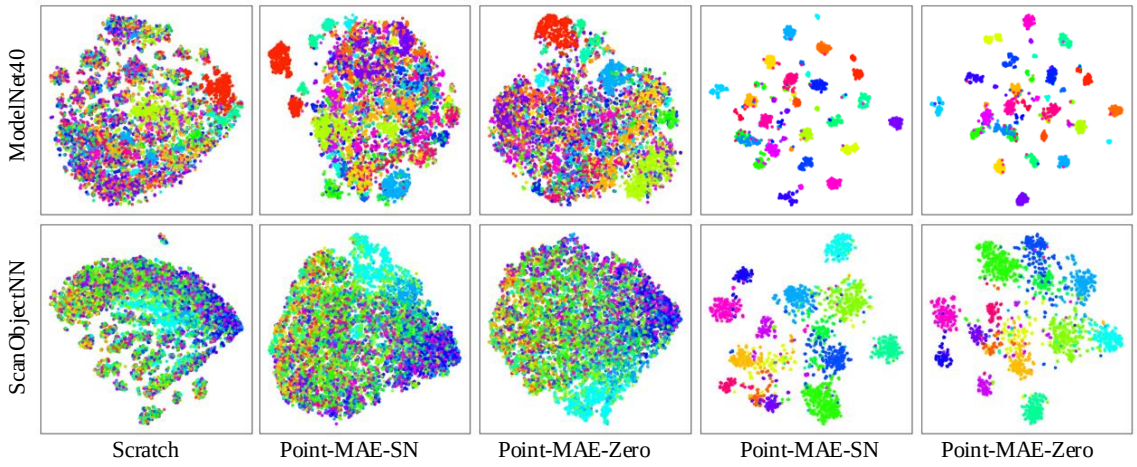


Figure 6. **Learning curves in downstream tasks.** We present validation accuracy (top row) and training curves (bottom row) in object classification tasks on ScanObjectNN (left column) and ModelNet40 (right column).

Notably, Point-MAE-Zero and Point-MAE-SN achieve comparable performance on downstream tasks when pretrained on datasets of the same size, regardless of differences in their pretraining data domains (i.e., synthetic



(a) 3D representation w/o fine-tuning (b) 3D representation w/ fine-tuning

Figure 7. **t-SNE visualization of 3D shape representations. (a)** Shows representations from transformer encoders: Scratch, Point-MAE-SN (ShapeNet), and Point-MAE-Zero (procedural shapes). **(b)** Displays fine-tuned representations for object classification on ModelNet40 (top) and ScanObjectNN (bottom). Each point represents a 3D shape while the color denotes the semantic categories.

shapes vs. ShapeNet). We include additional experiments in the appendix to ablate the effect of dataset size.

Efficiency of Transfer Learning. Fig. 6 shows the learning curves for training from scratch, Point-MAE-SN, and Point-MAE-Zero on shape classification tasks in the transfer learning setting. Both Point-MAE-SN and Point-MAE-Zero demonstrate faster training convergence and higher test accuracy compared to training from scratch. This trend is consistent across both ModelNet40 and ScanObjectNN benchmarks.

t-SNE Visualization. Fig. 7 visualizes the distribution of 3D shape representations from Point-MAE-SN and Point-MAE-Zero via t-SNE [35], before and after fine-tuning on specific downstream tasks. It also includes representations from a randomly initialized neural network as a reference.

First, compared to the representations from scratch, both Point-MAE-SN and Point-MAE-Zero demonstrate ***visually improved separation*** between different categories in the latent space. For example, this is evident in the red and light blue clusters on ModelNet40 and the blue and light blue clusters on ScanObjectNN. This indicates the effectiveness of self-supervised 3D representation learning via masked auto-encoding.

Second, when comparing representations after fine-tuning, both Point-MAE-SN and Point-MAE-Zero show ***much less clear separation*** between categories in the latent space. This raises the question of whether high-level semantic features are truly learned through the masked autoencoding pretraining scheme.

Finally, the t-SNE visualization reveals structural similarities between Point-MAE-Zero and Point-MAE-SN. Most categories ***lack clear separation*** in both models, except for the red and light blue clusters on ModelNet40 and the blue and light blue clusters on ScanObjectNN. This suggests that Point-MAE-Zero and Point-MAE-SN may have learned similar 3D representations, despite differences in the domains of their pretraining datasets. We provide more in-depth analysis in the supplementary material.

