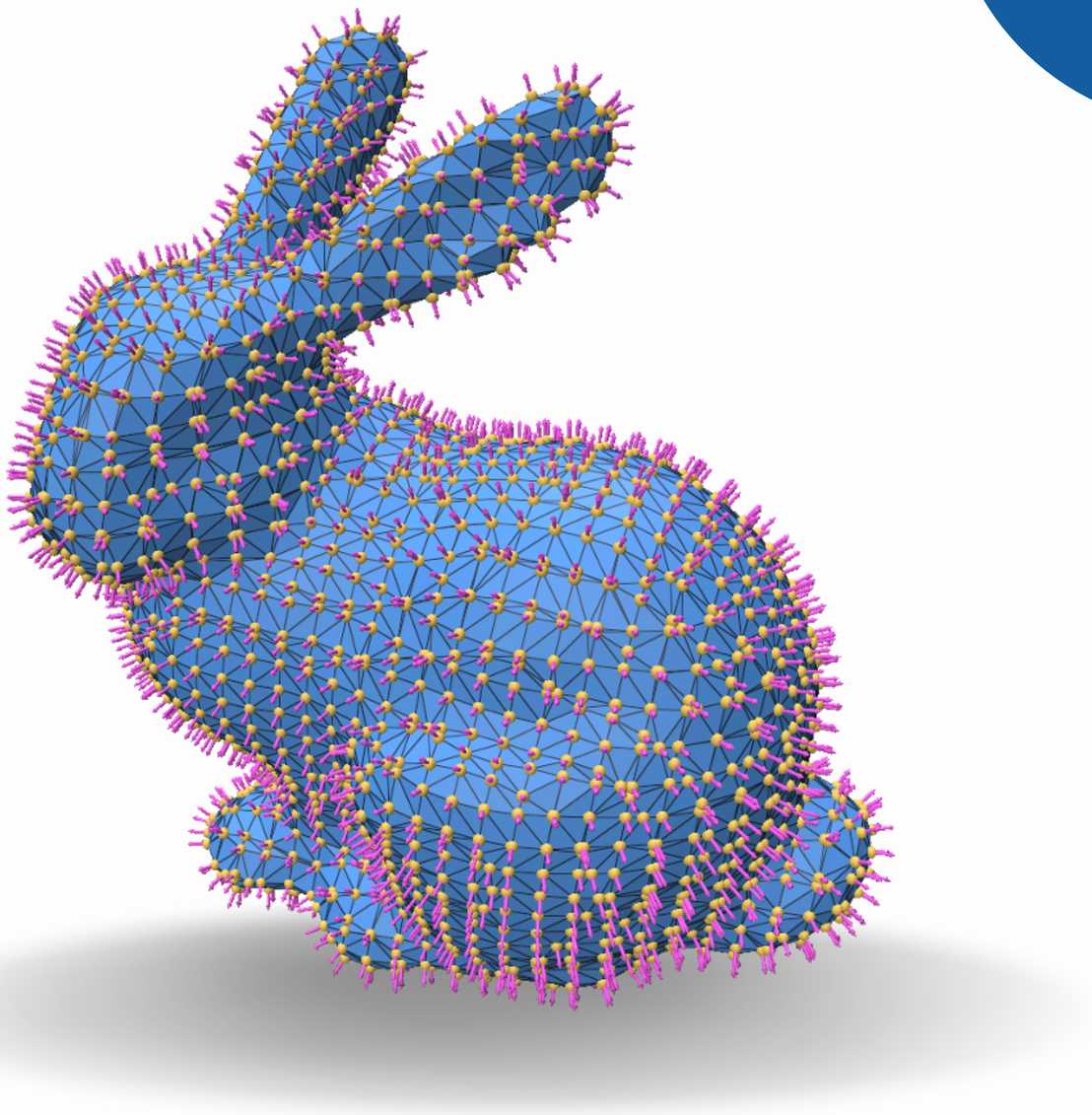
3D Point Cloud Classification with Graph Neural Networks

