PU-Transformer: Point Cloud Upsampling Transformer

Shi Qiu [1,2], Saeed Anwar [1,2], Nick Barnes [1]
[1] The Australian National University; [2] DATA61-CSIRO, Australia

ACCV 2022 Macau

Background

Point Cloud Upsampling:

◇ The aim is to generate dense point clouds from sparse input, where the generated data should recover the fine-grained structures at a higher resolution.

◇ Raw point cloud data has inherent properties of irregularity and sparsity, posing enormous challenges for further processings.

◇ The upsampled points are expected to lie on the underlying surfaces in a uniform distribution, benefiting downstream tasks for both 3D data visualization and visual analysis.

◇ Transformer has theoretic plausibility, practical feasibility and applicable adaptability in point cloud upsampling.

Contributions:

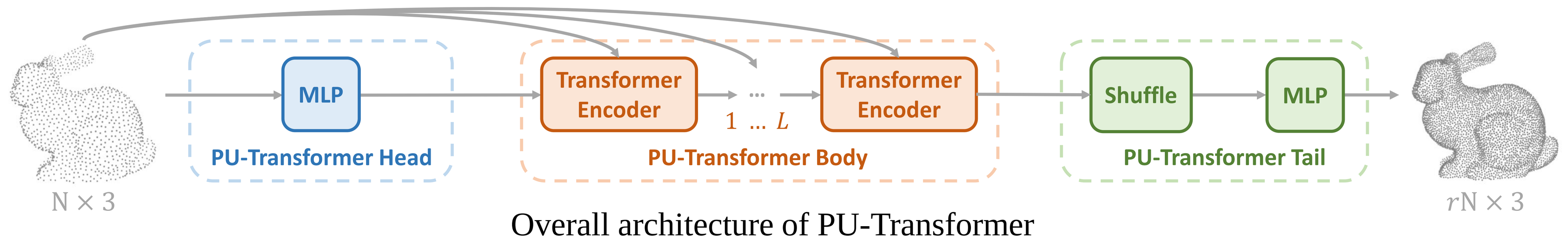
◇ We are the first to introduce a transformer-based model for point cloud upsampling.

◇ We quantitatively validate the effectiveness of the PU-Transformer by significantly outperforming the results of state-of-the-art point cloud upsampling networks on two benchmarks using three metrics.

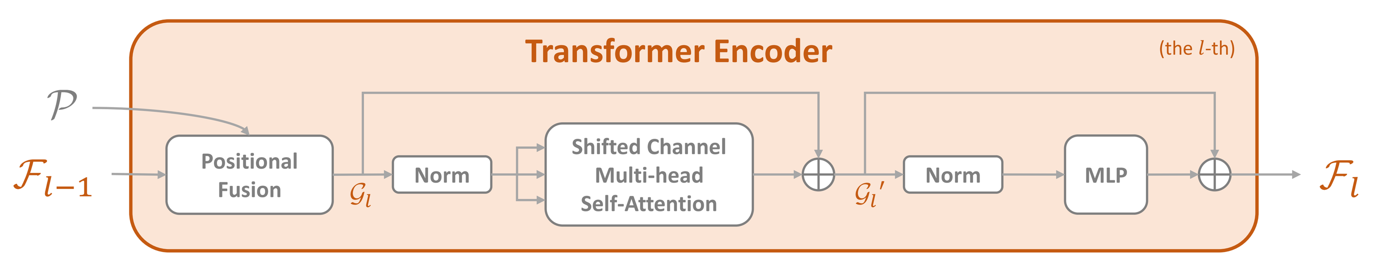
◇ The upsampled visualizations demonstrate the superiority of PU-Transformer for diverse point clouds.

Overall Architecture

The details of PU-Transformer:



Overall architecture of PU-Transformer



Components:

◇ **PU-Transformer Head:** To encode a preliminary feature map for the following operations. In practice, we only use a single layer MLP.

◇ **PU-Transformer Body:** To learn comprehensive feature representations of input point cloud using a cascaded set of Transformer Encoders. The detailed structure is shown in the lower chart.

◇ **PU-Transformer Tail:** To construct a dense resolution feature map by reforming the body's output via a channel-wise periodic shuffling operation.

