POINT TRANSFORMER V3: *SIMPLER FASTER STRONGER*

CHENG CHEN & ALEC DIGBY

EECE 571F

CONTENTS

PAPER OVERVIEW

- Publication Date: March 25, 2024
- Authors: Xiaoyang Wu[1,2], Li Jiang[3], Peng-Shuai Wang[4], Zhijian Liu[5], Xihui Liu[1], Yu Qiao[2], Wanli Ouyang[2], Tong He[2*], Hengshuang Zhao[1*]
- Affiliate Organizations: [1]HKU [2]SH AI Lab [3]CUHK(SZ) [4]PKU [5]MIT
- Key Takeaway: *By improving computational efficiency, point transformers can leverage the power of scaling. Scaling is more important than intricate design for model performance.*

PAPER OVERVIEW

- Progress is made here through an **engineering approach** rather than a theoretical one [1].
- Approximations are good if they unlock scaling through efficiency.
- Specific Innovations:
    - Replaces precise neighbor search (KNN) with serialized mapping.
    - Replaces complex attention patch interactions (ex. shift window) with a streamlined approach.
    - Eliminates reliance on relative positional encoding by using a sparse convolutional layer.
    - *All the key contributions of this paper are related to **computational efficiency**.*
- Improvement Over PTv2 [2]:
    - 3.3x inference speed[1]:
    - 10.2x reduction in memory usage [1].
    - Receptive field increased from 16 to 1024 points [1].
- Results: Achieves **state of the art accuracy** on benchmark datasets.

WHAT'S THE POINT?

- Perform tasks on 3d point data.
  - Classification
  - Part Segmentation
  - Semantic Segmentation (most useful)
- Applications:
  - Autonomous driving
  - Augmented reality
  - Robotics [3]

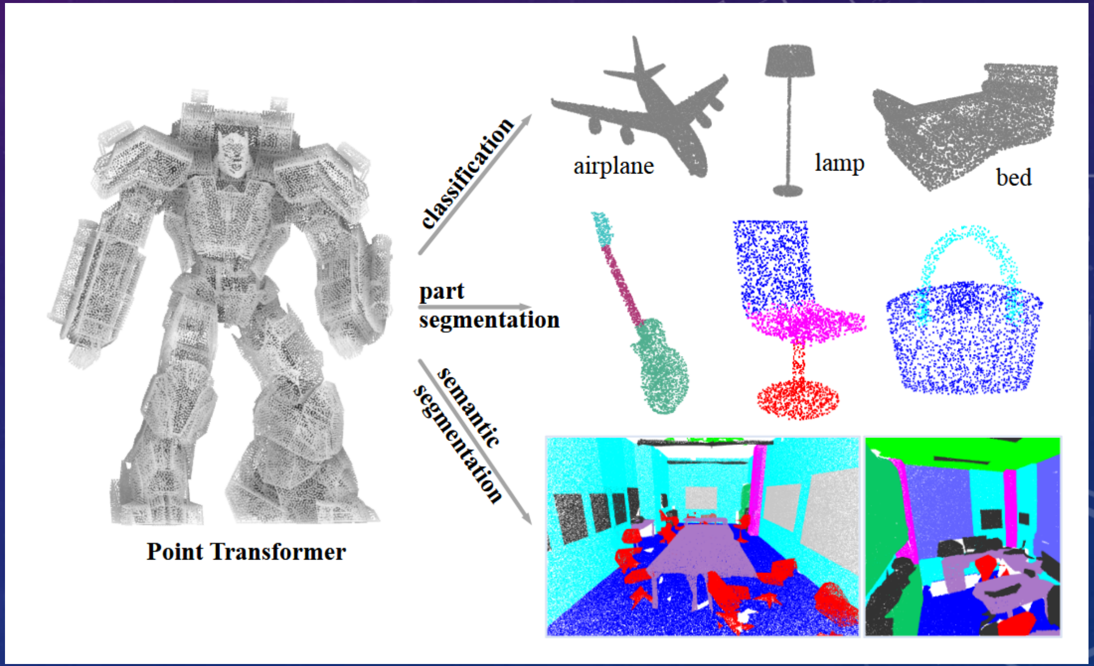


Image Credit: [3]

PREVIOUS METHODS BEFORE TRANSFORMERS

- Deep Sets [4]
- PointNet++ [5]
- Graph Methods [6]
- PointCNN [7]

POINT TRANSFORMERS

- Introduced in Point transformer v1 [3].
- Uses attention to aggregate neighborhoods (instead of fixed pooling). More expressive.
- Relaxes invariance internally while keeping it globally. (Previous methods such as PointNet++ [5] require invariance at every layer, transformers get around this.)
- Improved in Point Transformer v2 [2].

SCALING PRINCIPLE

- Scaling law for LLMS: “The loss scales as a power-law with model size, dataset size, and the amount of compute used for training” [8].
- Generally, **bigger is better**.
- In the case of point transformers, a larger **receptive field** provides the greatest performance gains, rather than parameter count, dataset size etc.
- Scaling is enabled through increased **computational efficiency.**

RECEPTIVE FIELD

- The region of the input that influences a point’s feature.
- Analogous to the context length of an LLM.
- Larger receptive field = better generalization capabilities [1].
- “PTv3 capitalizes on its inherent ability to scale the range of perception, expanding its receptive field from **16** to **1024** points while maintaining efficiency.” [1].

RECEPTIVE FIELD – CONVOLUTION ANALOGY

- In an image CNN, each layer expands the receptive field, moving from local features such as edges to larger features such as object parts.
- Point cloud transformers similarly identify local features within the receptive window and move to global features at later layers.
- Expanding the receptive window essentially allows us to jump to larger and more complex features, increasing the expressiveness of the model.