5. Discussion

In this work, we propose to learn 3D representations from synthetic data automatically generated using procedural 3D programs. We conduct an comprehensive empirical analysis of existing 3D SSLs and perform extensive comparisons with learning from well-curated, semantically meaningful 3D datasets.

We demonstrate that learning with procedural 3D programs performs comparably to learning from recognizable 3D models, despite the lack of semantic content in synthetic data. Our experiments highlights the importance of geometric complexity and dataset size in synthetic datasets for effective 3D representation learning. Our analysis further reveals that existing 3D SSLs primarily learns geometric structures (*e.g.*, symmetry) rather than high-level semantics.

This work has several limitations. For example, due to limited computational resources, we were unable to further scale up our experiments, such as by increasing the dataset size or conducting more detailed ablation studies on procedural 3D generation. Additionally, our findings may be influenced by potential biases in visualization tools (*e.g.*, t-SNE) or benchmarks (*e.g.*, data distribution and evaluation protocols). Furthermore, in 3D vision, the distinction between geometric structures and semantics remains an open question, as well-stated by Xie *et al.* [45]. This work also does not provide any novel representation learning method. Nevertheless, we hope our findings will inspire further exploration into self-supervised 3D representation learning.

References

[1] Manel Baradad, Richard Chen, Jonas Wulff, Tongzhou Wang, Rogerio Feris, Antonio Torralba, and Phillip Isola. Procedural image programs for representation learning. *Advances in Neural Information Processing Systems*, 35:6450–6462, 2022. 2, 3

[2] Manel Baradad Jurjo, Jonas Wulff, Tongzhou Wang, Phillip Isola, and Antonio Torralba. Learning to see by looking at noise. *Advances in Neural Information Processing Systems*, 34:2556–2569, 2021. 2, 3

[3] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021. 1

[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. *arXiv preprint arXiv:1512.03012*, 2015. 1, 2, 4

[5] Prakash Chandra Chhipa, Richa Upadhyay, Rajkumar Saini, Lars Lindqvist, Richard Nordenskjold, Seiichi Uchida, and Marcus Liwicki. Depth contrast: Self-supervised pretraining on 3dpm images for mining material classification. In *European Conference on Computer Vision*, pages 212–227. Springer, 2022. 2

[6] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21126–21136, 2022. 2

[7] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 4, 6

[8] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 1, 2

[9] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2

[10] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1, 3, 4

[11] Runpei Dong, Zekun Qi, Linfeng Zhang, Junbo Zhang, Jianjian Sun, Zheng Ge, Li Yi, and Kaisheng Ma. Autoencoders as cross-modal teachers: Can pretrained 2d image transformers help 3d representation learning?, 2023. 1, 2

[12] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision*, 129:3313–3337, 2021. 1, 2

[13] Ziyu Guo, Renrui Zhang, Longtian Qiu, Xianzhi Li, and Pheng-Ann Heng. Joint-mae: 2d-3d joint masked autoencoders for 3d point cloud pre-training, 2023. 2

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 1, 2

[15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 1, 2, 3, 4

[16] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts, 2021. 3

[17] Hanwen Jiang, Zexiang Xu, Desai Xie, Ziwen Chen, Haian Jin, Fujun Luan, Zhixin Shu, Kai Zhang, Sai Bi, Xin Sun, Jiuxiang Gu, Qixing Huang, Georgios Pavlakos, and Hao Tan. Megasynth: Scaling up 3d scene reconstruction with synthesized data. 2024. 1, 2, 3

[18] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. *arXiv preprint arXiv:2307.07635*, 2023. 3

[19] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31, 2018. 2, 4

[20] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018. 3

[21] Hao Liu, Minglin Chen, Yanni Ma, Haihong Xiao, and Ying He. Point cloud unsupervised pre-training via 3d gaussian splatting, 2024. 3

[22] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8895–8904, 2019. 1, 4

[23] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 3

[24] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 3

[25] Wufei Ma, Guanning Zeng, Guofeng Zhang, Qihao Liu, Letian Zhang, Adam Kortylewski, Yaoyao Liu, and Alan Yuille. Imagenet3d: Towards general-purpose object-level 3d understanding. *arXiv preprint arXiv:2406.09613*, 2024. 2

