AFEKA
The Academic College of Engineering in Tel Aviv

SCHOOL OF SOFTWARE ENGINEERING: INTELLIGENT SYSTEMS

Point Cloud Denoising Using Self-Supervised Transformers

Final Report Presentation

Superviser: Dr. Dovi Yellin

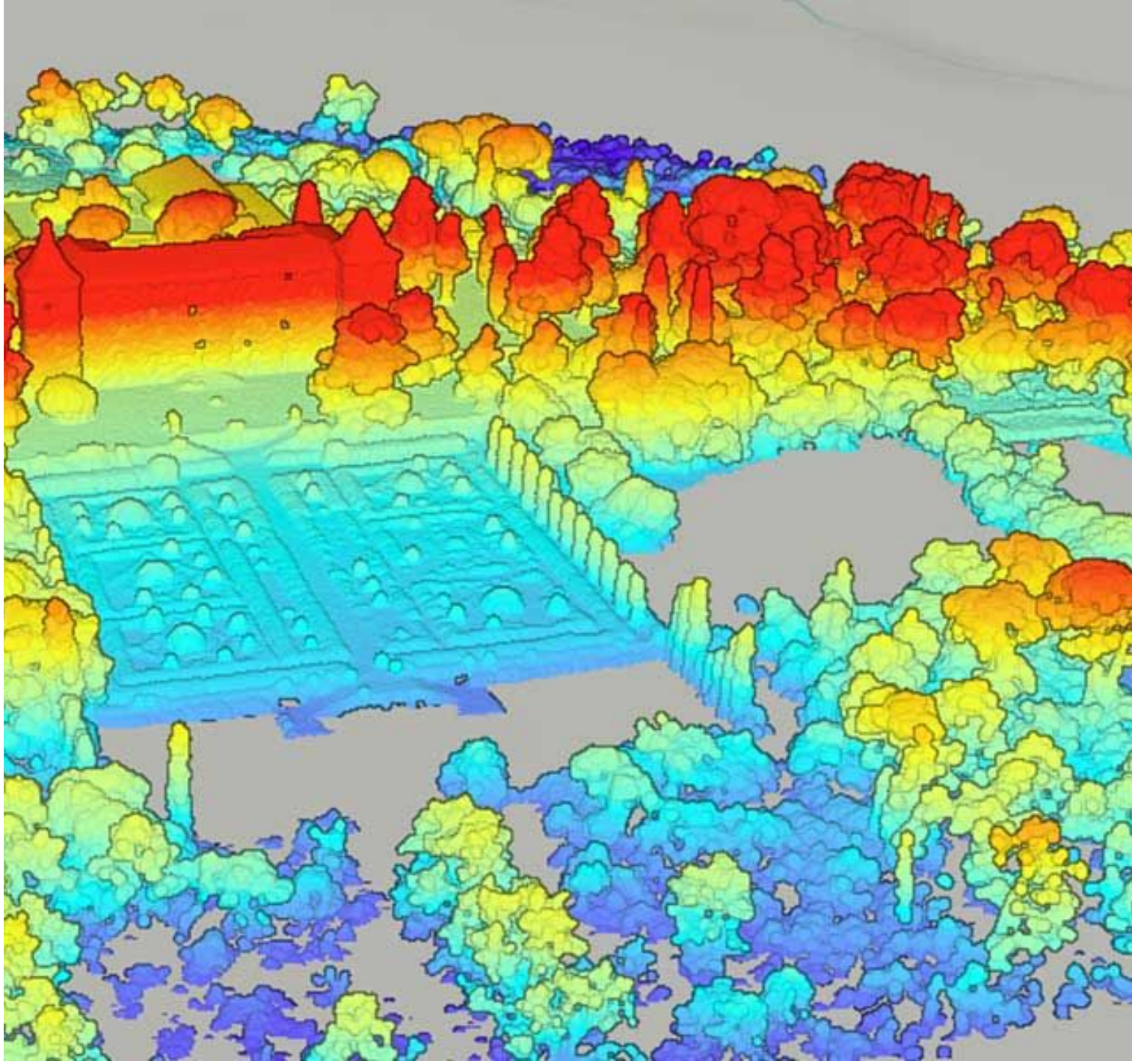
Adviser: Dr. Yehudit Aperstain

Presented by: Maoz Koren

20.06.2025

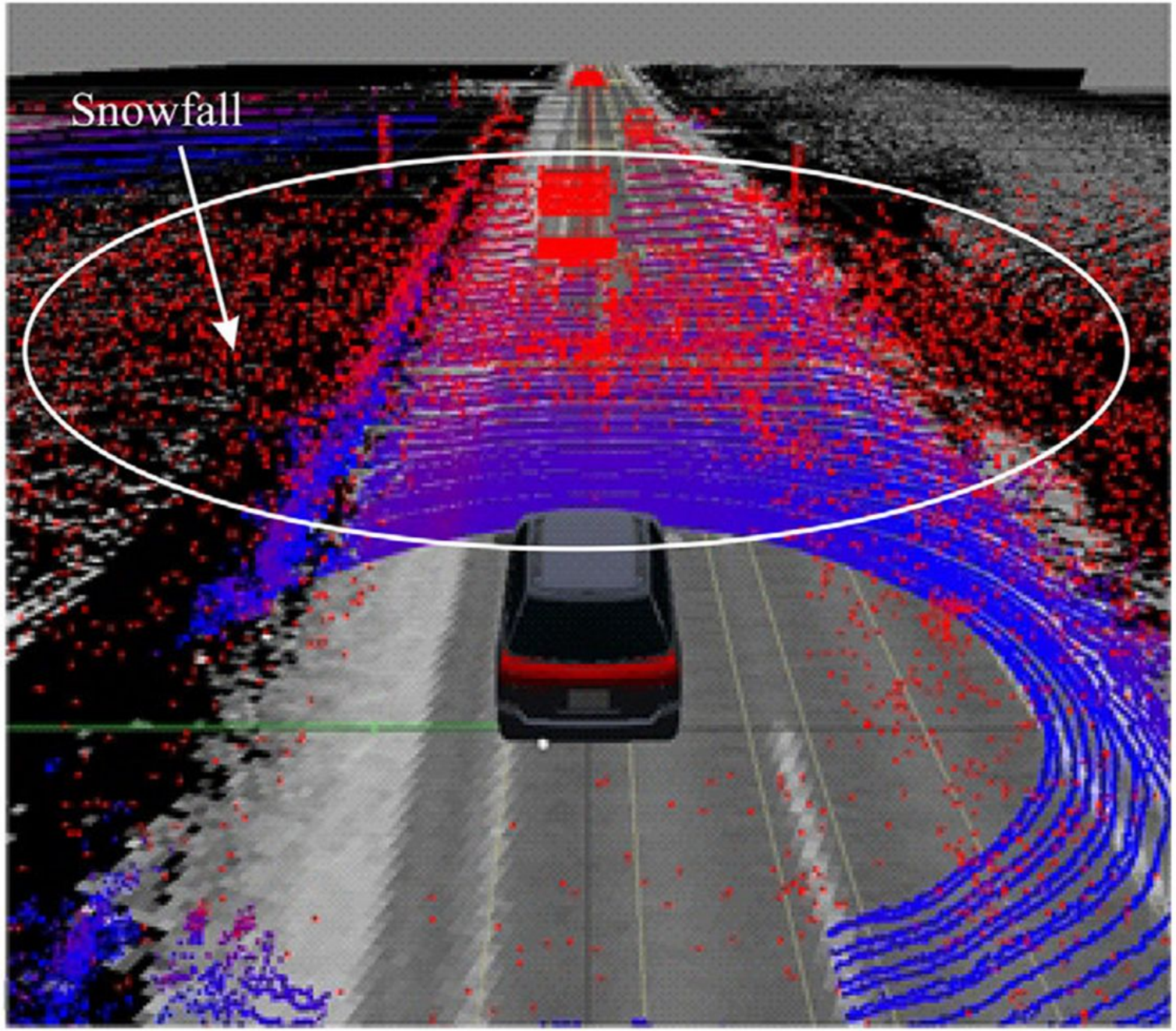
Motivation And Background

- Processing of Data collected from Light Detection and Ranging (LiDAR) systems is an integral part of many technological sectors, such as:
  - Urban planning
  - Autonomous driving
  - Mining and construction
  - Security and surveillance



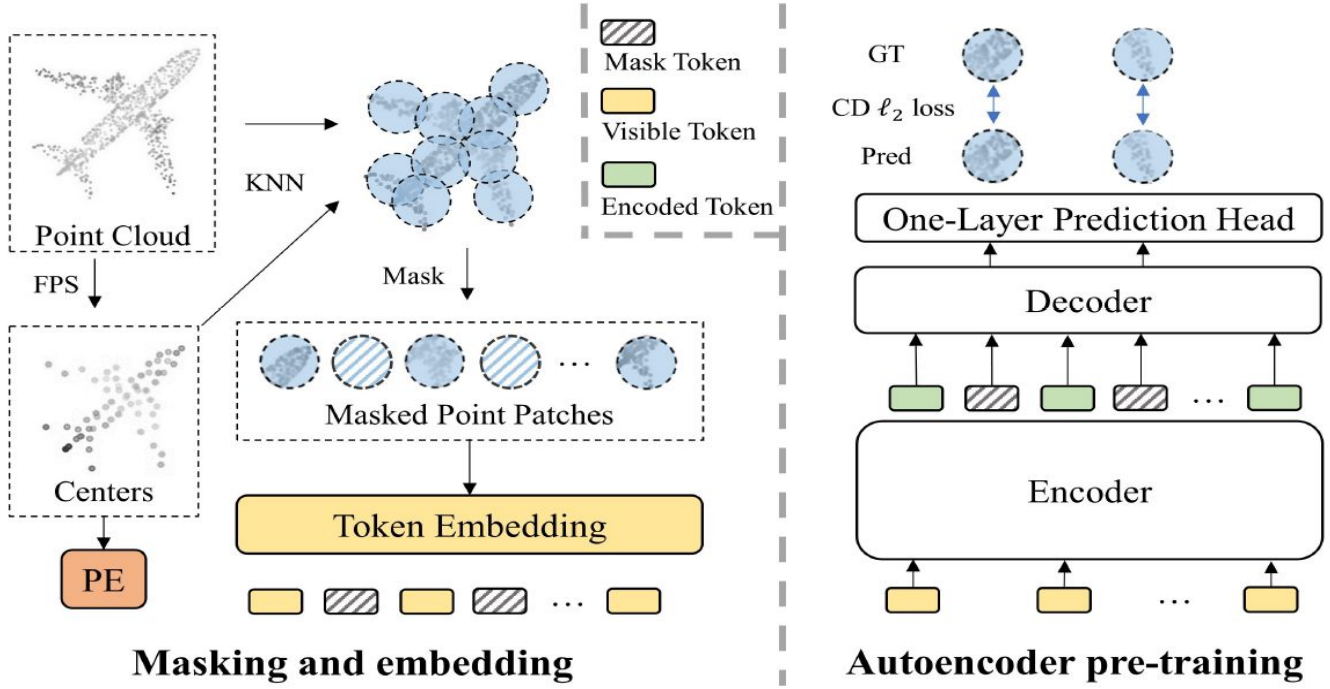
Research Problem

- LiDAR data is susceptible to noise. These noise factors, caused by various sources such as:
  - Atmospheric conditions
  - Sensor Limitations
  - Reflective properties of scanned objects
- Noise can compromise and impede precise data interpretation:
  - Object detection
  - Classification
  - Localization



Point Cloud Processing Techniques

Point-MAE is an Encoder-Decoder architecture, built out of transformer layers, that uses self-supervised learning for point cloud segmentation and classification, and includes Positional Embeddings (PE's) for allowing location information to be retained during the decoding phase for better results.



Point Cloud Processing Techniques

Point-MAE is an Encoder-Decoder architecture, built out of transformer layers, that uses self-supervised learning for point cloud segmentation and classification, and includes Positional Embeddings (PE's) for allowing location information to be retained during the decoding phase for better results.