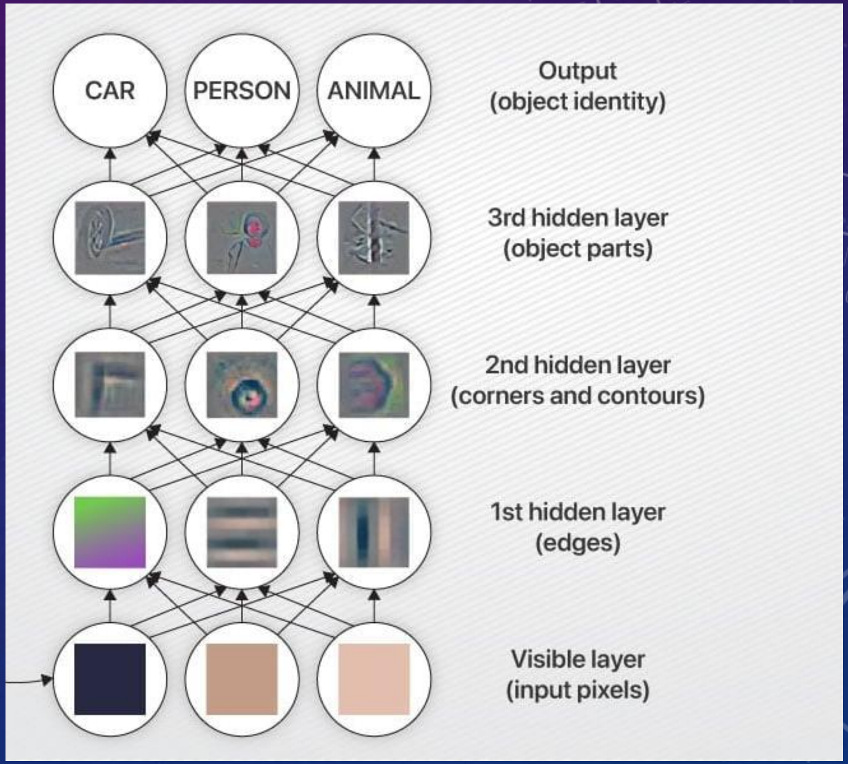

CAR
PERSON
ANIMAL
Output (object identity)
3rd hidden layer (object parts)
2nd hidden layer (corners and contours)
1st hidden layer (edges)
Visible layer (input pixels)

Image Credit: [9]

SPECIFIC OPTIMIZATIONS TO UNLOCK SCALING

- Removed relative positional encoding.
- Removed shift window attention.
- Replaced grouping algorithms with **point cloud serialization**. [1]

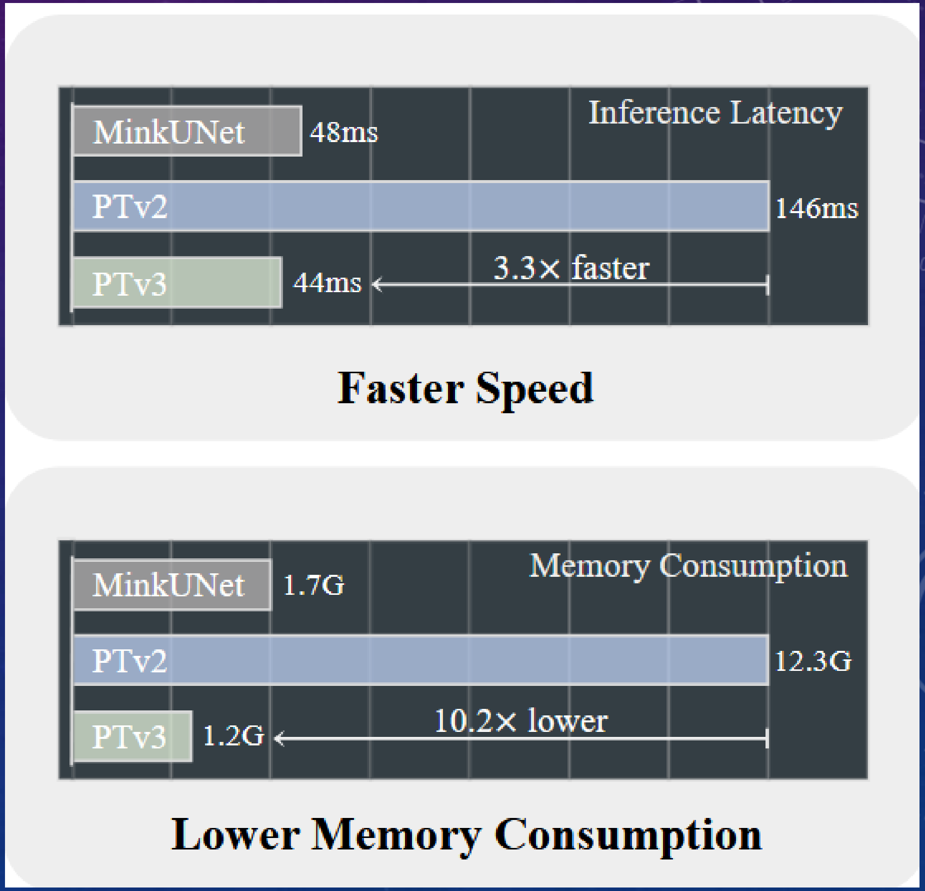


Image Credit: [1]

OPTIMIZATION: REMOVED RELATIVE POSITIONAL ENCODING

- Relative positional encoding ≈ large-kernel **Sparse Convolution**.
- Sparse convolution: Gives the network information about local geometry more efficiently.