By: Ahmed Assy, Mahmoud Abdellahi

Supervised By: Martin Ritzert

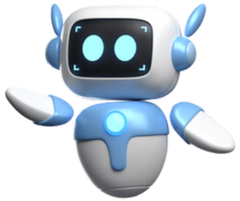


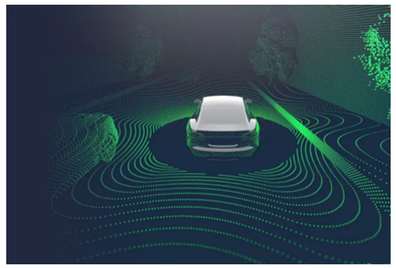
Overview

Introduction

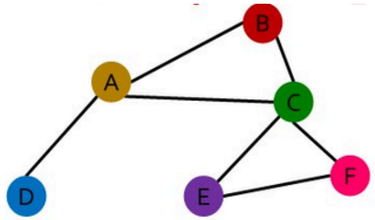
- **3D Point Cloud**

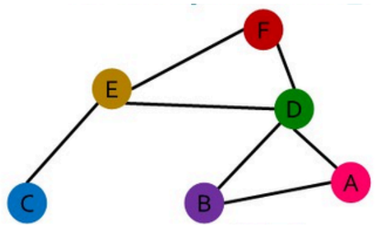
- **Applications**





- **Why Graph Methods**

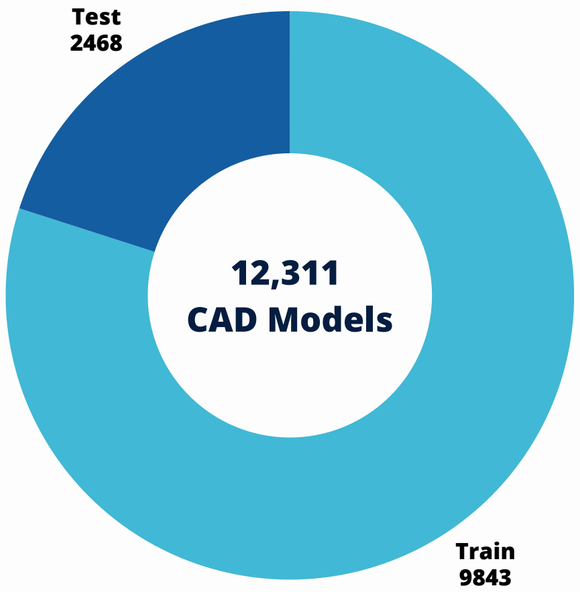




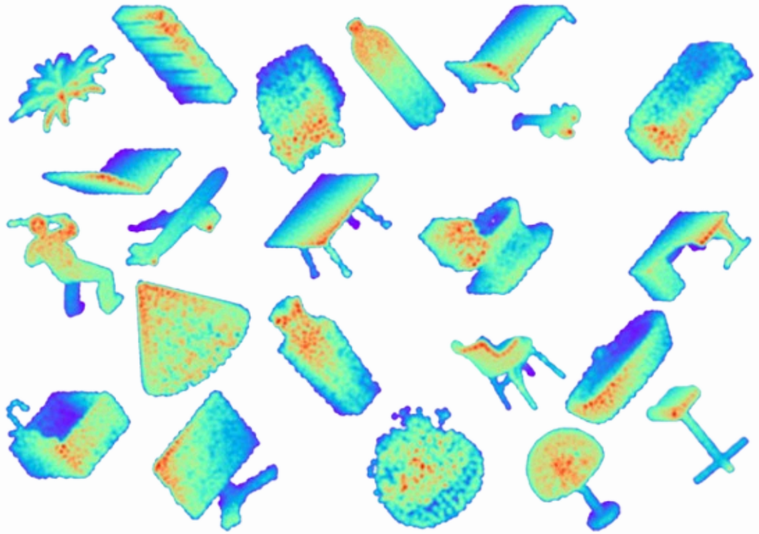
Dataset

ModelNet40

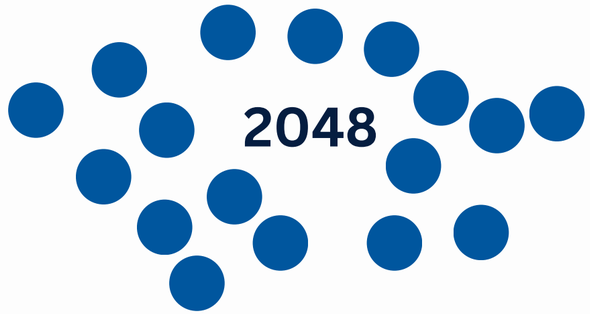
- Widely used 3D object dataset of 40 object categories, including chairs, tables, cars, etc.
- Created by Princeton University researchers.
- Benchmarking 3D deep learning models for tasks like classification, segmentation, and retrieval.

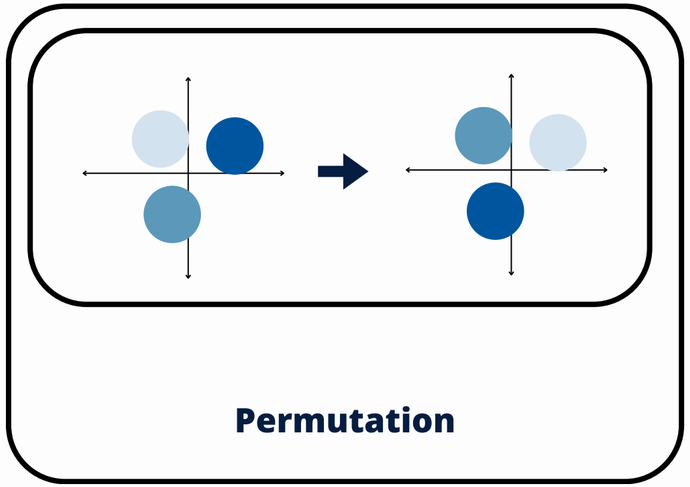


Class Name	Total	Class Name	Total	Class Name	Total	Class Name	Total
airplane	726	cup	99	laptop	169	sofa	780
bathtub	156	curtain	158	mantel	384	stairs	144
bed	615	desk	286	monitor	565	stool	110
bench	193	door	129	night_stand	286	table	492
bookshelf	672	dresser	286	person	108	tent	183
bottle	435	Plower_pot	169	piano	331	toilet	444
bowl	84	Glass_box	271	plant	340	tv_stand	367
car	297	guitar	255	radio	124	vase	575
chair	989	keyboard	165	range_hood	215	wardrobe	107
cone	187	lamp	144	sink	148	xbox	123