Positional Fusion Block

◇ **Geometric context:** original 3D coordinates $\mathcal{P} \in \mathbb{R}^{N\times 3}$
Feature context: a learned feature map $\mathcal{F} \in \mathbb{R}^{N\times C}$

◇ **K-nearest-neighbors:** $\mathcal{P}_j \in \mathbb{R}^{N\times k\times 3}$

◇ **Relative positions:** $\Delta\mathcal{P} = \mathcal{P}_j - \mathcal{P} \in \mathbb{R}^{N\times k\times 3}$
Relative features: $\Delta\mathcal{F} = \mathcal{F}_j - \mathcal{F} \in \mathbb{R}^{N\times k\times C}$

◇ **Local geometric context:** $\mathcal{G}_{geo} = \text{concat}\big[\underset{k}{\text{dup}}(\mathcal{P}); \Delta\mathcal{P}\big] \in \mathbb{R}^{N\times k\times 6}$
Local feature context: $\mathcal{G}_{feat} = \text{concat}\big[\underset{k}{\text{dup}}(\mathcal{F}); \Delta\mathcal{F}\big] \in \mathbb{R}^{N\times k\times 2C}$

◇ **Output point feature:** encoded by two MLPs ($\mathcal{M}$)

$$\mathcal{G} = \max_{k}\Big(\text{concat}\big[\mathcal{M}_{\Phi}(\mathcal{G}_{geo}); \mathcal{M}_{\Theta}(\mathcal{G}_{feat})\big]\Big) \in \mathbb{R}^{N\times C'}$$

SC-MSA Block

Shifted Channel Multi-head Self-Attention:

Algorithm 1: Shifted Channel Multi-head Self-Attention

input: a point cloud feature map: $\mathcal{I} \in \mathbb{R}^{N\times C'}$
output: the refined feature map: $\mathcal{O} \in \mathbb{R}^{N\times C'}$
others: channel-wise split width: w
channel-wise shift interval: d, $d < w$
the number of heads: M

$\mathcal{Q} = \text{Linear}(\mathcal{I})$ # Query Mat $\mathcal{Q} \in \mathbb{R}^{N\times C'}$
$\mathcal{K} = \text{Linear}(\mathcal{I})$ # Key Mat $\mathcal{K} \in \mathbb{R}^{N\times C'}$
$\mathcal{V} = \text{Linear}(\mathcal{I})$ # Value Mat $\mathcal{V} \in \mathbb{R}^{N\times C'}$
for $m \in \{1, 2, ..., M\}$ **do**
  $\mathcal{Q}_m = \mathcal{Q}[\,:\,, (m-1)d : (m-1)d + w]$;
  $\mathcal{K}_m = \mathcal{K}[\,:\,, (m-1)d : (m-1)d + w]$;
  $\mathcal{V}_m = \mathcal{V}[\,:\,, (m-1)d : (m-1)d + w]$;
  $\mathcal{A}_m = \text{softmax}(\mathcal{Q}_m {\mathcal{K}_m}^T)$;
  $\mathcal{O}_m = \mathcal{A}_m \mathcal{V}_m$;
end for
obtain: $\{\mathcal{O}_1, \mathcal{O}_2, ..., \mathcal{O}_M\}$
$\mathcal{O} = \text{Linear}\Big(\text{concat}\big[\{\mathcal{O}_1, \mathcal{O}_2, ..., \mathcal{O}_M\}\big]\Big)$

Comparisons to Multi-head Self-Attention:

◇ It is easier to integrate the information between the *connected* multi-head outputs, compared to using the *independent* multi-head results of regular MSA.

◇ SC-MSA can further enhance the channel-wise relations in the final output, better fulfilling an efficient and effective shuffling-based upsampling strategy than only using regular MSA's point-wise information.