Point Cloud Processing Techniques

Object Classification on ModelNet40 Dataset

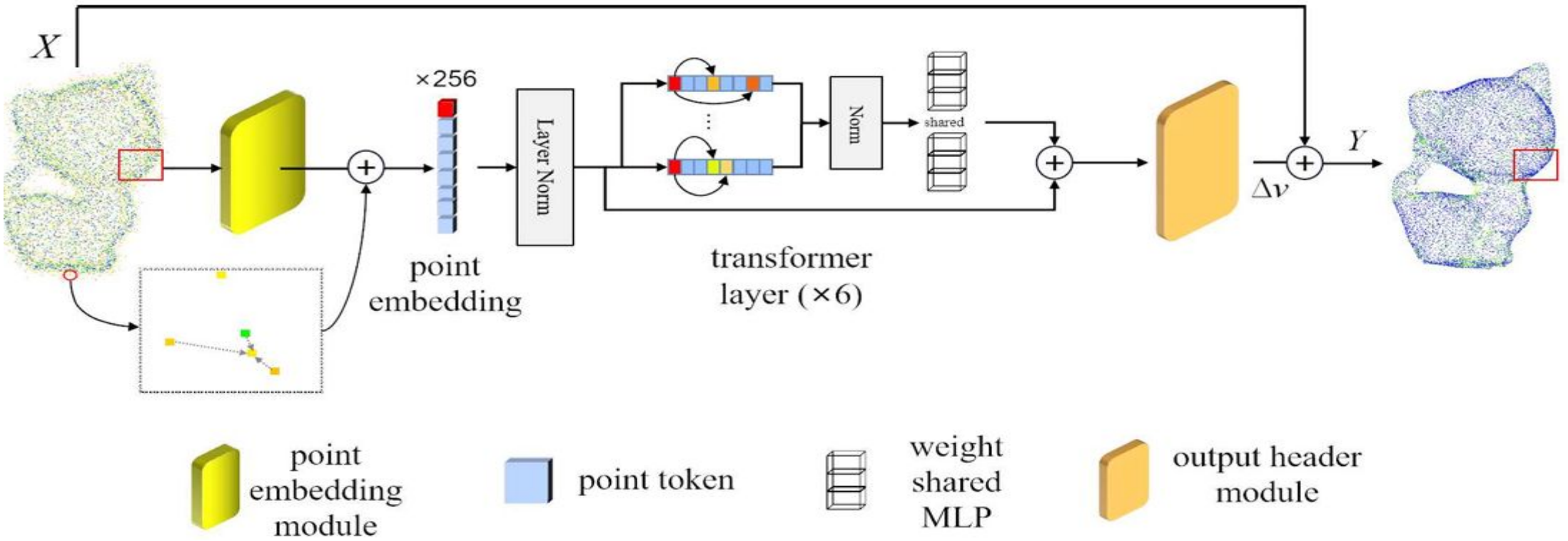
Note that object classification success rate in generative models is higher than in non-generative techniques

Self-supervised methods	Accuracy
OcCo [41]	93.0%
STRL [19]	93.1%
IAE [51]	93.7%
[ST]Transformer-OcCo [54]	92.1%
[ST]Point-BERT [54]	93.2%
[ST]**Point-MAE**	**93.8%**

Supervised methods	Accuracy
PointNet [29]	89.2%
PointNet++ [30]	90.7%
PointCNN [21]	92.5%
KPConv [38]	92.9%
DGCNN [21]	92.9%
RS-CNN [24]	92.9%
[T]PCT [16]	93.2%
[T]PVT [57]	93.6%
[T]PointTransformer [59]	93.7%
[ST]Transformer [54]	91.4%

Point Cloud Denoising Techniques

NoiseTrans. A supervised transformer-based encoder architecture for 3D point cloud object denoising (using supervised training on ModelNet40).



Research Objectives and Contributions

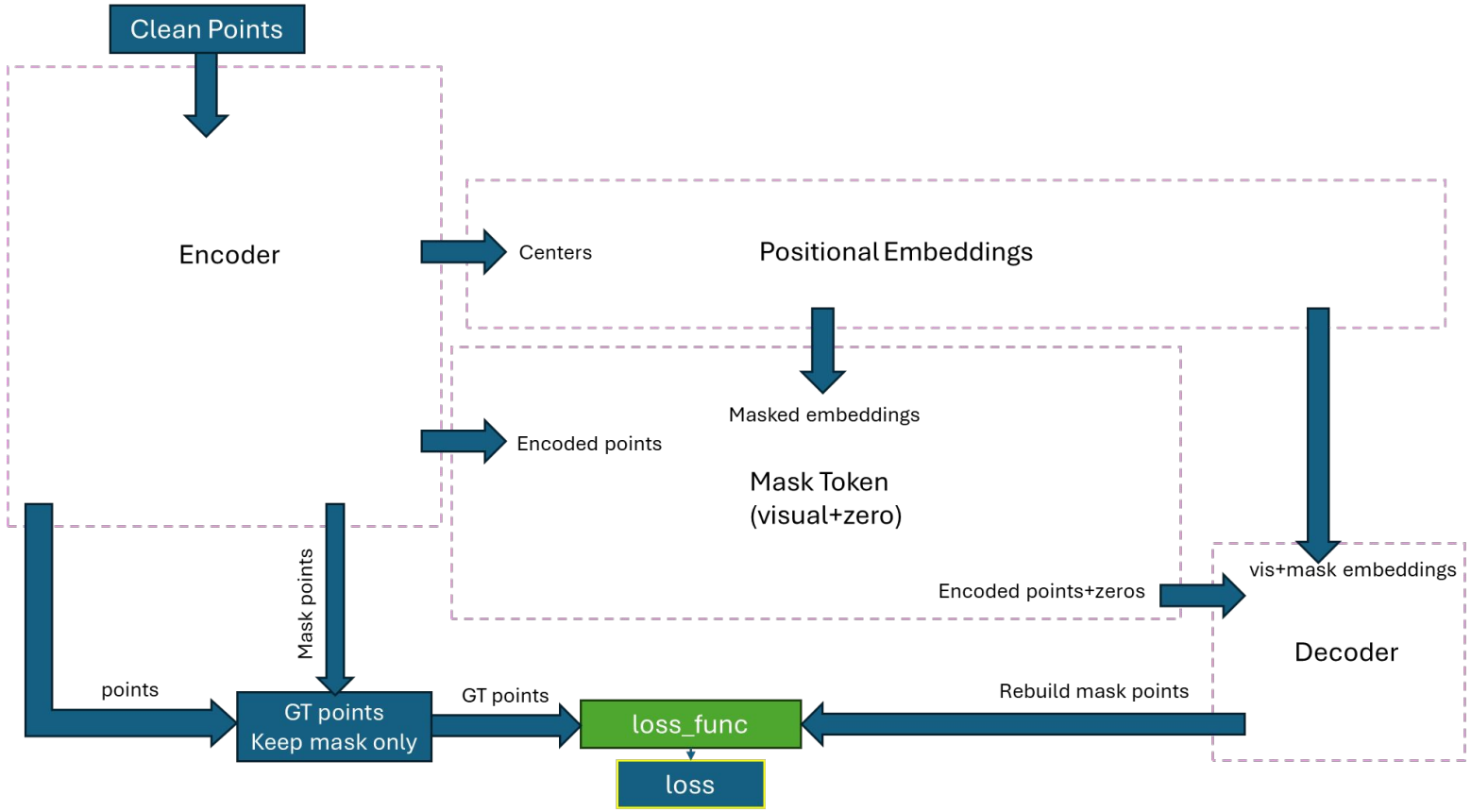
- Develop and optimize a denoising architecture for 3D point cloud scans based on a generative self-supervised learning framework (a novelty compared to previous supervised learning methods).
- Avoid reliance on large labeled datasets by combining self-supervised learning and lightweight supervised fine-tuning.
- This method aims to learn robust noise-resilient representations while preserving spatial fidelity after denoising.
- Model significantly outperforms baselines like NoiseTrans, PCNet, and MRPCA on standard evaluation metric.
- We hope to advance 3D scanned objects/scenes processing ability by creating a SOTA point cloud denoising model, as cleaner point clouds can lead to better obstacle detection, improved navigation, and safer movement for robots and vehicles.