[26] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point

cloud self-supervised learning. In *European conference on computer vision*, pages 604–621. Springer, 2022. 1, 2, 3, 4, 5, 6, 7

[27] 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*, pages 652–660, 2017. 2, 4, 5

[28] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2, 4, 5

[29] Zekun Qi, Runpei Dong, Guofan Fan, Zheng Ge, Xiangyu Zhang, Kaisheng Ma, and Li Yi. Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining. In *International Conference on Machine Learning (ICML)*, 2023. 2

[30] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10901–10911, 2021. 2

[31] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016. 3

[32] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2974–2983, 2018. 2

[33] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 3

[34] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, 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*, pages 1588–1597, 2019. 1, 4, 7

[35] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008. 8

[36] Chengyao Wang, Li Jiang, Xiaoyang Wu, Zhuotao Tian, Bohao Peng, Hengshuang Zhao, and Jiaya Jia. Groupcontrast: Semantic-aware self-supervised representation learning for 3d understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4917–4928, 2024. 3

[37] Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matt J Kusner. Unsupervised point cloud pre-training via occlusion completion. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9782–9792, 2021. 1, 2, 3, 4, 5

[38] 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 (tog)*, 38(5):1–12, 2019. 2, 4, 5

[39] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. 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*, pages 803–814, 2023. 2

[40] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems*, 35:33330–33342, 2022. 2, 4

[41] Xiaoyang Wu, Xin Wen, Xihui Liu, and Hengshuang Zhao. Masked scene contrast: A scalable framework for unsupervised 3d representation learning, 2023. 1, 2, 3, 4, 6

[42] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4840–4851, 2024. 2

[43] 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*, pages 1912–1920, 2015. 2

[44] 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*, pages 1912–1920, 2015. 4

[45] Desai Xie, Sai Bi, Zhixin Shu, Kai Zhang, Zexiang Xu, Yi Zhou, Sören Pirk, Arie Kaufman, Xin Sun, and Hao Tan. Lrm-zero: Training large reconstruction models with synthesized data. *arXiv preprint arXiv:2406.09371*, 2024. 1, 2, 3, 8

[46] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 574–591. Springer, 2020. 2

[47] Saining Xie, Jiatao Gu, Demi Guo, Charles R. Qi, Leonidas J. Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding, 2020. 3

[48] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In *Proceedings of the European conference on computer vision (ECCV)*, pages 87–102, 2018. 2, 4

[49] Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. Deep image-based relighting from optimal sparse samples. *ACM Transactions on Graphics (ToG)*, 37 (4):1–13, 2018. 1, 3

[50] Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, and Ravi Ramamoorthi. Deep view synthesis from sparse photometric images. *ACM Transactions on Graphics (ToG)*, 38(4):1–13, 2019. 1, 3

[51] Siming Yan, Zhenpei Yang, Haoxiang Li, Li Guan, Hao Kang, Gang Hua, and Qixing Huang. Iae: Implicit autoencoder for point cloud self-supervised representation learning. 2022. 3

[52] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381, 2024. 3

[53] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023. 1, 2

[54] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016. 1, 5

[55] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19313–19322, 2022. 1, 2, 3, 4, 5

[56] Xueyang Yu, Xinlei Chen, and Yossi Gandelsman. Learning video representations without natural videos. *arXiv preprint arXiv:2410.24213*, 2024. 2, 3

[57] Yaohua Zha, Huizhen Ji, Jinmin Li, Rongsheng Li, Tao Dai, Bin Chen, Zhi Wang, and Shu-Tao Xia. Towards compact 3d representations via point feature enhancement masked autoencoders, 2023. 2

[58] Yaohua Zha, Huizhen Ji, Jinmin Li, Rongsheng Li, Tao Dai, Bin Chen, Zhi Wang, and Shu-Tao Xia. Towards compact 3d representations via point feature enhancement masked autoencoders. *arXiv preprint arXiv:2312.10726*, 2023. 2

[59] Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, and Hongsheng Li. Point-m2ae: Multi-scale masked autoencoders for hierarchical point cloud pre-training. *arXiv preprint arXiv:2205.14401*, 2022. 1, 2, 3, 4, 5, 6

[60] Renrui Zhang, Liuhui Wang, Yu Qiao, Peng Gao, and Hongsheng Li. Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders, 2022. 2