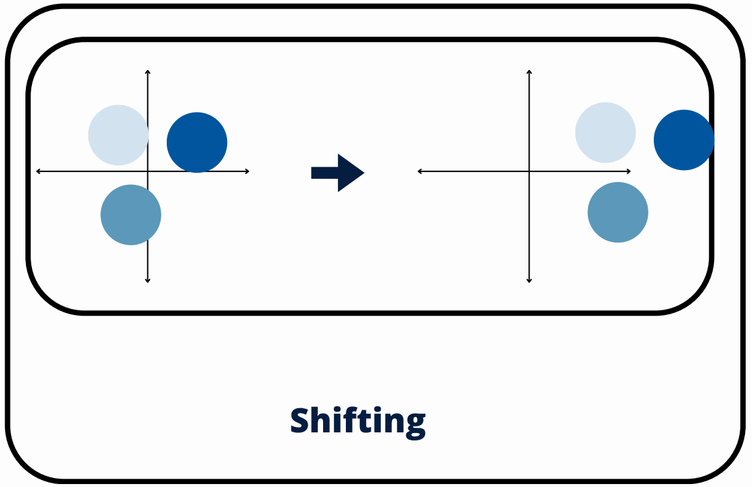


Data Processing

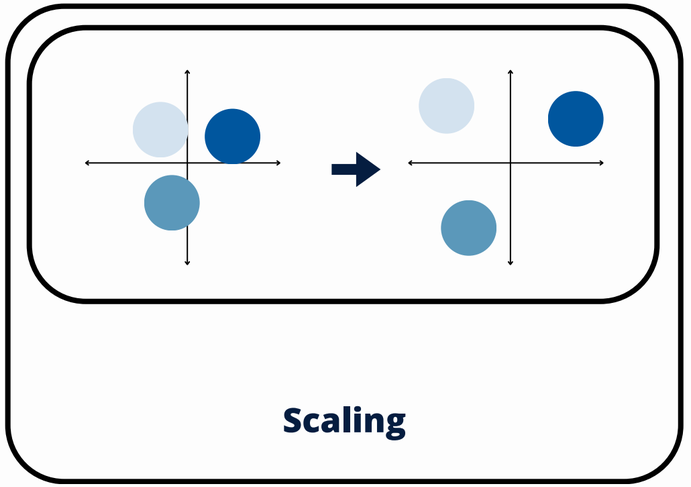




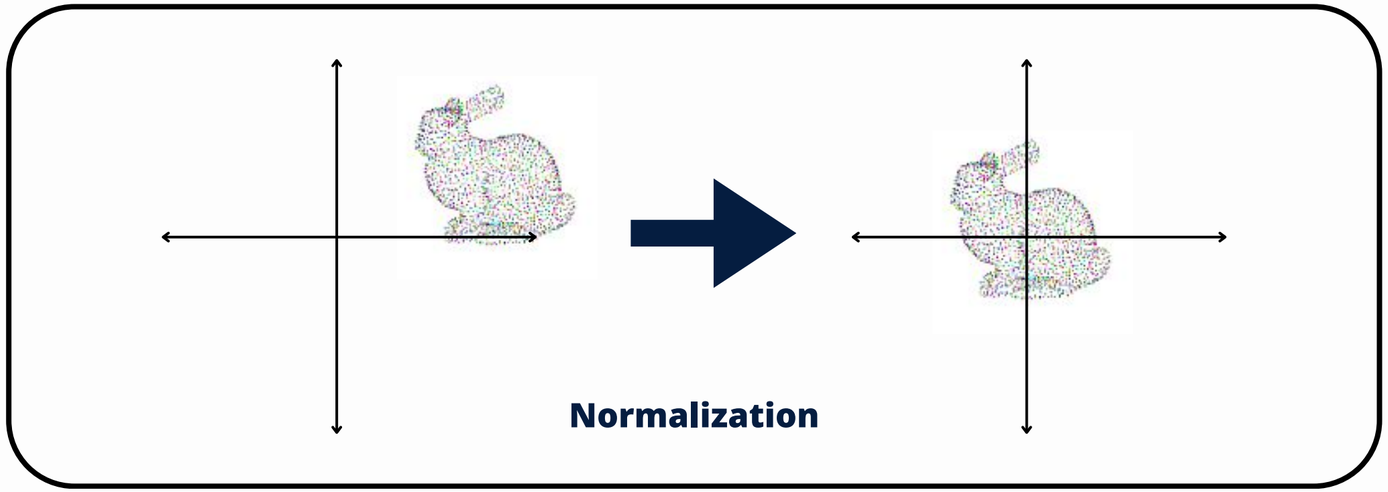
Permutation



Shifting



