

---

# Imagine with the Teacher: Complete Shape in a Multi-View Distillation Way

---

Zhanpeng Luo<sup>1</sup> Linna Wang<sup>1</sup> Guangwu Qian<sup>1</sup> Li Lu<sup>1</sup>

<sup>1</sup>Computer College, Sichuan University

{luozhanpeng, lenawang}@stu.scu.edu.cn

{g\_qian, luli}@scu.edu.cn

## Abstract

Point cloud completion aims to recover the completed 3D shape of an object from its partial observation caused by occlusion, sensor's limitation, noise, etc. When some key semantic information is lost in the incomplete point cloud, the neural network needs to infer the missing part based on the input information. Intuitively we would apply an autoencoder architecture to solve this kind of problem, which take the incomplete point cloud as input and is supervised by the ground truth. This process that develops model's imagination from incomplete shape to complete shape is done automatically in the latent space. But the knowledge for mapping from incomplete to complete still remains dark and could be further explored. Motivated by the knowledge distillation's teacher-student learning strategy, we design a knowledge transfer way for completing 3d shape. In this work, we propose a novel View Distillation Point Completion Network (VD-PCN), which solve the completion problem by a multi-view distillation way. The design methodology fully leverages the orderliness of 2d pixels, flexibleness of 2d processing and powerfulness of 2d network. Extensive evaluations on PCN, ShapeNet55, and MVP datasets confirm the effectiveness of our design and knowledge transfer strategy, both quantitatively and qualitatively. Committed to facilitate ongoing research, we will make our code publicly available.

## 1 Introduction

Point cloud completion plays an essential role in 3D vision applications and remains an active research topic in recent years. Throughout this period, a diverse range of learning-based approaches have been introduced, many of which have shown promising results. As research progresses, incremental improvements in result accuracy and model efficiency have been reported. [1, 2, 3, 4, 5], Despite these advancements, the inherent sparsity and significant structural incompleteness of captured point clouds continue to impede the efficacy of existing models in generating fully satisfactory outcomes[6]. We observe two intriguing problems in two fields in the point cloud processing and raise two questions about the current points completion task.

In the field of point cloud classification, the success of SimpleView surprise us[7, 8], which prove the efficacy and robustness of a 2d network in the point cloud processing. Even simple multi-view CNN with the depth map of projected point cloud, can achieve on par or better results than sophisticated state-of-the-art methods on ModelNet40. Looking back to the completion task, while the common solution is to use pointnet++[9] structure to extract information directly from the input point cloud[4, 6, 10], we are wondering if there is still potential for further enhancement? Since a multi-view CNN could do better on the classification problem, we are naturally infer that a CNN-based encoder could get a more descriptive representation for the incomplete input. This prompts the question: **Can we reproduce the Simpleview's success in the completion task?**

In the field of point cloud completion, intuitively we would apply an encoder-decoder structure to solve the completion-like problem implicitly. But recovering an unordered partial point cloud is a challenging task[11]. Existing approaches that rely on decoding a latent feature to recover the complete shape, often lead to the completed point cloud sometimes over-smoothing or losing details on edges. In order to lower the learning difficulty and recover better details, some methods are adopted to exploit the information from both the input and ground truth, such as concatenate the input on the generated point cloud and subsample into the original amount[12]. And another typical solution is to give the ground truth supervision multiple times, such as a coarse-to-fine paradigm.[13] But could we lower the learning difficulty one more step? Would the information from the ground truth be further explored? How can we make the dark knowledge in the latent space more explicit? This raises us the second question: **Can we leverage some kind of technique that further lower the learning difficulty in the completion task?**

First, to explore the first question is rather trivial. Following SimpleView[7], we could directly apply a multi-view CNN on a depth image group that the point cloud project to. But a simple shared 2d network for all images do not consider the relation between views. This is especially important for the completion tasks, because sometimes it is very hard for even for human to imagine a complete object from partial observation. Therefore we design a multi-view encoder and a decoder considering the information loss during the CNN pooling operation. Also, since transformation from point cloud to depth map would lose information, we reintroduce the original point cloud information during the decoder stage and use both 2d and 3d feature for point generation. This consists of the basic struture of our model.

Next, to explore a method to lower the learning difficulty, we are inspired by the knowledge distillation technique, the siamese network design, and the knowledge transfer strategy and adopt a teacher–student knowledge transfer strategy to exploit the rich information in the ground truth point cloud. Specifically, the teacher and student share the same structure. The teacher is trained with partial point cloud and complete depth map as input and is supervised by ground truth. After training a teacher who build well-established mapping from complete depth map and partial point cloud to complete point cloud, we froze the teacher and add a feature-level distillation supervision on student. At the student side, we use teacher’s pre-trained weight as initiation, froze all module except the multi-view encoder and train it with a smaller learning rate. The student would not only inherit teacher’s semantic understanding but also learn to gradually adapt to the incomplete input and mimic teacher’s feature and point prediction. By experiment, we have shown that the knowledge transfer strategy would produce better coarse point and better result.

With the help of this designing methodology and learning strategy, our method outperforms state-of-the-art completion networks on many widely used benchmarks. We would also like to summarize our contribution: We devise a new method for points completion achieving current state-of-the-art result on many widely-used dataset by solving two problems we observed. We also extend a new knowledge transfer training diagram on the point cloud completion task, which can be easily adopted for the related problems. We are committed to releasing code for reproducibility and future research.