[61] Xiangdong Zhang, Shaofeng Zhang, and Junchi Yan. Pcp-mae: Learning to predict centers for point masked autoencoders. *arXiv preprint arXiv:2408.08753*, 2024. 1, 2, 3, 4, 5, 6

[62] 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*, pages 16259–16268, 2021. 2, 4

Learning 3D Representations from Procedural 3D Programs

Supplementary Material

We provide more implementation details, additional experimental results, and visualizations in this supplementary material. In Sec. A, we present our training-from-scratch baseline, which surpasses the results reported in Point-MAE [26]. In Sec. B, we introduce a more rigorous evaluation protocol using a dedicated validation set instead of relying on the test set for validation. Moreover, in Sec. C, we provide linear probing results on ModelNet40 and three ScanObjectNN variants to compare the performance of Point-MAE-SN and Point-MAE-Zero. Additional visualizations are provided in Sec.D. Details on generating scene-level procedural 3D programs are in Sec.E. Lastly, implementation details for 3D object SSLs and 3D scene SSLs are presented in Sec. F.

A. Training-from-Scratch Baseline

For the training-from-scratch baseline, we report the results in the supplementary material for completeness. In the main text, we referenced the scores reported in the original Point-MAE paper for this baseline. However, our experiments suggest that training from scratch in our setup produced higher scores than those reported in Point-MAE [26] and Point-BERT [55]. The results from our training-from-scratch baseline are presented in this section, as shown in Tab. 6, Tab. 7, and Tab. 8.

We follow the same evaluation protocol as Point-MAE [26]. While the results we obtained from the training-from-scratch baseline consistently surpass the previously reported scores, there remains a significant gap between the performance of the training-from-scratch baseline and the pre-trained methods. This underscores the effectiveness of pre-training in enhancing model performance. Furthermore, pre-trained methods demonstrate much faster convergence compared to the training-from-scratch baseline.

Methods	ModelNet40	OBJ-BG	OBJ-ONLY	PB-T50-RS
Scratch [55]	91.4	79.86	80.55	77.24
Scratch*	93.4	87.44	82.03	81.99
Point-MAE-SN [26]	**93.8**	90.02	88.29	85.18
Point-MAE-Zero	93.0	**90.36**	**88.64**	**85.46**

Table 6. **Object Classification.** Classification accuracy (%) on ModelNet40 and three ScanObjectNN variants under the revised evaluation setup (*Higher is better*). Note Scratch* indicates baseline method's results we obtained.

Methods	5w/10s	5w/20s	10w/10s	10w/20s
Scratch [55]	87.8±5.2	93.3±4.3	84.6±5.5	89.4±6.3
Scratch*	92.7±3.5	95.5±3.1	88.5±5.1	92.0±4.5
Point-MAE-SN [26]	**96.3±2.5**	**97.8±1.8**	**92.6±4.1**	**95.0±3.0**
Point-MAE-Zero	95.4±2.5	97.7±1.6	91.3±5.1	95.0±3.5

Table 7. **Few-shot classification on ModelNet40.** We evaluate performance on four n-way, m-shot configurations. For example, 5w/10s denotes a 5-way, 10-shot classification task. The table reports the mean classification accuracy (%) and standard deviation across 10 independent runs for each configuration. **Top**: Results from existing methods for comparison. **Bottom**: Comparison with our baseline methods. Note Scratch* indicates baseline method's results we obtained.

Methods	mIoU_I	aero	bag	cap	car	chair	earphone	guitar	knife
Scratch [55]	85.1	82.9	**85.4**	87.7	78.8	90.5	**80.8**	91.1	**87.7**
Scratch*	84.0	84.3	83.1	**89.1**	80.6	91.2	74.5	**92.1**	87.3
Point-MAE-SN [26]	**86.1**	84.3	85.0	88.3	80.5	91.3	78.5	**92.1**	87.4
Point-MAE-Zero	**86.1**	**85.0**	84.2	88.9	**81.5**	**91.6**	76.9	**92.1**	87.6

Methods	lamp	laptop	motor	mug	pistol	rocket	skateboard	table
Scratch [55]	85.3	95.6	73.9	**94.9**	83.5	61.2	74.9	80.6
Scratch*	85.1	95.9	74.3	94.8	84.3	61.1	76.2	80.9
Point-MAE-SN [26]	**86.1**	**96.1**	75.2	94.6	84.7	63.5	77.1	**82.4**
Point-MAE-Zero	86.0	96.0	**77.8**	94.8	**85.3**	**64.7**	**77.3**	81.4

Table 8. **Part Segmentation Results.** We report the mean Intersection over Union (IoU) across all instances (mIoU_I) and the IoU (%) for each category on the ShapeNetPart benchmark (*higher values indicate better performance*). Note Scratch* indicates baseline method's results we obtained.

