Implicit Autoencoder for Point Cloud Self-supervised Representation Learning

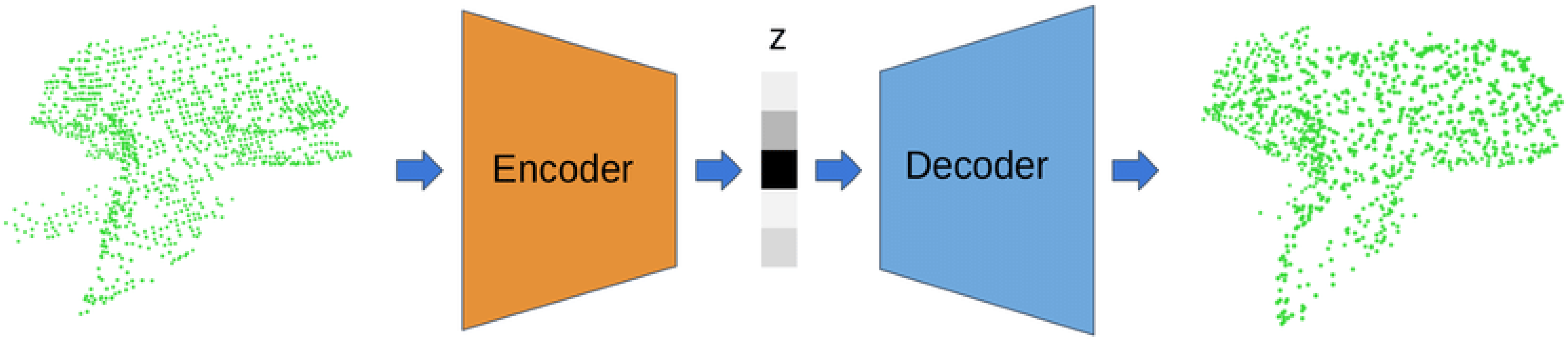
Siming Yan, Zhenpei Yang, Haoxiang Li, Li Guan, Hao Kang, Gang Hua, Qixing Huang

Motivation

Autoencoders - traditionally input = output = point cloud

But point clouds have sampling variations

as a result, a point-based AE forces (some) useless encoding.



Their solution

In broad strokes: decoder outputs a CONTINUOUS representation shared among different point cloud samplings of the same model.

Two strengths:

- Discards sampling variations in the output of decoder
- Minimizing discrepancy between two implicit functions DOES NOT require computing correspondence (e.g. using chamfer distance). Faster.

Explicit AE

$$\min_{\Theta,\Phi} d_{\text{exp}}((g_\Phi \circ f_\Theta)(\mathcal{P}^{\text{in}}), \mathcal{P}^{\text{gt}}_{\text{sub}})$$

Example of distance function can be the chamfer distance, which gives a distance between two point-clouds.

Implicit AE

$$\min_{\Theta,\Phi} d_{\text{imp}}((g_\Phi \circ f_\Theta)(x|\mathcal{P}^{\text{in}}), g_0(x))$$

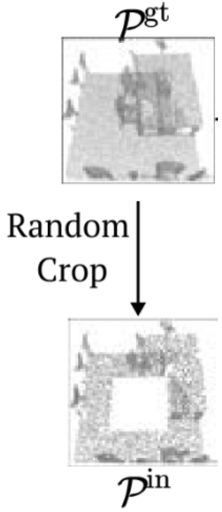
Groundtruth implicit function obtained as SDF, occupancy grid …etc. Choice of distance function is coupled with the type of implicit representation used.

Loss Function

$$\mathcal{L} = \frac{1}{N}\sum_{i=0}^{N} \|(g_\Phi \circ f_\Theta)(x_i|\mathcal{P}^{\text{in}}) - g_0(x_i)\|$$

Random crops

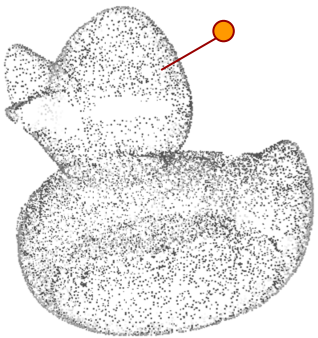
To capture high-level semantic features, random crop part of the input point cloud, then reconstruct missing parts. They show resulting point cloud can perform better on downstream tasks



How do you obtain ground truth implicits?