## 2 Related Work

**Point cloud completion** Like traditional methods for point cloud learning early attempts at 3D shape completion usually adopt intermediate aids (i.e., voxel, grids) [2, 14]. After the success of Point Net, the community put more attention on the point-based method. Recently, due to some new technique such as attention, and large model, research introduce multi-modality to the completion process, such as paired image[15], RGBD image[16] and text[17]. Within the completion model design, common solution is an encoder decoder architecture in a coarse-to-fine paradigm[13]. A set of coarse points usually first predicted from the input information. During each decoding stage, the coarse point would be upsampled and is supervised by the ground truth. SnowFlakeNe[4] summarize this as a “Feature Extraction – Seed Generation – Point Generation” paradigm. Recent work put lots of effort on the decoding part, or the point generation part. FBNet[18] adopts the feedback mechanism during refinement and generates points in a recurrent manner. LAKe-Net[19] integrates its surface-skeleton representation into the refinement stage. SVDFormer[6] identify a structure similarity between each stage. These improvements often would either increase the running time or increase model complexity. Also, other methods improve on the loss function chamfer distance[20, 21] to help better reconstruct shape.

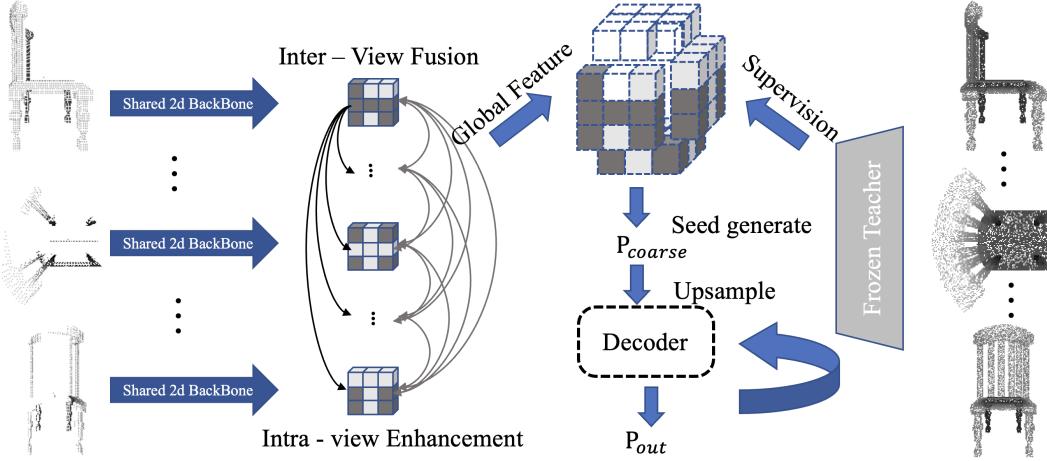


Figure 1: Model Overview

**Multi-View 3d Understanding** Early works in 3d shape learning usually adopt multi-view projection or 3D voxelization to transform the irregular point clouds into regular representations[22]. Before the invention of pointnet, we have to operate 2d/3d CNN on these regular representations. But recently, SimpleView[7] has drawn people’s attention again to the usage of 2d network in 3d understanding. SimpleView shows us a simple multi-view CNN could perform on par or better than other sophisticated point-based method in the classification problem with great cross-dataset generalization. Lots of work have been exploring the usage of multi-view projection in other 3d understanding task. Tackling 3D vision tasks with this indirect approach has two main advantages: (i) mature and transferable 2D computer vision models (CNNs[23], Transformers[24] etc.), and (ii) large and diverse labeled image datasets for pre-training (e.g., ImageNet[25]). Furthermore, recent advances in differentiable rendering allow for end-to-end deep learning pipelines that render multi-view images of the 3D data and process the images by CNNs/transfomers/diffusion[26] to obtain a more descriptive representation for the 3D data.

**Knowledge Distillation** Knowledge distillation [27], is first proposed as a model compression method, which pushes the student network to mimic the soft logits of the teacher network in the originally setting. However, there are few works introducing this technique directly to the point cloud analysis. The image based feature-level distillation would cause negative transfer because either point-feature alignment or point-pixel alignment fails for data’s irregularity and sparsity[8, 28]. Some methods provide a workaround for the problem. PointDistiller[28] apply the local distillation and the reweighted learning strategy , which cluster the local neighbouring points and extract information with dynamic graph convolution. ProtoTransfer[29] use a class-wise prototype distillation method to align student in a cross modality way. Let image give you more[8] choose to align teacher and student on the point cloud loss, rather than a feature level alignment. In our work, we utilize the regularity of pixels and can directly align teacher’s and student’s feature to transfer the knowledge.

### 3 Methodology

Here, we present the task formulation and an overview of VD-PCN.

#### 3.1 General Pipeline

Given a set incomplete unordered points  $P_{in} \in \mathbb{R}^{N_{in}*3}$ , the model need to recover a complete point set  $P_{out} \in \mathbb{R}^{N_{out}*3}$ . SnowFlakeNet[4] has formulated this task into three stages: feature extraction, seed generation and point upsampling. In the feature extraction stage, input point cloud is encoded into a global feature, which would be further used to generate a coarse point cloud  $P_{coarse} \in \mathbb{R}^{N_{coarse}*3}$ . as seed. Then in the upsampling stage, this coarse point cloud would finally be upsampled into the same resolution as  $P_{out}$ .