- Improvements:
  - replaces Relative Positional Encoding with a prepositive sparse conv layer + skip connection.
  - Relative Positional Encoding (RPE) -> Enhanced Conditional Positional Encoding (xCPE)

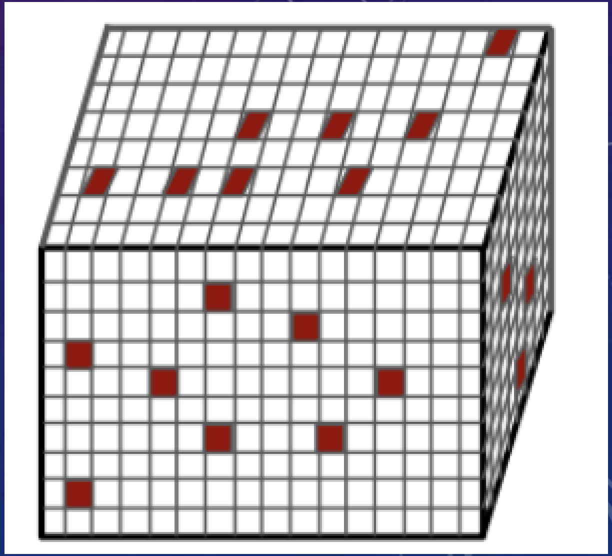


Image Credit: [10]

OPTIMIZATION: REMOVED SHIFT WINDOW ATTENTION

- “PTv3 replaces more complex attention patch interaction mechanisms, like shift-window (impeding the fusion of attention operators) and the neighborhood mechanism (causing high memory consumption), with a streamlined approach tailored for serialized point clouds.” [1]
- Shift Window:
  - Partition input into local windows to compute attention within each window.
  - Shift the windows so that points in different windows can interact.
- Cons:
  - Unlike images, point clouds are sparse and non-uniform, so partitioning it into a grid makes less sense.
  - “Impedes the fusion of attention operators”, meaning that larger context and features are broken up by windowing, requires multiple layers to propagate information.
- Improvements:
  - Point cloud serialization naturally does not create these window boundaries and artifacts.

OPTIMIZATION: MORE EFFICIENT GROUPING ALGORITHMS

- Computationally hard to compute attention across the entire point cloud, better to define a local group of neighbors.
- "PTv3 shifts from the traditional spatial proximity defined by K-Nearest Neighbors (KNN) query, accounting for 28% of the forward time. Instead, it explores the potential of serialized neighborhoods in point clouds, organized according to specific patterns." [1]
- K-Nearest Neighbors (KNN). $O(N^2)$.
- Radius Search, not a fixed number of neighbors. Also $O(N^2)$.
- KNN Responsible for 28% of forward time in PTV2. [1][2]

POINT CLOUD SERIALIZATION

- **Space-filling Curves**:
    - Use a 3D traversal pattern to serialize points into a sequence.
- Approximate neighborhoods are the points within a certain window of the resulting sequence.
- "The ordering effectively rearranges the points in a manner that respects the spatial ordering defined by the given space filling curve, which means that **neighbor points in the data structure are also likely to be close in space**." [1]

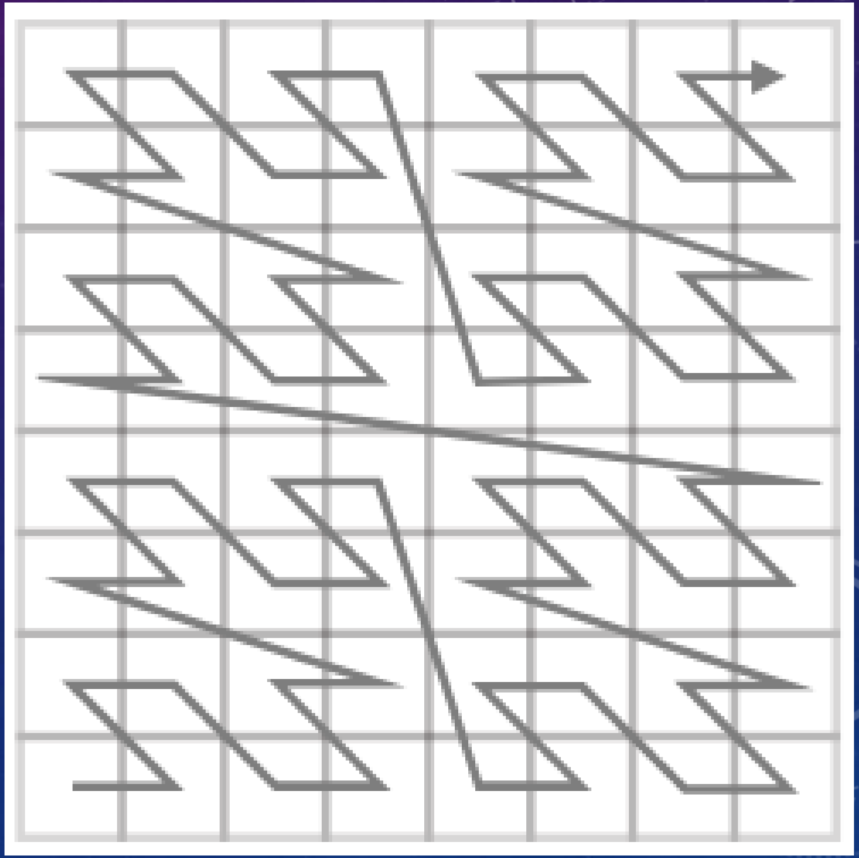
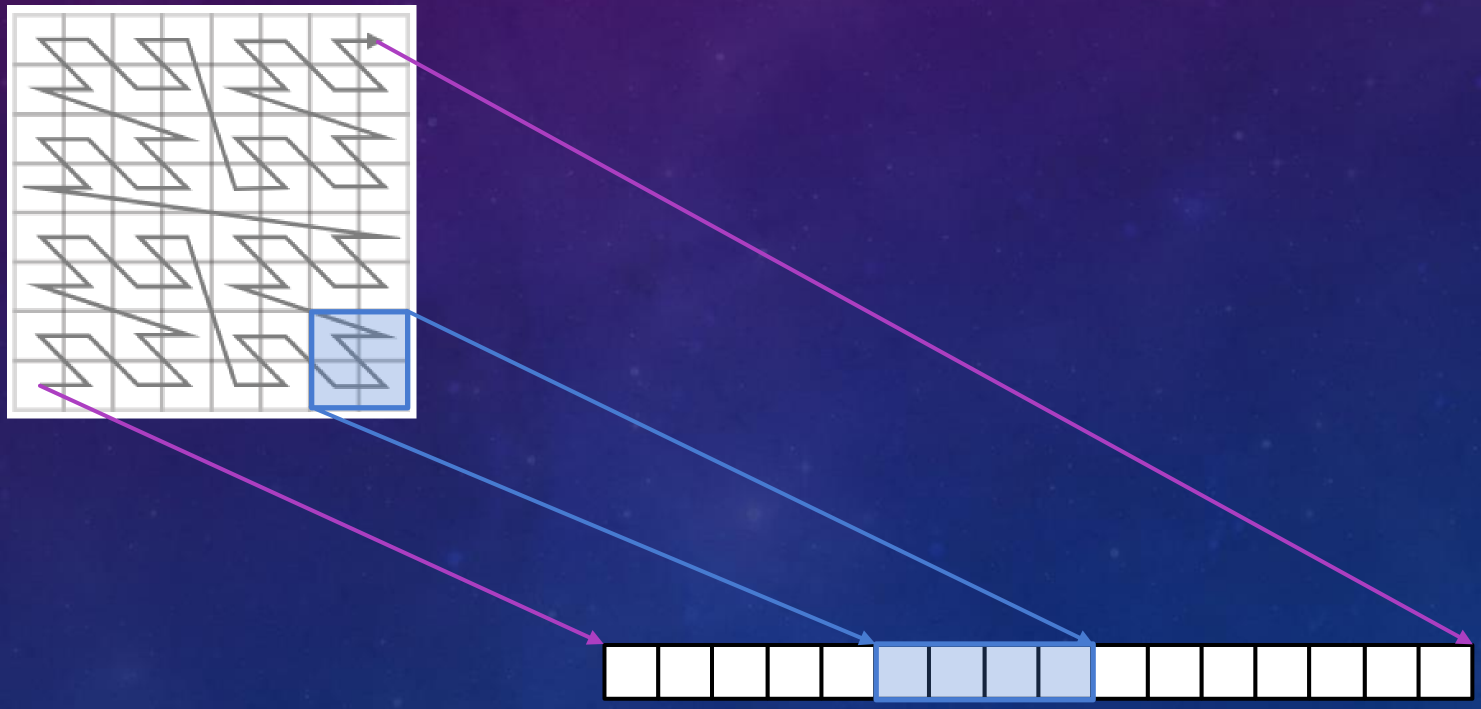


