HAL open science

Scalable 3D Panoptic Segmentation As Superpoint Graph Clustering

Damien Robert, Hugo Raguet, Loic Landrieu

To cite this version:

Damien Robert, Hugo Raguet, Loic Landrieu. Scalable 3D Panoptic Segmentation As Superpoint Graph Clustering. 11th International Conference on 3D Vision 2024 (3DV 2024), Mar 2024, Davos, Switzerland. hal-04398319v2

HAL Id: hal-04398319

https://hal.science/hal-04398319v2

Submitted on 9 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Scalable 3D Panoptic Segmentation As Superpoint Graph Clustering

Damien Robert[1, 2] damien.robert@ign.fr
Hugo Raguet[3] hugo.raguet@insa-cvl.fr
Loic Landrieu[2,4] loic.landrieu@enpc.fr

[1]CSAI, ENGIE Lab CRIGEN, France
[2] LASTIG, IGN, ENSG, Univ Gustave Eiffel, France
[3]INSA Centre Val-de-Loire Univ de Tours, LIFAT, France
[4]LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS, France

Abstract

We introduce a highly efficient method for panoptic segmentation of large 3D point clouds by redefining this task as a scalable graph clustering problem. This approach can be trained using only local auxiliary tasks, thereby eliminating the resource-intensive instance-matching step during training. Moreover, our formulation can easily be adapted to the superpoint paradigm, further increasing its efficiency. This allows our model to process scenes with millions of points and thousands of objects in a single inference. Our method, called SuperCluster, achieves a new state-of-the-art panoptic segmentation performance for two indoor scanning datasets: 50.1 *PQ (+*7.8*) for S3DIS Area 5, and* 58.7 *PQ (+*25.2*) for ScanNetV2. We also set the first state-of-the-art for two large-scale mobile mapping benchmarks: KITTI-360 and DALES. With only* 209*k parameters, our model is over* 30 *times smaller than the best-competing method and trains up to* 15 *times faster. Our code and pretrained models are available at* https://github.com/drprojects/superpoint_transformer.

1. Introduction

Understanding large-scale 3D environments is pivotal for numerous high-impact applications such as the creation of "digital twins" of extensive industrial facilities [24, 47, 53] or even the digitization of entire cities [29, 48, 73]. Extensive and comprehensive 3D analysis methods also benefit large-scale geospatial analysis, *e.g.* for land [59, 74] or forest surveys [18, 72], as well as building inventory [66] for country-scale mapping. These problems call for scalable models that can process large point clouds with millions of 3D points, accurately predict the semantics of each point, and recover all instances of specific objects, a task referred to as 3D panoptic segmentation [26].

Most existing 3D panoptic segmentation methods focus on sparse LiDAR scans for autonomous navigation [5, 15, 81]. Given the relevance of large-scale analysis for industry and practitioners, there is surprisingly little work on large-scale 3D panoptic segmentation [71]. Although they contain non-overlapping instance labels, S3DIS [3] and ScanNet [13] only have a few panoptic segmentation entries, and KITTI-360 [39] and DALES [58] currently have none.

Figure 1. **Large-Scale Panoptic Segmentation.** We present the results of SuperCluster for the entire Area 5 of S3DIS [4] (ceiling removed for visualization) with 9.2M points (78M pre-subsampling) and 1863 true "things" objects. Our model can process such large scan in one inference on a single V100-32GB GPU in 3.3 seconds and reach a state-of-the-art PQ of 50.1.

Large-scale 3D panoptic segmentation is particularly challenging due to the scale of scenes, often featuring millions of 3D points, and the diversity in objects—ranging from a few to thousands and with extreme size variability. Current methods typically rely on large backbone networks with millions of parameters, restricting their analysis to small scenes or portions of scenes due to their high memory consumption. Furthermore, training these models requires resource-intensive procedures, such as non-maximum suppression and instance matching. These costly operations prevent the

analysis of large scenes with many points or objects. Most methods also require a pre-set limit on the number of detectable objects, introducing unnecessary complexity and the risk of missing objects in large scenes. Although recent mask-based instance segmentation methods [57] have demonstrated high performance and versatility, they fail to scale effectively to large scenes, as they predict a binary mask that covers the entire scene for each proposed instance.

To address these limitations, we present Super-Cluster, a novel approach for large-scale and efficient 3D panoptic segmentation. Our model differs from existing methods in three main ways:

- **Scalable graph clustering:** We view the panoptic segmentation task as a scalable graph clustering problem, which can be resolved efficiently at a large scale without setting the number of predicted objects in advance.
- **Local supervision:** We use a neural network to predict the parameters of the graph clustering problem and supervise with auxiliary losses that do not require an actual segmentation. This allows us to avoid resource-intensive non-maximum suppression or instance matching steps.
- **Superpoint-only segmentation:** Our approach can easily be adapted to a superpoint-based approach. Feature computation, supervision, and prediction are conducted entirely at the superpoint level and never individual points, starkly decreasing their complexity.

These features make SuperCluster particularly resource-efficient, fast, and scalable, while ensuring high precision, as shown in Figure 1. Our primary contributions are as follows:

- **Large-scale panoptic segmentation:** SuperCluster significantly improves the panoptic segmentation state-of-the-art for two indoor scanning datasets: 50.1 PQ (+7.8) on S3DIS Fold5 [4], and 58.7 PQ (+25.2) on ScanNetV2 [13]. We also set the first panoptic state-of-the-art for S3DIS 6-fold and two large-scale benchmarks (KITTI-360 [39] and DALES [58]).
- **Fast and scalable segmentation:** SuperCluster contains only 209k trainable parameters (205k in the backbone), yet outperforms networks over 30 times larger. SuperCluster inference is also as fast as the fastest instance segmentation methods and trains up to 15 times faster: 4 h for one S3DIS fold and 6 h for ScanNet.

2. Related Work

The panoptic segmentation of point clouds with millions of points has received little attention from the 3D computer vision community. In this paper, we aim to address this gap.

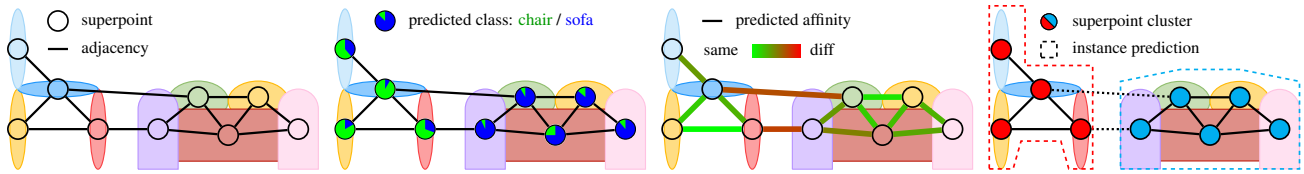
3D Panoptic and Instance Segmentation. Over the last few years, deep learning approaches for 3D point clouds have garnered considerable interest [16]. Autonomous driving, in particular, has been the focus of numerous studies, resulting in multiple proposed approaches for object detection [2, 76], as well as semantic [41, 77, 82], instance [79, 80] and panoptic segmentation [5, 15, 43, 81]. However, these methods consider sequences of sparse LiDAR acquisition, and focus on a small set of classes (pedestrians, cars).

For the panoptic segmentation of dense LiDAR point clouds, the volume of research is surprisingly small [71]. A limited number of studies have addressed the panoptic segmentation of indoor spaces using RGB-D images [45, 69]. Dense scans have primarily been used in the context of instance segmentation [19, 23, 46, 57, 64, 75]. However, while this task is related to panoptic segmentation, these methods often adopt specific strategies to maximize instance segmentation metrics [11, 71]. Moreover, many methods require specifying the maximum number of predicted instances in advance, a constraint that proves inefficient for small scenes and results in missing objects in large scenes. Additionally, when implementing a sliding-window strategy, the predicted instances must be stitched together using either heuristic techniques or resource-intensive post-processing. Lastly, the best-performing methods [46, 57] rely on a computationally expensive matching step between the predicted and true instances [8, 22, 75]. This process often depends on the Hungarian algorithm, which has cubic complexity in the number of objects and, therefore, cannot scale to large scenes.