Applying a 2d network to solve the completion problem, we first takes depth maps for the input point cloud from  $k$  viewpoints and get an image group, represented as  $I = \{I_i | i = 1, \dots, k\}$ ,  $I_i \in \mathbb{R}^{1 \times H \times W}$ . Then  $I$  would be sent into a shared 2d backbone, getting a multi-view feature group  $F = \{F_i | i = 1, \dots, k\}$ . To further equip our model with a panoramic view, we design a multi-view encoder to aggregate information from each viewpoint. Then the global feature was sent to generate seed and unsample until high resolution.

When the knowledge transfer strategy is applied, we reformulate the working pipeline as a "Multi-View Perceptual Encoder, Distilled Knowledge Transfer, and Dual Modality Decoder" process. The teacher and student share the same structure. Firstly, a model would be trained with partial point cloud and complete depth map and would be supervised by the ground truth. After the teacher build a well-established mapping from complete depth map to complete point cloud, we froze the teacher and apply a feature-level distillation supervision. Also, we initiate student with teacher's weight. and train with a smaller learning rate and epoch.

### 3.2 Model Design

As we describe in the general pipeline, VD-PCN consist of three part, the multi-view encoder, distilled knowledge transfer and dual-modality decoder.

**Multi-View Perceptual Encoder** As we discussed above, VD-PCN is initiated with a 2d backbone processing the  $k$  view depth images. We choose the left branch of a classical Unet[23] as our 2d backbone for its larger receptive field. And we do not choose the attention based vision transformer since we hope to build a model in a comparable parameter scale with point-based method. Processed by a shared 2d network, the  $k$  views are reshaped into the batch size dimension, sent to the Unet, and downsampled into a multi-view feature map  $F_v \in \mathbb{R}^{k \times C \times H_1 \times W_1}$ .

To enable our model with a global awareness of all the  $k$  views on one object, we design a multi-view encoder to aggregate each view's information. Inspired by Leap[30], we choose to use a cross attention on  $F_v$  to achieve a panoramic awareness for the object. Firstly, to find the relationship between distinct views, we design an Intra-View Fusion (IVF) layer, which do cross attention with  $f_0$  as query and  $f_{1\dots k}$  as key & value and update  $f_0$  into  $f'_0$ . This could be formulate as:

$$f'_0 = FFN(CrossAttention(K = f_0, Q \& V = f_{1\dots k}))$$

Following the global integration facilitated by the IVF Layer, the Intra-View Enhancement (IVE) Layer focuses on refining the feature maps within each view. Utilizing self-attention mechanisms, the IVE Layer enhances the relation within the view group. For the IVE layer, we use a self-attention mechanism where the key, query, and value are the image features group  $F$ . It can be formulated as:

$$F'_v = FFN(SelfAttention(Q \& K \& V = F_v))$$

where  $F'_v$  stands for the updated value for the whole feature group. And  $F_v$  is the original feature group with  $f_0$  updated. To maximize the effectiveness of our encoder, the IVF and IVE Layers undergo multiple iterations of refinement. This iterative process allows for the progressive enhancement of the global feature map. The updated feature map  $F'_v \in \mathbb{R}^{k \times C \times H_1 \times W_1}$  would be further max pooled into a global feature tensor  $F'_g \in \mathbb{R}^{k \times C}$ .

**Distilled Knowledge Transfer** Another core part of VD-PCN is the knowledge transfer distillation strategy. Initially, we train a teacher who share the same structure with student, which takes complete depth map and partial point cloud as input. The teacher is supervised by the ground truth. After building a well-established mapping from complete depth map to complete point cloud, we begin to train our model with a feature-level supervision by the teacher. During the experiment, we find that downsample the ground truth to make depth images to would produce better result. A reasonable guess is fewer points already contain enough information for completion[31] and a small amount of points would prevent object's structure occlusion and bring less noise.

We do not adopt the online knowledge distillation[32] to compare in a similar computation cost framework. Therefore, the teacher's multi-view encoder is frozen, takes the complete depth map as input and generate teacher's feature map  $F_v^T \in \mathbb{R}^{k \times C \times H_1 \times W_1}$  and max-pooled global feature tensor  $F_g^T \in \mathbb{R}^{k \times C}$ .

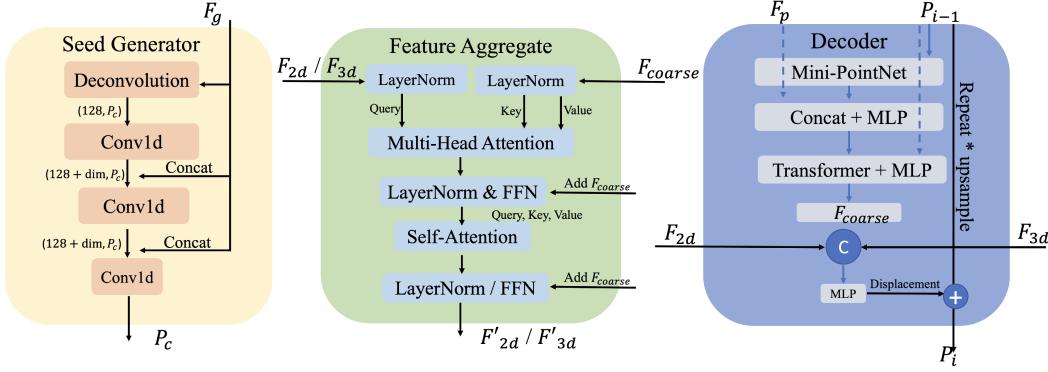


Figure 2: Model detail illustrated for three modules in the decoder part.

