PViT + Pix4Point

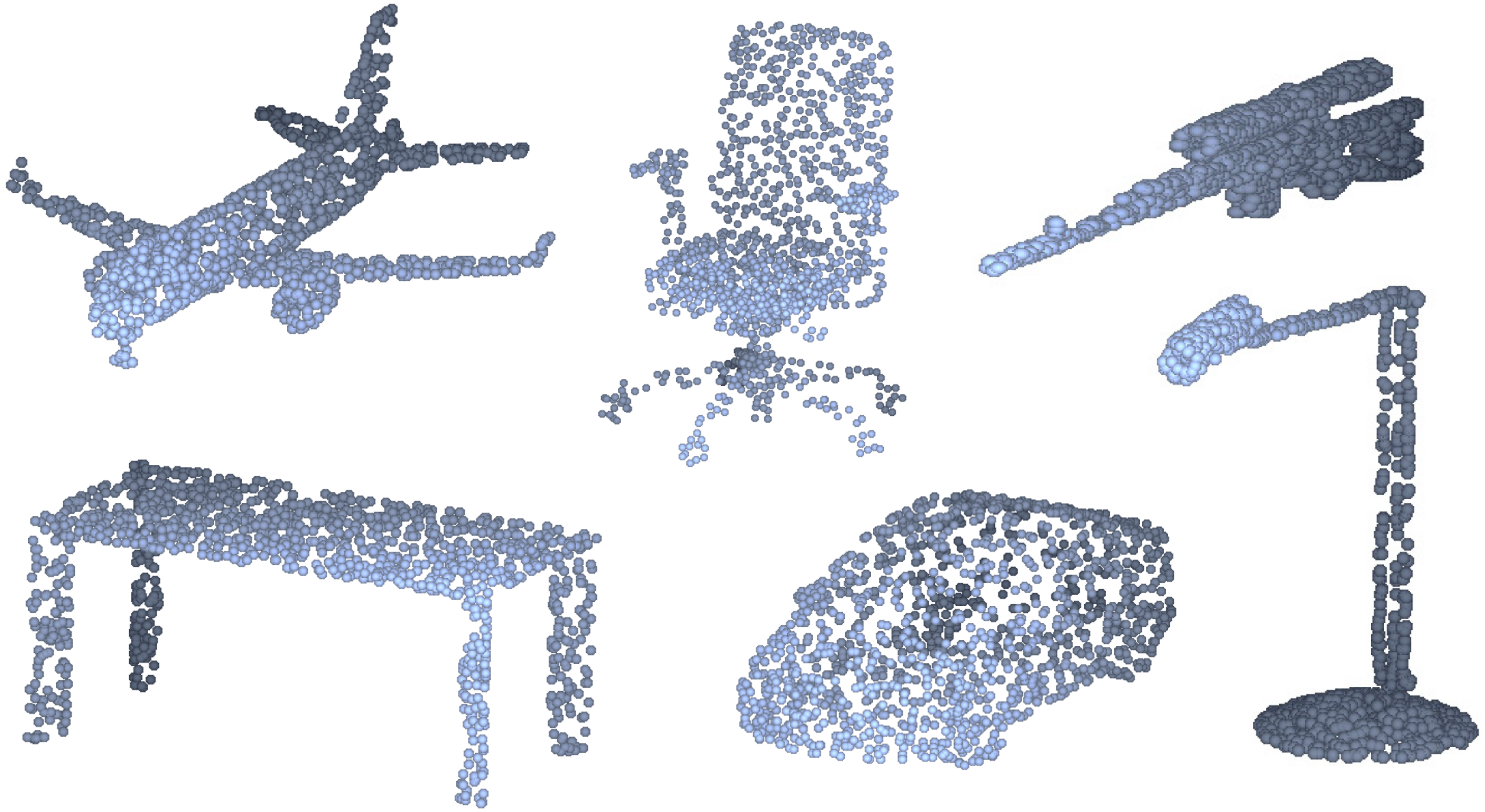
Image Pretrained Transformers for 3D Point Cloud Understanding

Anthony Baryshnikov @ December 7th 2022

Point clouds

Why are they interesting to us?

