Pointwise Partial Convolutions for Photo-Realistic Point Cloud Upsampling

Mihir Garimella, Prathik Naidu

{mihirg, prathikn}@stanford.edu

Introduction

- Point clouds are a popular and versatile way to represent 3D scenes
- However, raw sensor data (LiDAR, RGB-D) is often sparse and irregular, making it difficult to use with deep networks or where photo-realism matters
- **We explore upsampling both the geometric structure and photo-realistic visual information in a sparse colored 3D point cloud**
- We enable key applications in areas like VR, scene understanding, and autonomous driving

Related Work

- **Partial convolutions** (Liu et al., 2018) demonstrate impressive results for 2D image inpainting and super-resolution
- **Pointwise convolutions** (Hua et al., 2018) propose a simple convolution operator for 3D point clouds
- PU-Net (Yu et al., 2018) uses recent methods in 3D deep learning to upsample the geometric structure of single object point clouds

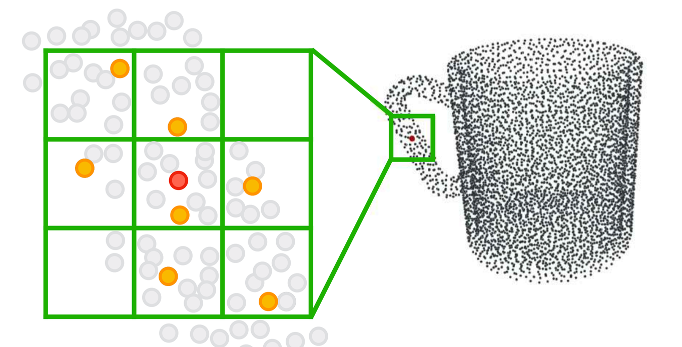
Dataset

- Matterport3D dataset contains 3D textured meshes from **90 indoor scenes**
- Preprocessing pipeline converts meshes to point clouds, creates (1.5m)^3 chunks, and extracts patches of 4096 points from each

Technical Approach

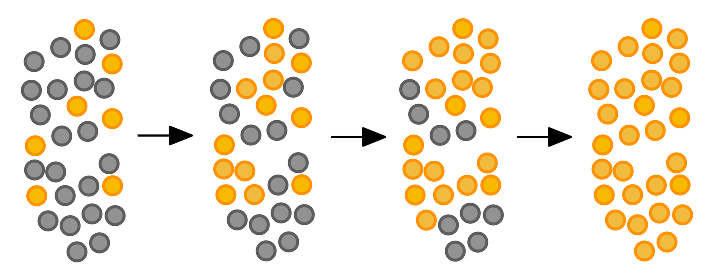
We **define a new operator** called the **pointwise partial convolution** ("pconv3p") that can operate directly on 3D point clouds with **incomplete feature information**.

1 Kernel multiplication



Center voxel grid kernel at each point 🔴, multiply feature values of valid neighbors 🟡 with kernel weights

2 3D mask update



Point becomes valid if it had at least one valid neighbor during kernel multiplication → network incrementally fills in missing feature values ⚪

Forward Pass

$$x_i^\ell = \sum_k \left(\frac{w_k}{\sum_{p_j \in \Omega_i(k)} m_j^{\ell-1}} \sum_{p_j \in \Omega_i(k), m_j \neq 0} x_j^{\ell-1} \right)$$

$$m_i^\ell = \begin{cases} 1 & \text{if } \sum_{p_j \in \Omega_i(k)} m_j^{\ell-1} \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

Backward Pass

$$\frac{\partial L}{\partial x_j^{\ell-1}} = \sum_k \sum_{i \in \Omega_j(k)} \frac{\partial L}{\partial x_i^\ell} \frac{\partial x_i^\ell}{\partial x_j^{\ell-1}}$$