B. More Rigorous Evaluation

In prior works [26, 55], the validation set was identical to the test set. The model was evaluated on the test set after every epoch, with the best-performing result selected. Such practices can artificially inflate performance metrics and fail to accurately reflect the model's ability to generalize to unseen data. While we followed this setup in the main text for fair comparisons, we also performed more rigorous evaluations using a dedicated validation set, which has no overlap with either the training set or the test sets. Specifically, we set aside 20% of the original test set as our validation set, leaving the remaining 80% as the new test set. We then report the test performance with checkpoints selected based on the validation accuracy.

As presented in Tab. 9, our new evaluation results are consistent with these reported in the main text — both Point-MAE-SN and Point-MAE-Zero outperform the training-from-scratch baseline across all four object classification tasks, while Point-MAE-Zero performs on par

with Point-MAE-SN. The performance gap between models with pretraining and models trained from scratch is particularly pronounced in the most challenging experiment, PB-T50-RS, compared to the other three classification tasks.

On ModelNet40, we observe that training-from-scratch and pre-trained methods achieve similar performance, though the results may vary across different runs. For example, in another run with a different random seed, training-from-scratch achieve 92.1% accuracy, which is lower than both Point-MAE-SN and Point-MAE-Zero. However, in all our experiments, pre-trained methods consistently converge significantly faster than training-from-scratch as shown in Fig. 6 in the main text.

Methods	ModelNet40	OBJ-BG	OBJ-ONLY	PB-T50-RS
Scratch* [55]	**92.95**	84.19	86.94	80.92
Point-MAE-SN [26]	92.22	**88.32**	87.97	83.83
Point-MAE-Zero	92.87	**88.32**	**88.31**	**84.73**

Table 9. **Object Classification.** Classification accuracy (%) on ModelNet40 and three ScanObjectNN variants under the revised evaluation setup. The original test set was split to create a new test set and validation set, and all models were re-evaluated using the updated splits. (*Higher is better*). Note Scratch* indicates baseline method's results we obtained and we do not apply voting in these experiments.

C. Linear Probing

In line with Point-BERT [55] and Point-MAE [26], we primarily report the performance of pre-trained methods in a transfer learning and few-shot learning setting in the main text, where the pre-trained model is fine-tuned with task-specific supervision. However, a common practice for benchmarking self-supervised learning methods is the linear probing, which trains a single linear layer while keeping the pretrained network backbone frozen. Below we provide details of our experimental setup and results.

Experimental Setup. Instead of fully fine-tuning the pre-trained model, we freeze the model's weights and train a single linear layer for the target task.

We include a training-from-scratch baseline, where we freeze the randomly initialized weights and train only a single linear layer, providing a point of comparison for the pre-trained models.

Methods	ModelNet40	OBJ-BG	OBJ-ONLY	PB-T50-RS
Scratch*	84.16	62.65	66.09	56.80
Point-MAE-SN [26]	**90.56**	**78.83**	**81.41**	**68.22**
Point-MAE-Zero	89.30	76.24	78.83	67.87

Table 10. **Object Classification.** Classification accuracy (%) on ModelNet40 and three ScanObjectNN variants under the revised evaluation setup (*Higher is better*). Note Scratch* indicates baseline method's results we obtained.

Results. We present linear probing results for object classification in Tab. 10. The gap between pre-trained models and the training-from-scratch baseline is noticeably larger in this setting. Interestingly, Point-MAE-SN outperforms Point-MAE-Zero in object classification under the linear probing setup, suggesting that semantically meaningful data may enable models to achieve a better understanding of 3D structures. However, the performance gap between Point-MAE-SN and Point-MAE-Zero remains relatively small. Based on the results in Tab. 1 in the main text, Point-MAE-SN appears to be a more effective choice for transfer learning tasks. We encourage future research to explore improved learning algorithms to take full advantage of data generated from procedure 3D programs.

