LCM: Locally Constrained Compact Point CloudModel for Masked Point Modeling

Yaohua Zha[1,2], Naiqi Li[1], Yanzi Wang[1], Tao Dai[3,*], Hang Guo[1], Bin Chen[4], Zhi Wang[1], Zhihao Ouyang[5], Shu-tao Xia[1,2]

[1]Tsinghua University; [2]Pengcheng Laboratory , [3]Shenzhen University; [4]Harbin Institute of Technology; [5]Bytedance Inc.

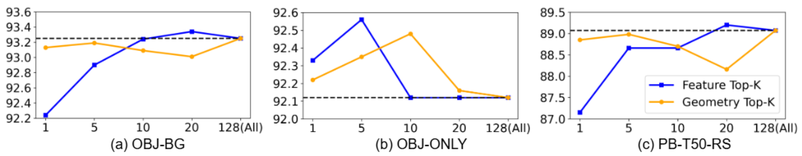
*Corresponding author

Motivation

➢ Two Inherent Issues of Transformers in Point Cloud

- Transformer require quadratic complexity and huge model sizes. In practical point cloud applications, models are often deployed on resource limited devices such as robots or VR headsets.
- The reconstruction potential of the Transformer for masked points with low information density is limited when used as a decoder in Masked Point Modeling (MPM).

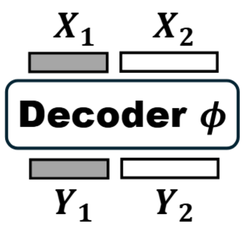
➢ Observation of Top-K Attention



Empirical observations of directly skipping attention computation for less important points.

- In point clouds, computing attention weights using the top-K most important tokens is generally more efficient than using all tokens.
- Compared to using top-K attention in a dynamic feature space, applying top-K attention in a static geometric space results in almost identical representational capacity while requiring fewer computations.

➢ An Information Theoretic Perspective for MPM Decoder



$$I(Y_2^M; X_1, X_2) \geq I(Y_2^T; X_1, X_2)$$

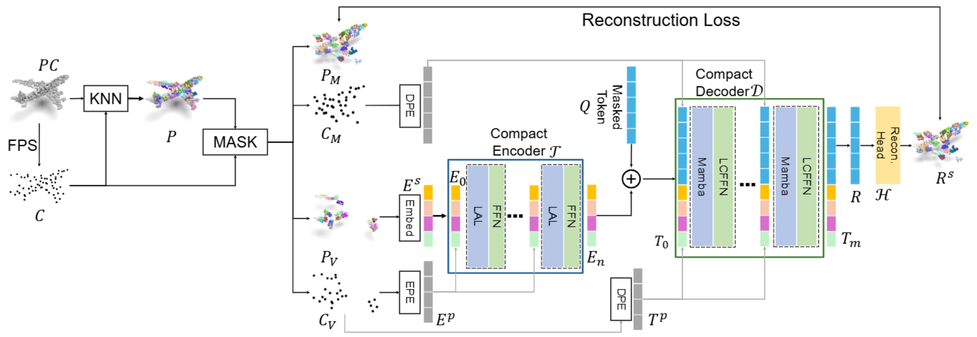
The mutual information based on the Mamba decoder is greater than that of the Transformer decoder.

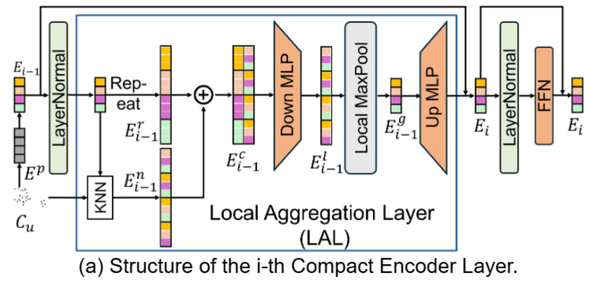
➢ Solution

💡 **Redundancy reduction** idea for optimizing efficiency.

