Point-BERT : Pre-training 3D Point Cloud Transformers with Masked Point Modeling

https://arxiv.org/pdf/2111.14819

0. Introduction

- 3D 포인트 클라우드 인식 분야에서는 기존에 복잡한 구조나 수작업 특징이 필요했음
- 2D 이미지·언어에서 성공한 Transformer 를 그대로 3D에 적용하기 어려웠던 이유는 3D 데이터의 구조적 특성 때문
- 논문은 BERT 방식의 사전학습을 3D 포인트 클라우드에 확장하기 위해 Point-BERT를 제안
- 주요 기여
  - 3D 포인트 클라우드를 로컬 패치 단위로 잘라 토큰으로 만드는 Tokenizer
  - 마스킹된 패치를 복원하는 Masked Point Modeling (MPM)
  - 순수 Transformer 기반 3D 학습 성능 향상

1. Overview

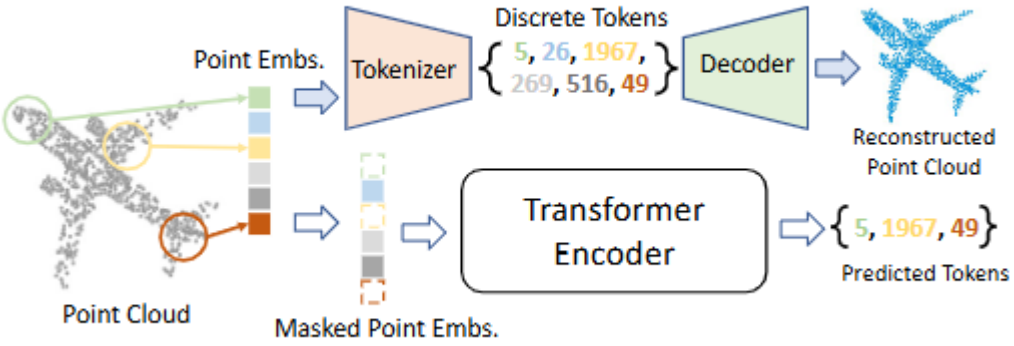
- 핵심 아이디어 : 포인트 클라우드를 **패치 토큰 시퀀스**로 변환해 Transformer 로 학습
- 처리 흐름 : 포인트 클라우드를 sub-cloud → 임베딩 → dVAE 로 discrete token → 일부 마스킹 → Transformer 로 예측
- neighbor-aggregation 없이 Transformer 기반 표현 학습 가능