D. Additional Visualization

In this section, we present additional qualitative comparisons between Point-MAE-SN and Point-MAE-Zero for Masked Point Cloud Completion under both *guided* and *unguided* settings. Additionally, we provide t-SNE visualizations for Point-MAE-Zero and Point-MAE-SN to further investigate its representational capabilities, focusing on whether it can effectively distinguish between different primitives.

D.1. Masked Point Cloud Completion

Additional qualitative results for masked point cloud completion, both guided and unguided, are shown in Figure 9.

D.2. More t-SNE Visualizations

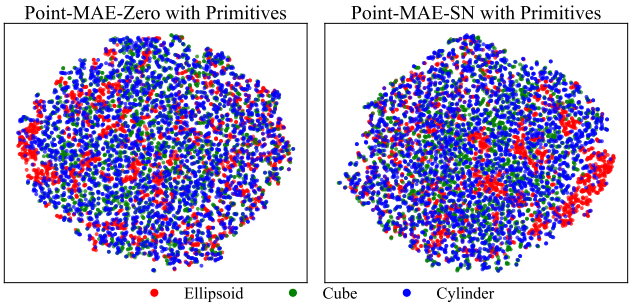


Figure 8. **t-SNE Visualization.** We visualize features extracted by Point-MAE-Zero (left) and Point-MAE-SN (right) for three primitive shapes — Ellipsoid, Cube, and Cylinder.

Point-MAE on Primitives Fig. 8 presents t-SNE visualizations of features extracted by Point-MAE-Zero and Point-MAE-SN for three primitives: ellipsoid, cube, and cylinder. Interestingly, the notable structural differences among these primitives are not reflected in the latent space of either Point-MAE-SN or Point-MAE-Zero. We hypothesize that this occurs because the learned 3D representations from both models primarily capture local structures rather than global shapes.

E. Details on Generating Scene-Level Procedural 3D Programs

We follow MegaSynth's [17] generation pipeline. First, the pipeline generates a scene floor plan. Each 3D box represents a different shape, with distinct colors indicating various object types. This defines the spatial structure of the scene. Next, we creates 3D objects by combining primitive shapes such as cubes and spheres. These objects undergo geometry augmentations, including scaling, deformation, and boolean operations, to introduce variations. Since we do not need texture and lighting, we omit the rest of the procedurals. Instead we sample points from each shape and randomly sample points from the layout in order to represent walls, ceilings and floors. We will release code to further assist the reproducibility of our results.

F. Implementation details.

We closely follow each baseline's open-source configuration. Similar to prior work for 3D shape pretraining, we sample each point cloud to $p = 1024$ points and divide it into $n = 64$ patches, with each patch containing $k = 32$ points via the KNN algorithm. The autoencoder consists of an encoder with 12 Transformer blocks and a decoder with 4 Transformer blocks, each block having a 384-dimensional hidden size and 6 attention heads. During pretraining, we randomly sample 1024 points per shape and apply standard random scaling and translation. We train for 300 epochs using the AdamW optimizer [24] with a cosine decay schedule [23], an initial learning rate of 0.001, weight decay of 0.05, and a batch size of 128.

For scene-level SSL MSC, we use SparseUNet34 as the backbone with a hierarchical encoder-decoder structure. The encoder consists of depths $[2, 3, 4, 6]$ and channels $[32, 64, 128, 256]$, while the decoder follows depths $[2, 2, 2]$ with channels $[256, 128, 64, 64]$. A kernel size of 3 is used in both encoding and decoding, with a pooling stride of $[2, 2, 2, 2]$. The pre-training phase employs SGD [31] with a cosine decay scheduler [23], an initial learning rate of 0.1, a weight decay of $1e^{-4}$, and a momentum of 0.8. We use a batch size of 32 and train on ScanNet for 600 epochs, with 6 warmup epochs. For fine-tuning, we use a similar SGD [31] with cosine decay setup [23], but with a learning rate of 0.05, weight decay of $1e^{-4}$, and momentum of 0.9. The batch size is 48, with 40 warmup epochs, and the model is trained for 600 epochs. We will release code to further assist the reproducibility of our results.