Then the student, who take a partial point cloud depth image would mimic the teacher’s behaviour. In specific, we align  $F_v^S$  with  $F_v^T$  in a  $L_2$  loss and  $F_g^S$  with  $F_g^T$  in a  $L_1$  loss.

$$\mathcal{L}_{KD} = \tau_1 \cdot \sum_s |P_v^T - P_v^S|^2 + \tau_2 \cdot \sum_s |P_g^T - P_g^S|$$

Here we choose L2 loss for  $F_v$  and L1 loss for  $F_g$ . By earlier experiments, different type of loss and scale factor  $\tau_1$  and  $\tau_2$  seem to achieve similar effect at the end. Therefore we simply set  $\tau_1 = \tau_2 = 1$  and introduce two type of loss in the distillation stage.

And here, we also wanted to discuss the distillation design in the point completion task. As far as we know, we are the first one to design a knowledge distillation method on the point completion task. This is because a naive knowledge distillation method would cause negative transfer for the unorderliness of the point cloud. A feature-level alignment strategy would fail when the teacher’s input do not have the same order as student’s[28]. But in VD-PCN, we apply CNN on projection depth images, which leverage the orderliness of pixels to align teacher’s and student’s feature to achieve a knowledge transfer effect.

**Dual-Modality Decoder** The decoder generate seed from  $F_g$  and upsample in a coarse to fine way until final high-resolution point cloud. Following SnowFlakeNet, we adapt a seed generator, a deconvolution followed by residual connected MLP would generate the first coarse point cloud . And to lower the difficulty to generate seed. A common technique when generating  $P_c$  is to concatenate the input on the  $P_c$  and subsample it into the same resolution it with  $P_c$ .

At  $i_{th}$  stage, the coarse points  $P_{i-1}$  from the previous stage is utilized. A mini-pointnet would first project the coarse into a high-dimensional space, then we concatenate the global feature on these points. After that, we apply a transformer to strengthen the mapping between  $F_{coarse}$  and  $P_{coarse}$ .

At the feature aggregation module, the key ideas in the decoder design is to re-introduce the lost information from two modality. The first information loss is in the transformation from point cloud  $P_n$  to multi-view depth images  $I$  and the second information loss is the pooling operation from  $F_v$  to  $F_p$ . Therefore we use a feature aggregation module to introduce 2d and 3d information into the upsampling process. In detail, we sent apply a cross attention module with  $F_{coarse}$  as query and  $F_p$  /  $F_v$  as Key & Value. After feature aggregation,  $F_{2d}$  and  $F_{3d}$  are concatenated together and sent into a MLP to predict the displacement for the upsampled point cloud. Then we simply repeat  $P_{i-1}$  and add the predicted displacement to produce  $P_i$ .

### 3.3 Training Loss

In our implementation, we use Chamfer distance as the primary loss function. We supervise all predicted point sets  $P_c \dots P_n$  with ground truth  $P_{gt}$ . And we add the distillation loss  $\mathcal{L}_{kd}$ . The scale factor is chosen as 1 during our experiment. The training loss can be formulated as:

$$\mathcal{L} = \mathcal{L}_{CD}(P_c, P_{gt}) + \sum_n \mathcal{L}_{CD}(P_i, P_{gt}) + \tau_0 \cdot \mathcal{L}_{kd}$$

## 4 Experiment

### 4.1 Experiment Setting

**Dataset** We empirically verify the merit of VD-PCN on PCN, ShapeNet-55 and MVP dataset. All of these datasets are created from ShapeNet. We follow the standard protocols into evaluate models on these three datasets. In particular, ShapeNet-55 contains 55 categories and includes 41, 952 and 10, 518 shapes in the training and testing sets, respectively. The evaluations on ShapeNet-55 are conducted on the point cloud data masked with a ratio of 25%50% and 75%, accordingly formulating the completion task at three difficulty levels of simple (S), moderate (M) and hard (H). The MVP[33] dataset contain 62400 sample from 2400 CAD models for training and 41600 samples for 1600 CAD models for testing. MVP dataset also provide 4 different resolution (2048, 4096, 9192, 16384). We report the highest resolution in the paper and provide other in the supplementary material.

**Implementation Detail** We implement our VD-PCN on the PyTorch platform[34]. The number of depth camera viewpoint  $k$  is 6, with resolution  $[H, W] = 224 \times 224$ . The 2d backbone Unet downsample the input in 16 times, producing a  $[H1, W1] = 14 \times 14$  and the channel  $c = 512$ . We choose set the number of coarse point  $P_c = 128$ . During experiment, we find that set  $P_c$  to 128 or 256 would produce better performance, comparing to 512 and 1024. We take two stage for coarse to fine point generation. In training process, we choose the AdamW optimizer[35] with weight decay = 0.0005. For the teacher, the learning rate is set as 0.0002 and we train it with 200 epoch. For the distillation training, we choose a small learning rate at 0.0001 and train it with 200 epoch. We use the L1 Chamfer Distance as the point cloud training loss, a MSE loss for the teacher / student’s global feature alignment and L1 loss for teacher / student’s visual feature loss. All the scale factor  $\tau_0, \tau_1, \tau_2$  are set to 1.