Experiment

PU1K Dataset:

Methods	Model (MB)	Time ($\times 10^{-3}$s)	Param. ($\times 10^{3}$)	Results ($\times 10^{-3}$) CD ↓	HD ↓	P2F ↓
PU-Net	10.1	8.4	812.0	1.155	15.170	4.834
MPU	6.2	8.3	76.2	0.935	13.327	3.551
PU-GACNet	–	–	**50.7**	0.665	9.053	2.429
PU-GCN	**1.8**	**8.0**	76.0	0.585	7.577	2.499
Dis-PU	13.2	10.8	1047.0	0.485	6.145	1.802
Ours	18.4	9.9	969.9	**0.451**	**3.843**	**1.277**

PUGAN's Dataset:

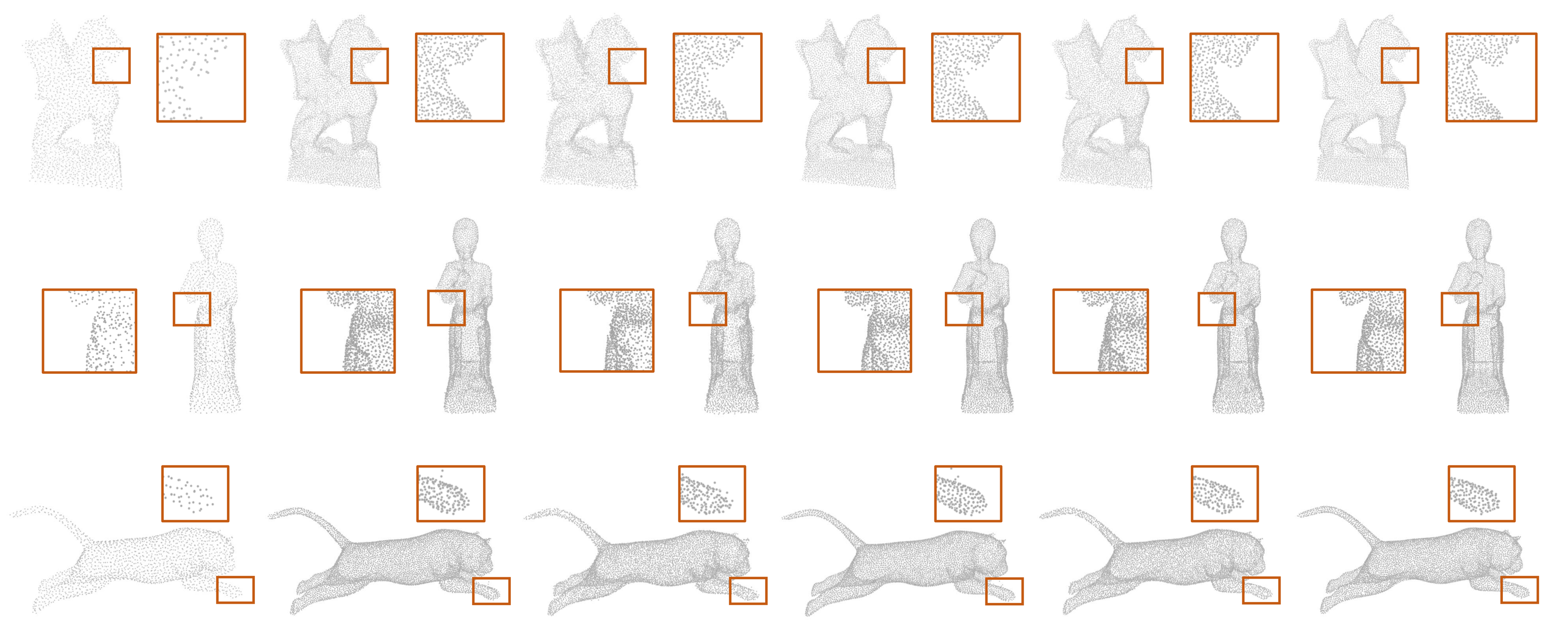
Methods	4× Upsampling			16× Upsampling		
	CD ↓	HD ↓	P2F ↓	CD ↓	HD ↓	P2F ↓
PU-Net	0.844	7.061	9.431	0.699	8.594	11.619
MPU	0.632	6.998	6.199	0.348	7.187	6.822
PU-GAN	0.483	5.323	5.053	0.269	7.127	6.306
PU-GCN	0.357	5.229	3.628	0.256	5.938	3.945
Dis-PU	0.315	4.201	4.149	**0.199**	4.716	4.249
Ours	**0.273**	**2.605**	**1.836**	0.241	**2.310**	**1.687**

