SeRP: Self Supervised Representation Learning Using Perturbed Point Clouds

Siddhant Garg | Mudit Chaudhary

COMPSCI 674: Intelligent Visual Computing
Final Project Presentation
Content

● Motivation and Introduction
● Related Works
● Methodology
  ○ Point Cloud Perturbation
  ○ SeRP-PointNet
  ○ SeRP-Transformer
  ○ VASP-Transformer
● Experiments and Results
● Conclusion and Future Work
Motivation

1. 3D models are trained from scratch unlike 2D vision models that are pre-trained on ImageNet.
2. Annotating 3D data is time consuming.
3. Lots of 3D data available in the form on raw points clouds.
4. Self-Driving Cars, Robotics
   a. SSL can help in learning world-knowledge
   b. Large annotated datasets, or sensors may cause redundancies. (Supervised methods may not be scalable.)
SeRP framework works as follows:

1. **Perturbation**: Point cloud patches are perturbed using Gaussian noise.
2. **Autoencoder**:
   a. Encoder: Encodes the latent representation of the point cloud
   b. Decoder: Reconstructs the original point cloud using the latent representations.
3. **Representations**: Encoder representations are used for downstream tasks.
Related Works

Point-BERT

Yu et al. 2021: Point-BERT: Pre-training 3D Point Cloud Transformers with Masked Point Modeling

Point-MAE

Y. Pang et al 2022. Point-MAE: Masked Autoencoders for Point Cloud Self-Supervised Learning
Methodology
Point Cloud Perturbation

1. Randomly sample 20 points
2. Apply KNN to find 20 nearest points
3. Sample Gaussian noise with mean 0 and standard deviation of 0.03 for each group
4. Add the sampled gaussian noise to each of the groups
SeRP-PointNet
SeRP-PointNet modifies the existing PointNet architecture as shown below.
It performs self-supervision using two tasks:

1. **Classification**: Classifies whether the point is perturbed or not
2. **Reconstruction**: Reconstructs the perturbed point to its original position
SeRP-PointNet

For classification, we use a fully-connected layer (n, 2) as the classification head and use cross-entropy loss. For reconstruction, we employ a fully-connected layer (n, 3) as the reconstruction head.

For reconstruction, we provide two methods:

1. δ-learning: The reconstruction head predicts δ, i.e. the 3-coordinate differences between the ground truth point cloud and the perturbed point cloud. To calculate the loss, we use a Mean Squared Loss function.

2. CDL2-learning: The reconstruction head predicts the 3-d coordinates directly from the per-point features. For this method, we use Chamfer Distance L2 loss shown below.

\[
\mathcal{L}(P, \hat{P}) = \frac{1}{|\hat{P}|} \sum_{x \in P} \min_{y \in \hat{P}} \|x - y\|_2^2 + \frac{1}{|P|} \sum_{x \in \hat{P}} \min_{y \in P} \|x - y\|_2^2
\]
SeRP-Transformer
SeRP-Transformer

Perturbed Point Cloud

FPS
+ KNN

PointNet Tokenizer

[CLS]

Token Embeddings
SeRP-Transformer

[CLS]

Token Embeddings

Transformer Encoder
Decoder with Unmasked Self-Attention

Encoder

Decoder

Reconstructed Point Cloud

Transformers: Vaswani et al. 2017 - Attention Is All You Need
SeRP-Transformer Learning Objective

- Reconstruction Loss: Chamfer L2 Loss function

\[
\mathcal{L}(P, \hat{P}) = \frac{1}{|\hat{P}|} \sum_{x \in \hat{P}} \min_{y \in P} \|x - y\|_2^2 + \frac{1}{|P|} \sum_{x \in P} \min_{y \in \hat{P}} \|x - y\|_2^2
\]
Vector Quantization

\[ z_q(x) = \arg \min_{z_q \in \mathcal{Z}} \| z_e(x) - z_q(x) \|_2 \]

VASP: Vector-Quantized Autoencoder for Self-Supervised Representation Learning for Point Clouds

VQ-VAE: Neural Discrete Representation Learning
Vector-Quantization Objectives

\[ \mathcal{L}_{VQ} = \log p(x | z_q(x)) + \alpha \| \text{sg}[z_e(x)] - e \|_2^2 + \beta \| z_e(x) - \text{sg}[e] \|_2^2 \]

- Reconstruction Loss:
  \[ \mathcal{L}(P, \hat{P}) = \frac{1}{|\hat{P}|} \sum_{x \in \hat{P}} \min_{y \in P} \| x - y \|_2^2 + \frac{1}{|P|} \sum_{x \in P} \min_{y \in \hat{P}} \| x - y \|_2^2 \]

- Embedding Loss: \[ \| \text{sg}[z_e(x)] - e \|_2^2 \] Move embeddings to encoder output.