**Evaluation metrics** We employ the L1 Chamfer Distance as the evaluation metrics for the PCN and L2 Chamfer Distance for ShapeNet-55, MVP datasets. We also report the common used F score for evalution. On the ShapeNet55, there are three test difficulty and we also report an average Chamfer Distance on table 4.

### 4.2 Evaluation on benchmarks

Table 1: Performance comparison in terms of L1 Chamfer Distance  $\times 10^3$  ( $CD_{L1}$ ) on the PCN dataset. The Chamfer Distance performances of each category and the averaged result across all categories are all listed. (Lower  $CD_{L1}$  is better)

Method	Plane	Cabinet	Car	Chair	Lamp	Sofa	Table	Boat	$CD_{L1}$
FoldingNet[36]	9.49	15.80	12.61	15.55	16.41	15.97	13.65	14.99	14.31
PCN[1]	5.50	22.70	10.63	8.70	11.00	11.34	11.68	8.59	9.64
GRNet[2]	6.45	10.37	9.45	9.41	7.96	10.51	8.44	8.04	8.83
PoinTr[3]	4.75	10.47	8.68	9.39	7.75	10.93	7.78	7.29	8.38
SnowFlakeNet[4]	4.29	9.16	8.08	7.89	6.07	9.23	6.55	6.40	7.21
SeedFormer[10]	3.85	9.05	8.06	7.06	5.21	8.85	6.05	5.85	6.74
AnchorFormer[5]	3.70	8.94	7.57	7.05	5.21	8.40	6.03	5.81	6.59
SVDFormer[6]	3.62	8.79	7.46	6.91	5.33	8.49	5.90	5.83	6.54
<b>Ours</b>	<b>3.52</b>	<b>8.51</b>	<b>7.36</b>	<b>6.55</b>	<b>5.04</b>	<b>8.27</b>	<b>5.73</b>	<b>5.58</b>	<b>6.32</b>

**Result on PCN** Table 1 summarizes the  $CD_{l1}$  comparisons on eight categories of the PCN[1] dataset. VD-PCN consistently outperforms all baselines in terms of both per-category  $CD_{l1}$  and the averaged distance. In general, lower Chamfer Distance indicates more accurate reconstructive shape. Specifically, our model achieves the averaged  $CD_{l1}$  6.32 which reduces the Chamfer Distance of the best competitor SVDFormer[6] by 0.22. Though both of SVDFormer and VD-PCN rebuild the 3D

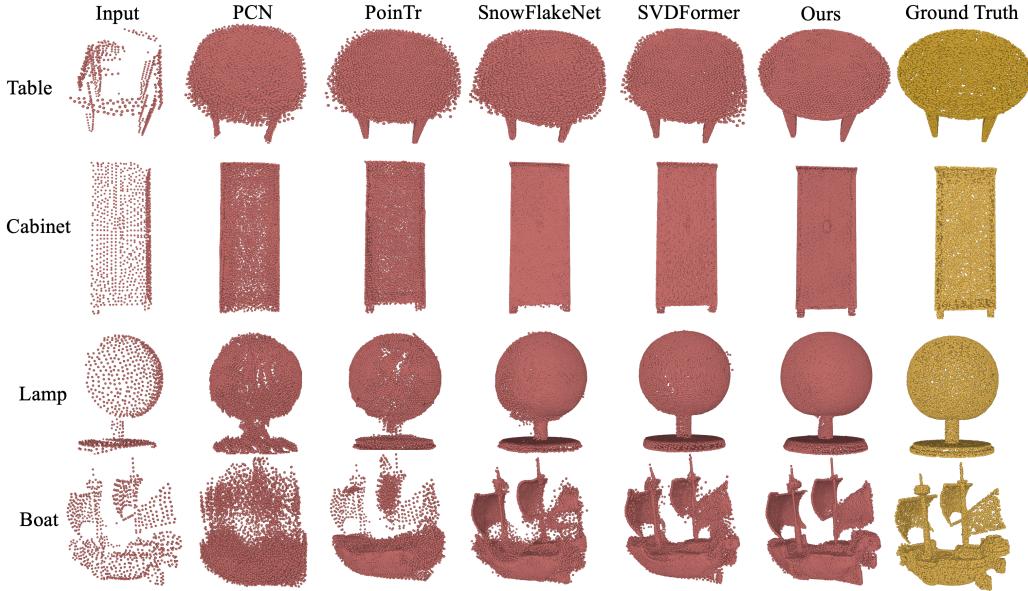


Figure 3: Visualizartion on the PCN Dataset.

shape with a multi-view projection way. SVDFormer utilize the combination of point information and image information while VD-PCN fully leverage the 2d information and the orderliness of 2d pixels. As indicated by the results, the knowledge transfer help our model predict better coarse point  $P_c$ .

We also provide a qualitative analysis on PCN dataset. Figure 3 further visualizes the point cloud completion results of four rather difficult categories. We choose two easier categories and two harder categories. In particular, for the table, VD-PCN is the only model predicting a round desktop, comparing with other arts. For cabinet and lamp, VD-PCN predicts fine-grained detail (i.e. the handle on the cabinet) and less noise. Also, VD-PCN performs better on the difficult boat, not only recovers more detail from limited input information but also brings less noise. We also provide visualization on other categories and failure case analysis in the supplementary material.