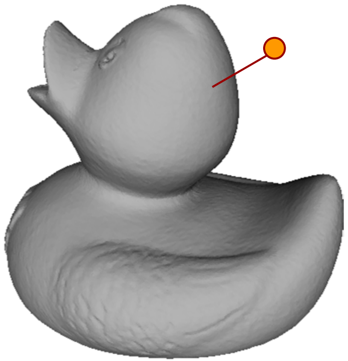
Real data:

- Compute closest distance between query point and groundtruth points
- Use unsigned distance function



Synthetic data:

- Signed distance function obtained from underlying water-tight meshes.



Evaluation

- Focus on a pretrain-then-transfer setting
- Pretraining:
  - ShapeNet: synthetic dataset -> access to water-tight meshes
  - ScanNet: real indoor scenes -> grountruths are approximated
- Downstream tasks
  - ShapeNet: Shape Classification
  - ScanNet: Indoor 3D object Detection, 3D semantic segmentation

Shape Classification

Method	ModelNet40
3D-GAN [49]	83.3%
Latent-GAN [1]	85.7%
SO-Net [21]	87.3%
MAP-VAE [16]	88.4%
Jigsaw* [42]	84.1%
FoldingNet* [54]	90.1%
Orientation* [36]	90.7%
STRL* [20]	90.9%
OcCo* [47]	89.7%
IAE(ours)	**92.1%**

Table 1: **Linear evaluation for shape classification on ModelNet40**. Note that to make a fair comparison, different * methods use the same DGCNN encoder backbone.

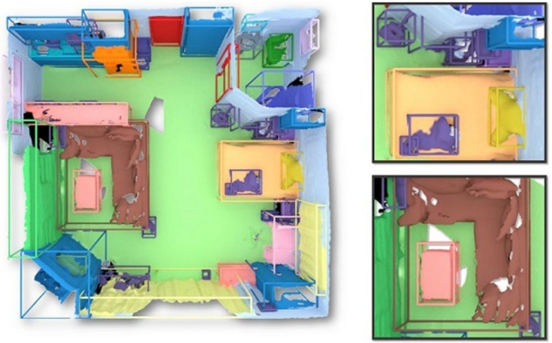
Category	Method	ModelNet40
Supervised	PointNet [38]	89.2%
	PointNet++ [39]	90.7%
	PointCNN [22]	92.2%
	KPConv [44]	92.9%
	DGCNN [48]	92.9%
	PointTransform [59]	93.7%
Self-Supervised	FoldingNet [54]	93.1%
	STRL [20]	93.1%
	OcCo [47]	93.0%
	IAE(ours)	**93.7%**

Table 2: **Shape classification fine-tuned results on ModelNet40.** Supervised learning methods train the model from scratch. Self-supervised methods use the pre-trained models as the initial weight for supervised fine-tuning. All the self-supervised methods shown here use the same DGCNN encoder backbone.

Object Detection in 3D Scenes

Method	ScanNet		SUN RGB-D	
	AP_{50}	AP_{25}	AP_{50}	AP_{25}
VoteNet [37]	33.5	58.6	32.9	57.7
STRL [20]	38.4	59.5	35.0	58.2
RandomRooms [40]	36.2	61.3	35.4	59.2
PointContrast [53]	38.0	59.2	34.8	57.5
DepthContrast[2] [58]	39.1	**62.1**	35.4	**60.4**
IAE (Ours)	**39.8**	61.5	**36.0**	**60.4**

Table 3: **3D object detection results.** We fine-tuned our pre-trained model on ScanNetV2 and SUN-RGBD validation set using a popular detection framework, VoteNet [37]. We show mean of average precision(mAP) across all semantic classes with 3D IoU threshold 0.25 and 0.5. Our method outperforms prior work across most metrics.



Main takeaway: transferable! Unlike other methods for self-supervision in the past in this task setting.