- Commitment Loss: \[ \| z_e(x) - \text{sg}[e] \|_2^2 \] To commit the encoder output to an embedding and limit its output space.

\text{sg}: \text{stop-gradient}
Experiments and Results
We perform two sets of experiments:

1. **Pre-training:** We pre-train our autoencoders on ShapeNet55 dataset.
   a. We sample and perturb 1024 points from each point cloud with gaussian noise (0.0, 0.03).
   b. We use AdamW optimizer with initial learning rate 0.001 with cosine annealing and batch size 128.
   c. SeRP-Transformer is trained for 300 epochs while SeRP-PointNet is trained for 100 epochs.

2. **Downstream Evaluation:** We use the pre-trained encoders and finetune it on ModelNet40 classification task to evaluate the performance gain from pre-training.
Experiments – ShapeNet Pre-training

ShapeNet Point Cloud

Sampled Point Cloud
Experiments – ShapeNet Pre-training

Sampled Point Cloud

Perturbed Point Cloud
## Results – Example Reconstructions

<table>
<thead>
<tr>
<th>Ground Truth</th>
<th>Perturbed</th>
<th>Reconstructed</th>
<th>Ground Truth</th>
<th>Perturbed</th>
<th>Reconstructed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PointNet</strong></td>
<td><img src="image1" alt="PointNet Perturbed" /></td>
<td><img src="image2" alt="PointNet Reconstructed" /></td>
<td><img src="image3" alt="PointNet Ground Truth" /></td>
<td><img src="image4" alt="PointNet Perturbed" /></td>
<td><img src="image5" alt="PointNet Reconstructed" /></td>
</tr>
<tr>
<td><strong>SeRP Transformer</strong></td>
<td><img src="image6" alt="SeRP Transformer Perturbed" /></td>
<td><img src="image7" alt="SeRP Transformer Reconstructed" /></td>
<td><img src="image8" alt="SeRP Transformer Ground Truth" /></td>
<td><img src="image9" alt="SeRP Transformer Perturbed" /></td>
<td><img src="image10" alt="SeRP Transformer Reconstructed" /></td>
</tr>
<tr>
<td><strong>VASP</strong></td>
<td><img src="image11" alt="VASP Perturbed" /></td>
<td><img src="image12" alt="VASP Reconstructed" /></td>
<td><img src="image13" alt="VASP Ground Truth" /></td>
<td><img src="image14" alt="VASP Perturbed" /></td>
<td><img src="image15" alt="VASP Reconstructed" /></td>
</tr>
</tbody>
</table>
Results – Downstream Evaluation

### Table 1: Downstream evaluation results using SeRP-Transformer

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ModelNet40</th>
<th>ShapeNet</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy</td>
<td>Gain</td>
</tr>
<tr>
<td>Scratch</td>
<td>88.48</td>
<td>-</td>
</tr>
<tr>
<td>SeRP-Net</td>
<td>89.1</td>
<td>0.62 ↑</td>
</tr>
<tr>
<td>VASP</td>
<td>87.85</td>
<td>-0.63 ↓</td>
</tr>
</tbody>
</table>

### Table 2: Downstream evaluation results using SeRP-PointNet

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ModelNet40</th>
<th>ShapeNet</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy</td>
<td>Gain</td>
</tr>
<tr>
<td>Scratch</td>
<td>82.97</td>
<td>-</td>
</tr>
<tr>
<td>δ learn</td>
<td>84.1</td>
<td>1.13 ↑</td>
</tr>
<tr>
<td>cd l2 learn</td>
<td>84.06</td>
<td>1.09 ↑</td>
</tr>
</tbody>
</table>
Conclusion

1. We presented a self-supervised learning paradigm to learn latent representations of point cloud data.
2. Pre-trained models performed better on the downstream tasks in-comparison to the models trained from scratch noticing a 0.5-1% performance gain on ModelNet40 and ShapeNet55 classification tasks.
Future Works

1. Propose a more challenging strategy to perturb point clouds by sampling centers using Far Point Sampling (FPS).
2. Compare the existing approach with traditional variational inference and discrete variational inference methods.
Thank you