**Result on ShapNet 55** The test set of ShapeNet-55 can be classified into three levels of difficulty: simple (S), moderate (M), and hard (H), which correspond to different numbers of missing points (2,048, 4,096, and 6,144). Under all the difficulty, the the input points are subsample to 2048 considering the fixed width of a neural network. However, since we are processing the 2d images, we can accept more points than the point-based methods, which is a natural advantage, but for fairness, we still choose a common evaluation methods. The quantitative results are presented in table 2. The results under different difficulty levels are shown in performance comparison in terms of L2 Chamfer Distance  $\times 10^3$  ( $CD_{L2}$ ) on the ShapeNet-55 dataset. The per-category L2 Chamfer Distance results are reported on 5 categories with most training samples and 5 categories with the least training samples.

Method	Table	Chair	Plane	Car	Sofa	Birdhouse	Bag	Remote	Keyboard	Rocket	$CD_{L2}$
FoldingNet[36]	2.53	2.81	1.43	1.98	2.48	4.71	2.79	1.44	1.24	1.48	3.12
TopNet[37]	2.21	2.53	1.14	2.18	2.36	4.83	2.93	1.49	0.95	1.32	2.91
PCN[1]	2.13	2.29	1.02	1.85	2.06	4.50	2.86	1.33	0.89	1.32	2.66
GRNet[2]	1.63	1.88	1.02	1.64	1.72	2.97	2.06	1.09	0.89	1.03	1.97
PoinTr[3]	0.81	0.95	0.44	0.91	0.79	1.86	0.93	0.53	0.38	0.57	1.09
SeedFormer[10]	0.72	0.81	0.40	0.89	0.71	1.51	0.79	0.46	0.36	0.50	0.92
AnchorFormer[5]	0.58	0.67	0.33	0.69	0.58	1.35	0.64	0.36	0.27	0.42	0.76
<b>Ours</b>	<b>0.52</b>	<b>0.60</b>	<b>0.39</b>	<b>0.62</b>	<b>0.49</b>	<b>1.15</b>	<b>0.53</b>	<b>0.33</b>	<b>0.24</b>	<b>0.41</b>	<b>0.70</b>

Table 2: Comparision on the ShapeNet 55 dataset

**Result on MVP** We further test our model on the MVP dataset, The MVP dataset[33] is a multi-view partial point cloud dataset which consists of 16 categories of high-quality partial/complete point clouds. For each complete CAD shape, 26 partial point clouds are generated by selecting 26 camera poses which are uniformly distributed on a unit sphere. The resolution of depth camera used MVP is higher to stimulate the real scene. We show a long table at 16384 resolution and provide other resolution at the supplementary material. Measure with  $CD_{l2}$ , our model out perform strong baseline FBNet. The result is listed in table 3.

Methods	airplane	cabinet	car	chair	lamp	sofa	table	watercraft	bed	bench	bookshelf	bus	guitar	motorbike	pistol	skateboard	Avg.
PCN	2.95	4.13	3.04	7.07	14.93	5.56	7.06	6.08	12.72	5.73	6.91	2.46	1.02	3.53	3.28	2.99	6.02
TopNet	2.72	4.25	3.40	7.95	17.01	6.04	7.42	6.04	11.60	5.62	8.22	2.37	1.33	3.90	3.97	2.09	6.36
MSN	2.07	3.82	2.76	6.21	12.72	4.74	5.32	4.80	9.93	3.89	5.85	2.12	0.69	2.48	2.91	1.58	4.90
CRN	1.59	3.64	2.60	5.24	9.02	4.42	5.45	4.26	9.56	3.67	5.34	2.23	0.79	2.23	2.86	2.13	4.30
GRNet	1.61	4.66	3.10	4.72	5.66	4.61	4.85	3.53	7.82	2.96	4.58	2.97	1.28	2.24	2.11	1.61	3.87
NSFA	1.51	4.24	2.75	4.68	6.04	4.29	4.84	3.02	7.93	3.87	5.99	2.21	0.78	1.73	2.04	2.14	3.77
VRCNet	1.15	3.20	2.14	3.58	5.57	3.58	4.17	2.47	6.90	2.76	3.45	1.78	0.59	1.52	1.83	1.57	3.06
SnowflakeNet	0.96	3.19	2.27	3.30	4.10	3.11	3.43	2.29	5.93	2.29	3.34	1.81	0.50	1.72	1.54	2.13	2.73
FBNet	0.81	2.97	2.18	2.83	2.77	2.86	2.84	1.94	<b>4.81</b>	1.94	2.91	1.67	0.40	1.53	1.29	1.09	2.29
<b>Ours</b>	<b>0.65</b>	<b>2.82</b>	<b>2.11</b>	<b>2.58</b>	<b>2.74</b>	<b>2.69</b>	<b>2.72</b>	<b>1.82</b>	<b>4.82</b>	<b>1.65</b>	<b>2.85</b>	<b>1.55</b>	<b>0.34</b>	<b>1.38</b>	<b>1.24</b>	<b>0.96</b>	<b>2.16</b>

Table 3: Point cloud completion results on MVP dataset (16384 points) in terms of per-point L2 Chamfer distance ( $\times 10^4$ ).