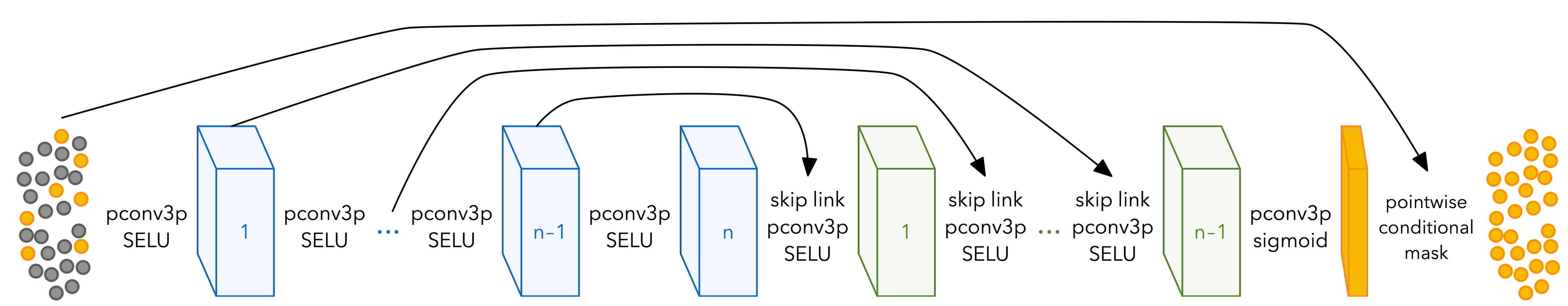
$$\frac{\partial x_i^\ell}{\partial x_j^{\ell-1}} = \sum_k \left(\frac{w_k}{\sum_{p_j \in \Omega_i(k)} m_j^{\ell-1}} \sum_{p_j \in \Omega_i(k), m_j \neq 0} 1 \right)$$

$$\frac{\partial L}{\partial w_k} = \sum_i \frac{\partial L}{\partial x_i^\ell} \frac{\partial x_i^\ell}{\partial w_k}$$

$$\frac{\partial x_i^\ell}{\partial w_k} = \frac{1}{\sum_{p_j \in \Omega_i(k)} m_j^{\ell-1}} \sum_{p_j \in \Omega_i(k), m_j \neq 0} x_j^{\ell-1}$$

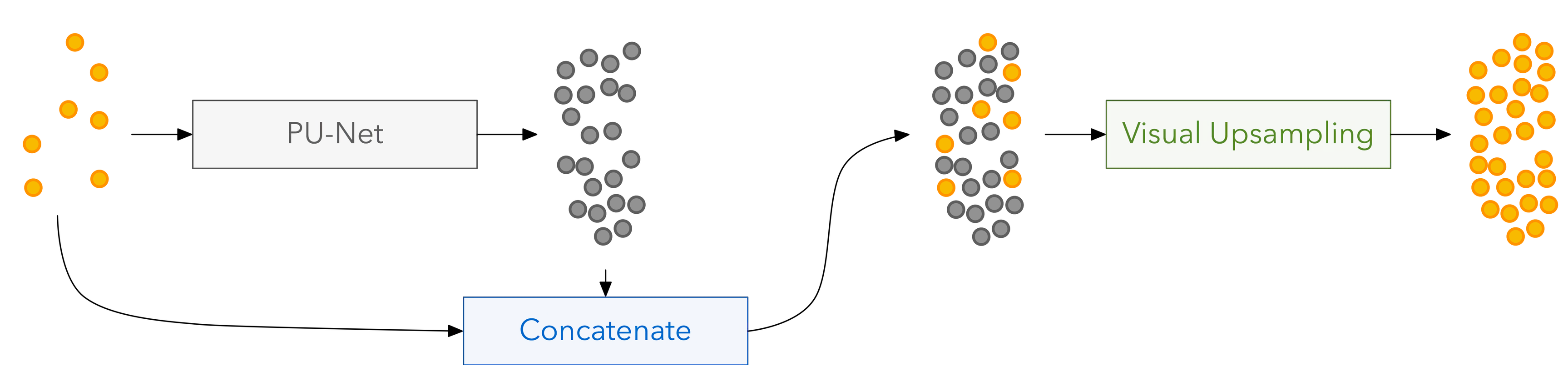
(implemented as **custom TF op**, ~800 lines of CUDA)

For photo-realistic visual upsampling, we design a **"flattened U-Net" architecture**, an **adaptation of U-Net for point clouds**. It removes the downsampling/upsampling in U-Net but preserves skip links (implemented as pointwise feature concatenation + mask addition).