Image Credit: [1]

POINT CLOUD SERIALIZATION



POINT CLOUD SERIALIZATION

- $O(N^2) \rightarrow O(N)$
- Uses multiple space filling curves to smooth out approximation errors of any given curve.
- "In our implementation, we do not physically re-order the point clouds, but rather, we record the mappings generated by the serialization process. This strategy maintains compatibility with various serialization patterns and provides the flexibility to transition between them efficiently." [1]

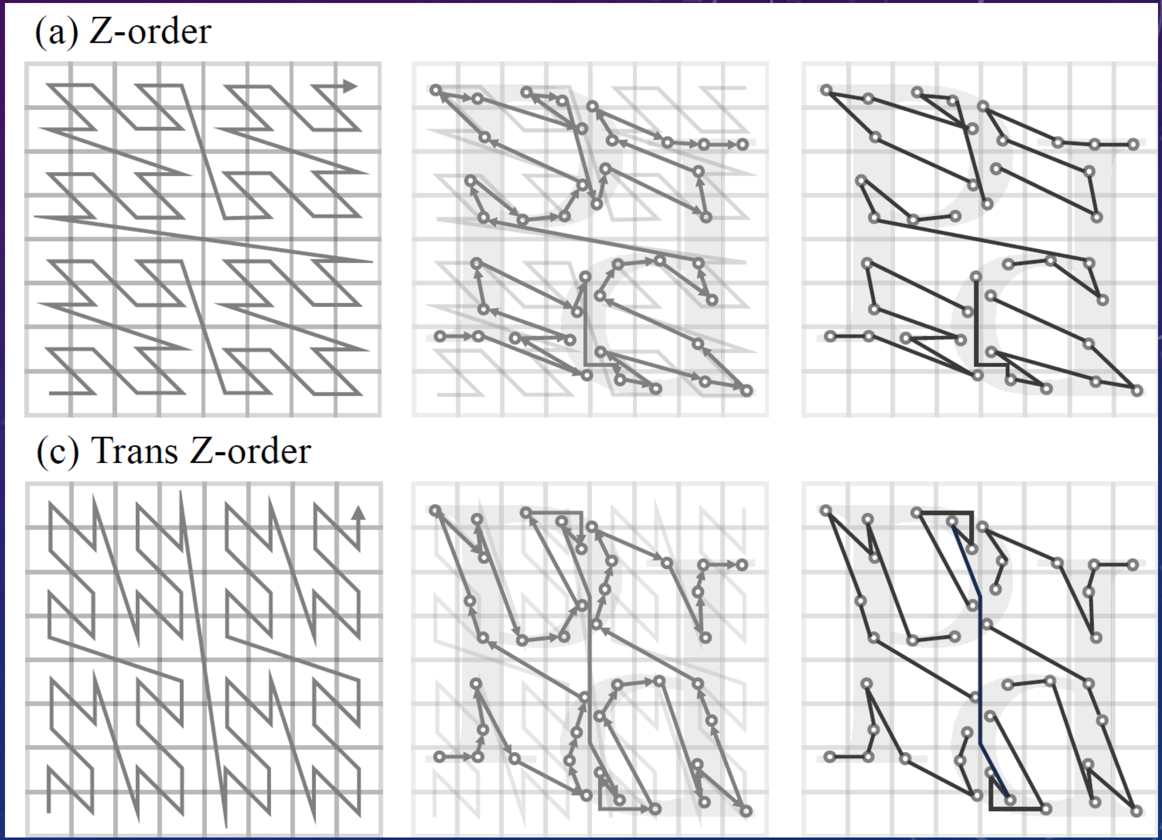


Image Credit: [1]

PATCH ATTENTION

- **Patch Grouping**
  - Partitioning 1D sequences into fixed-size segments.