Methods	$CD_{l2} - S$	$CD_{l2} - M$	$CD_{l2} - H$	$CD_{avg}$	F-Score% $\uparrow$
FoldingNet[1]	2.67	2.66	4.05	3.12	0.082
TopNet[37]	2.26	2.16	4.30	2.91	0.126
PCN[38]	1.94	1.96	4.08	2.66	0.133
GRNet[13]	1.35	1.71	2.85	1.97	0.238
PoinTr[39]	0.58	0.88	1.79	1.09	0.464
SeedFormer[33]	0.50	0.77	1.49	0.92	0.472
AnchorFormer[5]	0.41	0.61	1.26	0.76	<b>0.558</b>
SVDFFormer[6]	0.48	0.70	1.30	0.83	0.451
CRA-PCN [40]	0.48	0.71	1.37	0.85	-
<b>Ours</b>	<b>0.36</b>	<b>0.56</b>	<b>1.19</b>	<b>0.70</b>	<b>0.558</b>

Table 4: Comparison of methods on the MVP dataset based on  $CD-\ell_2$  and F-Score.

### 4.3 Analysis of VD-PCN

In this section, we do an accuracy-complexity trade-off test, evaluate the effectiveness of the distillation method and analyse different distillation strategy. We also discuss the motivation and current limitation of our work.

**Accuracy-Complexity Trade-Offs** We report the run-time FLOPs and inference time in Tab 5 and discuss the time consumption for the teacher training phase. All methods were evaluated on a single NVIDIA GeForce GTX 4090Ti graphic card. For fair comparisons, we disable gradient calculation and use point clouds with a resolution of 2,048. VD-PCN needs additional computation cost at teacher’s training while the distillation process is faster than training from scratch. And the teacher neither appear at the inference time nor provide any information. From the results on table 5, we see VD-PCN can achieve better trade-offs in performance and complexity. The time indicate the inference running time in unit input.

**Analysis of the different distillation strategy** Also we experiment with different distillation strategy. In the CNN pooling operation, feature map  $F_v \in \mathbb{R}^{k \times C \times H_1 \times W_1}$  would be further max pooled into a gloval feature tensor  $F_g \in \mathbb{R}^{k \times C}$ . In the table 6, we find that apply both  $f_v$  and  $f_g$  would produce better result, which also correspond with empirical analysis.

**Limitation** The major limitation of our work is the lack of study of a real-world scenario for the proposed framework. All the common used dataset (i.e. PCN, ShapeNet 55, MVP) are artificially

Methods	FLOPs (G)	Time (ms)	$CD_{l1}$
PoinTr[3]	18.81	15.34	8.38
SeedFormer[10]	107.51	16.50	6.74
AnchorFormer[5]	<b>14.54</b>	23.71	6.59
SVDFormer[6]	50.14	17.09	6.54
<b>Ours</b>	105.49	<b>15.00</b>	<b>6.32</b>

Table 5: Run-time analysis for FLOPs and inference time, both are the smaller the better.

Model	$f_v$	$f_g$	$CD_{l1}$	F-Score% $\uparrow$
A			6.55	0.841
B	✓		6.38	0.850
C		✓	6.34	0.851
D	✓	✓	<b>6.32</b>	<b>0.852</b>

Table 6: Ablation studies on the distillation strategy. Model A do not apply the knowledge transfer and share the exact model with other variants.

derived from ShapeNet[41]. In future work, we will focus on extending the work to real acquisitions, such as point cloud scene and LiDAR point cloud.

## 5 Conclusion

In this work, we presented VD-PCN, a novel approach for 3D point cloud completion that synergistically combines multi-view 2D representations with a knowledge distillation strategy. By leveraging mature 2D CNN models and the orderliness of 2D pixel data, our method achieves superior performance over prior point-based methods on various benchmarks. The multi-view encoder provides a panoramic understanding of the partial point cloud, while the distillation from a well-trained teacher model guides the student towards accurate completions. The following dual-modality decoder that integrates both 2D and 3D information for upsampling the coarse point cloud predictions to high resolution. Extensive experiments demonstrate the superiority of our method. Moreover, It will be an interesting future direction to extend the knowledge transfer strategy to similar completion-like tasks.

## References

- [1] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 international conference on 3D vision (3DV)*, pages 728–737. IEEE, 2018.
- [2] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. Grnet: Gridding residual network for dense point cloud completion. In *European Conference on Computer Vision*, pages 365–381. Springer, 2020.
- [3] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12498–12507, 2021.
- [4] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5499–5509, 2021.
- [5] Zhikai Chen, Fuchen Long, Zhaofan Qiu, Ting Yao, Wengang Zhou, Jiebo Luo, and Tao Mei. Anchorformer: Point cloud completion from discriminative nodes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13581–13590, 2023.
- [6] Zhe Zhu, Honghua Chen, Xing He, Weiming Wang, Jing Qin, and Mingqiang Wei. Svdformer: Complementing point cloud via self-view augmentation and self-structure dual-generator. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14508–14518, 2023.
- [7] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a simple and effective baseline. In *International Conference on Machine Learning*, pages 3809–3820. PMLR, 2021.
- [8] Xu Yan, Heshen Zhan, Chaoda Zheng, Jiantao Gao, Ruimao Zhang, Shuguang Cui, and Zhen Li. Let images give you more: Point cloud cross-modal training for shape analysis. *Advances in Neural Information Processing Systems*, 35:32398–32411, 2022.
- [9] 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.
- [10] Haoran Zhou, Yun Cao, Wenqing Chu, Junwei Zhu, Tong Lu, Ying Tai, and Chengjie Wang. Seedformer: Patch seeds based point cloud completion with upsample transformer. In *European conference on computer vision*, pages 416–432. Springer, 2022.
- [11] Ren-Wu Li, Bo Wang, Lin Gao, Ling-Xiao Zhang, and Chun-Peng Li. High-fidelity point cloud completion with low-resolution recovery and noise-aware upsampling. *Graphical Models*, 126:101173, 2023.
- [12] Zhe Zhu, Liangliang Nan, Haoran Xie, Honghua Chen, Jun Wang, Mingqiang Wei, and Jing Qin. Csdn: Cross-modal shape-transfer dual-refinement network for point cloud completion. *IEEE Transactions on Visualization and Computer Graphics*, 2023.
- [13] Xiaogang Wang, Marcelo H Ang Jr, and Gim Hee Lee. Cascaded refinement network for point cloud completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 790–799, 2020.
- [14] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2018.
- [15] Xuancheng Zhang, Yutong Feng, Siqi Li, Changqing Zou, Hai Wan, Xibin Zhao, Yandong Guo, and Yue Gao. View-guided point cloud completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15890–15899, 2021.

