Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs (SUPPLEMENTARY)

1. Model Details

Voxelization. We pre-process input point clouds with voxelization subsampling by computing per-voxel mean positions and observations over a regular 3D grid (5 cm bins for Semantic3D and 3 cm bins for S3DIS dataset). The resulting semantic segmentation is interpolated back to the original point cloud in a nearest neighbor fashion. Voxelization helps decreasing the computation time and memory requirement, and improves the accuracy of the semantic segmentation by acting as a form of geometric and radiometric denoising as well (Table 4 in main paper). The quality of further steps is practically not affected, as superpoints are usually strongly subsampled for embedding during learning and inference anyway (Subsection 3.3 in the main paper).

Geometric Partition. We set regularization strength $\mu = 0.8$ for Semantic3D and $\mu = 0.03$ for S3DIS, which strikes a balance between semantic homogeneity of superpoints and the potential for their successful discrimination (S3DIS is composed of smaller semantic parts than Semantic3D). In addition to five geometric features f (linearity, planarity, scattering, verticality, elevation), we use color information o for clustering in S3DIS due to some classes being geometrically indistinguishable, such as boards or doors.

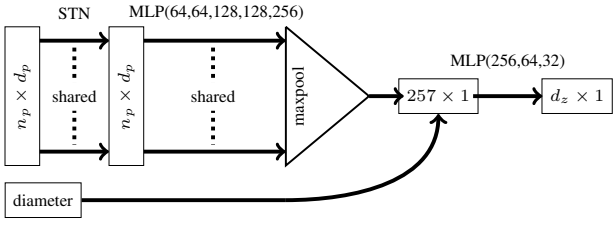

Figure 1: The PointNet embedding n_p d_p-dimensional samples of a superpoint to a d_z-dimensional vector.

PointNet. We use a simplified shallow and narrow PointNet architecture with just a single Spatial Transformer Network (STN), see Figure 1. We set $n_p = 128$ and $n_{\text{minp}} = 40$. Input points are processed by a sequence of MLPs (widths 64, 64, 128, 128, 256) and max pooled to a single vector of 256 features. The scalar metric diameter is appended and the result further processed by a sequence of MLPs (widths 256, 64, d_z=32). A residual matrix $\Phi \in \mathbb{R}^{2\times 2}$ is regressed by STN and $(I + \Phi)$ is used to transform XY coordinates of input points as the first step. The architecture of STN is a "small PointNet" with 3 MLPs (widths 64, 64, 128) before max pooling and 3 MLPs after (widths 128, 64, 4). Batch Normalization [1] and ReLUs are used everywhere. Input points have d_p=11 dimensional features for Semantic3D (position p_i, color o_i, geometric features f_i), with 3 additional ones for S3DIS (room-normalized spatial coordinates, as in past work [3]).

Segmentation Network. We use embedding dimensionality $d_z = 32$ and $T = 10$ iterations. ECC-VV is used for Semantic3D (there are only 15 point clouds even though the amount of points is large), while ECC-MV is used for S3DIS (large number of point clouds). Filter-generating network Θ is a MLP with 4 layers (widths 32, 128, 64, and 32 or 32^2 for ECC-VV or ECC-MV) with ReLUs. Batch Normalization is used only after the third parametric layer. No bias is used in the last layer. Superedges have $d_f = 13$ dimensional features, normalized by mean subtraction and scaling to unit variance based on the whole training set.

Training. We train using Adam [2] with initial learning rate 0.01 and batch size 2, *i.e.* effectively up to 1024 superpoints per batch. For Semantic3D, we train for 500 epochs with stepwise learning rate decay of 0.7 at epochs 350, 400, and 450. For S3DIS, we train for 250 epochs with steps at 200 and 230. We clip gradients within $[-1, 1]$.

2. CRF-ECC

In this section, we describe our adaptation of CRF-RNN mean field inference by Zheng *et al.* [6] for post-processing PointNet embeddings in SPG, denoted as unary potentials U_i here.

The original work proposed a dense CRF with pairwise potentials Ψ defined to be a mixture of m Gaussian kernels as $\Psi_{ij} = \mu \sum_m w_m K_m(F_{ij})$, where μ is label compati-

bility matrix, w are parameters, and K are fixed Gaussian kernels applied on edge features.

We replace this definition of the pairwise term with a Filter generating network Θ [4] parameterized with weights W_e, which generalizes the message passing and compatibility transform steps of Zheng *et al.* . Furthermore, we use superedge connectivity $\mathcal{E}$ instead of assuming a complete graph. The pseudo-code is listed in Algorithm 1. Its output are marginal probability distributions Q. In practice we run the inference for $T = 10$ iterations.