- **Patch Interaction**
  - **Shift Dilation**
  - **Shift Patch**
  - **Shift Order**
  - **Shuffle Order**

- **The Trade-off**: Efficiency & Scalability over minor loss in neighborhood precision.

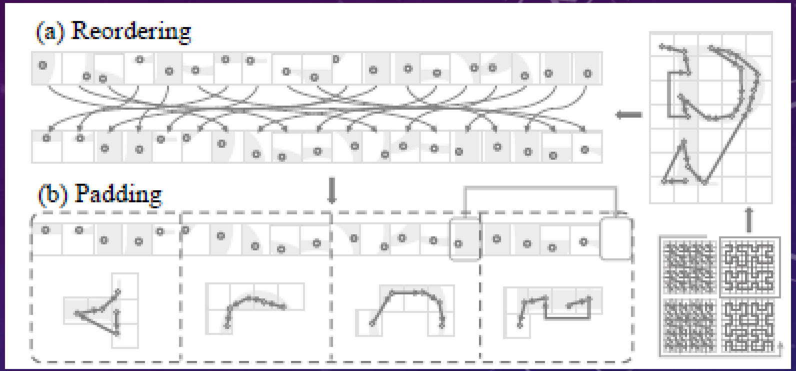


Image Credit: [1]

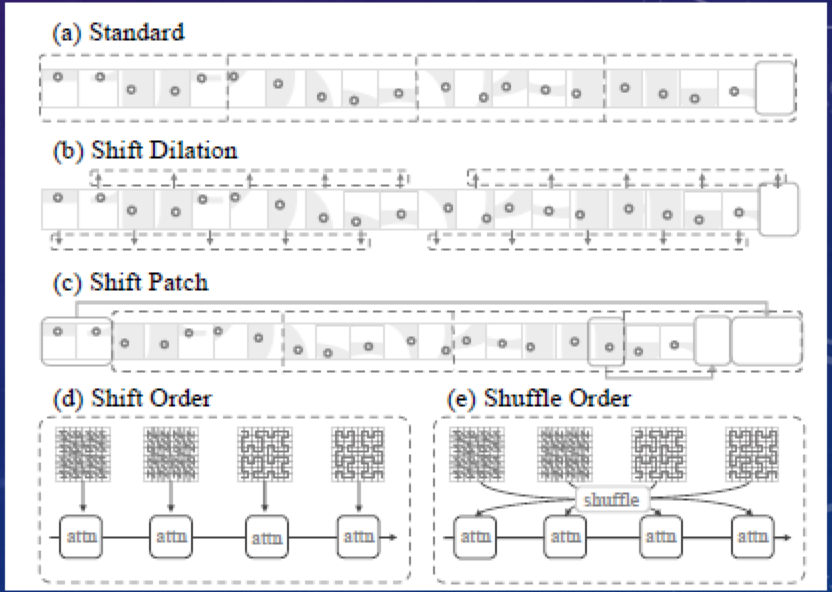


Image Credit: [1]

“BREAKING THE CURSE OF PERMUTATION INVARIANCE”

- “Current point transformers encounter challenges in scaling when adhering to the request of permutation invariance.” [1]
- From lecture on deep sets: a permutation invariant layer will have less trainable parameters, which bottlenecks the model. [10].
- “…we move away from the traditional paradigm, which treats point clouds as unordered sets. Instead, we choose to “break” the constraints of permutation invariance by serializing point clouds into a structured format.” [1]
- Point cloud serialization achieves overall permutation invariance without requiring invariance internally.
- Any ordering of the same points will map to the same serialization (permutation invariant). After this, the model treats it as a sequence instead (not permutation invariant).

PAPER RESULTS

- Outperforms all previous algorithms on several indoor and outdoor semantic segmentation datasets.
- Achieves large reductions in memory use and increase in speed over PTv2.