Methods

- SOTA models for point cloud processing and denoising use **supervised** transformers to capture the context of the scanned object and filter out noise.
- In this project, we use transformer layers and **self-supervised** learning, along with a small scale classification network to denoise 3D point clouds.

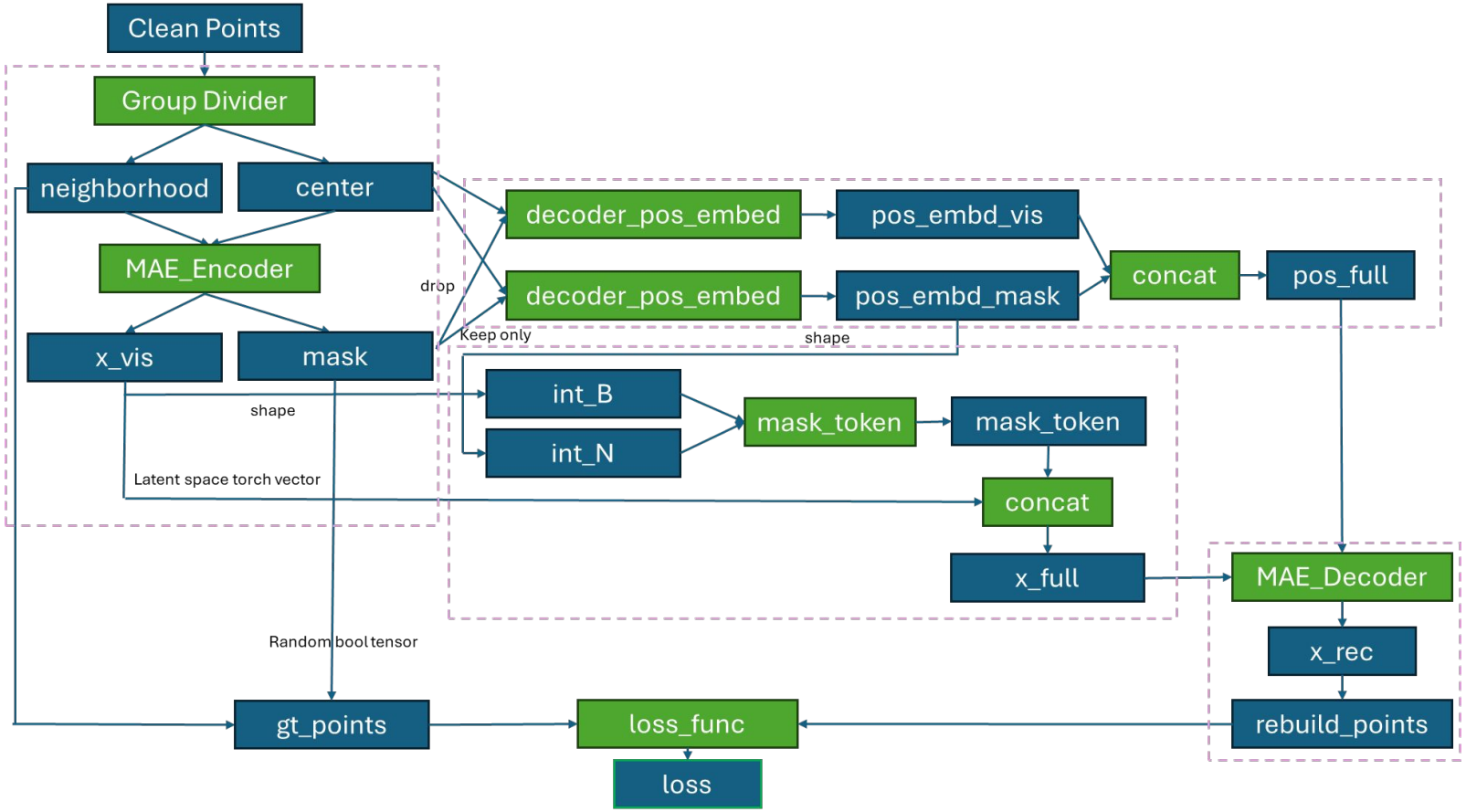
Methods

- We have selected Point-MAE as our skeleton code. Let's visualize the original (Point-MAE) architecture in greater detail:



Methods

- Point clouds are divided into groups using FPS and KNN
- Centers, Tokens fed to encoder
- Random masking in encoder
- The unmasked/masked tokens fed into the decoder with center points



Methods

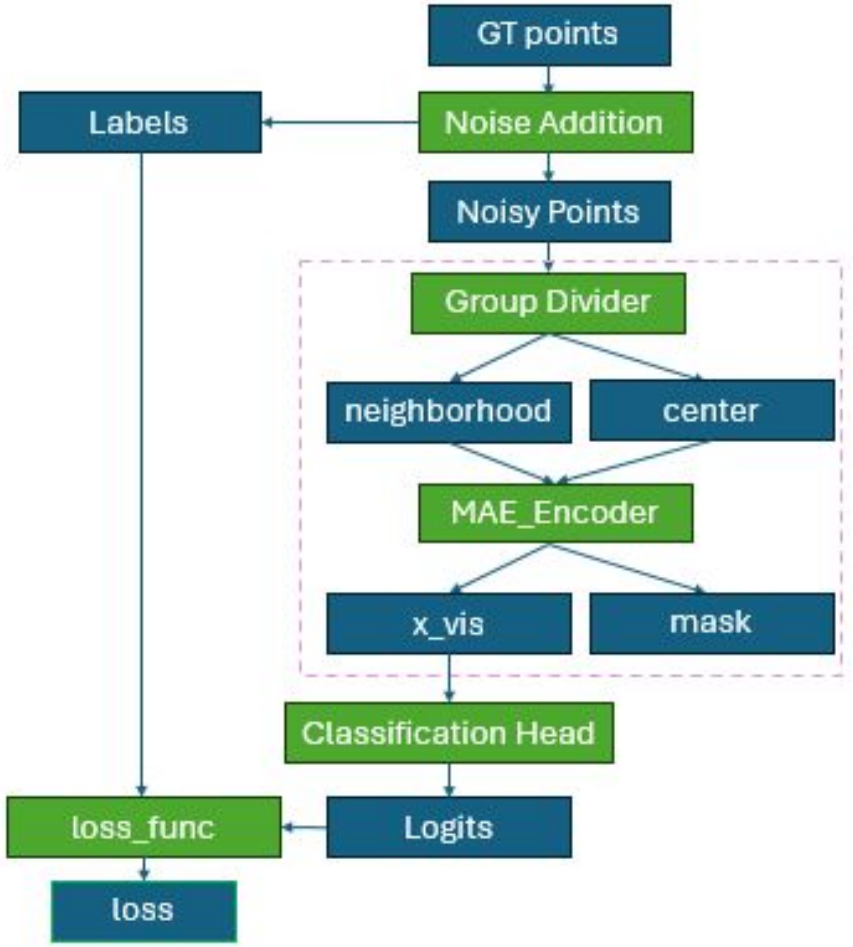
- Various architectures and noise injection methods where explored
- Added noise after tokenization during training — **model learned to denoise and reconstruct.**
- During inference, shape deformation occurred — model misinterpreted noise as structure.
- Found that noise alters token (center point) arrangement, hindering denoising
- **Root cause: tokenization unaffected by noise during training but altered during inference.**