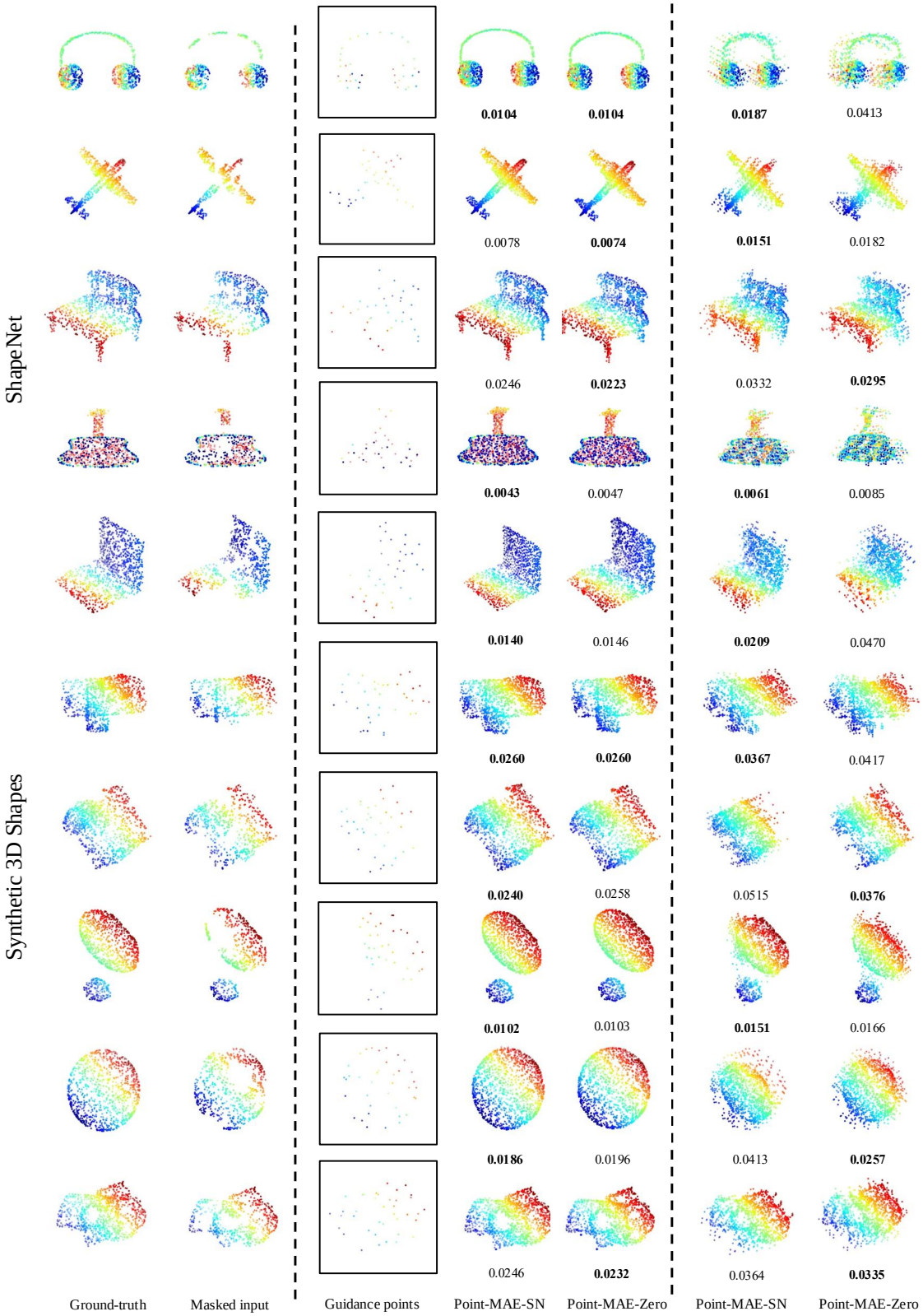


Figure 9. **Masked Point Cloud Completion.** This figure visualizes shape completion results with Point-MAE-SN and Point-MAE-Zero on the test split of ShapeNet and procedurally synthesized 3D shapes. **Left**: Ground truth 3D point clouds and masked inputs with a 60% mask ratio. **Middle**: Shape completion results using the centers of masked input patches as guidance, following the training setup of Point-MAE [26]. **Right**: Point cloud reconstructions without any guidance points. The L_2 Chamfer distance (*lower is better*) between the predicted 3D point clouds and the ground truth is displayed below each reconstruction.