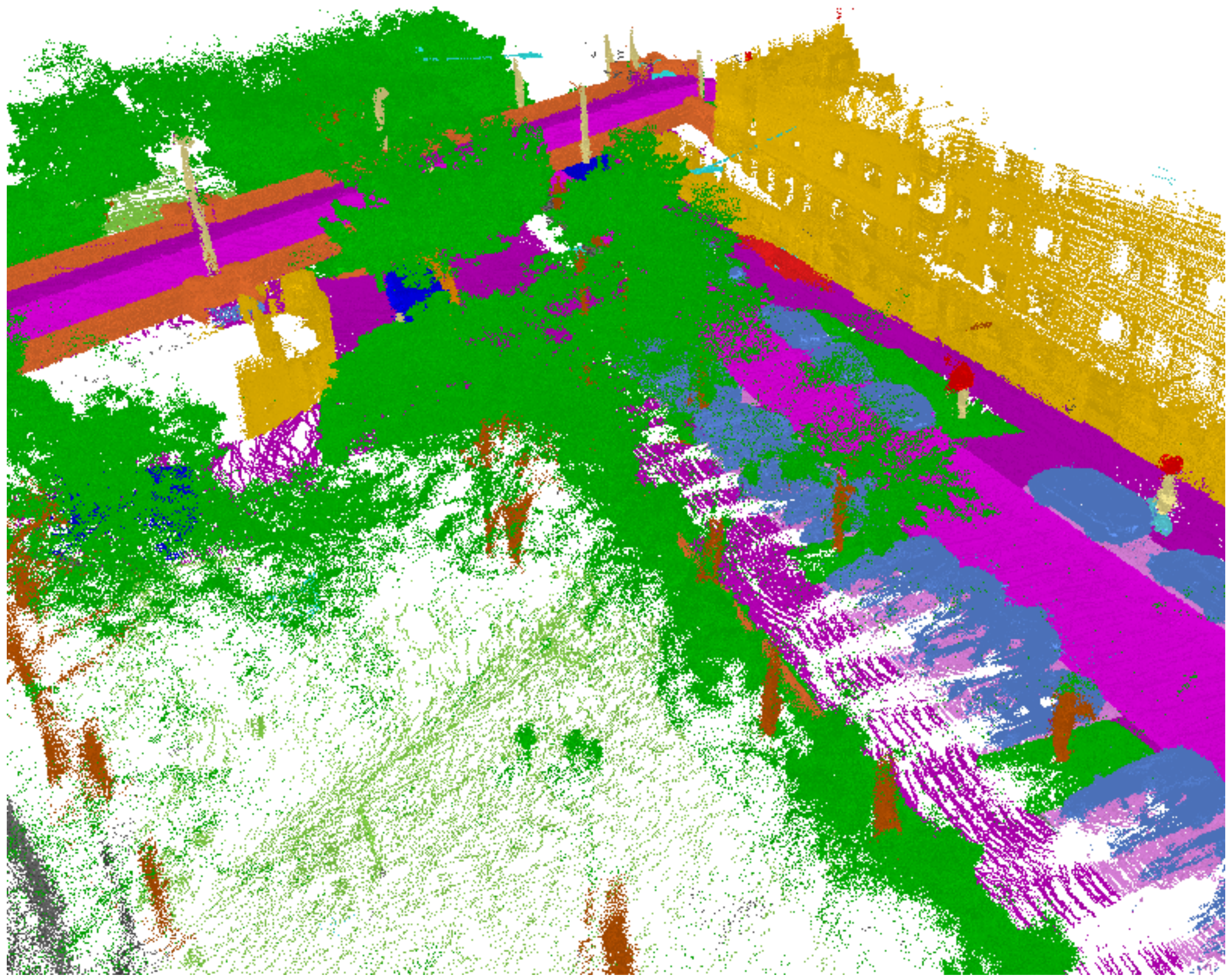
- Universal 3D object representation.
- Used in robotics and self-driving cars.
- No information loss.
- Easer to get than other formats.