Methods

- Solution: Use ViT architecture approach
- Create grid of fixed center points
- Train encoder-decoder with noise added **in entire unit-sphere** (Pre-training phase)

Methods

- Fine-tuning phase: Use pre-trained encoder to train a 2-layer **classification** network.
- Freeze all encoder layers accept last two.
- Final layers remain trainable to
  allow adaptation for the classification task.



Loss Function

- We train the modified encoder-decoder with Chamfer Distance L2 as the loss function
- CD L2 enables the model to learn point cloud structure, allowing the classification head to distinguish between GT points and noise.

$$\mathrm{CD} = \frac{1}{|\mathbf{S}_1|} \sum_{\mathbf{x} \in \mathbf{S}_1} \min_{\mathbf{y} \in \mathbf{S}_2} \|\mathbf{x} - \mathbf{y}\|_2^2 + \frac{1}{|\mathbf{S}_2|} \sum_{\mathbf{y} \in \mathbf{S}_2} \min_{\mathbf{x} \in \mathbf{S}_1} \|\mathbf{y} - \mathbf{x}\|_2^2$$

- We use BCE (Binary Cross Entropy) as a loss function for training the classification head
- BCE directly supports binary label separation, based on the embedded representations from the trained encoder.

Loss Function

- Chamfer Distance is a measure of the sum of average distances between each point in the original dataset and the nearest neighbor point in the generated dataset.
  (Some variations of chamfer distance use sum of square distances)

$$\text{chamfer}(P_1, P_2) = \frac{1}{2n}\sum_{i=1}^{n}|x_i - \text{NN}(x_i, P_2)| + \frac{1}{2m}\sum_{j=1}^{n}|x_j - \text{NN}(x_j, P_1)|$$

$P_1 = \{x_i \in \mathbb{R}^3\}_{i=1}^{n}$ and $P_2 = \{x_j \in \mathbb{R}^3\}_{j=1}^{m}$ Are the original and generated point clouds

To evaluate denoising, chamfer distance is calculated between the ground truth and the denoised point cloud

Dataset

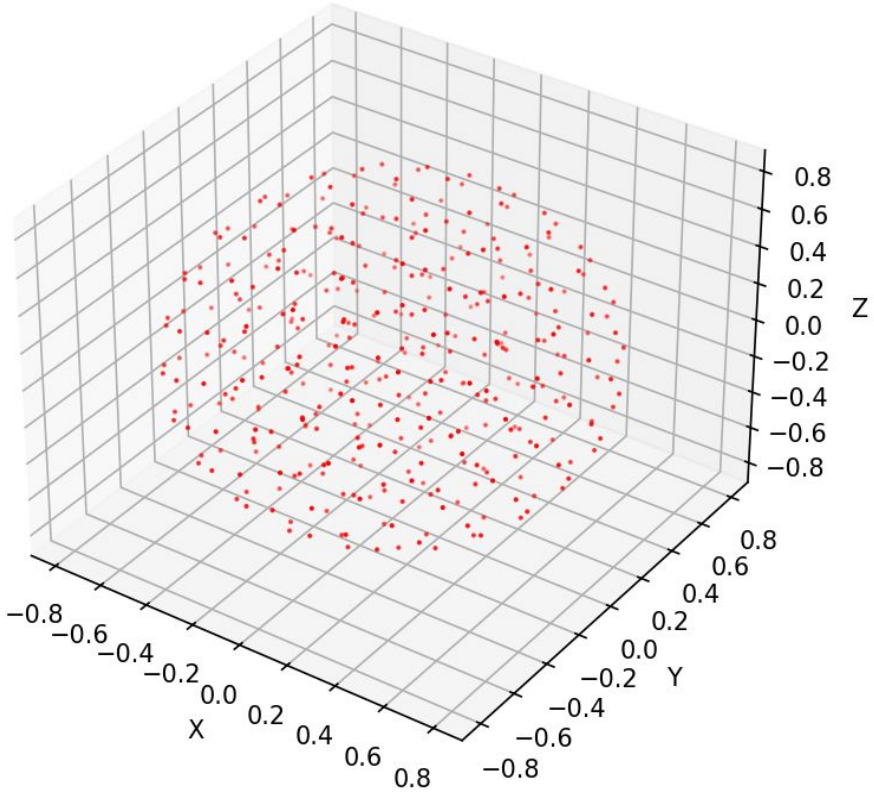
- **Training**: ShapeNet55 – A dataset that covers 55 common object categories with about 51,300 unique 3D models. The 3D objects in this dataset are exact, with no noise and clutter.
- Data is normalized to the unit sphere.
- Gaussian noise is synthetically created and added to the point cloud before being fed into the model.

Dataset

- **Testing**: At inference, synthetic noise will be added to the point cloud in the same way, using two different noise creation strategies, as will be described in the experiments section.
- To compare our model's performance to other work, a dataset closely resembling PU-Net was created.
- Test set combines ShapeNet55 and ModelNet40, which constitute the majority of the data sources used in PU-Net.
- This newly built dataset preserves the key characteristics and diversity of the original, ensuring a fair and meaningful comparison with baseline models.