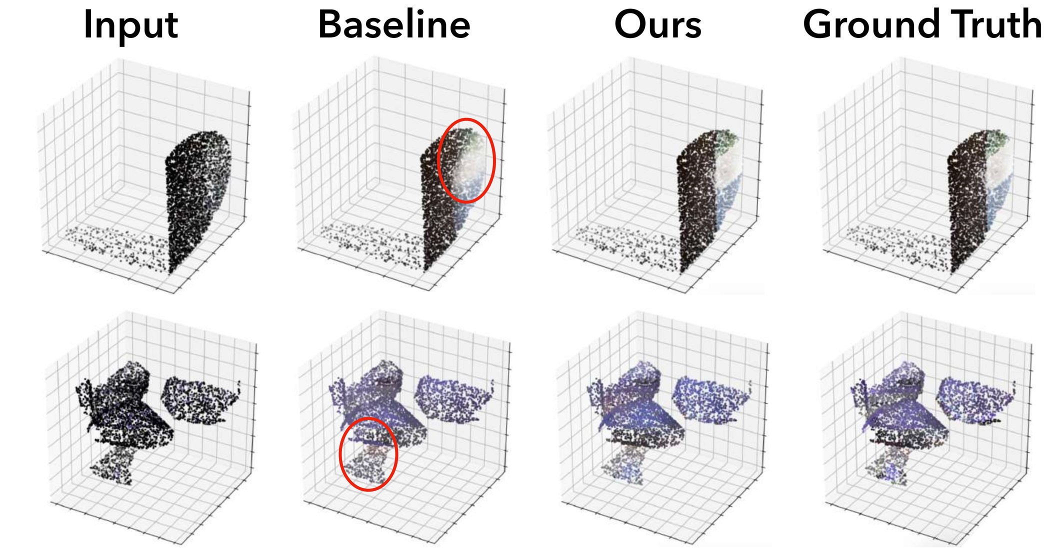
For visual upsampling task: n = 14 (28 layers), perceptual loss function (Lab color difference per point), 9 features per point at each intermediate layer

For geometric upsampling, we modify PU-Net to work on point clouds of entire scenes. Then, we combine our modified PU-Net with our visual upsampling network to create an **end-to-end pipeline** for **upsampling both geometric and visual details of point clouds.**



Results

Visual Upsampling



Randomly dropped color for **80% of points**

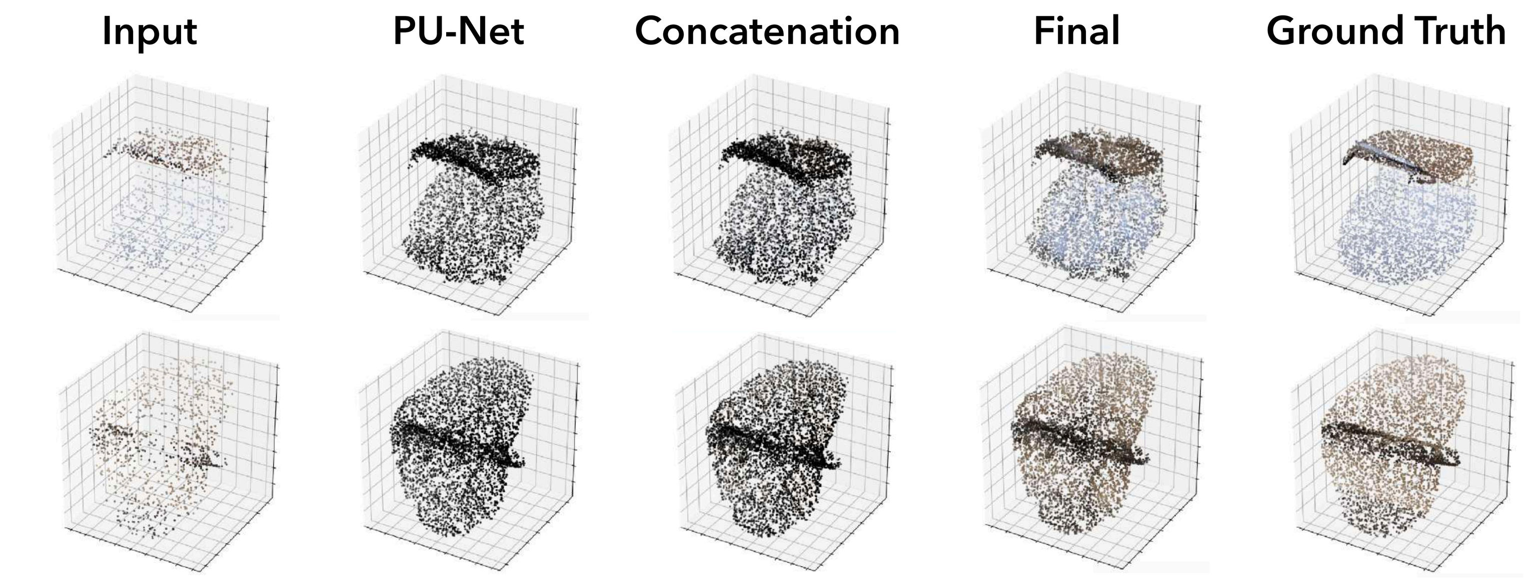
(Baseline: Weighted per-channel average of RGB colors of 9 nearest known neighbors)