Point clouds

Issues

- Very hard to get compared to other modalities.
- Very hard to clean the data from impurities.
- **Extremely** hard to annotate the data.



Transformers

Recap

- Work really well for images and text (ViT, GPT, you name it).
- Multimodal architecture.
- Scale better with more data.

Transformers + Point Clouds

Issues

- Transformers need a lot of data.
- Previous approaches don’t work too well with point clouds.
- Convolutions dominate the field.
- Let’s fix it!

What if we fix the architecture?

That should work

- We lose multimodal abilities.
- We will no longer scale that well with extra data.
- Prone to more overfitting.
- ST — standard transformer architecture.

What to improve?

Overview

- The tokenizer.
- The decoder.
- Find more data.
- The backbone must stay the same.

Proposed method

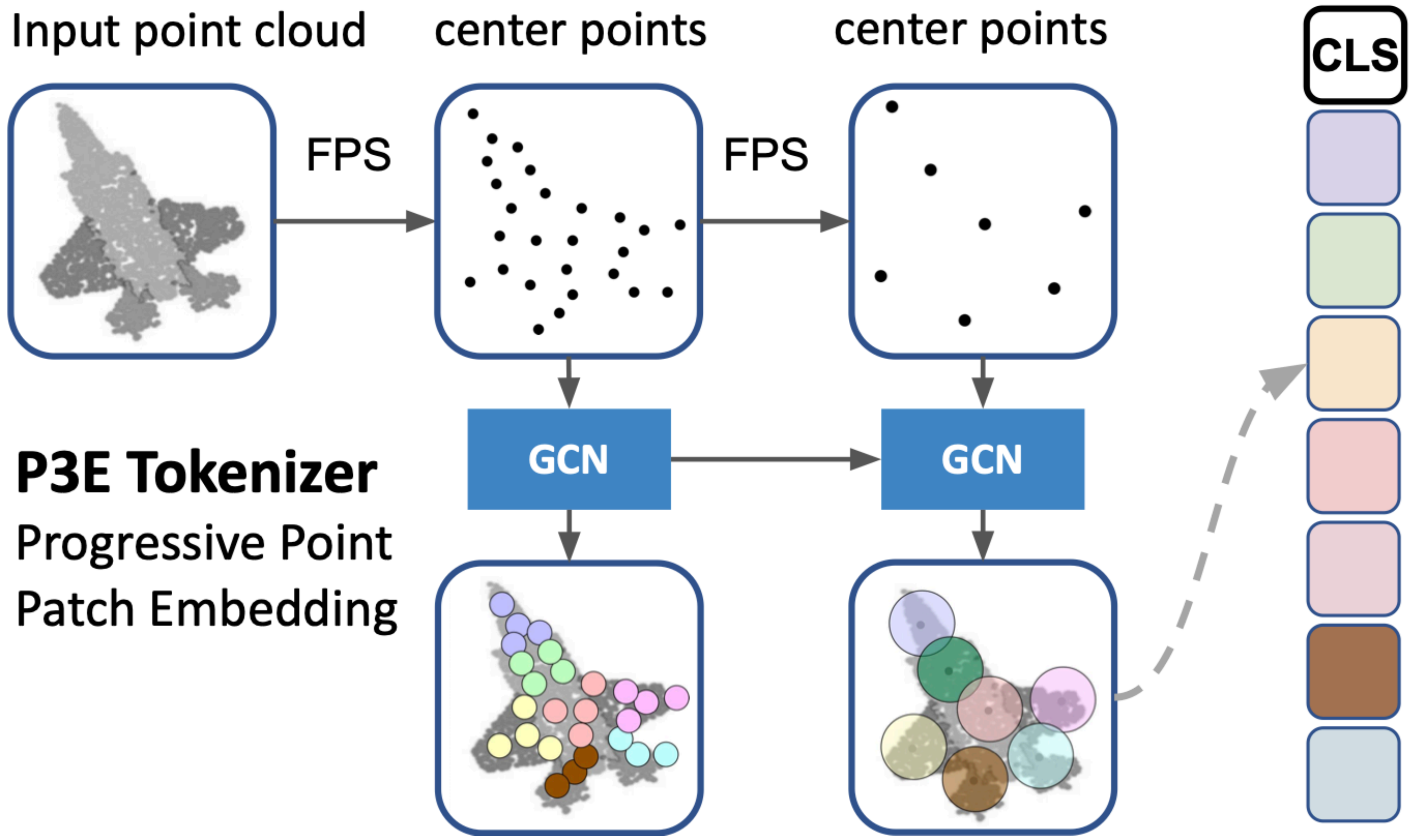
New tokenizer

Old method:

- Primitive patching.
- Simple feature extraction.