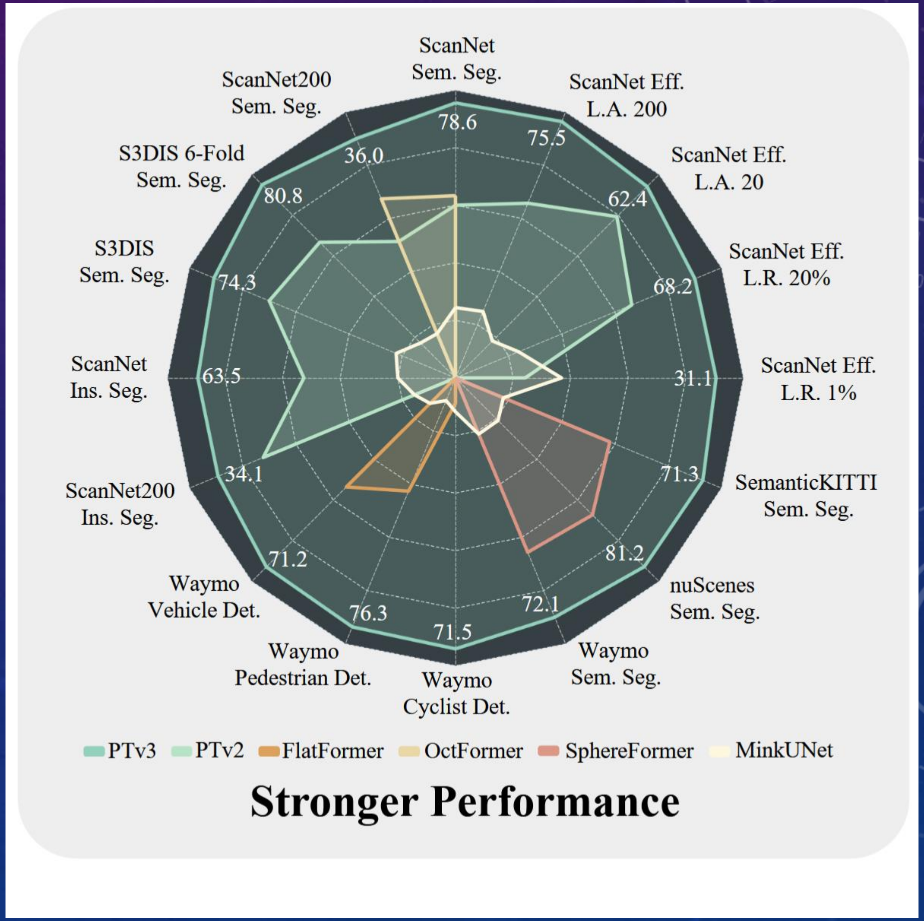
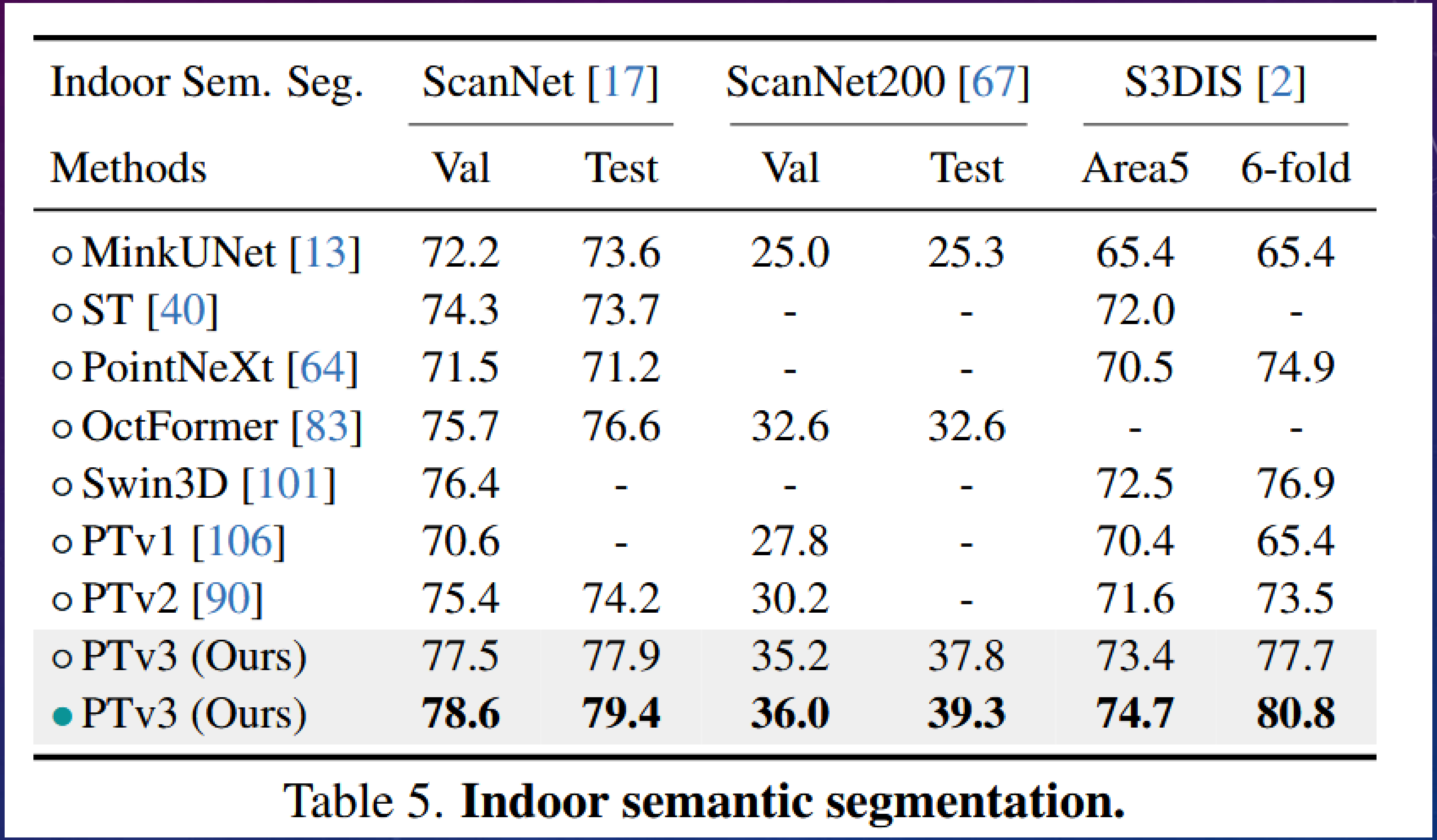


