Self-Supervised Pretraining of 3D Features on any Point-Cloud

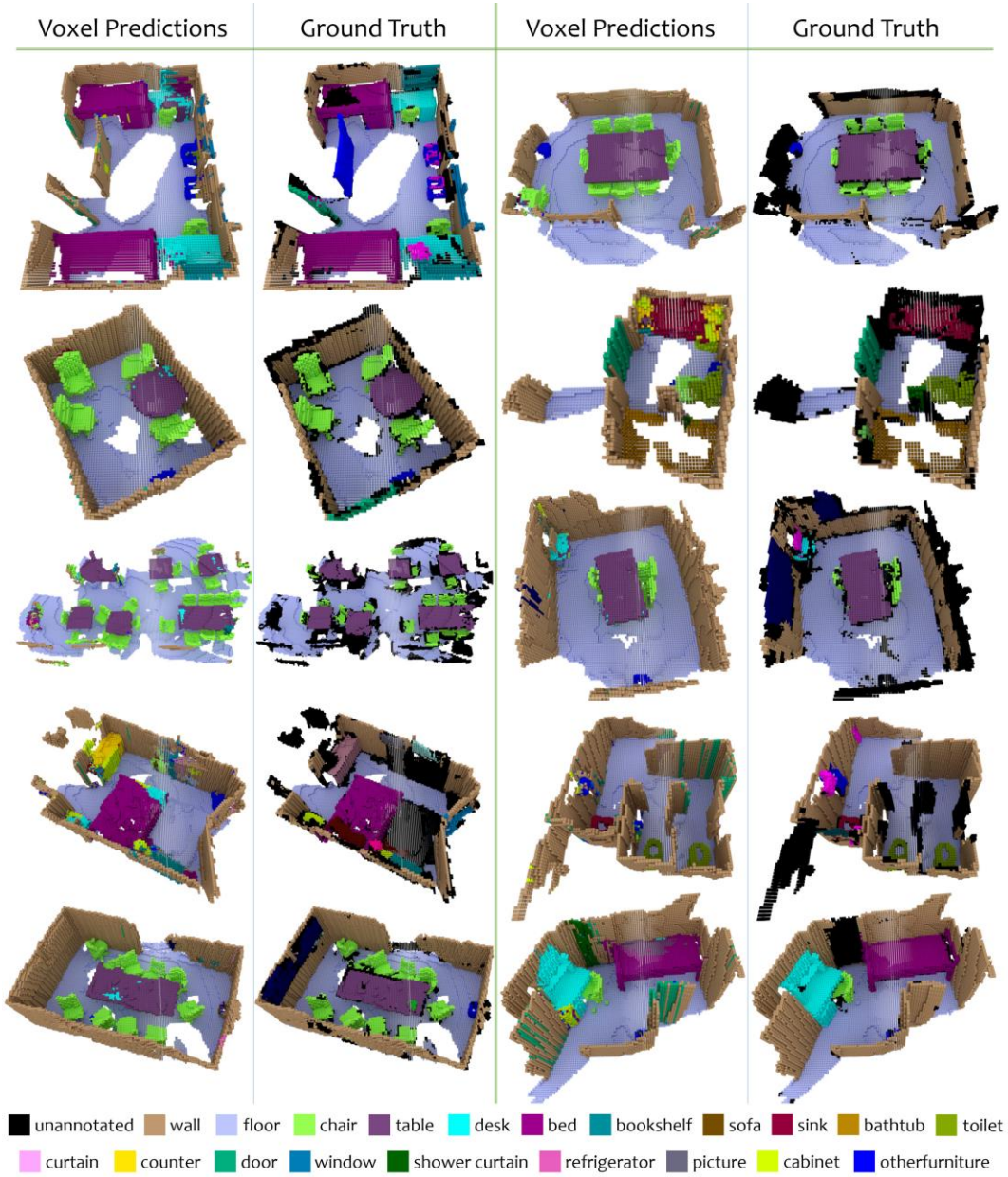
Paper Review

Isabel Taylor

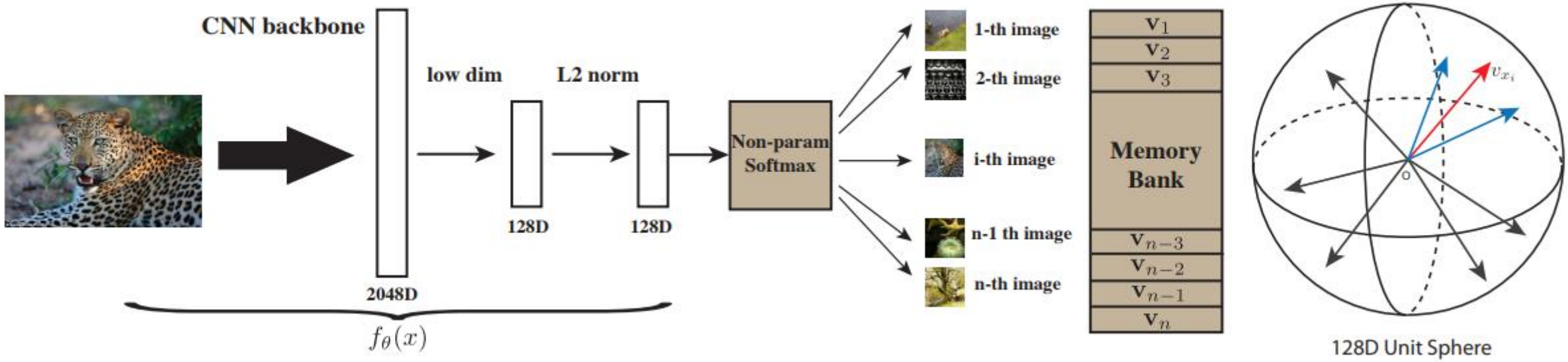
February 17th, 2021

Current Problem

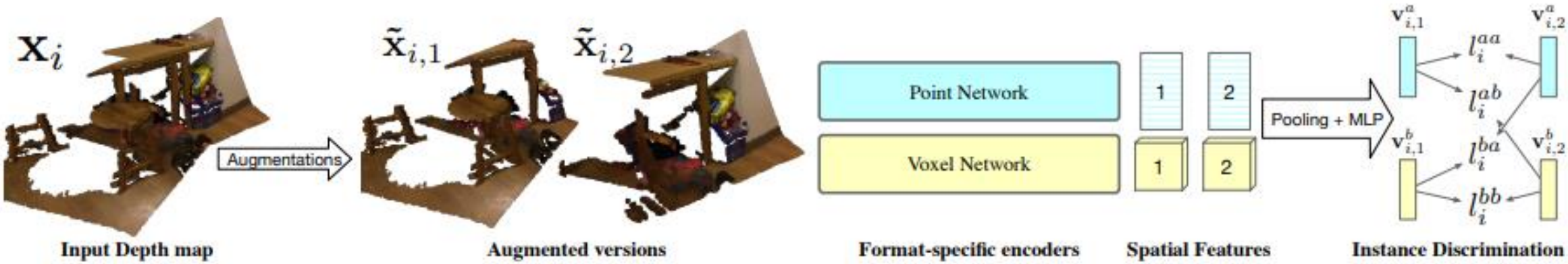
- Pretraining on a large set of data is necessary for good performance for many vision tasks.
- Large sets of annotated 3D data is difficult to acquire and time consuming to label.
- 3D reconstruction requires registering and aligning multiple static depth maps



Background on Citation 107



DepthContrast



Contrastive Learning and Extension to Multiple Formats

$$l_i = -\log\left(\frac{\exp\left(\frac{v_{i,1}^T v_{i,2}}{\tau}\right)}{\exp\left(\frac{v_{i,1}^T v_{i,2}}{\tau}\right) + \sum_{j\neq i}^{K}\exp\left(\frac{v_{i,1}^T v_j}{\tau}\right)}\right)$$

$$l_i^{ab} = -\log\left(\frac{\exp\left(\frac{v_{i,1}^{a\,T} v_{i,2}^b}{\tau}\right)}{\exp\left(\frac{v_{i,1}^{a\,T} v_{i,2}^b}{\tau}\right) + \sum_{j\neq i}^{K}\exp\left(\frac{v_{i,1}^{a\,T} v_j^b}{\tau}\right)}\right)$$

Combined Loss

$$l_i^{ab} = -\log\left(\frac{\exp\left(\frac{v_{i,1}^{a\,T} v_{i,2}^{b}}{\tau}\right)}{\exp\left(\frac{v_{i,1}^{a\,T} v_{i,2}^{b}}{\tau}\right) + \sum_{j\neq i}^{K} \exp\left(\frac{v_{i,1}^{a\,T} v_{j}^{b}}{\tau}\right)}\right)$$

$$L_i = l_i^{ab} + l_i^{ba} + l_i^{aa} + l_i^{bb}$$

Across Format

Within Format

Experimenting Details

- Fine-tuning on downstream tasks
  - Object classification, semantics segmentation, object detection
  - full scenes/object centric; using different 3D sensors; single/multi-view; real/synthetic; indoor/outdoor
- ScanNet dataset
  - 2.5 million RGB-D scans for more than 1500 indoor scenes
  - Extract about 190k RGB-D scans from video sequences
  - No camera calibration or 3D registration models
  - Operate on single-view depth maps
  - Preform data augmentation