Superpoint-Based 3D Analysis. The strategy of partitioning large 3D point clouds into groups of adjacent and homogeneous points, called superpoints, has been used successfully for point cloud oversegmentation [32, 40, 49], semantic segmentation [21, 35, 56], and object detection [14, 17]. Our approach shares similarities with some superpoint-based approaches for 3D instance segmentation [38, 60]. However, these methods are limited in scalability due to their reliance on point-wise encoders. Furthermore, the work by Sun *et al*. [60] employs a Hungarian-type instance matching scheme and allocates a binary mask to each predicted instance, covering the entire scene and drastically limiting the number of detected instances. Liang *et al*. [38] resort to quadratic-complexity agglomerative clustering to merge superpoints, and heavy postprocessing to refine and score superpoints. In contrast, our method employs a fast graph clustering approach [27, 34], which does not require any instance matching or post-processing steps.

3. Method

Our objective is to perform panoptic segmentation of a large 3D point cloud $\mathcal{P}$ with potentially numerous and broad objects. For clarity, we first present our graph clustering formulation at the point level. We then explain how our approach can be supervised solely with per-node and pre-edge targets, making its training particularly efficient. Finally, we detail how our method can be easily generalized to superpoints to further increase its scalability.



(a) Superpoint graph computation (b) Semantic prediction (c) Object-agreement prediction (d) Graph clustering

Figure 2. **SuperCluster.** We illustrate the sequence of operations of SuperCluster for a simplified scene with two objects: a chair and a sofa. **Sub-figure (a)** showcases the first stage of our process, where the point cloud is partitioned into connected *superpoints* with simple geometric shapes. In **Sub-figure (b)**, we predict a semantic class distribution for each superpoint. In **Sub-figure (c)**, we predict the object agreement for each pair of adjacent superpoints, indicating the likelihood that they belong to the same object. Finally, **Sub-figure (d)** showcases the output of a graph clustering problem which merges superpoints with compatible class distribution and object agreement while cutting edges at the transition between objects. The resulting superpoint clusters define the instances of a panoptic 3D segmentation.

Problem Statement. Consistently with the image panoptic segmentation setup [26], each point $p \in \mathcal{P}$ is associated with its position, a semantic label $\text{cls}(p) \in [1, C]$ with C the total number of classes, and an object index $\text{obj}(p) \in \mathbb{N}$. Points identified with a "thing" label (*e.g.*, chair, car) are given an index that uniquely identifies this object. Conversely, points with a "stuff" label (*e.g.*, road, wall) are assigned an index *shared by all points with the same class* within $\mathcal{P}$. Our goal is to recover the class and object index of all points in $\mathcal{P}$.

3.1. Panoptic Segmentation as Graph Clustering

We propose viewing the panoptic segmentation task as grouping adjacent points with compatible class and object predictions. We formulate this task as an optimization problem structured by a graph. Specifically, we connect the points of $\mathcal{P}$ to their K-nearest neighbors, forming a graph $\mathcal{G} = (\mathcal{P}, \mathcal{E})$ where $\mathcal{E} \subset \mathcal{P} \times \mathcal{P}$ denotes these connections.

Spatial-Semantic Regularization. We use a neural network to associate each point p with a probabilistic class prediction $x_p^{\text{class}} \in [0, 1]^C$. The architecture and supervision of this network are detailed in Section 3.2. A simple way to obtain a panoptic segmentation would be to group spatially adjacent points with the same class prediction $\arg\max_c x_{p,c}^{\text{class}}$. However, this approach neglects object structure, potentially causing two problems: erroneously merging adjacent same-class objects and unwanted object fragmentation due to the probabilistic nature of the prediction x^{class}.

To address this last issue, we aim to enforce the spatial consistency of object prediction. We introduce the signal x, defined for each point p as the channelwise concatenation of its position x_p^{pos} and its semantic prediction: $x_p = [x_p^{\text{class}}, x_p^{\text{pos}}]$. We propose to compute a piecewise-constant approximation $y^\star$ of x with an energy minimization problem regularized by the graph cut [7] between its constant components [36]. This approach aligns with well-established practices in 2D [37, 44] and 3D [28] analyses, and leads to the following optimization problem:

$$y^\star = \underset{y \in \mathbb{R}^{(C+3) \times |\mathcal{P}|}}{\arg\min} \sum_{p \in \mathcal{P}} d(x_p, y_p) + \lambda \sum_{(p,q) \in \mathcal{E}} w_{p,q} [y_p \neq y_q] \ , \quad (1)$$

where $[a \neq b] := 0$ if $a = b$ and 1 otherwise, $\lambda > 0$ is a parameter controlling the regularization strength, and $w_{p,q}$ is a nonnegative weight associated with edge (p, q), see below.

The dissimilarity function d takes into account both the spatial and semantic nature of x:

$$d(x_p, y_p) = H(y_p^{\text{class}}, x_p^{\text{class}}) + \eta \| x_p^{\text{pos}} - y_p^{\text{pos}} \|^2 \ , \quad (2)$$

where y_p^{class} is the first C coordinates of y_p and y_p^{pos} the last 3, and $\eta \geq 0$ a parameter. The term $H(x, y)$ denotes the cross-entropy between two distributions: $H(x, y) = -\sum_{c=1}^{C} y_c \log(x_c)$.

Object-Guided Edge Weights. The edge weight $w_{p,q}$ determines the cost of predicting an object transition between p and q. Designing appropriate edge weights is critical to differentiate between objects of the same class that are spatially adjacent, such as rows of chairs or cars in traffic. Edge weights should encourage cuts along probable object transitions and prevent cuts within objects.

To facilitate this, we propose to train a neural network to predict an object agreement $a_{p,q} \in [0, 1]$ for each edge (p, q) in $\mathcal{E}$. This value represents the probability that both points belong to the same object. We then determine the edge weight $w_{p,q} \in [0, \infty]$ as follows:

$$w_{p,q} = a_{p,q} / (1 - a_{p,q} + \epsilon) \ , \quad (3)$$

with $\epsilon > 0$ a fixed parameter. High values of $w_{p,q}$ discourage cuts between points p and q that are confidently predicted to belong to the same object. Conversely, a smaller $w_{p,q}$ means that cuts between edges with probable transition $a_{p,q}$ are not heavily penalized.

Graph Clustering. The constant components of the solution $y^\star$ of Eq. (1) define a clustering $\mathcal{K}$ of $\mathcal{P}$. Clusters $\mathcal{K}$ contain spatially adjacent points with compatible semantics, and their contours should follow predicted object transitions.

Converting to a Panoptic Segmentation. We can derive a panoptic segmentation from the clusters $\mathcal{K}$. For each cluster, we calculate the average point distribution of its constituent points and select the class with the highest probability. We then associate a unique object index to each cluster k predicted as a "thing" class. Likewise, we assign to each cluster classified as "stuff" an index shared by all clusters predicted as the same class. Finally, each individual point is labeled with the class and object index of its respective cluster.

Optimization. The optimization problem expressed in Eq. (1) is widely explored in the graph optimization literature. Referred to as the *generalized minimal partition problem* [34], this problem is related to the Potts models [50] and image partitioning techniques [37, 44]. We adapt the parallel ℓ_0-cut pursuit algorithm [36, 54] to the dual spatial-semantic nature of the regularized signal. The resulting algorithm is particularly scalable and can handle graphs with hundreds of millions of edges on a standard workstation. This allows us to process large point clouds in one inference without the need for tiling and instance stitching post-processing.

3.2. Local Supervision

A major benefit of our approach is that it can be entirely supervised with local auxiliary tasks: all losses described in this section are sums of simple functions depending on one or two points at the time. In particular, we bypass the computationally expensive step of matching true instances with their predicted counterparts.

Recall from Section 3.1 that we can obtain a panoptic segmentation by predicting the parameters of a graph clustering problem: the semantic predictions x_p^{class} and the object agreements $a_{p,q}$. These quantities are both derived from a common pointwise embedding $\{e_p\}_{p\in\mathcal{P}}$, computed by a neural network.