New method (P3E):

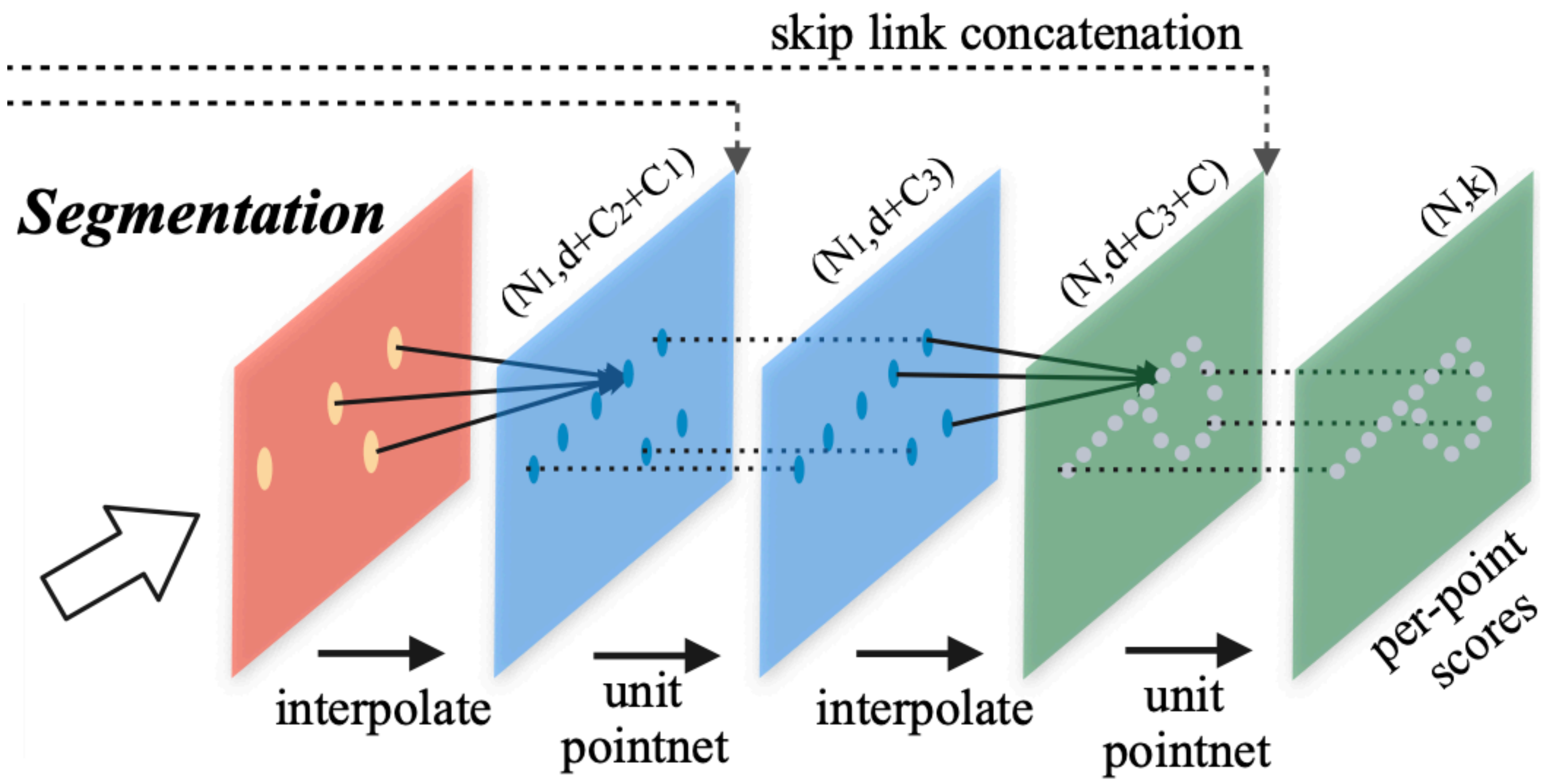
- Multiple patching steps.
- GCN for feature extraction.
- Uses relative features and positions.