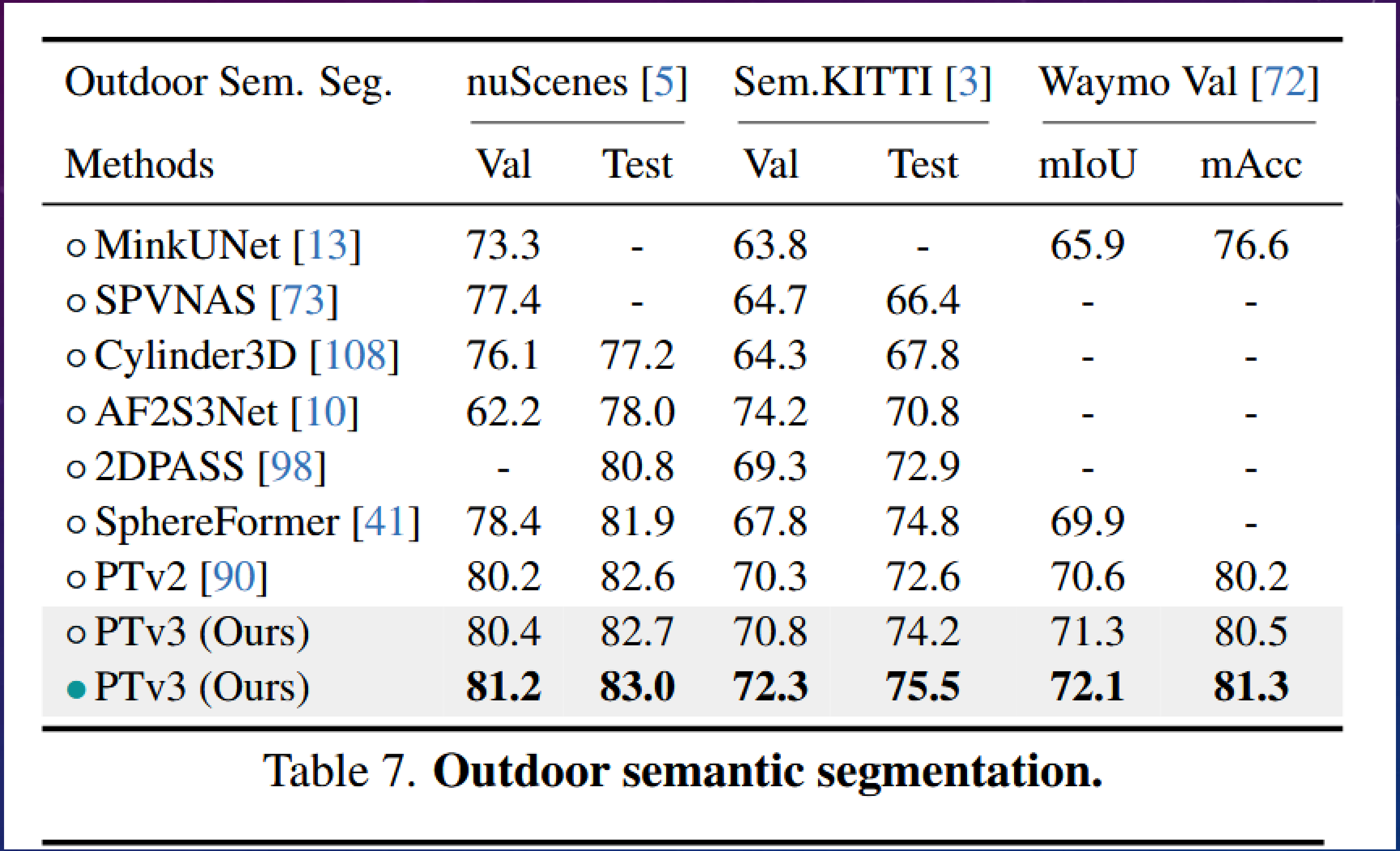
Image Credit: [1]



Indoor Sem. Seg.	ScanNet [17]		ScanNet200 [67]		S3DIS [2]	
Methods	Val	Test	Val	Test	Area5	6-fold
○ MinkUNet [13]	72.2	73.6	25.0	25.3	65.4	65.4
○ ST [40]	74.3	73.7	-	-	72.0	-
○ PointNeXt [64]	71.5	71.2	-	-	70.5	74.9
○ OctFormer [83]	75.7	76.6	32.6	32.6	-	-
○ Swin3D [101]	76.4	-	-	-	72.5	76.9
○ PTv1 [106]	70.6	-	27.8	-	70.4	65.4
○ PTv2 [90]	75.4	74.2	30.2	-	71.6	73.5
○ PTv3 (Ours)	77.5	77.9	35.2	37.8	73.4	77.7
● PTv3 (Ours)	**78.6**	**79.4**	**36.0**	**39.3**	**74.7**	**80.8**

Table 5. **Indoor semantic segmentation.**

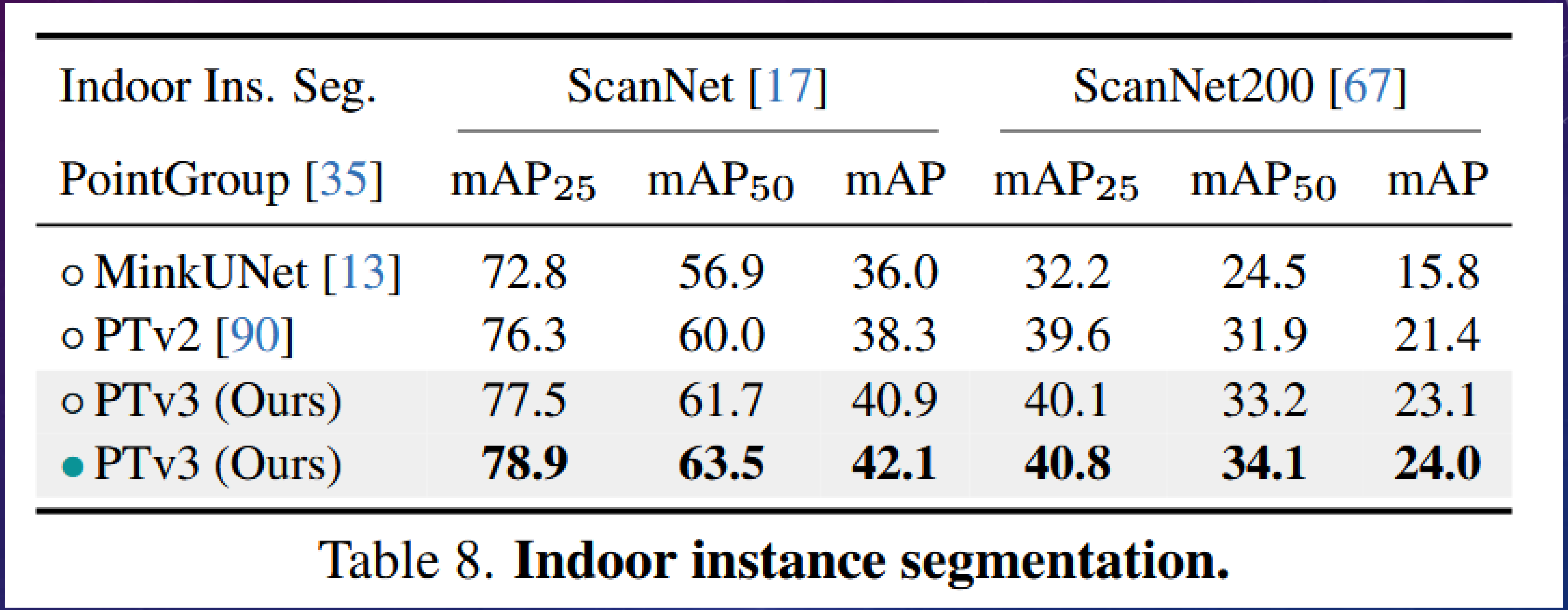
Image Credit: [1]