Predicting Semantics. We predict the class distribution $x_p^{\text{class}} = \text{softmax}(\phi^{\text{class}}(e_p))$ with ϕ^{class} a Multi-Layer Perceptron (MLP). This distribution is supervised by its cross-entropy against the true class $\text{cls}(p)$:

$$\mathcal{L}_p^{\text{class}} = H(x_p^{\text{class}}, \mathbf{1}(\text{cls}(p))) \ , \tag{4}$$

with $\mathbf{1}(c) \in \{0,1\}^C$ the one-hot embedding of class c.

Predicting Object Agreement. To predict the object agreement $a_{p,q}$ between two adjacent points $(p,q) \in \mathcal{E}$, we employ an MLP ϕ^{object} whose input is a symmetric combination of the points' embedding vectors:

$$a_{p,q} = \text{sigmoid}\left(\phi^{\text{object}}\left([(e_p+e_q)/2, \mid e_p - e_q \mid]\right)\right) \ , \tag{5}$$

where $|\cdot|$ refers to the termwise absolute value. The true object agreement $\hat{a}_{p,q}$ is assigned the value of 1 if $\text{obj}(p) = \text{obj}(q)$ and 0 otherwise. The prediction of $a_{s,t}$ can be seen as *a binary edge classification problem* as inter- and intra-object edges [32], and is supervised with the cross-entropy between true and predicted object agreements:

$$\mathcal{L}_{p,q}^{\text{object}} = H(\text{Bern}(a_{p,q}), \text{Bern}(\hat{a}_{p,q})) \ , \tag{6}$$

where $\text{Bern}(a)$ denote the Bernoulli distribution parametrized by $a \in [0,1]$.

Loss Function. We combine the two losses above into a single objective $\mathcal{L}$:

$$\mathcal{L} = \frac{1}{\mid \mathcal{P} \mid} \sum_{p\in\mathcal{P}} \mathcal{L}_p^{\text{class}} + \frac{1}{\mid \mathcal{E} \mid} \sum_{(p,q)\in\mathcal{E}} \mathcal{L}_{p,q}^{\text{object}} \ , \tag{7}$$

with $\mid \mathcal{E} \mid$ and $\mid \mathcal{P} \mid$ the total number of edges and 3D points, respectively.

3.3. Extension to Superpoints

In this section, we discuss the extension of our method to a superpoint-based approach for enhanced scalability.

Motivation. We aim to design a panoptic segmentation method that can scale to large 3D point clouds. While the formulation presented in the previous section is advantageous, it still requires computing embeddings and predictions for each individual point, which can be memory intensive and limits the volume of data that can be processed simultaneously. We propose to group adjacent points with similar local geometry and color into *superpoints*, and to only compute embeddings and predictions for superpoints and not individual points. By doing so, we drastically reduce the computational and memory requirements of our method, enabling it to handle larger 3D point clouds at once.

Computing Superpoints. We partition the point cloud $\mathcal{P}$ into a set of non-overlapping superpoints $\mathcal{S}$. We use the partition method implemented by Robert *et al.* in SPT [56], which defines superpoints as the constant components of a low-surface piecewise constant approximation of geometric and radiometric point features.

Although the superpoints $\mathcal{S}$ form a high-purity oversegmentation of $\mathcal{P}$, some superpoints can span multiple objects. To account for this, we associate each superpoint s with its *majority-object* $\text{obj}(s)$ defined as the most common object index within its points: $\text{obj}(s) = \text{mode}\{\text{obj}(p) \mid p \in s\}$. Likewise, we define $\text{cls}(s) = \text{mode}\{\text{cls}(p) \mid p \in s\}$.

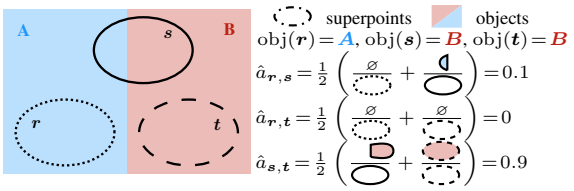


Figure 3. **Superpoint Object Agreement.** We compute for each pair of adjacent superpoint (s, t) an object agreement score $\hat{a}_{s,t}$. This value is defined by the average overlap ratio between s and t and their majority-objects $\text{obj}(t)$ and $\text{obj}(s)$, see Eq. (8).

Adapting Graph Clustering. Our clustering step can be directly adapted by substituting the point set $\mathcal{P}$ with the superpoint set $\mathcal{S}$, and defining the graph $\mathcal{G}$ by connecting superpoints with adjacent points following the approach of SPT [56]. We replace the point position x_p^{pos} by the coordinates of the superpoints' centroids x_s^{pos}. All other steps remain unchanged.

Superpoint Embedding. We use a superpoint-embedding network to compute the superpoint features e_s for $s \in \mathcal{S}$. We employ the Superpoint Transformer model [56] for its efficiency and ability to leverage large spatial context. More details on this design choice are provided in the Appendix.

Superpoint Semantic Supervision. We supervise the semantic superpoint prediction x_s^{class} with Eq. (4) where we replace $\text{cls}(p)$ with $\text{cls}(s)$.

Superpoint Object Agreement Supervision. While the true object agreement $\hat{a}_{p,q}$ between two points is binary, the agreement between superpoints spans a continuum. As illustrated in Figure 3, we quantify this agreement as:

$$\hat{a}_{s,t} = \frac{1}{2}\left(\frac{|\, s \cap \mathcal{P}_{|\text{obj}(t)} \,|}{|\, s \,|} + \frac{|\, t \cap \mathcal{P}_{|\text{obj}(s)} \,|}{|\, t \,|}\right), \quad (8)$$

where $\mathcal{P}_{|o} := \{p \in \mathcal{P} \mid \text{obj}(p) = o\}$ is the set of points of $\mathcal{P}$ with the object index o, and $|\, s \,|$ is the count of 3D points in s. We can now supervise the predicted object agreement $a_{s,t}$ with Eq. (6) unchanged.

4. Experiments

We first present the datasets and metrics used for evaluation in Section 4.1, then our main results and their analysis in Section 4.2, and finally an ablation study in Section 4.3.

4.1. Datasets and Metrics

Datasets. We present the four datasets used in this paper.

- **S3DIS** [4]. This indoor scanning dataset consists of 274 million points distributed across 271 rooms in 6 building floors—or areas. We do not use the provided room partition, as they require significant manual processing and may not translate well to other environments such as open offices, industrial sites, or mobile mapping. Instead, we merge all rooms in the same area and treat each floor as one single large-scale acquisition [9, 61].
  We follow the standard evaluation protocol, using the area 5 as a test set and implementing 6-fold cross-validation. In line with Xiang *et al.*'s [71] proposal, we treat all 13 classes as "thing". However, certain classes, such as *walls*, *ceiling*, and *floors*, are susceptible to arbitrary division due to room splitting, making their evaluation somewhat inconsistent. As a result, we also present panoptic metrics in which these three classes are considered as "stuff".
- **ScanNet** [13]. This dataset consists of 237M 3D points organized in 1501 medium-scale indoor scenes. We evaluate SuperCluster on ScanNet's open test set, as the hidden test set is not evaluated for panoptic segmentation. We use for "things" the class evaluated in the instance segmentation setting: *bathtub*, *bed*, *bookshelf*, *cabinet*, *chair*, *counter*, *curtain*, *desk*, *door*, *other furniture*, *picture*, *refrigerator*, *shower curtain*, *sink*, *sofa*, *table*, *toilet*, and *window*. The *walls* and *floor* class are designated as "stuff".
- **KITTI-360** [39]. Containing over 100k mobile mapping laser scans from an outdoor urban environment, we utilize the *accumulated point clouds* format, which aggregates multiple sensor rotations to form 300 extensive scenes with an average of more than 3 million points. We train on 239 scenes and evaluate it on the remaining 61. *Building* and *cars* classes are treated as "thing" while the remaining 13 are classified as "stuff".
- **DALES** [63]. This large-scale aerial scan data set spans 10 km^2 and contains 500 millions of 3D points organized along 40 urban and rural scenes, of which we use 12 for evaluation. The "thing" classes are *buildings*, *cars*, *trucks*, *power lines*, *fences*, and *poles*. *Ground* and *vegetation* evaluated as "stuff".