Proposed method

New decoder

- How to restore the original number of points?
- Let's use feature propagation from PointNet++.
- Take the points at the hierarchy level.
- Interpolate the features using kNN from previous layer.
- Concatenate global information: [CLS] + global max pool.



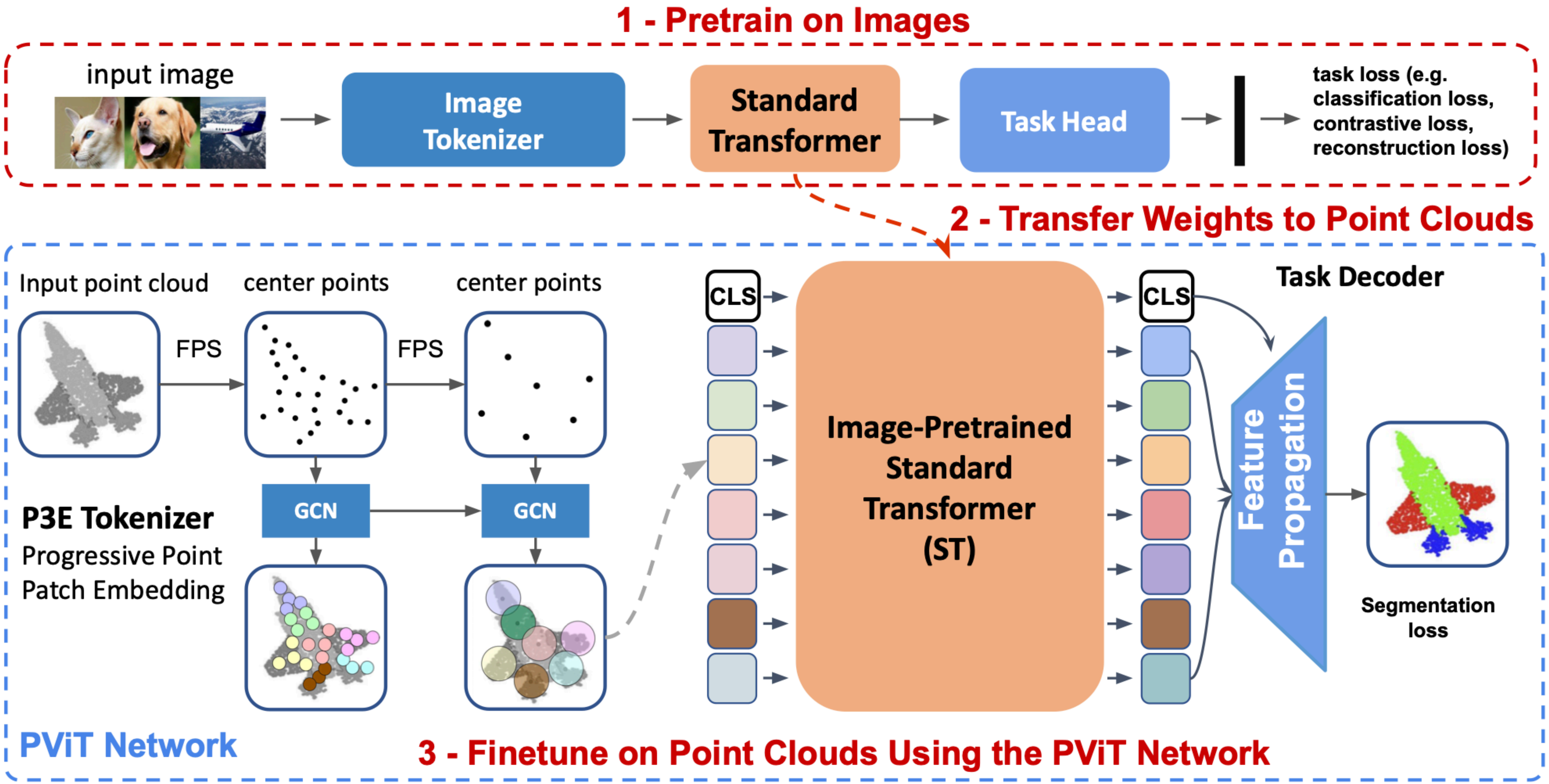
Proposed method

Image pre-training

- We want more data.
- Let’s use pre-trained image transformers as the initialization.
- They learn universal token interactions.
- We only need a few fine-tuning epochs.
- Sketchy explanation.

Proposed method

Overview



Results

Semantic segmentation on S3DIS

- Instance-level semantic segmentation.
- Large scale scenes.
- Much better than previous transformer approaches.
- Almost on par with CNN state-of-the-art.
- Much fewer parameters.

Method	**mIoU** (%)	**mAcc** (%)	**Params.** M
PointNet [38]	41.1	49.0	3.6
PointNet++ [39]	53.5	-	1.0
DeepGCN [27]	52.5	-	3.6
PVCNN [31]	59.0	-	-
KPConv [48]	67.1	72.8	15.0
ASSANet-L [41]	66.8	-	-
PCT [17]	61.3	67.7	-
Point Transformer [65]	70.4	76.5	7.8
PointNeXt [42]	70.5	76.8	41.6
Standard Transformer [63]	60.0	68.6	27.1
Point-BERT [63]	60.8	69.9	27.1
PViT	64.4 (+4.4)	69.9 (+1.3)	23.7
PViT+Pix4Point	69.6 (+9.6)	75.2 (+6.6)	23.7

Results

3D Part Segmentation on ShapeNetPart

- Smaller point clouds.
- Better than previous transformer approaches.