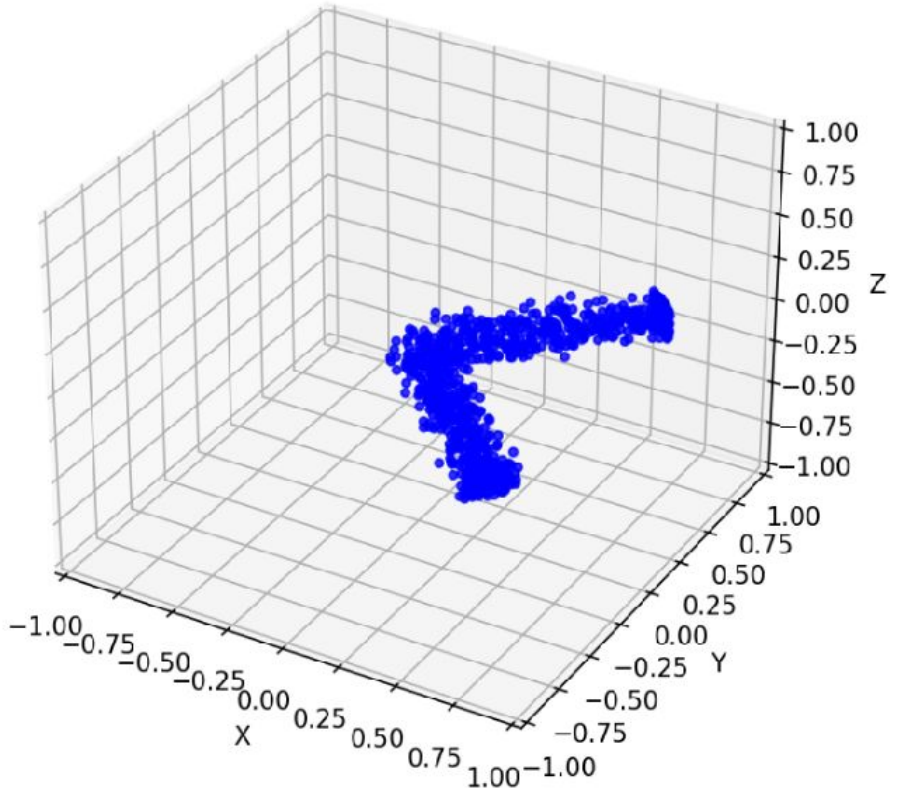
Experiments

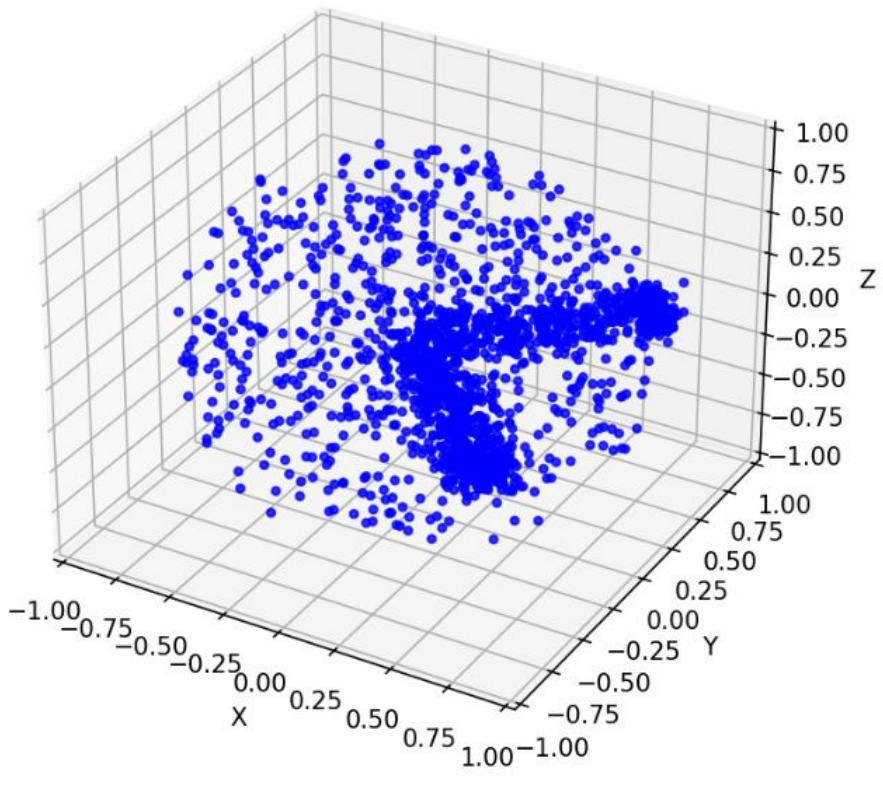
- Create grid of fixed center points.



Experiments

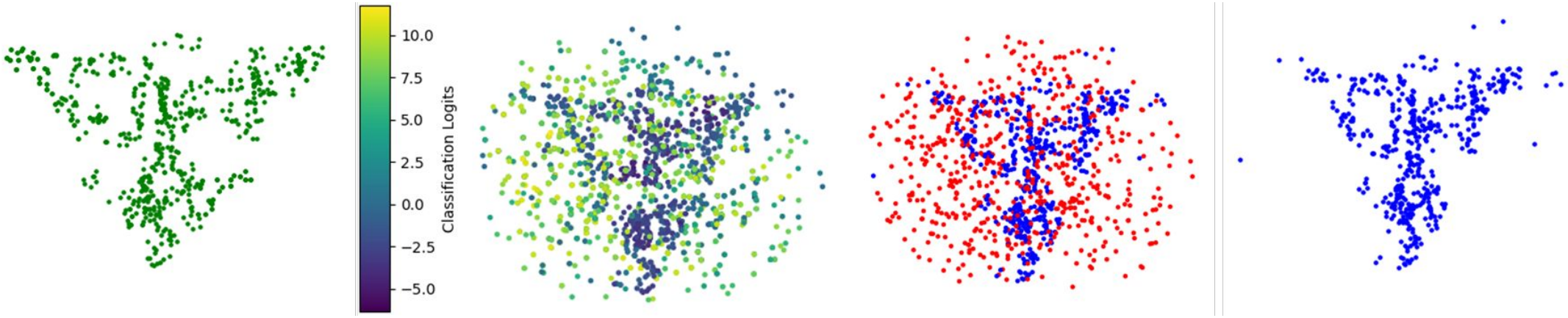
- Create grid of fixed center points.
- Noise addition in pre-training phase.