Overall Results

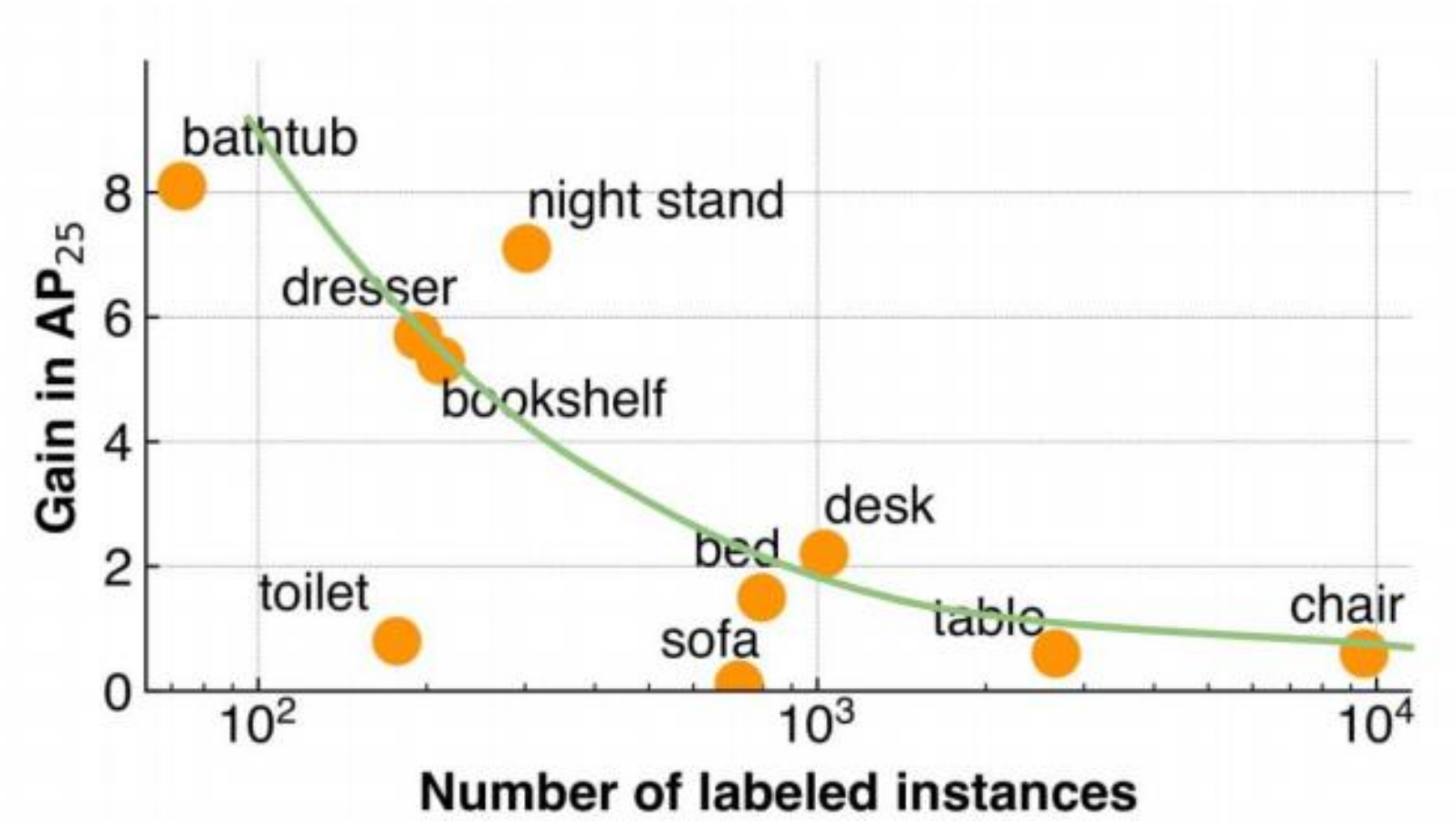
Dataset	Stats	Task	Gain of DepthContrast
Self-supervised Pretraining			
ScanNet-vid [18]	190K single-view depth maps (Indoor)		
Redwood-vid [16]	370K single-view depth maps (Indoor/Outdoor)		
Transfer tasks			
ScanNet [18]	1.2K train, 312 val (Indoor)	Det.	**+3.6% mAP**
		Seg.	**+0.9% mIOU**†
SUNRGBD [84]	5.2K train, 5K val (Indoor)	Det	**+3.3% mAP**
S3DIS [4]	199 train, 67 val (Indoor)	Det	**+12.1% mAP**
		Seg.	**+2.4% mIOU**
Synthia [74]	19.8K train, 1.8K val (Synth.)	Seg.	**+2.4% mIOU**
Matterport3D [10]	1.4K train, 232 val (Indoor)	Det.	**+3.9% mAP**
ModelNet [106]	9.8K train, 2.4K val (Synth.)	Cls.	**+3.1% Acc**†