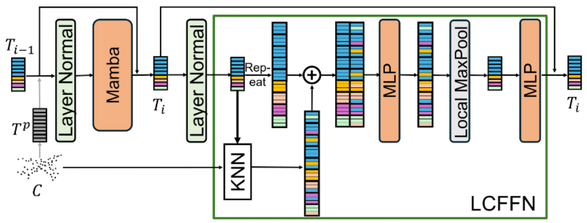
💡 **Mamba decoder** for reconstruction.

Method





(a) Structure of the i-th Compact Encoder Layer.



(b) Structure of the i-th Mamba-based Decoder Layer.

Locally Constrained Compact Encoder

$$\boldsymbol{E_i} = \boldsymbol{E_{i-1}} + l_i(n_i^1(\boldsymbol{E_{i-1}}), \boldsymbol{C_u})$$
$$\boldsymbol{E_i} = \boldsymbol{E_i} + f_i(n_i^2(\boldsymbol{E_i}))$$

Locally Constrained Mamba-based Decoder

$$\boldsymbol{T_i} = \boldsymbol{T_{i-1}} + s_i(n_i^1(\boldsymbol{T_{i-1}}))$$
$$\boldsymbol{T_i} = \boldsymbol{T_i} + f_i^l(n_i^2(\boldsymbol{T_i}), \boldsymbol{C})$$

Experiments

➢ Object Classification

Method	Pretrain	#Params(M)	FLOPs(G)	ScanObjectNN OBJ-BG	ScanObjectNN OBJ-ONLY	ScanObjectNN PB-T50-RS
Supervised Learning Only						
◦ PointNe [39]	✘	3.5	0.5	73.3	79.2	68.0
◦ PointNet++ [40]	✘	1.5	1.7	82.3	84.3	77.9
◦ PointMLP [32]	✘	12.6	31.4	-	-	85.2
◦ Transformer [46]	✘	22.1	4.8	86.75	86.92	80.78
◦ PointMamba [27]	✘	12.3	-	88.30	87.78	82.48
◦ SFR [63]	✘	-	-	-	-	87.80
• Transformer [46]	✘	22.1	4.8	91.95	91.39	86.65
• LCM (Ours)	✘	2.7(↓ 88%)	1.3(↓ 73%)	92.77(↑ 0.82)	91.54(↑ 0.15)	87.75(↑ 1.10)
Self-Supervised Learning						
◦ Point-BERT [60]	MPM	22.1	4.5	87.43	88.12	83.07
◦ MaskPoint [28]	MPM	22.1	4.5	89.30	88.10	84.30
◦ Point-MAE [37]	MPM	22.1	4.8	90.02	88.29	85.18
◦ Point-MAE w/ IDPT [64]	MPM	23.3	7.1	91.22	90.02	84.94
◦ Point-MAE w/ DAPT [70]	MPM	22.7	5.0	90.88	90.19	85.08
◦ Inter-MAE [29]	MPM	22.1	4.8	88.70	89.60	85.40
◦ Point-M2AE [65]	MPM	12.9	7.9	91.22	88.81	86.43
◦ ACT [8]	MPM	22.1	4.8	93.29	91.91	88.21
◦ PointGPT-B [5]	GPT	120.5	36.2	93.60	92.50	**89.60**
◦ PointMamba [27]	MPM	12.3	-	93.29	91.91	88.17
• Point-BERT [60]	MPM	22.1	4.5	92.48	91.60	87.91
• MaskPoint [28]	MPM	22.1	4.5	92.17	91.69	87.65
• Point-MAE [37]	MPM	22.1	4.8	92.67	92.08	88.27
• Point-M2AE [65]	MPM	12.9	7.9	93.12	91.22	88.06
• ACT [8]	MPM	22.1	4.8	92.08	91.70	87.52
• Point-BERT **w/ LCM**	MPM	3.1 (↓ 86%)	2.5 (↓ 44%)	93.55 (↑ 1.07)	92.43 (↑ 0.83)	88.57 (↑ 0.66)
• MaskPoint **w/ LCM**	MPM	3.1 (↓ 86%)	2.5 (↓ 44%)	93.31 (↑ 1.14)	91.98 (↑ 0.29)	87.75 (↑ 0.10)
• Point-MAE **w/ LCM**	MPM	2.7 (↓ 88%)	**1.3** (↓ 73%)	**94.51** (↑ 1.84)	92.75 (↑ 0.67)	88.87 (↑ 0.60)
• Point-M2AE **w/ LCM**	MPM	**2.5** (↓ 81%)	6.7 (↓ 15%)	93.83 (↑ 0.71)	92.41 (↑ 1.19)	88.38 (↑ 0.32)
• ACT **w/ LCM**	MPM	3.1 (↓ 86%)	2.8 (↓ 42%)	94.13 (↑ 2.05)	**92.66** (↑ 0.96)	88.57 (↑ 1.05)