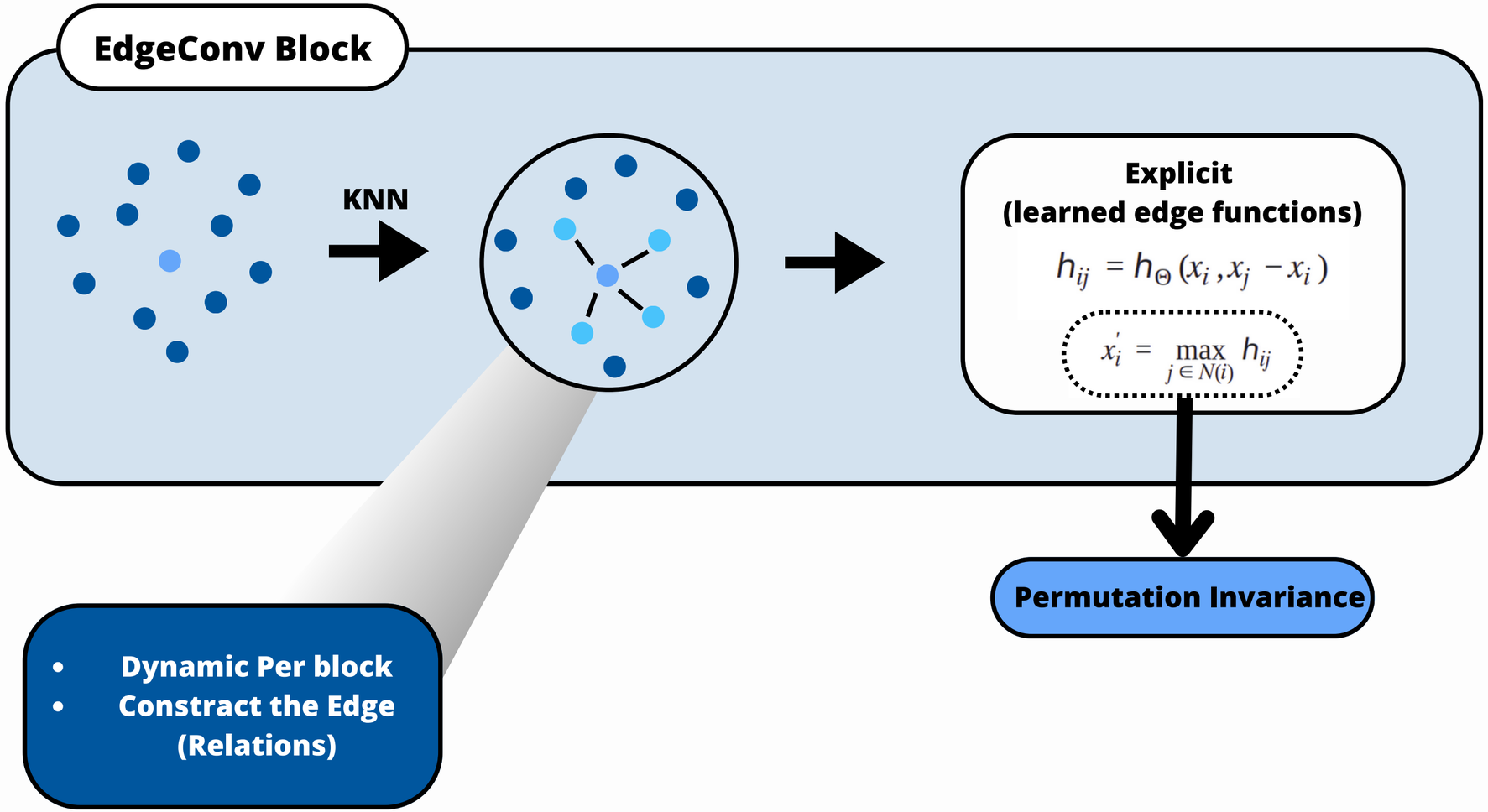
Scaling



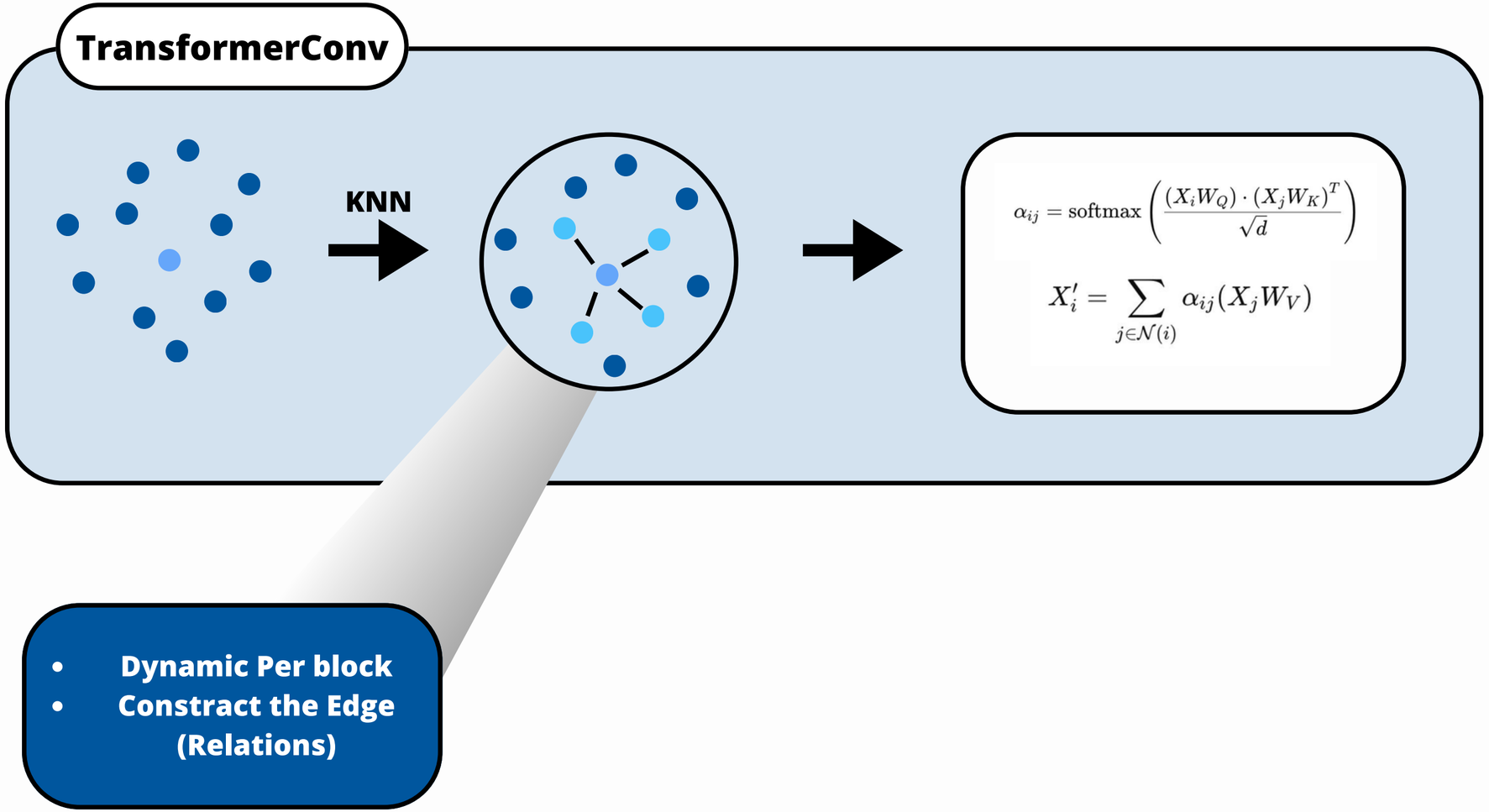
Normalization

Methodology

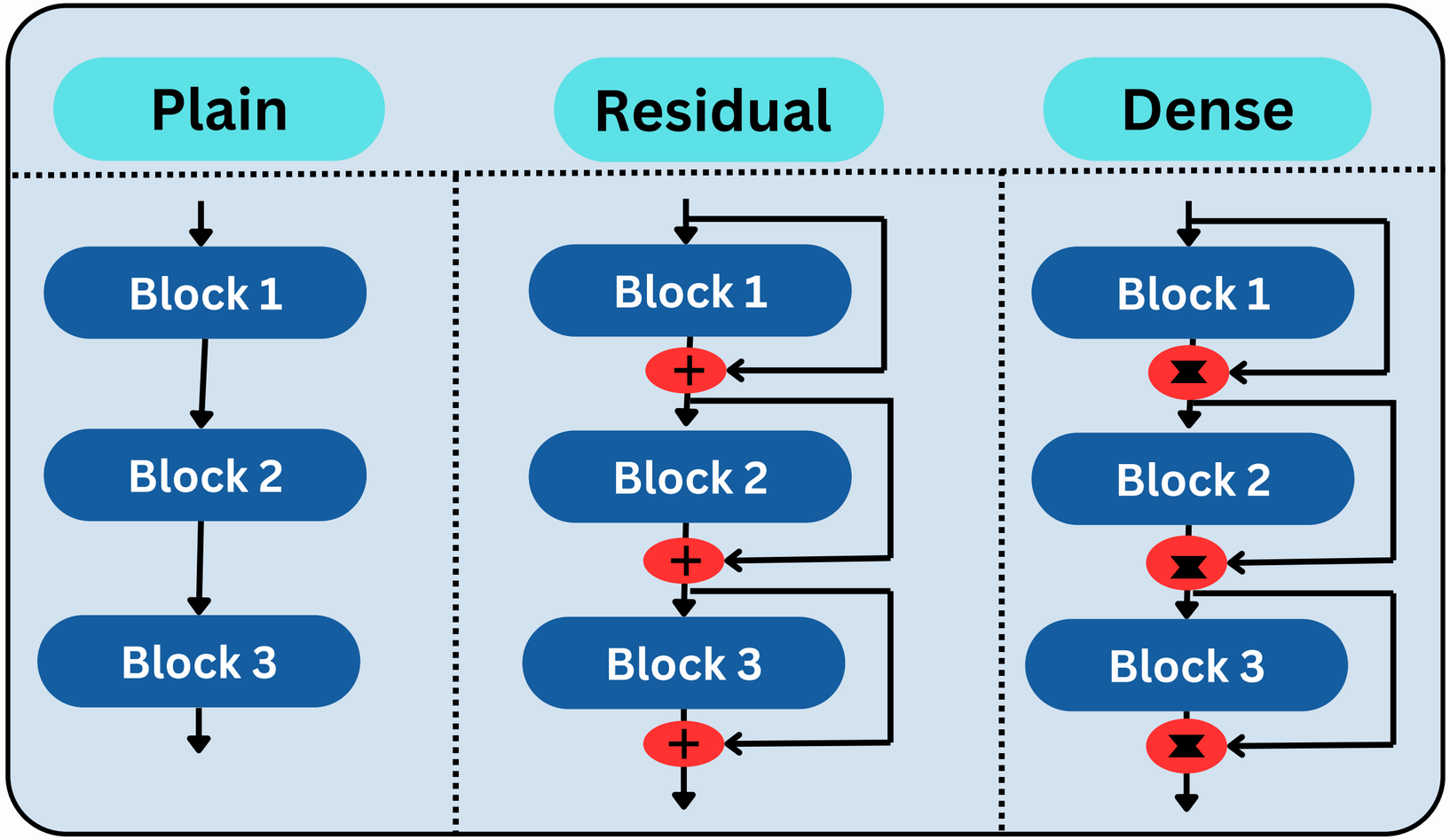
EdgeConv


EdgeConv Block
KNN
Explicit
(learned edge functions)
$h_{ij} = h_\Theta(x_i, x_j - x_i)$
$x_i' = \max_{j \in N(i)} h_{ij}$
Permutation Invariance
Dynamic Per block
Constract the Edge
(Relations)