- [16] Emanuele Aiello, Diego Valsesia, and Enrico Magli. Cross-modal learning for image-guided point cloud shape completion. *Advances in Neural Information Processing Systems*, 35:37349–37362, 2022.
- [17] Yoni Kasten, Ohad Rahamim, and Gal Chechik. Point cloud completion with pretrained text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.
- [18] Xuejun Yan, Hongyu Yan, Jingjing Wang, Hang Du, Zhihong Wu, Di Xie, Shiliang Pu, and Li Lu. Fbnet: Feedback network for point cloud completion. In *European Conference on Computer Vision*, pages 676–693. Springer, 2022.
- [19] Junshu Tang, Zhijun Gong, Ran Yi, Yuan Xie, and Lizhuang Ma. Lake-net: Topology-aware point cloud completion by localizing aligned keypoints. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1726–1735, 2022.
- [20] Fangzhou Lin, Yun Yue, Ziming Zhang, Songlin Hou, Kazunori Yamada, Vijaya Kolachalam, and Venkatesh Saligrama. Infocd: A contrastive chamfer distance loss for point cloud completion. *Advances in Neural Information Processing Systems*, 36, 2024.
- [21] Fangzhou Lin, Yun Yue, Songlin Hou, Xuechu Yu, Yajun Xu, Kazunori D Yamada, and Ziming Zhang. Hyperbolic chamfer distance for point cloud completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14595–14606, 2023.
- [22] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2016.
- [23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.
- [24] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [25] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [26] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [27] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- [28] Linfeng Zhang, Runpei Dong, Hung-Shuo Tai, and Kaisheng Ma. Pointdistiller: structured knowledge distillation towards efficient and compact 3d detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21791–21801, 2023.
- [29] Pin Tang, Hai-Ming Xu, and Chao Ma. Prototransfer: Cross-modal prototype transfer for point cloud segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3337–3347, 2023.
- [30] Hanwen Jiang, Zhenyu Jiang, Yue Zhao, and Qixing Huang. Leap: Liberate sparse-view 3d modeling from camera poses. *arXiv preprint arXiv:2310.01410*, 2023.
- [31] Xianzu Wu, Xianfeng Wu, Tianyu Luan, Yajing Bai, Zhongyuan Lai, and Junsong Yuan. Fsc: Few-point shape completion. *arXiv preprint arXiv:2403.07359*, 2024.
- [32] Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. Online knowledge distillation via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11020–11029, 2020.

- [33] Liang Pan, Xinyi Chen, Zhongang Cai, Junzhe Zhang, Haiyu Zhao, Shuai Yi, and Ziwei Liu. Variational relational point completion network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8524–8533, 2021.
- [34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
- [35] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [36] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 206–215, 2018.
- [37] Lyne P Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. Topnet: Structural point cloud decoder. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 383–392, 2019.
- [38] Minghua Liu, Lu Sheng, Sheng Yang, Jing Shao, and Shi-Min Hu. Morphing and sampling network for dense point cloud completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11596–11603, 2020.
- [39] Liang Pan. Ecg: Edge-aware point cloud completion with graph convolution. *IEEE Robotics and Automation Letters*, 5(3):4392–4398, 2020.
- [40] Yi Rong, Haoran Zhou, Lixin Yuan, Cheng Mei, Jiahao Wang, and Tong Lu. Cra-pcn: Point cloud completion with intra-and inter-level cross-resolution transformers. *arXiv preprint arXiv:2401.01552*, 2024.
- [41] 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.

## 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: In the abstract and introduction, we share two observation in the completion task and give out solution.

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: In the 4.3 Analysis of model part, we give an discussion about the limitation of current work.

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: [No]

Justification: There is no theoretical result in this paper.

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: We would provide code, training detail, error bar, and model weight for Reproducibility.

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: [Yes]

Justification: Yes, we would provide open access.

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: We have specified all the training details in.

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: We conduct multiple times of experiment and ablation study and provide the error bar to verify the statistical significance.

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: For the experiments, we mainly finish it on a 4 \* Nvidia A800 GPU. But 4\* Nvidia 3090 GPU can reproduce the whole experiment. Training on the PCN dataset would cost 1.5day.

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 have checked the NeurIPS code of Ethics and the whole research was the research conducted in the paper conform with 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: This application paper do not talk about broader 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: There is no societal impact of the work performed.

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: The creators and original owners are properly credited.

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](https://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 give our code, model, training detail and licence in the supplementary material and would provide it in a structured template later.

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: The paper does not involve crowdsourcing nor 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: The paper does not involve crowdsourcing nor research with human subjects.

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.