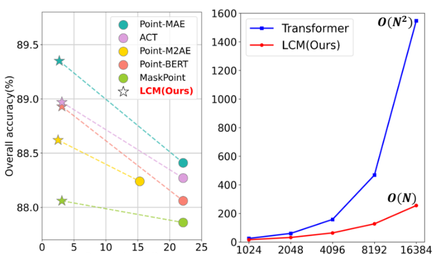
➢ Object detection

Methods	Pretrain	AP_{25}	AP_{50}
Supervised Learning Only			
VoteNet [38]	✘	58.6	33.5
3DETR [34](*baseline*)	✘	62.1	37.9
Transformer [46]	✘	60.5	40.6
LCM (**Ours**)	✘	63.8 (↑ 3.3)	46.4 (↑ 5.8)
Self-Supervised Learning			
PointContrast [58]	CL	58.5	38.0
STRL [25]	CL	-	38.4
Point-BERT [60]	MPM	61.0	38.3
PiMAE [4]	MPM	62.6	39.4
Point-MAE† [37]	MPM	59.5	41.2
Point-M2AE† [65]	MPM	60.0	41.4
ACT [8]	MPM	63.8	42.1
DepthContrast [69]	CL	64.0	42.9
MaskPoint [28]	MPM	64.2	42.1
Point-BERT [60] **w/ LCM**	MPM	**65.3** (↑ 4.3)	**47.3** (↑ 9.0)
Point-MAE [37] **w/ LCM**	MPM	64.7 (↑ 5.2)	47.2 (↑ 6.0)
Point-M2AE [65] **w/ LCM**	MPM	63.5 (↑ 3.5)	44.0 (↑ 2.6)
ACT [8] **w/ LCM**	MPM	65.0 (↑ 1.2)	45.8 (↑ 3.7)
MaskPoint [28] **w/ LCM**	MPM	65.3 (↑ 1.1)	46.3 (↑ 4.2)

➢ Part segmentation

Methods	Pretrain	mIoU_c	mIoU_I
Supervised Learning Only			
PointNet++ [40]	✘	81.9	85.1
DGCNN [50]	✘	82.3	85.2
Transformer [46]	✘	83.9	86.0
LCM (Ours)	✘	84.6 (↑ 0.7)	86.3 (↑ 0.3)
Self-Supervised Learning			
Transformer-OcCo [48]	CL	83.4	85.1
PointContrast [58]	CL	-	85.1
CrossPoint [1]	CL	-	85.5
Point-BERT [60]	MPM	84.1	85.6
IDPT [64]	MPM	83.8	85.9
MaskPoint [28]	MPM	84.4	86.0
Point-MAE [37]	MPM	84.2	86.1
ACT [8]	MPM	84.7	86.1
PointGPT-S [5]	MPM	84.1	86.2
PointGPT-B [5]	MPM	84.5	86.4
Point-M2AE [65]	MPM	84.9	86.5
Point-BERT [60] **w/ LCM**	MPM	85.0 (↑ 0.9)	86.5 (↑ 0.9)
MaskPoint [28] **w/ LCM**	MPM	85.1 (↑ 0.7)	86.6 (↑ 0.6)
Point-MAE [37] **w/ LCM**	MPM	**85.1** (↑ 0.9)	86.6 (↑ 0.5)
Point-M2AE [65] **w/ LCM**	MPM	85.0 (↑ 0.1)	86.5 (-)
ACT [8] **w/ LCM**	MPM	85.0 (↑ 0.3)	**86.7** (↑ 0.6)

➢ LCM v.s. Transformer



(a) Model Accuracy and Parameters

(b) Complexity growth curve

Additional Resources

paper code