Det.: Object Detection, Seg: Semantic Segmentation
Cls: Classification, Synth.: Synthetic, †Results in supplemental.

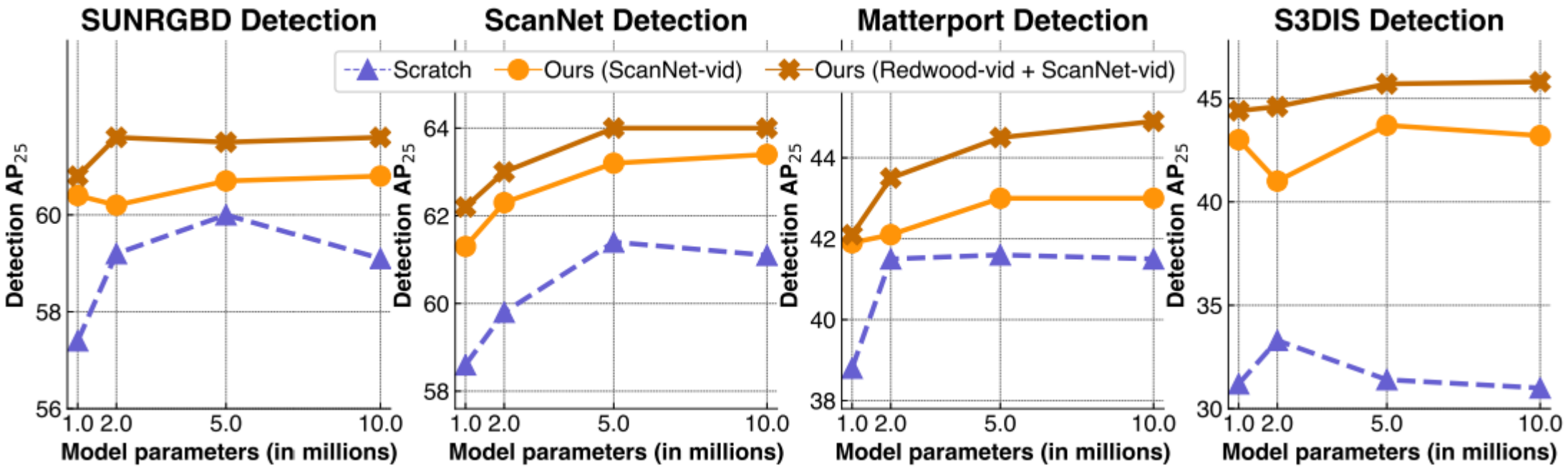
Importance of Augmentation

Task	VoteNet [67]	+Rand. Cuboid	+Rand. Drop
ModelNet Linear (Accuracy)	80.6	**85.4**	85.0
SUNRGBD Detection (mAP)	58.6	59.5	**60.7**

Long Tail Classes



Scaling Model and Pretraining Data



Multiple Input Formats

- 3 variants
  - Within format
    - For voxel methods, does not improve consistently from scratch
  - Across format
    - For point and voxel methods, improves performance
  - Combined
    - For voxel method, improves by 4% over the within format loss
    - Holds across different pretraining data and architecture
    - Comparable to the PointContrast method

$$L_i = l_i^{ab} + l_i^{ba} + l_i^{aa} + l_i^{bb}$$

Across Format

Within Format

Multiple Input Formats

Loss	Point Transfer		Voxel Transfer	
	SUNRGBD	ScanNet	S3DIS	Synthia
Scratch	57.4	58.6	68.2	78.9
Within Format only	60.4 (+3.0)	61.3 (+1.7)	66.5 (-2.7)	80.1 (+1.2)
Across format only	60.0 (+2.6)	61.1 (+2.5)	69.9 (+1.7)	81.2 (+2.3)
Both (Ours)	**60.7** (+3.3)	**62.2** (+3.6)	**70.6** (+2.4)	**81.3** (+2.4)
PointContrast [109]	57.5	59.2	70.9	83.1