Outdoor Sem. Seg.	nuScenes [5]		Sem.KITTI [3]		Waymo Val [72]	
Methods	Val	Test	Val	Test	mIoU	mAcc
○ MinkUNet [13]	73.3	-	63.8	-	65.9	76.6
○ SPVNAS [73]	77.4	-	64.7	66.4	-	-
○ Cylinder3D [108]	76.1	77.2	64.3	67.8	-	-
○ AF2S3Net [10]	62.2	78.0	74.2	70.8	-	-
○ 2DPASS [98]	-	80.8	69.3	72.9	-	-
○ SphereFormer [41]	78.4	81.9	67.8	74.8	69.9	-
○ PTv2 [90]	80.2	82.6	70.3	72.6	70.6	80.2
○ PTv3 (Ours)	80.4	82.7	70.8	74.2	71.3	80.5
● PTv3 (Ours)	**81.2**	**83.0**	**72.3**	**75.5**	**72.1**	**81.3**

Table 7. **Outdoor semantic segmentation.**

Image Credit: [1]



Indoor Ins. Seg.	ScanNet [17]			ScanNet200 [67]		
PointGroup [35]	mAP25	mAP50	mAP	mAP25	mAP50	mAP
○ MinkUNet [13]	72.8	56.9	36.0	32.2	24.5	15.8
○ PTv2 [90]	76.3	60.0	38.3	39.6	31.9	21.4
○ PTv3 (Ours)	77.5	61.7	40.9	40.1	33.2	23.1
● PTv3 (Ours)	**78.9**	**63.5**	**42.1**	**40.8**	**34.1**	**24.0**

Table 8. **Indoor instance segmentation.**

Image Credit: [1]

PERSONAL THOUGHTS – ROTATION AND SCALE INVARIANCE (ALEC)

- KNN – rotation and scale invariant.
- Radius search – rotation invariant.
- Point cloud serialization – technically not rotation or scale invariant.
- Using multiple space filling curves reduces this issue, and dataset augmentation would help further.
- Overall, the paper says that the benefits of scaling up the model due to improved efficiency outweigh the drawbacks of such approximations.

PERSONAL THOUGHTS (CHENG)

- Scalability & efficiency First
  - scaling potential > intricate module design
  - Receptive field
  - Utilize GPU characteristics
- Structuring Unstructured Data

- Questions:
  - Generalizability of 3D-to-1D Serialization
  - What is the 'living legacy' of this work if Transformers are replaced?

- Suggestions
  - Ultimate Optimization: Hardware-Aware Co-Design

REFERENCES

- [1] X. Wu, L. Jiang, P.-S. Wang, Z. Liu, X. Liu, Y. Qiao, W. Ouyang, T. He, and H. Zhao, “Point Transformer V3: Simpler, Faster, Stronger,” *arXiv preprint arXiv:2312.10035*, Dec. 2023.
- [2] X. Wu, Y. Lao, L. Jiang, X. Liu, and H. Zhao, “Point Transformer V2: Grouped Vector Attention and Partition-based Pooling,” *arXiv preprint arXiv:2210.05666*, Oct. 2022.
- [3] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, “Point Transformer,” *arXiv preprint arXiv:2012.09164*, Dec. 2020.
- [4] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Póczos, R. Salakhutdinov, and A. Smola, “Deep Sets,” *arXiv preprint arXiv:1703.06114*, Mar. 2017.
- [5] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space,” *arXiv preprint arXiv:1706.02413*, Jun. 2017.
- [6] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, “Dynamic Graph CNN for Learning on Point Clouds,” *arXiv preprint arXiv:1801.07829*, Jan. 2018.
- [7] Y. Li, R. Bu, M. Sun, and B. Chen, “PointCNN: Convolution on X-Transformed Points,” *arXiv preprint arXiv:1801.07791*, Jan. 2018.
- [8] Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D., “Scaling Laws for Neural Language Models,” *arXiv preprint arXiv:2001.08361*, Jan. 2020, doi: 10.48550/arXiv.2001.08361.
- [9] AnalytixLabs, “Convolutional Neural Networks – Definition, Architecture, Types, Applications, and more,” *AnalytixLabs Blog*, Jan. 08, 2024. [Online].
- [10] R. Liao, “Lecture 2: Invariance, Equivariance, and Deep Learning Models for Sets/Sequences,” *EECE 571F: Advanced Topics in Deep Learning*, University of British Columbia, Vancouver, BC, Canada, Winter Term 2, 2025. [Online].
- [11] https://zhuanlan.zhihu.com/p/382365889