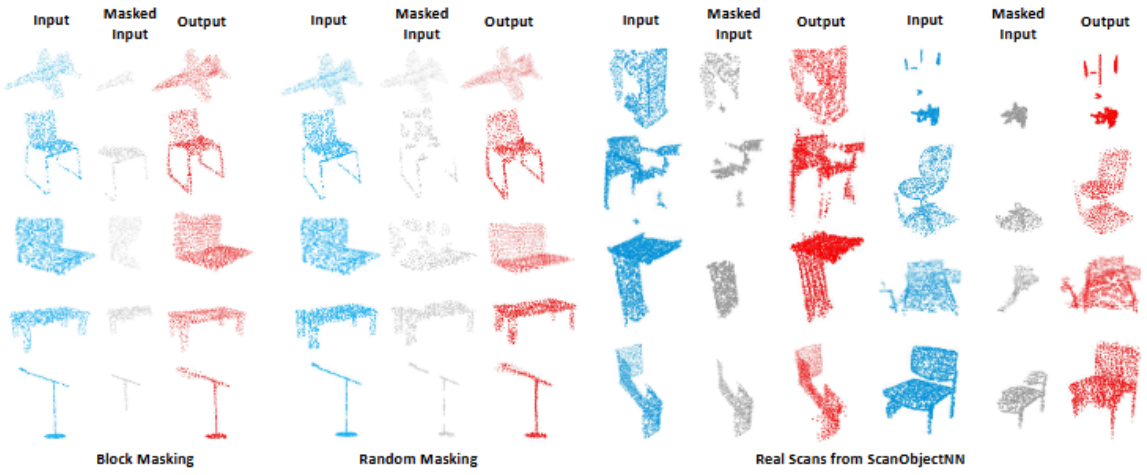
2. Challenges

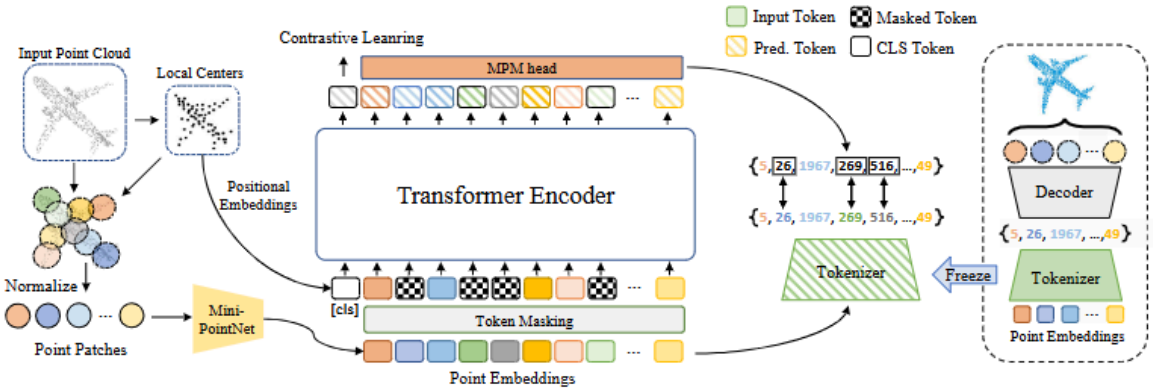
- 3D 데이터는 이미지·텍스트처럼 “단어 사전”이 없어 토큰화가 어려움
- 모든 포인트를 토큰으로 쓰면 self-attention 비용 폭증
- 3D 데이터의 형태 다양성으로 일반화 어려움

- 기하학 + 의미정보를 모두 Transformer 로 학습하기 어렵기 때문에 적절한 pre-training 과제 필요

3. Method







- **Point Tokenization**
  - FPS 로 중심점 선택 → 주변 kNN 으로 sub-cloud 생성
  - 각 sub-cloud 에 mini-PointNet 적용 → 임베딩
  - dVAE 로 discrete token 생성 → 3D vocabulary 형성
- **Masked Point Modeling (MPM)**
  - 일부 패치 마스킹
  - Transformer 가 마스킹된 토큰을 예측하게 학습
  - Patch Mixing 으로 서로 다른 샘플 패치를 섞어 일반화 향상
  - contrastive loss 도 함께 사용해 semantic 정보 강화
- **Transformer Backbone**
  - 표준 Transformer encoder
  - 입력은 sub-cloud 임베딩 + positional embedding

4. Experiments

- Pre-training : ShapeNet 사용
- 1024 포인트 → 64 sub-cloud (각 32 포인트)
- dVAE vocabulary 크기 8192
- Transformer depth 12, feature 384, head 6
- 마스크 비율 25% ~ 45%
- Downstream : classification, part segmentation, few-shot, transfer 등