Results on KITTI

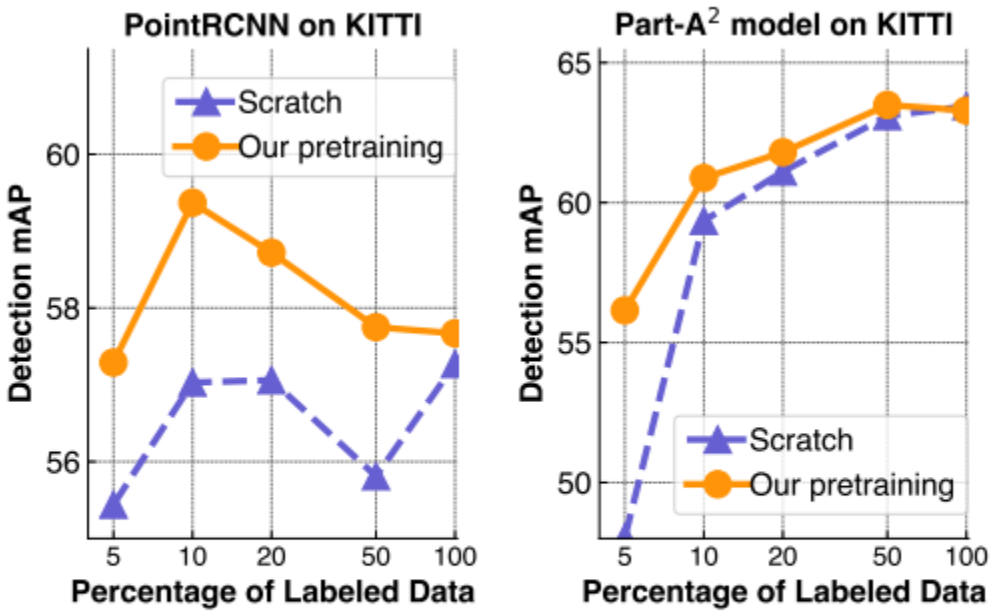


Figure 11: Label-efficiency evaluation for KITTI `pedestrian` detection at moderate difficulty level. We use the val split of the KITTI dataset.

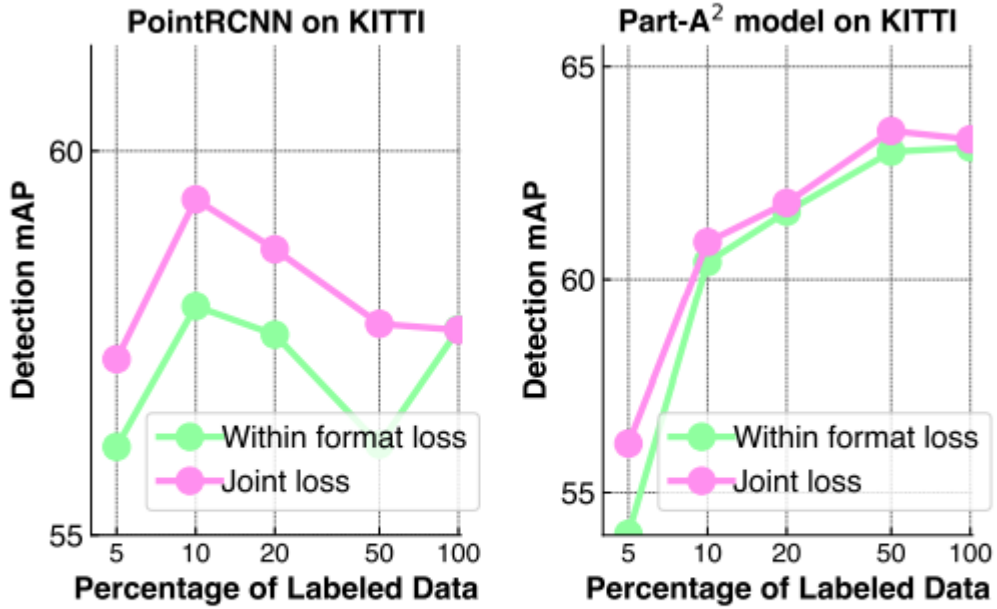


Figure 12: Comparison between within format and joint training loss. Label-efficiency evaluation for `pedestrian` detection at moderate difficulty level of the KITTI val split

Discussion

- The topic of data augmentation has been spoken about before in this class, demonstrating its usefulness. What other ways could 3D data be augmented in (than the ones mentioned)?

- The main advantage of this method is the generality of both the input data. What are some potential limitations of this generalization?

- Contrastive learning is a very popular method, what are some limitations of it? What are some possible other uses or benefits?