Algorithm 1 CRF-ECC

$Q_i \leftarrow \mathrm{softmax}(U_i)$
while not converged **do**
  $\hat{Q}_i \leftarrow \sum_{j|(j,i)\in\mathcal{E}} \Theta(F_{ji,\cdot}; W_e) Q_j$
  $\breve{Q}_i \leftarrow U_i - \hat{Q}_i$
  $Q_i \leftarrow \mathrm{softmax}(\breve{Q}_i)$
end while

3. Extended Ablation Studies

In this section, we present additional set of experiments to validate our design choices and present their results in Table 1.

a) Spatial Transformer Network. While STN makes superpoint embedding orientation invariant, the relationship with surrounding objects are still captured by superedges, which are orientation variant. In practice, STN helps by 4 mIoU points.

b) Geometric Features. Geometric features f_i are computed in the geometric partition step and can therefore be used in the following learning step for free. While PointNets could be expected to learn similar features from the data, this is hampered by superpoint subsampling, and therefore their explicit use helps (+4 mIoU).

c) Sampling Superpoints. The main effect of subsampling SPG is regularization by data augmentation. Too small a sample size leads to disregarding contextual information (-4 mIoU) while too large a size leads to overfitting (-2 mIoU). Lower memory requirements at training is an extra benefit. There is no subsampling at test time.

d) Long-range Context. We observe that limiting the range of context information in SPG harms the performance. Specifically, capping distances in G_{vor} to 1 m (as used in PointNet [3]) or 5 m (as used in SegCloud[1] [5]) worsens the performance of our method (even more on our Semantic 3D validation set).

e) Input Gate. We evaluate the effect of input gating (IG) for GRUs as well as LSTM units. While a LSTM unit achieves higher score than a GRU (-3 mIoU), the proposed IG reverses this situation in favor of GRU (+1 mIoU). Unlike the standard input gate of LSTM, which controls the information flow from the hidden state and input to the cell, our IG controls the input even before it is used to compute all other gates.

[1]Furthermore, SegCloud divides the inference into cubes without overlap, possibly causing inconsistencies across boundaries.

a) Spatial transf.	no	**yes**		
mIoU	58.1	62.1		
b) Geometric features	no	**yes**		
mIoU	58.4	62.1		
c) Max superpoints	256	**512**	1024	
mIoU	57.9	62.1	60.4	
d) Superedge limit	1 m	5 m	∞	
mIoU	61.0	61.3	62.1	
e) Input gate	LSTM	LSTM+IG	GRU	**GRU+IG**
mIoU	61.0	61.0	57.5	62.1
f) Regularization μ	0.01	0.02	**0.03**	0.04
# superpoints	785 010	385 091	251 266	186 108
perfect mIoU	90.6	88.2	86.6	85.2
mIoU	59.1	59.2	62.1	58.8
g) Superpoint size	1-40	40-128	128-1000	$\geq$ 1000
proportion of points	7%	14%	27%	52%

Table 1: Ablation study of design decisions on S3DIS (6-fold cross validation). Our choices in bold.

Model	mAcc	mIoU
Best	73.0	62.1
no mean offset	72.5	61.8
no offset deviation	71.7	59.3
no centroid offset	74.5	61.2
no len/surf/vol ratios	71.2	60.7
no point count ratio	72.7	61.7

Table 2: Ablation study of superedge features on S3DIS (6-fold cross validation).

f) Regularization Strength μ. We investigate the balance between superpoints' discriminative potential and their homogeneity controlled by parameter μ . We observe that the system is able to perform reasonably over a range of SPG sizes.

g) Superpoint Sizes. We include a breakdown of superpoint sizes for $\mu = 0.03$ in relation to hyperparameters $n_{\text{minp}} = 40$ and $n_p = 128$, showing that 93% of points are in embedded superpoints, and 79% in superpoints that are subsampled.

Superedge Features. Finally, in Table 2 we evaluate empirical importance of individual superedge features by removing them from Best. Although no single feature is crucial, the most being offset deviation (+3 mIoU), we remind the reader than without any superedge features the network performs distinctly worse (NoEdgeFeat, -22 mIoU).

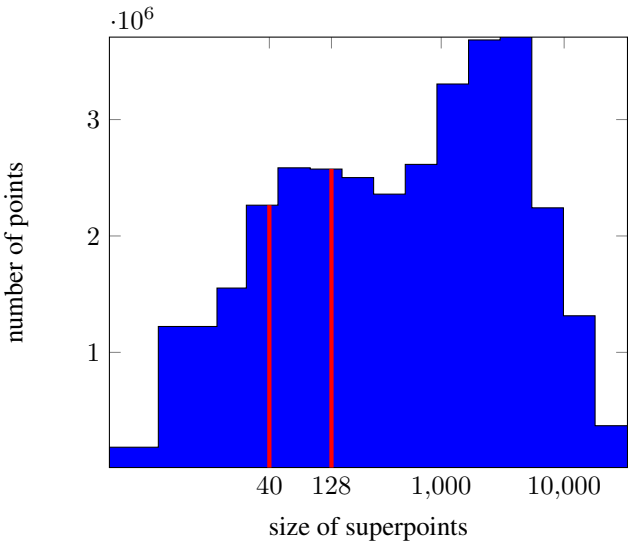

Figure 2: Histogram of points contained in superpoints of different size (in log scale) on the full S3DIS dataset. The embedding threshold n_{minp} and subsampling threshold n_p are marked in red.

4. Video Illustration

We provide a video illustrating our method and qualitative results on S3DIS dataset, which can be viewed at https://youtu.be/Ijr3kGSU_tU.

References

[1] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 1

[2] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 1

[3] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017. 1, 2

[4] M. Simonovsky and N. Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *CVPR*, 2017. 2

[5] L. P. Tchapmi, C. B. Choy, I. Armeni, J. Gwak, and S. Savarese. SEGCloud: Semantic segmentation of 3D point clouds. *arXiv preprint arXiv:1710.07563*, 2017. 2

[6] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015. 1