Ablation Study:

models	PU-Transformer Body		PU-Transformer Tail	Results ($\times 10^{-3}$)		
	Positional Fusion	Attention Type		CD ↓	HD ↓	P2F ↓
A_1	None	SC-MSA	Shuffle	0.605	6.477	2.038
A_2	$\mathcal{G}_{geo}$	SC-MSA	Shuffle	0.558	5.713	1.751
A_3	$\mathcal{G}_{feat}$	SC-MSA	Shuffle	0.497	4.164	1.511
B_1	$\mathcal{G}_{geo}$ & $\mathcal{G}_{feat}$	SA	Shuffle	0.526	4.689	1.492
B_2	$\mathcal{G}_{geo}$ & $\mathcal{G}_{feat}$	OSA	Shuffle	0.509	4.823	1.586
B_3	$\mathcal{G}_{geo}$ & $\mathcal{G}_{feat}$	MSA	Shuffle	0.498	4.218	1.427
C_1	$\mathcal{G}_{geo}$ & $\mathcal{G}_{feat}$	SC-MSA	MLPs	1.070	8.732	2.467
C_2	$\mathcal{G}_{geo}$ & $\mathcal{G}_{feat}$	SC-MSA	DupGrid	0.485	3.966	1.380
C_3	$\mathcal{G}_{geo}$ & $\mathcal{G}_{feat}$	SC-MSA	NodeShuffle	0.505	4.157	1.404
Full	$\mathcal{G}_{geo}$ & $\mathcal{G}_{feat}$	SC-MSA	Shuffle	**0.451**	**3.843**	**1.277**

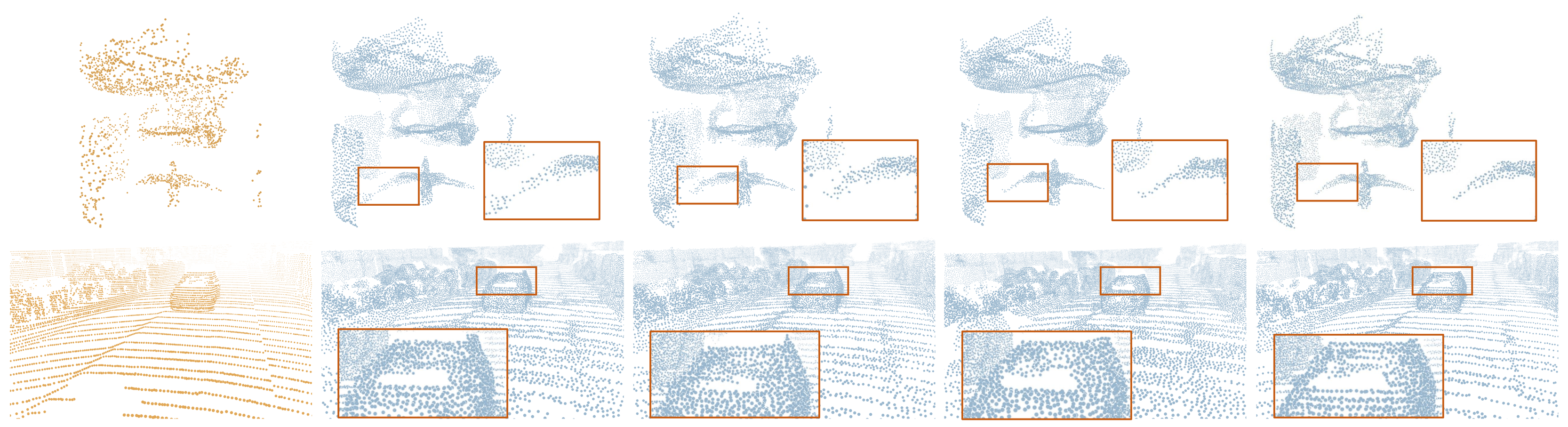
Visualization

Upsampling Synthetic Point Clouds:



(a) Input (b) PU-GAN (c) PU-GCN (d) Dis-PU (e) PU-Transformer (f) Ground-Truth

Upsampling Real-world Point Clouds:



(a) Input (b) PU-GAN (c) PU-GCN (d) Dis-PU (e) PU-Transformer

(collected from the ScanObjectNN and SemanticKITTI datasets)