Table 1. **S3DIS Area 5.** We report the semantic (*SS*) and panoptic segmentation results of the top-performing semantic segmentation methods on the fifth area of S3DIS, as well as panoptic segmentation approaches implemented by Xiang *et al.* [71]. We provide two panoptic metrics by considering all classes as "things" (*PS - no "stuff"*) and with *wall*, *ceiling* and *floor* as "stuff" (*PS*).

	Size	SS	PS - no "stuff"			PS		
	$\times 10^6$	mIoU	PQ	RQ	SQ	PQ	RQ	SQ
Semantic segmentation models								
SPT [56]	**0.21**	68.9	-	-	-	-	-	-
Point Trans.[78]	7.8	70.4	-	-	-	-	-	-
PointNeXt-XL [52]	41.6	71.1	-	-	-	-	-	-
Strat. Trans. [31, 68]	8.0	**72.0**	-	-	-	-	-	-
Panoptic segmentation models								
Xiang *et al.* [71]	0.13							
+ PointNet++ [51]	+3.0	58.7	24.6	32.6	68.2	-	-	-
+ Minkowski [12]	+37.9	63.8	39.2	48.0	74.9	-	-	-
+ KPConv [61]	+14.1	65.3	41.8	51.5	74.7	-	-	-
PointGroup [23] in [71]	7.7	64.9	42.3	52.0	74.7	-	-	-
SuperCluster (ours)	**0.21**	68.1	**50.1**	**60.1**	**76.6**	**58.4**	**68.4**	**77.8**

Table 2. **S3DIS 6-Fold.** We report the 6-Fold cross-validated semantic and panoptic segmentation results on S3DIS. No panoptic methods were evaluated in this setting to the best of our knowledge.

	Size	SS	PS - no "stuff"			PS		
	$\times 10^6$	mIoU	PQ	RQ	SQ	PQ	RQ	SQ
Semantic segmentation models								
DeepViewAgg [55]	41.2	74.7	-	-	-	-	-	-
Strat. Trans. [31, 68]	8.0	74.9	-	-	-	-	-	-
PointNeXt-XL [52]	41.6	74.9	-	-	-	-	-	-
SPT [56]	**0.21**	**76.0**	-	-	-	-	-	-
Panoptic segmentation models								
SuperCluster (ours)	**0.21**	75.3	**55.9**	**66.3**	**83.8**	**62.7**	**73.2**	**84.8**

Evaluation Metrics. Recognition Quality (RQ) assesses object identification and classification. Segmentation Quality (SQ) evaluates the alignment between target and predicted object segmentations. Panoptic Quality (PQ) combines both measures. We also compute the semantic segmentation performance by associating points with their superpoint's class and computing the mean Intersection over Union (mIoU).

Model Parameterization. Our backbone for the S3DIS and DALES datasets is a small SPT-64 model [56] with 205k parameters. We use a larger SPT-128 (791k parameters) for KITTI-360 and a slightly modified model for ScanNet (1M) parameters. SuperCluster adds two small MLP ϕ^{class} and ϕ^{object} for a total of 4.4k parameters for S3DIS and DALES, and 8.8k parameters for KITTI-360 and ScanNet.

Our training batches are composed of 4 randomly sampled cylinders with a radius of 7 m for S3DIS, 50 m for KITTI and DALES, and entire scenes for ScanNet. Partition parameters are adjusted so that $\mathcal{S}/\mathcal{P} \sim 30$ for S3DIS, DALES, and KITTI-360, and 20 for ScanNet.

We can tune the graph clustering parameters after training to optimize the PQ on the training set: λ in Eq. (1), η in Eq. (2), and ϵ in Eq. (3). As the clustering step is particularly efficient, we can evaluate tens of values in a few minutes. More details are provided in the Appendix.

4.2. Results and Analysis

We compare our method quantitatively with state-of-the-art models in Table 1 to 5. We also report a runtime analysis in Table 6 and qualitative illustrations in Figure 4.

S3DIS. We report in Table 1 the performance of our algorithm evaluated for Area 5 of the S3DIS dataset. Compared to several baselines for panoptic segmentation, our model shows a notable improvement with a PQ boost of $+7.8$ points and a mIoU increase of $+3.2$ points. Remarkably, our model is more than 33 times smaller than the highest performing model. Furthermore, we compute panoptic metrics by treating *wall*, *ceiling*, and *floor* as "stuff" classes to account for their arbitrary boundaries. To the best of our knowledge, we are the first to report panoptic results evaluated with 6-Fold cross-validation on S3DIS, in Table 2.

Table 3. **ScanNetv2 Val.** We report the Semantic Segmentation (*SS*) and Panoptic Segmentation (*PS*) performance for various methods on the open test set of ScanNetv2. † code and models unavailable.

	Size	SS	PS		
	$\times 10^6$	mIoU	PQ	RQ	SQ
Semantic segmentation models					
KPConv [61]	14.1	69.2	-	-	-
Point Trans [78]	7.8	70.6	-	-	
Point Trans. v2 [70]	11.3	**75.4**	-	-	-
OctFormer [67]	44.0	**75.7**	-	-	-
Panoptic segmentation models					
SceneGraphFusion [65, 69]	2.9	-	31.5	42.2	72.9
PanopticFusion [45]	†	-	33.5	45.3	73.0
SuperCluster (ours)	**1.0**	66.1	**58.7**	**69.1**	**84.1**

Table 4. **KITTI-360.** We report the Semantic Segmentation (*SS*) and Panoptic Segmentation (*PS*) performance for various methods on the open test set of KITTI-360. No panoptic methods were evaluated on this dataset to the best of our knowledge.

	Size	SS	PS		
	$\times 10^6$	mIoU	PQ	RQ	SQ
Semantic segmentation models					
Minkowski [12]	37.9	58.3	-	-	-
DeepViewAgg [78]	41.2	62.1	-	-	-
SPT [56]	**0.78**	**63.5**	-	-	-
Panoptic segmentation models					
SuperCluster (ours)	**0.79**	62.1	**48.3**	**58.4**	**75.1**

Despite its smaller size, our model achieves high semantic segmentation performance, reaching near state-of-the-art performance on the Area 5 and 6-fold evaluations.

ScanNet. As shown in Table 3, SuperCluster significantly improves the state-of-the-art of panoptic segmentation by 25.2 PQ points. Our model does not perform as well as large networks designed for semantic segmentation but provides decent results with a small backbone of only 1M parameters.

DALES and KITTI-360. SuperCluster is the first capable of processing the large tiles of the DALES and KITTI-360 datasets, thus establishing the first panoptic state-of-the-art for these datasets given in Table 4 and Table 5.

Inference and Training Speed. In Table 6, we compare the inference speed of our approach with state-of-the-art instance and panoptic segmentation algorithms. As we use a 1080Ti GPU to replicate the setting used to measure most of the approaches' speed (a Titan-X), the values are not entirely comparable. Still, our model is on par with the fastest methods and offers superior scalability.

Table 5. **DALES.** We report the Semantic Segmentation (*SS*) and Panoptic Segmentation (*PS*) performance for various methods on the open test set of DALES. No panoptic methods were evaluated on this dataset to the best of our knowledge.

	Size	SS	PS		
	$\times 10^6$	mIoU	PQ	RQ	SQ
Semantic segmentation models					
ConvPoint [6]	4.7	67.4	-	-	-
PointNet++ [51]	3.0	68.3	-	-	-
SPT [56]	**0.21**	79.6	-	-	-
KPConv [61]	14.1	**81.1**	-	-	-
Panoptic segmentation models					
SuperCluster (ours)	**0.21**	77.3	**61.2**	**68.6**	**87.1**

Table 6. **Runtime.** We compare the speed of our model to various instance and panoptic segmentation models. We report the time spent in the backbone network (first number) and performing panoptic segmentation (second number) on ScanNet Val. scans. $\star$ optional CRF post-processing.