MSE Evaluation

Baseline: 124.3568

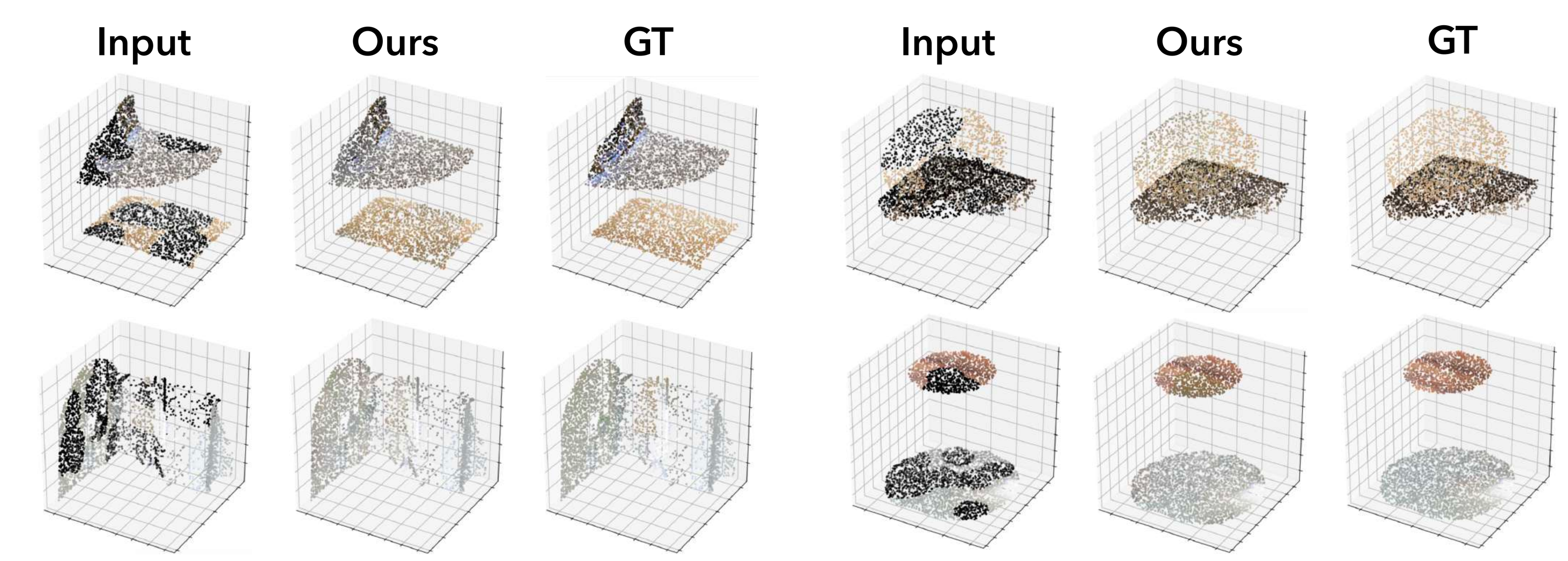
Ours: 85.2012

End-to-End Upsampling Pipeline



Conclusion

- First to explore photo-realistic upsampling in 3D
- Learned deep feature representations for 3D point clouds with novel operator and architecture
- **Future:** train geometric and visual upsampling networks together for higher accuracy, **hole filling**



After training for just 3 epochs, same operator and architecture demonstrate **impressive preliminary results** on point cloud hole filling task