- Almost on par with CNN state-of-the-art.

Table 2. **Part Segmentation on ShapeNetPart.**

Method	Ins. mIoU	cls. mIoU	Params.
PointNet [38]	83.7	80.4	3.6
PointNet++ [39]	85.1	81.9	1.0
DGCNN [53]	85.2	82.3	1.3
KPConv [48]	86.4	85.1	15.0
CurveNet [57]	86.8	-	-
ASSANet-L [41]	86.1	-	-
PCT [17]	86.4	-	-
Point Transformer [65]	86.6	83.7	7.8
PointMLP [35]	86.1	84.6	12.6
StratifiedFormer [25]	86.6	85.1	-
PointNeXt [42]	87.0	85.2	22.5
ST [63]	85.1	83.4	27.1
Point-BERT [63]	85.6	84.1	27.1
Point-MAE [37]	86.1	84.2	27.1
PViT	85.7 (+0.6)	83.7 (+0.3)	23.8
PViT+Pix4Point	**86.8** (+1.7)	**85.6** (+2.2)	23.8

Results

Object Classification

- New state-of-the-art.

Table 3. **3D Object Classification on ScanObjectNN PB_T50_RS.**

Method	**OA** (%)	**mAcc** (%)	**Params.** M
PointNet [38]	68.2	63.4	3.5
PointNet++ [39]	77.9	75.4	1.5
PointCNN [29]	78.5	75.1	0.6
DGCNN [53]	78.1	73.6	1.8
PointMLP [35]	86.4	83.9	13.2
PointNeXt [42]	87.7	85.8	1.4
Standard Transformer [63]	77.2	-	22.1
Point-BERT [63]	83.1	-	22.1
Point-MAE [37]	85.2	-	22.1
PViT	85.7 (+8.5)	83.5	22.7
PViT+Pix4Point	**87.9** (+10.7)	**86.7**	22.7

Conclusions

What to remember?

- Transformers are good.
- If transformers aren’t good then you’re the issue.
- We can use the same backbone in all domains.
- Image pre-training works better than pre-training on smaller 3D datasets.
- Maybe generalizes to other fields.