	Hardware	Runtime in ms
Instance segmentation methods		*average per scan on Val*
PointGroup [23]	Titan X	452 = 128 + 324
SoftGroup [64]	Titan X	345 = 152 + 148
HAIS [10]	Titan X	339 = 154 + 185
Mask3D [57]	Titan X	339
ISBNet [46]	Titan X	**237** = 152 + 85
SuperCluster (ours)	1080Ti	**238** = 193 + 45
Panoptic segmentation methods		*for* `scene0645_01`
PanopticFusion [45]	2×1080Ti	485 = 317 + 168 (+ 4500*)
SuperCluster (ours)	1080Ti	**482** = 376 + 106

None of the reported runtimes include the method's pre-processing times. Thanks to SPT's efficiency, our entire pre-processing, including the superpoint partition, is faster or equivalent to all existing 3D segmentation methods [56].

Our model can be trained in an amount of time comparable to its backbone SPT for semantic segmentation [55]. One fold of S3DIS takes just under 4 hours, which is substantially quicker than most existing semantic, instance, or panoptic segmentation models. For example, PointTransformer [78] trains for 63 h and Stratified Transformer [31] 216 GPU-h. SuperCluster trains on 6 h on ScanNet, compared to 78 h for Mask3D [57] and 20 h for ISBNet [46].

4.3. Ablation Study

We evaluate the impact of our design choice by performing several experiments whose results are given in Table 7. More experiments are provided in the Appendix.

Constant Edge Weights. Replacing all edge weights with a constant value of 1 yields a drop of 4.2 PQ points. This experiment shows the benefit of learning object transitions.

Table 7. **Ablation Study.** We report the performance of different experiments on S3DIS Area 5 with *wall*, *ceiling* and *floor* as "stuff".

Experiment	PS		
	PQ	RQ	SQ
Best Model	58.4	68.4	77.8
Constant Edge Weights	54.2	64.2	76.6
Offset Prediction	57.1	65.2	77.1
Smaller Superpoints	56.6	64.6	78.6
Superpoint Oracle	93.4	99.7	93.7
Clustering Oracle	83.6	91.7	90.8

Offset Prediction. Several *bottom-up* [20] segmentation approaches [17, 23, 30, 71] propose clustering points by shifting their positions towards the predicted position of the object centroid. To reproduce this strategy, we adjust the position of x_s^{pos} in x along a vector that predicts the center of the majority object. We supervise this prediction with the L1 loss, as it produced the best results among several alternatives that we examined. Despite our efforts, this approach did not improve the results: -1.3 PQ points. We attribute this to the size diversity of objects observed in large-scale scenes (corridors, buildings), resulting in an unstable prediction.

Smaller Superpoints. To demonstrate the benefits of using superpoints, we consider a finer partition with $\mathcal{S}/\mathcal{P} \sim 15$ instead of 30. This requires training with smaller 3 m cylinders instead of 7, decreasing the performance by -1.8 PQ points. This result illustrates that the superpoint paradigm is central to our approach.

Superpoint Oracle. Using superpoints greatly improves the efficiency and scalability of SuperCluster. However, since the predictions are made at the superpoint level and never for individual 3D points, the semantic and object purity of the superpoints can restrict the model's performance. To evaluate this impact, we define the superpoint oracle, which assigns to each superpoint s the class and index of its majority object $\text{obj}(s)$. The resulting performance provides an upper bound of what our model could potentially achieve. The high performance of this oracle (93.4 PQ) indicates that very little precision is lost by working with superpoints.

Clustering Oracle. In a similar vein, we calculate the upper bound of our model by computing the results of the graph clustering with perfect network predictions: x^{class} is set as the one-hot-encoding of the class of the majority object, and the object agreement is set to its true value: $a_{p,q} = \hat{a}_{p,q}$. The performance of this oracle (83.6 PQ) shows that our scalable clustering formulation does not significantly compromise the model's precision in its current regime.

Limitations. Our approach, while efficient, is not devoid of constraints. The functional minimized in Eq. (1) is non-continuous and nondifferentiable, which hinders the compu-

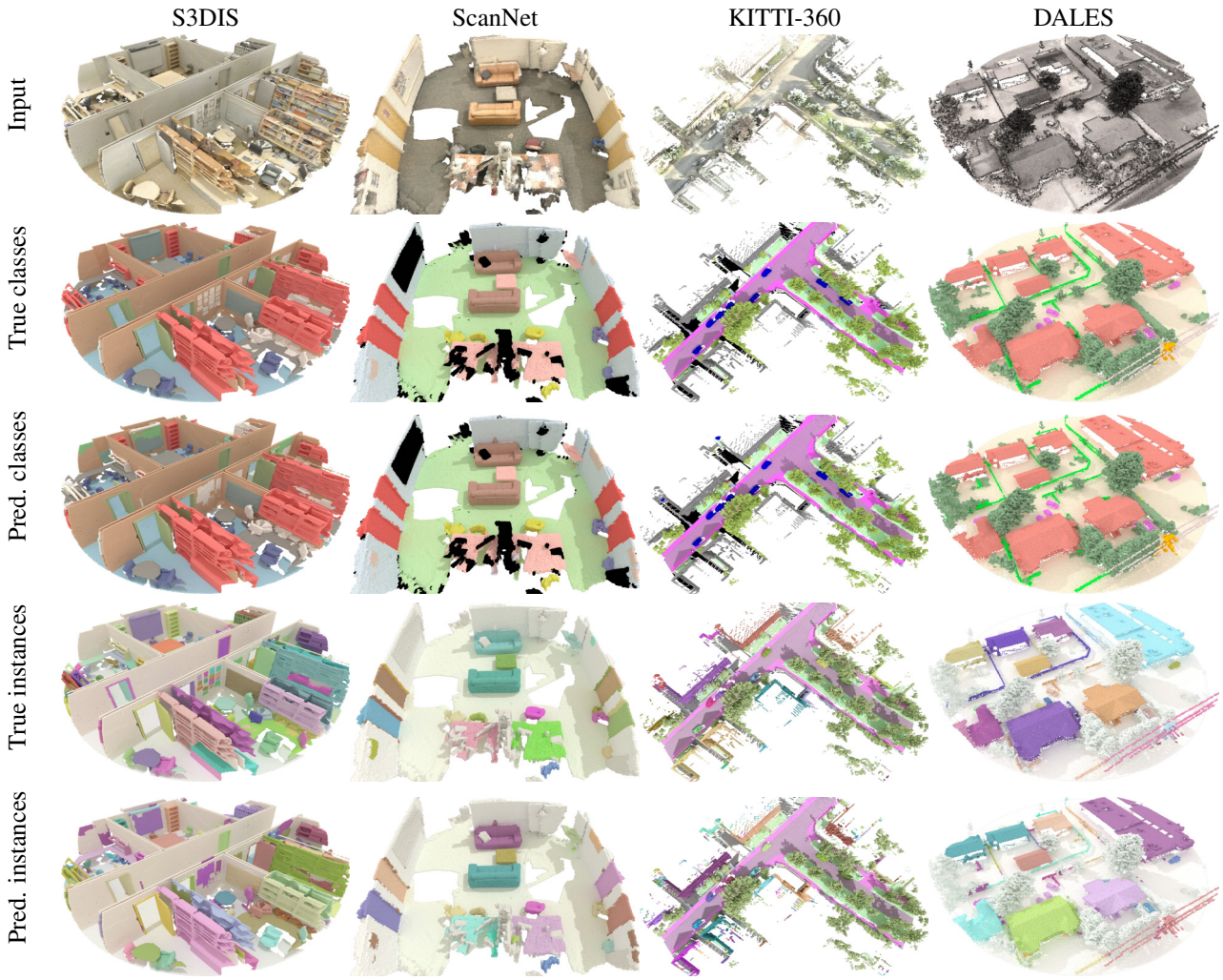


Figure 4. **Qualitative Results.** We present the panoptic predictions of our model for the four considered datasets. The scenes' size corresponds to a single batch item during training. "Stuff" classes are represented with a lower opacity.

tation of gradients and the possibility of learning the partition. Nevertheless, this aspect lends itself to the speed and simplicity of our training process. Although our approach can run on diverse acquisition setups, the superpoint partition is sensitive to low point density and may fail for sparse scans as visible on the edge of some KITTI-360 acquisitions.