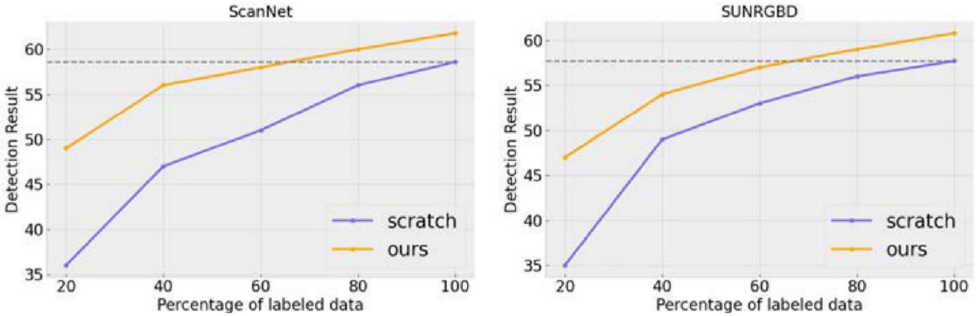
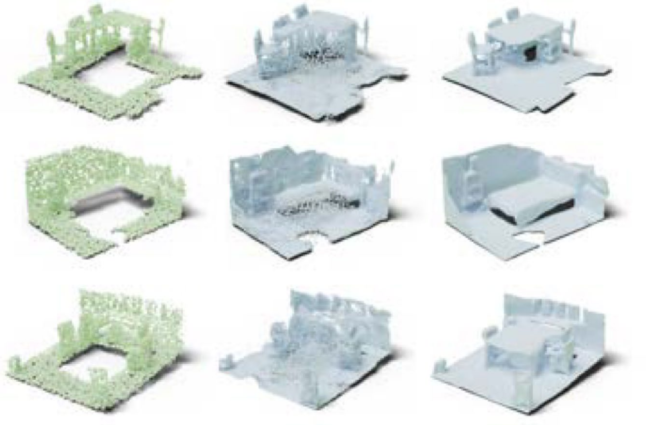


Fig. 4: **Label efficiency training.** We pre-train our model on ScanNet and then fine-tune on ScanNet and SUN RGB-D separately. During fine-tuning, different percentages of labeled data are used. Our pre-training model outperforms training from scratch and achieves nearly the same result with only 60% labeled data.

Decoder	Method	ModelNet40
Explicit	FoldingNet [54]	90.1%
	OcCo [47]	89.7%
	SnowflakeNet [52]	89.9%
Implicit	OccNet [26]	91.5%
	Conv-OccNet [35]	**92.1%**

Decoder	Functions	ModelNet40
Explicit	Point Cloud	90.1%
Implicit	Occ Value	91.3%
	UDF	91.7%
	SDF	**92.1%**

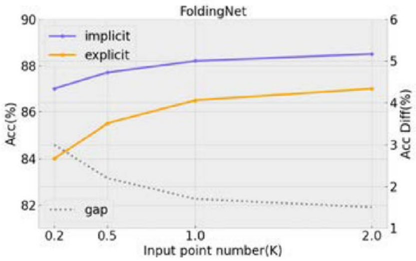
Table 6: Left: **Ablation study on different decoder model.** On ModelNet40, we show linear evaluation results. Our implicit auto-encoder formulations can be improved upon explicit counterpart under various decoder models. Right: **Ablation study on implicit function.** For explicit representation, we use FoldingNet as the decoder. For implicit representation, we experimented with Occupancy Value(Occ Value), Unsigned Distance Function(UDF), and Signed Distance Function(SDF) and find consistent improvement over explicit representation.

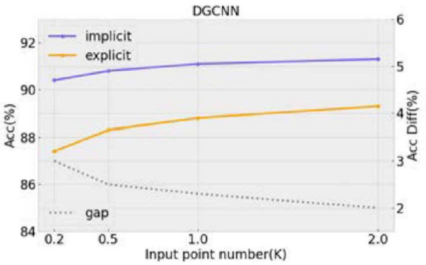


(a) Input (b) SnowflakeNet (c) IAE(ours)

Pros

More robust to point cloud resolutions





No need to compute difference between two sets, saves on compute time.

Allows for better self-supervision for a range of downstream tasks.

Cons

Need groundtruth implicit from point cloud, which is an additional preprocessing step.

Additionally, sampling has to be done across the VOLUME, instead of the SURFACE.