TransformerConv



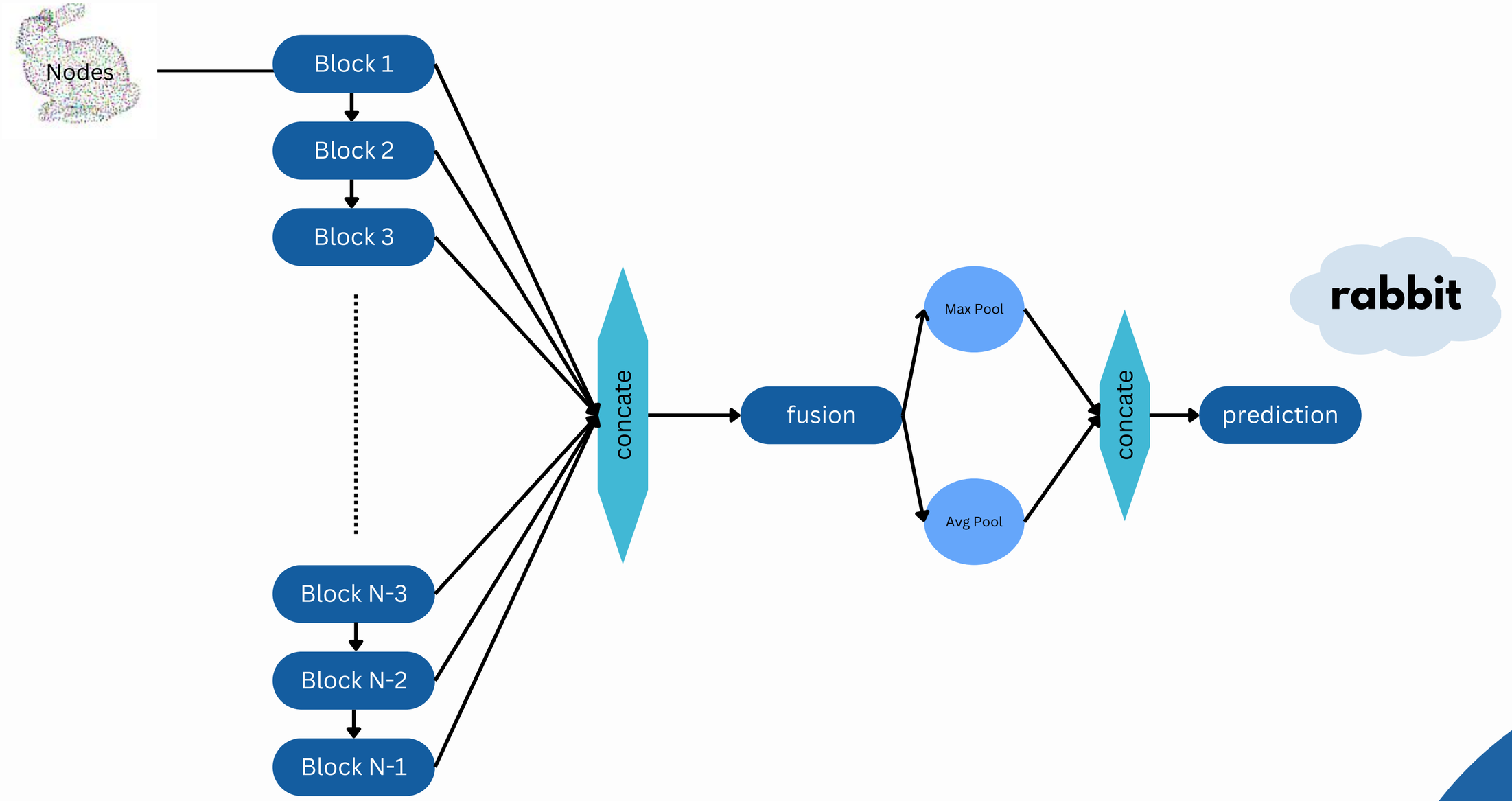
Blocks Connections



$\mathbf{X}' = \mathbf{X} + F(\mathbf{X})$

$\mathbf{X}' = \text{concat}(\mathbf{X}_0, \mathbf{X}_1, ..., \mathbf{X}_l)$

Model Architecture



Training Parameters

	Experiment 1	Experiment 2
Block	EdgeConv	TransformerConv
Connections	residual blocks	dense blocks
Number of blocks	14	7
Epochs	400	200
Learning Rate	1e-3	1e-3
GPU	4 V100	4 V100
Batch Size	32	32