We use a lightweight SPT network to ensure maximum scalability. This network, while expressive, is not the most powerful existing architecture. There is a potential for improved results using more resource-intensive networks.

Since our model does not improve the semantic segmentation performance of the backbone model (SPT) in any of our experiments, we conclude that our local panoptic supervision scheme does not help semantic segmentation.

5. Conclusion

In this paper, we introduced SuperCluster, a novel approach for 3D panoptic segmentation of large-scale point clouds. We propose a new formulation of this task as a scalable graph clustering problem, bypassing some of the most compute-intensive steps of current panoptic segmentation methods. Our results across multiple benchmarks, including S3DIS, ScanNet, KITTI-360, and DALES, demonstrate that our model achieves state-of-the-art performance while being significantly smaller, scalable, and easier to train.

Despite the considerable industrial applications, large-scale panoptic segmentation has been relatively unexplored by the 3D computer vision community. We hope that our positive results and the state-of-the-art we established on new datasets and settings will encourage the development of future panoptic approaches for large-scale 3D scans.

Acknowledgements. This work was funded by ENGIE Lab CRIGEN and ANR project READY3D ANR-19-CE23-0007. It used HPC resources from GENCI–IDRIS (Grant 2023-AD011013388R1).

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *TPAMI*, 2012. 12

[2] Simegnew Yihunie Alaba and John E Ball. A survey on deep-learning-based LiDAR 3D object detection for autonomous driving. *Sensors*, 2022. 2

[3] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3D semantic parsing of large-scale indoor spaces. *CVPR*, 2016. 1

[4] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3D Scene Graph: A structure for unified semantics, 3D space, and camera. *ICCV*, 2019. 1, 2, 5

[5] Mehmet Aygun, Aljosa Osep, Mark Weber, Maxim Maximov, Cyrill Stachniss, Jens Behley, and Laura Leal-Taixé. 4D panoptic LiDAR segmentation. *CVPR*, 2021. 1, 2

[6] Alexandre Boulch. ConvPoint: Continuous convolutions for point cloud processing. *Computers & Graphics*, 2020. 7

[7] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *TPAMI*, 2001. 3

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *ECCV*, 2020. 2

[9] Thomas Chaton, Nicolas Chaulet, Sofiane Horache, and Loic Landrieu. Torch-Points3D: A modular multi-task framework for reproducible deep learning on 3D point clouds. *3DV*, 2020. 5

[10] Shaoyu Chen, Jiemin Fang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Hierarchical aggregation for 3D instance segmentation. *ICCV*, 2021. 7

[11] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. *CVPR*, 2022. 2

[12] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal convnets: Minkowski convolutional neural networks. *CVPR*, 2019. 5, 6, 12

[13] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. *CVPR*, 2017. 1, 2, 5

[14] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3D-MPA: Multi-proposal aggregation for 3D semantic instance segmentation. *CVPR*, 2020. 2

[15] Whye Kit Fong, Rohit Mohan, Juana Valeria Hurtado, Lubing Zhou, Holger Caesar, Oscar Beijbom, and Abhinav Valada. Panoptic nuScenes: A large-scale benchmark for LiDAR panoptic segmentation and tracking. *Robotics and Automation Letters*, 2022. 1, 2

[16] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3D point clouds: A survey. *TPAMI*, 2020. 2

[17] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. OccuSeg: Occupancy-aware 3D instance segmentation. *CVPR*, 2020. 2, 7

[18] Marius Hauglin, Johannes Rahlf, Johannes Schumacher, Rasmus Astrup, and Johannes Breidenbach. Large scale mapping of forest attributes using heterogeneous sets of airborne laser scanning and national forest inventory data. *Forest Ecosystems*, 2021. 1

[19] Tong He, Chunhua Shen, and Anton van den Hengel. DyCo3D: Robust instance segmentation of 3D point clouds through dynamic convolution. *CVPR*, 2021. 2

[20] Yong He, Hongshan Yu, Xiaoyan Liu, Zhengeng Yang, Wei Sun, Yaonan Wang, Qiang Fu, Yanmei Zou, and Ajmal Mian. Deep learning based 3D segmentation: A survey. *arXiv preprint arXiv:2103.05423*, 2021. 7

[21] Le Hui, Jia Yuan, Mingmei Cheng, Jin Xie, Xiaoya Zhang, and Jian Yang. Superpoint network for point cloud oversegmentation. *ICCV*, 2021. 2, 12

[22] Ding Jia, Yuhui Yuan, Haodi He, Xiaopei Wu, Haojun Yu, Weihong Lin, Lei Sun, Chao Zhang, and Han Hu. DETRs with hybrid matching. *CVPR*, 2023. 2

[23] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. PointGroup: Dual-set point grouping for 3D instance segmentation. *CVPR*, 2020. 2, 5, 7, 13

[24] Yuchen Jiang, Shen Yin, Kuan Li, Hao Luo, and Okyay Kaynak. Industrial applications of digital twins. *Philosophical Transactions of the Royal Society A*, 2021. 1

[25] Xin Kang, Chaoqun Wang, and Xuejin Chen. Region-enhanced feature learning for scene semantic segmentation. *arXiv preprint arXiv:2304.07486*, 2023. 12

[26] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollar. Panoptic segmentation. *CVPR*, 2019. 1, 3, 13

[27] Vladimir Kolmogorov and Ramin Zabin. What energy functions can be minimized via graph cuts? *TPAMI*, 2004. 2

[28] Patrick Labatut, J-P Pons, and Renaud Keriven. Robust and efficient surface reconstruction from range data. *Computer Graphics Forum*, 2009. 3

[29] Florent Lafarge and Clément Mallet. Creating large-scale city models from 3D-point clouds: A robust approach with hybrid representation. *ICCV*, 2012. 1

[30] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and Martin R Oswald. 3D instance segmentation via multi-task metric learning. *ICCV*, 2019. 7

[31] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3D point cloud segmentation. *CVPR*, 2022. 5, 6, 7, 12

[32] Loic Landrieu and Mohamed Boussaha. Point cloud oversegmentation with graph-structured deep metric learning. *CVPR*, 2019. 2, 4, 12

[33] Loic Landrieu and Guillaume Obozinski. Cut Pursuit: Fast algorithms to learn piecewise constant functions. *AISTATS*, 2016. 12

[34] Loic Landrieu and Guillaume Obozinski. Cut Pursuit: Fast algorithms to learn piecewise constant functions on general weighted graphs. *SIAM Journal on Imaging Sciences*, 2017. 2, 4

[35] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. *CVPR*, 2018. 2, 12

[36] Loic Landrieu, Hugo Raguet, Bruno Vallet, Clément Mallet, and Martin Weinmann. A structured regularization framework for spatially smoothing semantic labelings of 3D point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2017. 3, 4

[37] Yvan G Leclerc. Constructing simple stable descriptions for image partitioning. *IJCV*, 1989. 3, 4

[38] Zhihao Liang, Zhihao Li, Songcen Xu, Mingkui Tan, and Kui Jia. Instance segmentation in 3D scenes using semantic superpoint tree networks. *CVPR*, 2021. 2

[39] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D. *TPAMI*, 2022. 1, 2, 5

[40] Yangbin Lin, Cheng Wang, Dawei Zhai, Wei Li, and Jonathan Li. Toward better boundary preserved supervoxel segmentation for 3D point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2018. 2

[41] Romain Loiseau, Mathieu Aubry, and Loïc Landrieu. Online segmentation of LiDAR sequences: Dataset and algorithm. *ECCV*, 2022. 2

[42] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. *ICML*, 2013. 13

[43] Rohit Mohan and Abhinav Valada. EfficientPS: Efficient panoptic segmentation. *IJCV*, 2021. 2

[44] David Bryant Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 1989. 3, 4

[45] Gaku Narita, Takashi Seno, Tomoya Ishikawa, and Yohsuke Kaji. PanopticFusion: Online volumetric semantic mapping at the level of stuff and things. *IROS*, 2019. 2, 6, 7