Experiments

- Testing using same noise addition method as in training on (adding noise at the entire unit sphere)



Experiments

- Comparison with baseline models (perturbed by Gaussian noise with standard deviations from 1% to 3% of the radius of the bounding sphere.
- Evaluation metric: Chamfer Distance (CD) L2

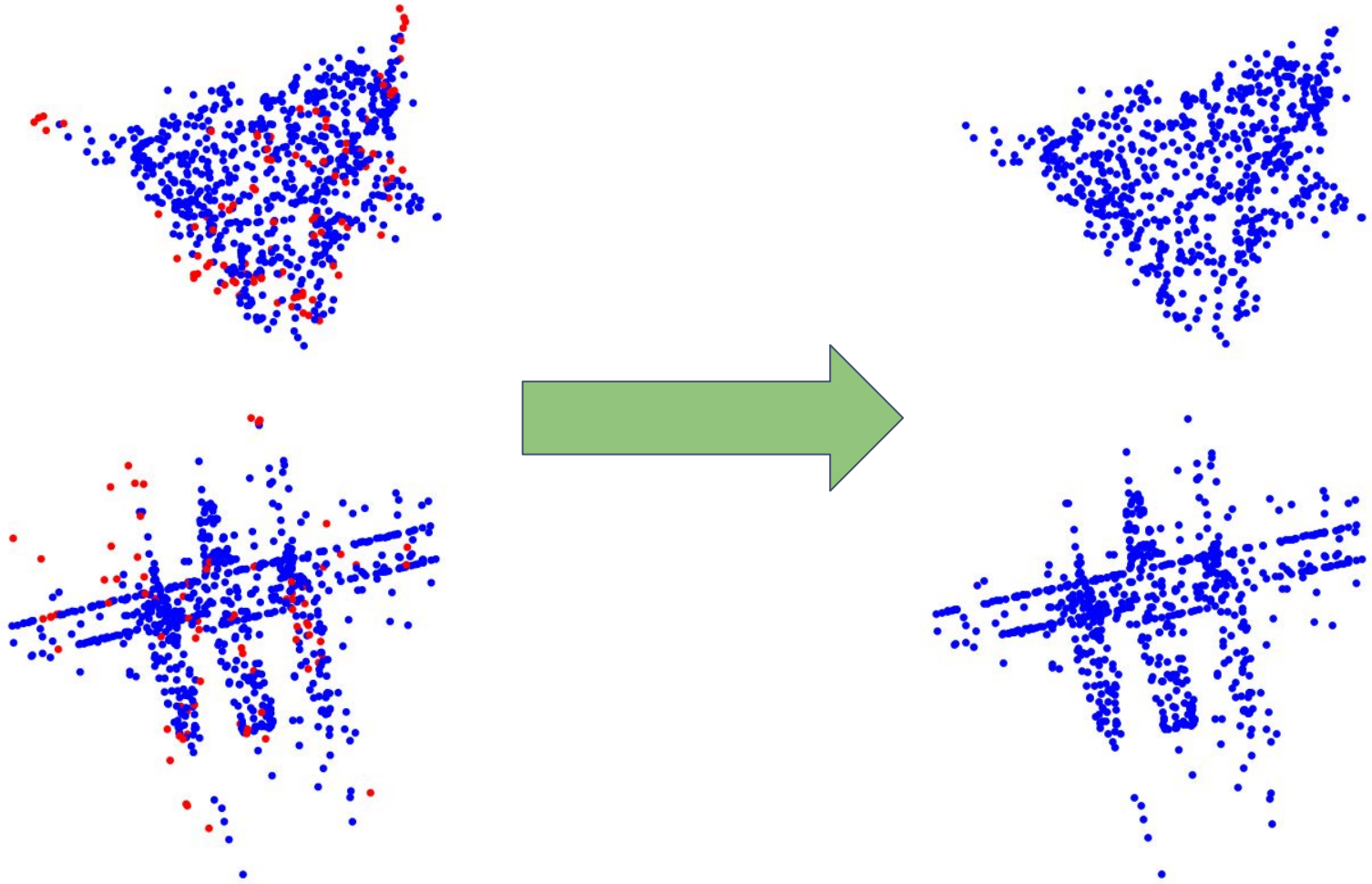
Experiments

Noise Level	**1%**	**2%**	**3%**
Noisy point Cloud	3.544	7.620	12.838
MRPCA	3.016	3.764	5.103
GLR	2.945	3.695	4.872
PCNet	3.426	7.352	12.851
DMR	4.425	4.968	5.885
Score	2.427	3.545	4.795
NoiseTrans	2.288	3.251	4.070
Ours	**0.18**	**0.66**	**0.76**

Experiments

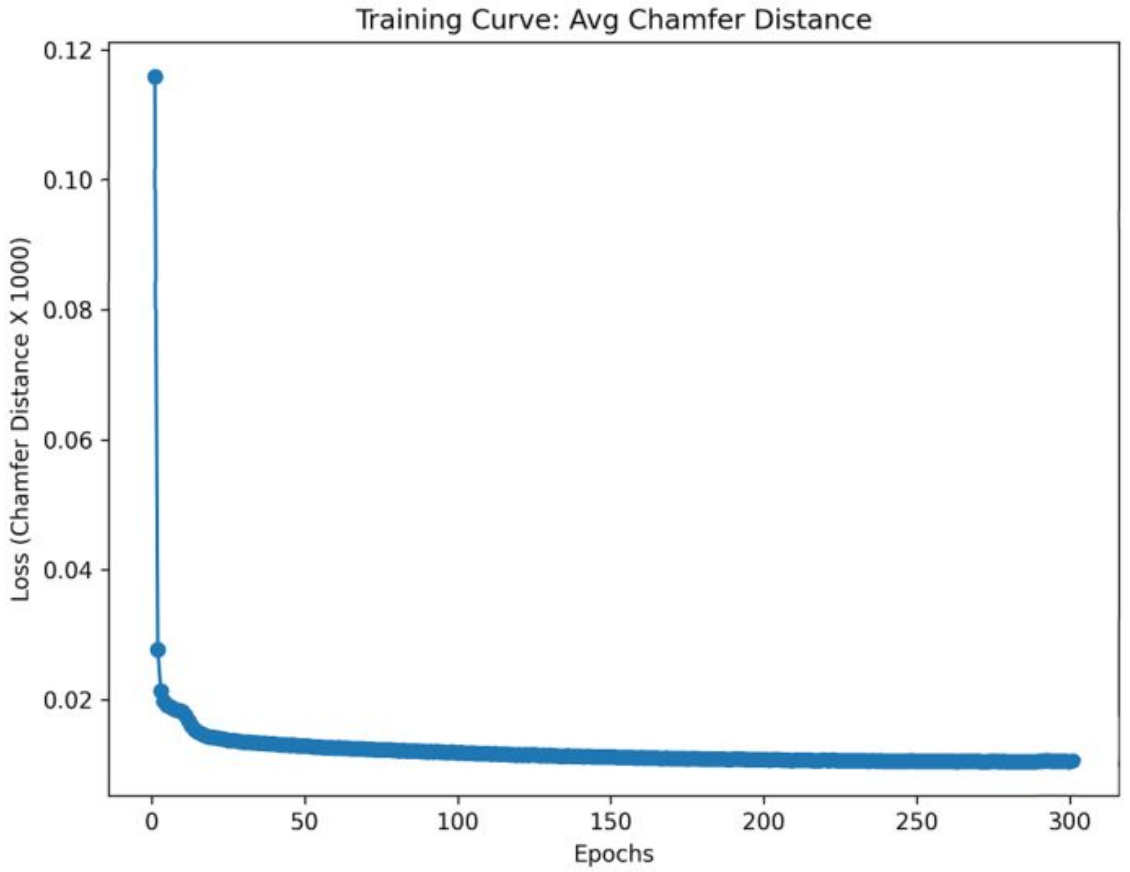
- Our approach:
  - Avoids regenerating already-accurate ground truth (GT) points.
  - Preserves original spatial fidelity of clean data.
  - Prevents distortions caused by full reconstruction methods.
  - Maintains geometric integrity while effectively denoising.
  - Only noisy points are filtered or labeled — GT points remain unchanged.
  - Results in lower Chamfer Distance at inference.