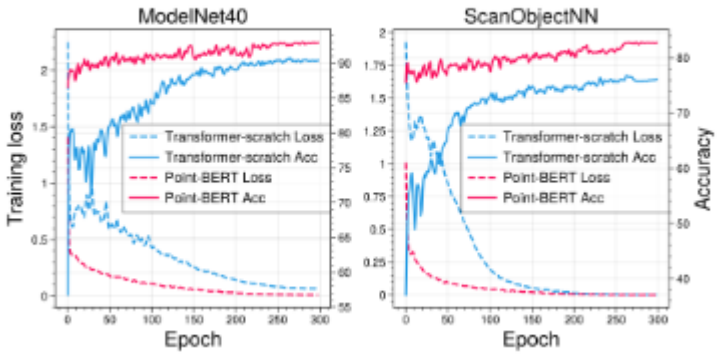
5. Results

Method	#point	Acc.
PointNet [34]	1k	89.2
PointNet++ [35]	1k	90.5
SO-Net [22]	1k	92.5
PointCNN [23]	1k	92.2
DGCNN [54]	1k	92.9
DensePoint [24]	1k	92.8
RSCNN [38]	1k	92.9
KPConv [46]	~6.8k	92.9
[T] PCT [11]	1k	93.2
[T] PointTransformer [65]	–	93.7
[ST] NPCT [11]	1k	91.0
[ST] Transformer	1k	91.4
[ST] Transformer + OcCo [52]	1k	92.1
[ST] Point-BERT	1k	93.2
[ST] Transformer	4k	91.2
[ST] Transformer + OcCo [52]	4k	92.2
[ST] Point-BERT	4k	93.4
[ST] Point-BERT	8k	**93.8**

Methods	mIoU$_C$	mIoU$_I$	aero	bag	cap	car	chair	earphone	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skateboard	table
PointNet [34]	80.39	83.7	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9	80.8	95.3	65.2	93	81.2	57.9	72.8	80.6
PointNet++ [35]	81.85	85.1	82.4	79	87.7	77.3	90.8	71.8	91	85.9	83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6
DGCNN [54]	82.33	85.2	84	83.4	86.7	77.8	90.6	74.7	91.2	87.5	82.8	95.7	66.3	94.9	81.1	63.5	74.5	82.6
Transformer	83.42	85.1	82.9	85.4	87.7	78.8	90.5	80.8	91.1	87.7	85.3	95.6	73.9	94.9	83.5	61.2	74.9	80.6
Transformer-OcCo	83.42	85.1	83.3	85.2	88.3	79.9	90.7	74.1	91.9	87.6	84.7	95.4	75.5	94.4	84.1	63.1	75.7	80.8
Point-BERT	**84.11**	**85.6**	84.3	84.8	88.0	79.8	91.0	81.7	91.6	87.9	85.2	95.6	75.6	94.7	84.3	63.4	76.3	81.5

Methods	OBJ-BG	OBJ-ONLY	PB-T50-RS
PointNet [34]	73.3	79.2	68.0
SpiderCNN [58]	77.1	79.5	73.7
PointNet++ [35]	82.3	84.3	77.9
PointCNN [23]	86.1	85.5	78.5
DGCNN [54]	82.8	86.2	78.1
BGA-DGCNN [49]	–	–	79.7
BGA-PN++ [49]	–	–	80.2
Transformer	79.86	80.55	77.24
Transformer-OcCo	84.85	85.54	78.79
Point-BERT	**87.43**	**88.12**	**83.07**

Pretext tasks	MPM	Point Patch Mixing	Moco	Acc.
Model A				91.41
Model B	✓			92.58 ↑
Model C	✓	✓		92.91 ↑
Model D	✓	✓	✓	93.24 ↑
Augmentation	**mask type**	**mask ratio**	**replace**	**Acc.**
Model B	block mask	[0.25, 0.45]	No	92.58
Model B	block mask	[0.25, 0.45]	Yes	91.81 ↓
Model B	rand mask	[0.25, 0.45]	No	92.34 ↓
Model B	block mask	[0.55, 0.85]	No	92.52 ↓
Model D	block mask	[0.25, 0.45]	No	93.16
Model D	block mask	[0.25, 0.45]	Yes	92.58 ↓
Model D	rand mask	[0.25, 0.45]	No	92.91 ↓
Model D	block mask	[0.55, 0.85]	No	92.59 ↓



- ModelNet40 : 93.8% 정확도
- ScanObjectNN : 어려운 설정에서도 83.1% 수준
- Few-shot 학습에서 큰 효과
- Ablation 결과 :
  - MPM → 성능 상승
  - MPM + Patch Mixing → 추가 상승
  - MPM + Patch Mixing + contrastive → 최고 성능
- 포인트 수 증가에 따라 성능 안정적 향상

6. Insight

- 3D에서도 BERT 스타일 학습이 효과적이라는 첫 강력한 증거
- discrete 토큰화 + masked modeling 조합이 핵심
- few-shot, 도메인 전이에서 특히 강함
- 하지만 전체 파이프라인이 비교적 무거움