[46] Tuan Duc Ngo, Binh-Son Hua, and Khoi Nguyen. ISNBET: A 3D point cloud instance segmentation network with instance-aware sampling and box-aware dynamic convolution. *CVPR*, 2023. 2, 7

[47] Steven A Niederer, Michael S Sacks, Mark Girolami, and Karen Willcox. Scaling digital twins from the artisanal to the industrial. *Nature Computational Science*, 2021. 1

[48] Timea Nochta, Li Wan, Jennifer Mary Schooling, and Ajith Kumar Parlikad. A socio-technical perspective on urban analytics: The case of city-scale digital twins. *Journal of Urban Technology*, 2021. 1

[49] Jeremie Papon, Alexey Abramov, Markus Schoeler, and Florentin Worgotter. Voxel cloud connectivity segmentation-supervoxels for point clouds. *CVPR*, 2013. 2, 12

[50] Renfrey Burnard Potts. Some generalized order-disorder transformations. *Mathematical Proceedings of the Cambridge Philosophical Society*, 1952. 4

[51] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *NeurIPS*, 2017. 5, 7

[52] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. PointNeXt: Revisiting PointNet++ with improved training and scaling strategies. *NeurIPS*, 2022. 5, 6, 12

[53] Camillo Quattrocchi, Daniele Di Mauro, Antonino Furnari, and Giovanni Maria Farinella. Panoptic segmentation in industrial environments using synthetic and real data. *ICIAP*, 2022. 1

[54] Hugo Raguet and Loic Landrieu. Parallel cut pursuit for minimization of the graph total variation. *ICML Workshop on Graph Reasoning*, 2019. 4

[55] Damien Robert, Bruno Vallet, and Loic Landrieu. Learning multi-view aggregation in the wild for large-scale 3D semantic segmentation. *CVPR*, 2022. 6, 7

[56] Damien Robert, Hugo Raguet, and Loic Landrieu. Efficient 3D semantic segmentation with superpoint transformer. *ICCV*, 2023. 2, 4, 5, 6, 7, 11, 12

[57] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3D: Mask transformer for 3D semantic instance segmentation. *ICRA*, 2023. 2, 7, 13

[58] Nina M Singer and Vijayan K Asari. DALES Objects: A large scale benchmark dataset for instance segmentation in aerial LiDAR. *IEEE Access*, 2021. 1, 2

[59] Julien Smeeckaert, Clément Mallet, Nicolas David, Nesrine Chehata, and Antonio Ferraz. Large-scale classification of water areas using airborne topographic LiDAR data. *Remote sensing of environment*, 2013. 1

[60] Jiahao Sun, Chunmei Qing, Junpeng Tan, and Xiangmin Xu. Superpoint transformer for 3D scene instance segmentation. *AAAI*, 2023. 2

[61] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. KPConv: Flexible and deformable convolution for point clouds. *ICCV*, 2019. 5, 6, 7, 12

[62] Anirud Thyagharajan, Benjamin Ummenhofer, Prashant Laddha, Om Ji Omer, and Sreenivas Subramoney. SegmentFusion: Hierarchical context fusion for robust 3D semantic segmentation. *CVPR*, 2022. 12

[63] Nina Varney, Vijayan K Asari, and Quinn Graehling. DALES: A large-scale aerial LiDAR data set for semantic segmentation. *CVPR Workshops*, 2020. 5

[64] Thang Vu, Kookhoi Kim, Tung M Luu, Thanh Nguyen, and Chang D Yoo. SoftGroup for 3D instance segmentation on point clouds. *CVPR*, 2022. 2, 7

[65] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3D semantic scene graphs from 3D indoor reconstructions. *CVPR*, 2020. 6

[66] Cheng Wang, Chenglu Wen, Yudi Dai, Shangshu Yu, and Minghao Liu. Urban 3D modeling with mobile laser scanning: A review. *Virtual Reality & Intelligent Hardware*, 2020. 1

[67] Peng-Shuai Wang. OctFormer: Octree-based transformers for 3D point clouds. *SIGGRAPH*, 2023. 6

[68] Qi Wang, Shengge Shi, Jiahui Li, Wuming Jiang, and Xiangde Zhang. Window normalization: Enhancing point cloud understanding by unifying inconsistent point densities. 2022. 5, 6

[69] Shun-Cheng Wu, Johanna Wald, Keisuke Tateno, Nassir Navab, and Federico Tombari. SceneGraphFusion: Incre-

mental 3D scene graph prediction from RGB-D sequences. *CVPR*, 2021. 2, 6

[70] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point Transformer V2: Grouped vector attention and partition-based pooling. *NeurIPS*, 2022. 6

[71] Binbin Xiang, Yuanwen Yue, Torben Peters, and Konrad Schindler. A review of panoptic segmentation for mobile mapping point clouds. *arXiv preprint arXiv:2304.13980*, 2023. 1, 2, 5, 7

[72] Dandan Xu, Haobin Wang, Weixin Xu, Zhaoqing Luan, and Xia Xu. LiDAR applications to estimate forest biomass at individual tree scale: Opportunities, challenges and future perspectives. *Forests*, 2021. 1

[73] Fan Xue, Weisheng Lu, Zhe Chen, and Christopher J Webster. From LiDAR point cloud towards digital twin city: Clustering city objects based on gestalt principles. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2020. 1

[74] Wai Yeung Yan, Ahmed Shaker, and Nagwa El-Ashmawy. Urban land cover classification using airborne LiDAR data: A review. *Remote Sensing of Environment*, 2015. 1

[75] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3D instance segmentation on point clouds. *NeurIPS*, 2019. 2

[76] Georgios Zamanakos, Lazaros Tsochatzidis, Angelos Amanatiadis, and Ioannis Pratikakis. A comprehensive survey of LiDAR-Based 3D Object detection methods with deep learning for autonomous driving. *Computers & Graphics*, 2021. 2

[77] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. PolarNet: An improved grid representation for online LiDAR point clouds semantic segmentation. *CVPR*, 2020. 2

[78] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point Transformer. *ICCV*, 2021. 5, 6, 7

[79] Yiming Zhao, Xiao Zhang, and Xinming Huang. A technical survey and evaluation of traditional point cloud clustering methods for LiDAR panoptic segmentation. *ICCV Workshop*, 2021. 2

[80] Dingfu Zhou, Jin Fang, Xibin Song, Liu Liu, Junbo Yin, Yuchao Dai, Hongdong Li, and Ruigang Yang. Joint 3D instance segmentation and object detection for autonomous driving. *CVPR*, 2020. 2

[81] Zixiang Zhou, Yang Zhang, and Hassan Foroosh. Panoptic-PolarNet: Proposal-free LiDAR point cloud panoptic segmentation. *CVPR*, 2021. 1, 2

[82] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3D convolution networks for LiDAR segmentation. *CVPR*, 2021. 2

Appendix

In this appendix, we introduce our interactive visualization tool in Section A-1, our source code in Section A-2. We provide details about the superpoint encoder backbone SPT in Section A-3 and implementation details in Section A-4. In Section A-5, we discuss the evaluation of SuperCluster on instance segmentation, before comparing in Section A-6 the scalability of the Hungarian algorithm with our graph clustering formulation. We then provide an illustration of how many points can be segmented at once with SuperCluster in Section A-7. Finally, we provide detailed class-wise results and illustrate the colormaps of each dataset in Section A-8.

A-1. Interactive Visualization

Our project page `https://drprojects.github.io/supercluster` offers interactive visualizations of our method. As shown in Figure A-1, we can visualize samples from the datasets with different point attributes and from any angle. These visualizations were instrumental in designing and validating our model; we hope that they will also facilitate the reader's understanding.

A-2. Source Code

We make our source code publicly available at `https://github.com/drprojects/superpoint_transformer`. Our method is developed in PyTorch and relies on PyTorch Geometric, PyTorch Lightning, and Hydra.

A-3. Superpoint-Based Backbone

As mentioned in Section 3.3, our panoptic segmentation method conveniently extends to superpoint-based methods. In particular, we discuss here our choice of using Superpoint Transfomer [56] for both computing superpoint partitions and learning superpoint features.