Experiments



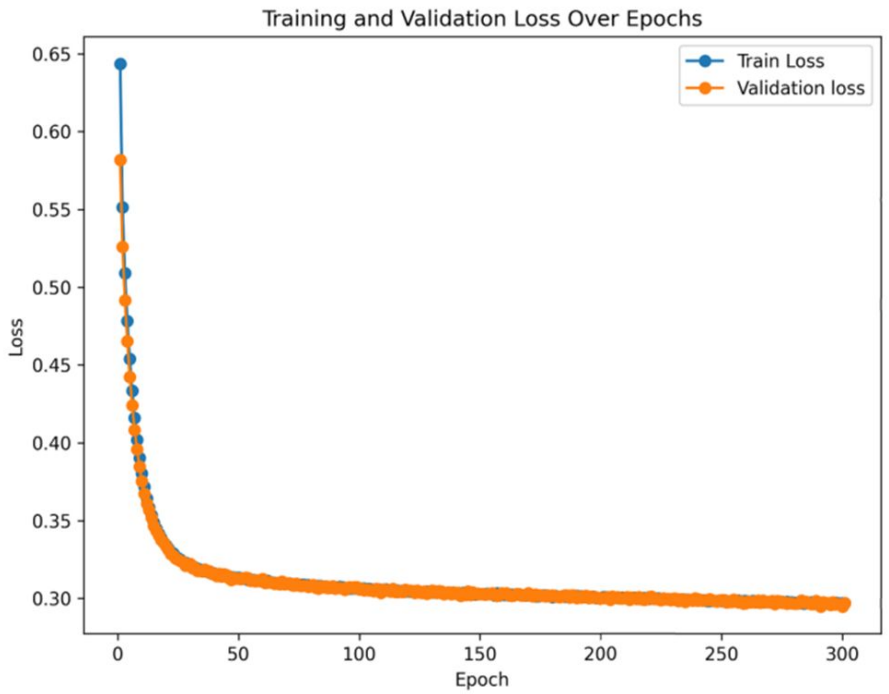
Results

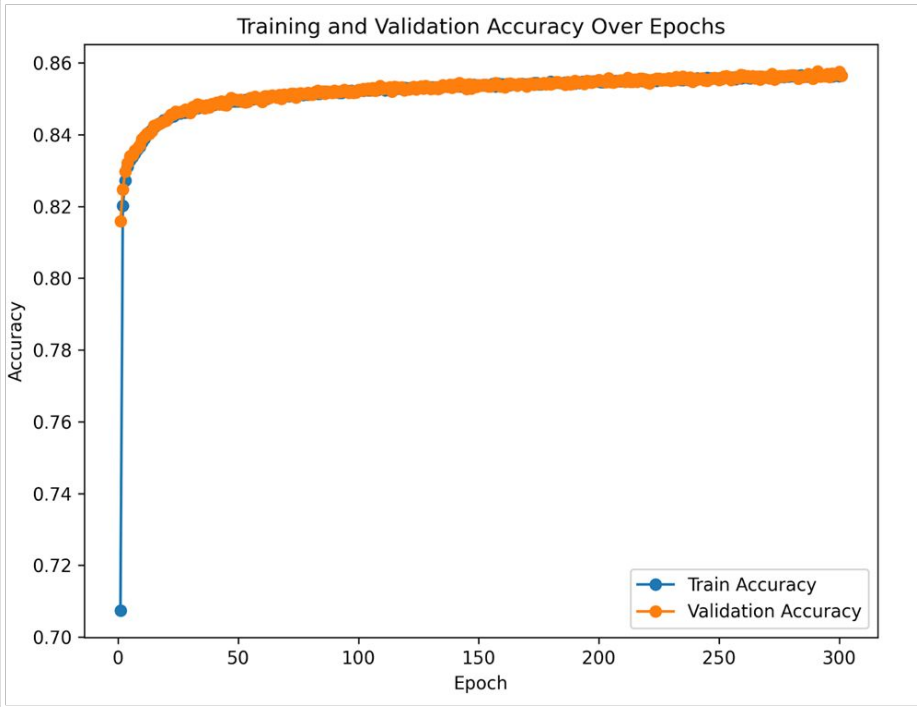
- Pre-training: final average loss value of 0.0106



Results

- Fine-tuning: maximum classification accuracy of 86%





Results

- Pre-training phase is stable and shows convergence to a final average loss value of 0.0106.
- Fine-tuning (classification training) phase is stable as well.
- Maximum classification accuracy achieved during training is 86%
- It can be noticed that the maximum classification accuracy achieved during training is 86%.
- Noise addition method affects the maximum achievable accuracy, as noise is added during training to the entire unit sphere, including inside of the point cloud shape, becoming inseparable to ground-truth points that are next to it.

Discussion

- Introduced a Transformer-based framework for denoising LiDAR point clouds.
- Combined self-supervised pre-training with supervised fine-tuning using a classification head.
- Modified Point-MAE architecture for point-wise noise classification, preserving geometric fidelity.
- Replaced unstable neighborhood sampling with a fixed 3D grid for consistent tokenization.
- Ensured reliable learning of structural features across varying noise levels.

Results and Evaluation

- Performance Highlights:
  - Achieved stable and generalizable denoising across noise intensities.
  - Outperformed reconstruction-based methods (e.g., NoiseTrans) in Chamfer Distance and visual quality.
  - Visualization confirms effective noise removal without distorting ground truth (GT) points.
- Limitations:
  - Some false negatives observed—noisy points misclassified as structure.
  - Boundary sensitivity remains a challenge near ambiguous regions.

Conclusion

- Proposed a label-efficient, scalable denoising approach for 3D point clouds.
- Demonstrated strong alignment between training and validation metrics.
- Preserved original spatial fidelity of clean ground-truth, making the method suitable for real-world tasks like segmentation and object detection.

Future Work

- Improved Classification: Explore deeper or attention-based heads; add uncertainty modeling.
- Better Loss Functions: Combine spatial and semantic cues for finer discrimination.
- Robustness: Train on diverse, unlabeled real-world data for stronger generalization.
- Task Integration: Connect denoising directly with segmentation and detection pipelines.