- **Optimizer:** Adamw
- **Learning rate scheduler:** CosineAnnealingLR

Results

Results

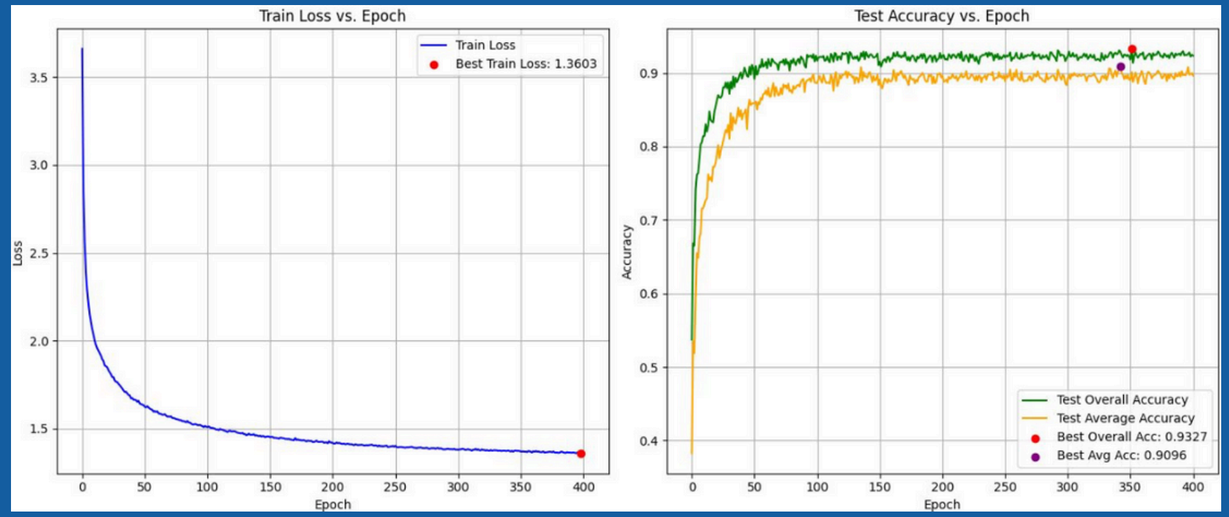
EdgeConv Result

- **Test Time**: 7.55 ms per point cloud.
- **Number of Parameters**: 2.2M
- **Test Overall Accuracy**: 93.27%
- **Test Average Class Accuracy**: 90.09%

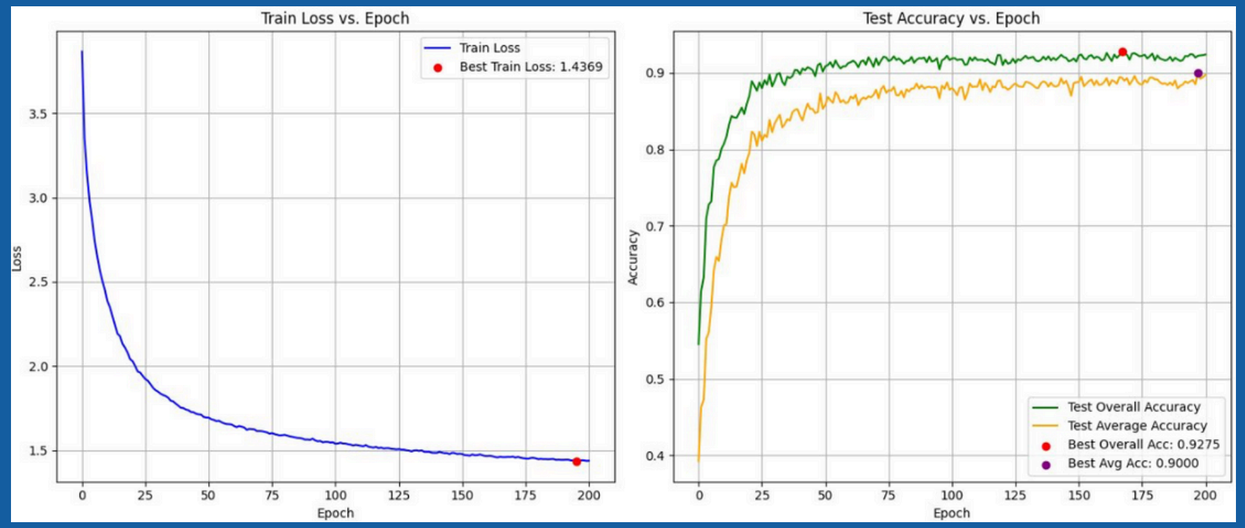
TransformerConv Result

- **Test Time**: 8.34 ms per point cloud.
- **Number of Parameters**: 15.8M
- **Test Overall Accuracy**: 92.75%
- **Test Average Class Accuracy**: 89.36%

EdgeConv

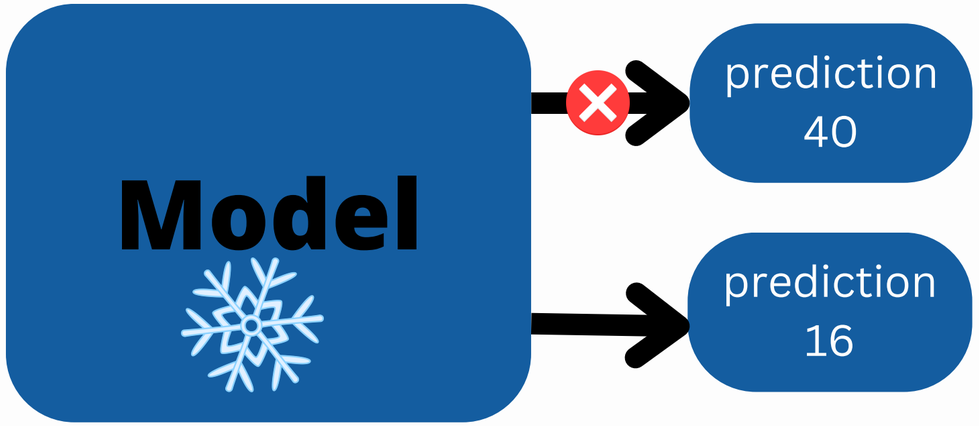


TransformerConv



ShapeNetPart

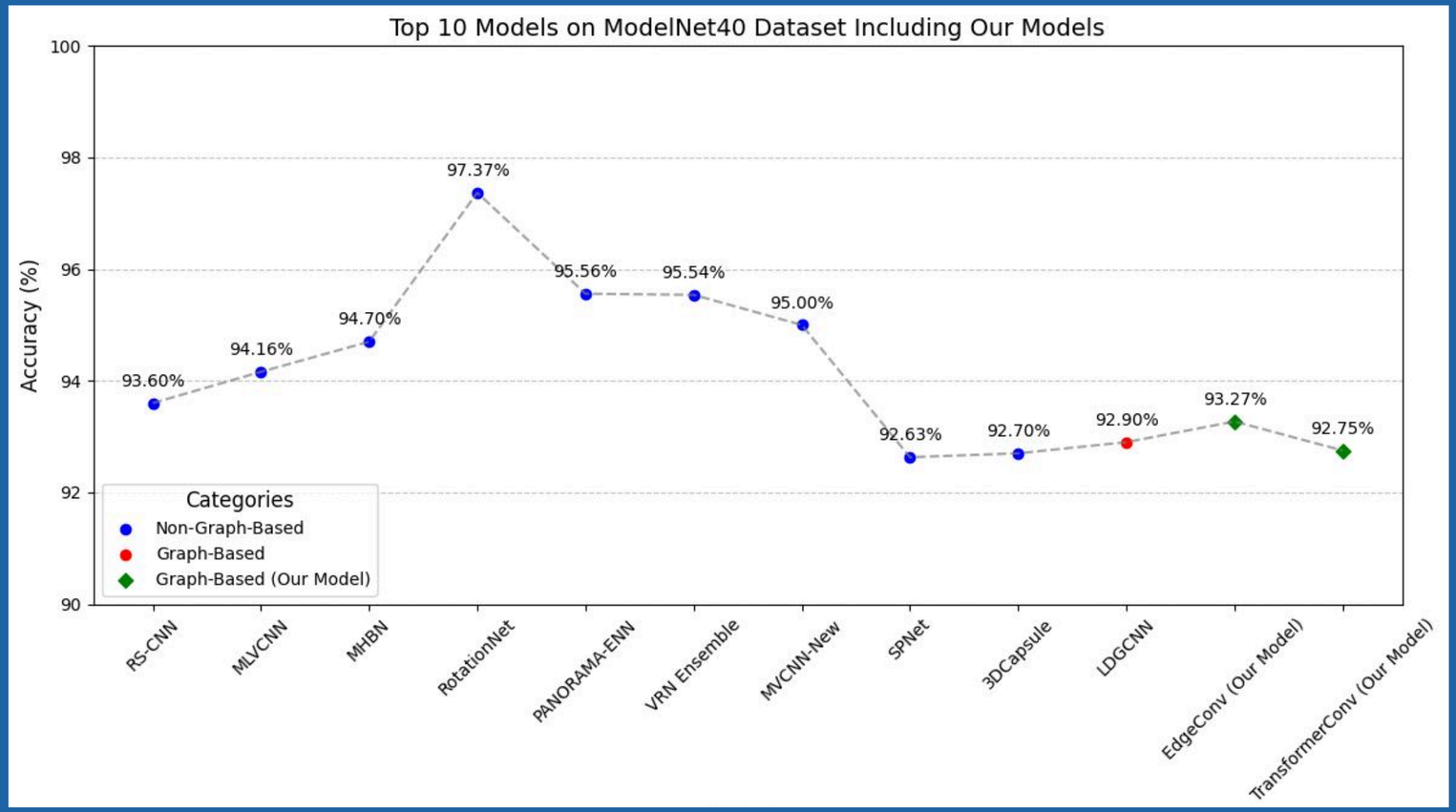
- **16 Classes**: The dataset contains part-level annotations for 16 object categories.
- **Total Samples**: 16,881 3D models, each with part segmentation labels.
- **Part-Level Segmentation**: Each model is annotated with parts, where each part has a label indicating the specific part (e.g., wing, seat, handle, etc.).



97.6% accuracy in 8 epochs with stabilized loss.

Class Name	Count	Class Name	Count	Class Name	Count
Airplane	1,225	**Chair**	3,084	**Bag**	1,413
Earphone	1,269	**Guitar**	1,070	**Cap**	1,155
Car	2,743	**Lamp**	1,383	**Laptop**	1,028
Motorbike	1,012	**Mug**	1,180	**Pistol**	1,036
Rocket	1,037	**Skateboard**	1,048	**Table**	1,017
TV	1,039				

SOTA Comparison on ModelNet40



Conclusion

Conclusion

01 We achieved good results in 3D point cloud classification using Graph Neural Networks.

02 Our model showed strong performance, particularly on ModelNet40, achieving competitive accuracy.

03 Further improvements can be made by utilizing additional datasets like ShapeNet, which provides more diverse object categories and fine-grained part segmentation.

04 Future work could explore alternative graph architectures and data augmentation techniques to enhance robustness and generalization.

THANK YOU!

Any Questions