Superpoint Transformer. Superpoint Transformer (SPT) is a superpoint-based transformer architecture for the efficient semantic segmentation of large-scale 3D point clouds. The authors propose a fast algorithm to build a hierarchical superpoint partition, whose implementation runs 7 times faster than previous superpoint-based approaches. Additionally, SPT relies on a self-attention mechanism to capture the relationships between superpoints at multiple scales, achieving state-of-the-art performance on S3DIS, KITTI-360, and DALES.

As a memory- and compute-efficient approach capable of producing superpoint representations for very large scenes, we found Superpoint Transformer to be the ideal backbone for scalable 3D panoptic segmentation endeavor.

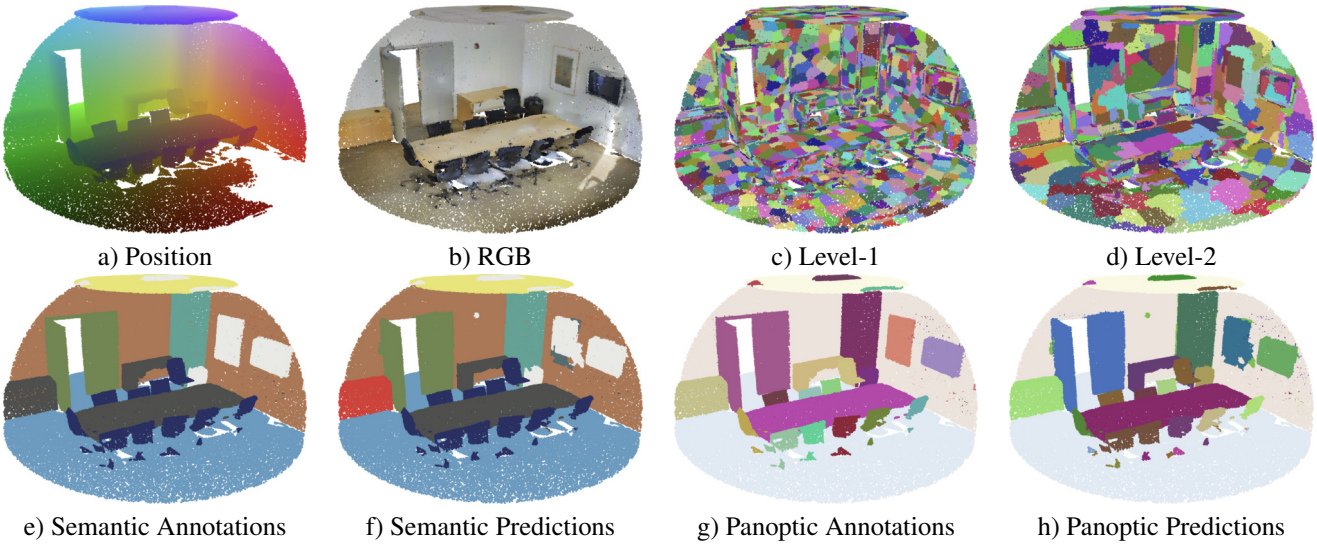


Figure A-1. **Interactive Visualization.** Our interactive viewing tool allows for the manipulation and visualization of point cloud samples colorized according to their position (a), radiometry (b), partition level (c,d), semantic annotations (e) and predictions (f), and panoptic annotations (g) and predictions (h).

Alternative Partitions. Alternative methods could be considered for computing the superpoint partition. Clustering-based methods such as VCCS [49] draw inspiration from SLIC [1] and use k-means on point features, under local adjacency constraints. However, these k-means-based methods rely on a fixed number of randomly initialized clusters, proscribing the processing of point clouds of arbitrary size and geometric complexity.

On the other hand, we use the implementation proposed in SPT, which itself derives from Landrieu *et al*. [35]. These papers cast point cloud oversegmentation as a structured optimization problem and use the cut-pursuit [33] algorithm to generate superpoints. This scalable approach does not make any assumption on the number of superpoints and produces a partition whose granularity adapts to the 3D geometry.

Alternative Backbones. One may consider different architectures to produce superpoint-wise or point-wise representations. Other methods for embedding superpoints exist [21, 25, 32, 35, 62], but Robert *et al*. [56] demonstrates superior performance and efficiency.

Alternatively, one may choose to adopt a per-point paradigm and rely on established models such as KPConv [61], MinkowskiNet [12], Stratified Transformer [31], or PointNeXt [52]. Although expressive, these models are memory and compute-intensive and can only handle small point clouds at once. For example, in an indoor setting such as S3DIS or ScanNet, SPT can process entire buildings as a whole, while these methods can only handle a few rooms simultaneously, which limits their applicability for large-scale panoptic segmentation.

Table A-1. **Graph Clustering Parameters.** We provide the graph clustering parameters used for each dataset.

Dataset	λ	η	ϵ
S3DIS	10	5.10^{-2}	10^{-4}
S3DIS - no "stuff"	20	5.10^{-2}	10^{-4}
ScanNet	20	5.10^{-2}	10^{-4}
KITTI-360	10	5.10^{-2}	10^{-4}
DALES	20	5.10^{-2}	10^{-4}

A-4. Implementation Details

In this section, we provide the exact parameterization of the SuperCluster architecture used for our experiments. For simplicity, we represent an MLP by the list of its layer widths: [in_channels, hidden_channels, out_channels].

Backbone. Our backbone model is Superpoint Transformer [56] with minor modifications, described below. We use SPT-64 for S3DIS and DALES and SPT-128 for KITTI360 and ScanNet.

For all datasets, we reduce the output dimension of the point encoder ϕ^0_{enc} from 128 to 64 [56]. We find that this does not affect SPT performance while reducing its memory requirements. For ScanNet, we find that using 32 heads instead of 16 and setting $D_{\text{adj}} = 64$ instead of 32 [56] improves the performance.

Object Agreement Head. The object agreement prediction head ϕ^{object} is a normalization-free MLP with LeakyReLU activations [42] and layers $[2D, 32, 16, 1]$, where D is the output feature dimension of the backbone (*i.e.* 64 for S3DIS and DALES, and 128 for ScanNet and KITTI-360).

Graph Clustering. As mentioned in Section 4.1, since our supervision framework does not require solving the clustering problem of Equation 1 during training, we tune the graph clustering parameters on the train set only after training. In Table A-1 we detail the tuned parameters for each dataset.

A-5. Instance Segmentation Evaluation

While methods predicting a panoptic segmentation could also provide an instance segmentation, we argue that their instance segmentation metrics are not directly comparable to the ones of methods dedicated to instance segmentation.

Firstly, instance segmentation metrics allow overlap between proposals, and not all points need to be in a predicted instance. Thus, instance segmentation methods can predict multiple instances per true object and avoid predicting in ambiguous or complex areas. In contrast, panoptic segmentation methods assign exactly one object label to each point [26].

Secondly, instance segmentation metrics require a confidence score for each proposal, which has a substantial influence on performance [23, 4.2.2]. Typically, instance segmentation methods learn this score with a dedicated network [23]. While Mask3D [57] derives this score from the semantic and mask confidences of the prediction, these are supervised by an explicit matching between the true and proposed instances. SuperCluster is precisely designed to avoid this matching step and never explicitly builds instances during training.

In summary, while it is technically possible to evaluate the predictions of SuperCluster with instance segmentation metrics, their comparison with dedicated methods would not be fair. A more equitable evaluation of SuperCluster on instance segmentation would require a dedicated post-processing step and an instance scoring mechanism, which falls outside the scope of this paper.

A-6. Scalability of Matching Step

In this section, we provide an experimental evaluation of the cost of the matching step using the Hungarian algorithm.

Experimental Protocol. We measure the time it takes our method to perform the panoptic segmentation step (Equation 1) for scenes of various sizes. Our goal is to compare this processing time with the matching step of conventional approaches. Given that we deal with scenes containing a large number of objects, far beyond what the encoders of these methods can handle in inference, we generate synthetic cost matrices to simulate the matching process. We compute these matrices for different combinations of the number of true objects `n_true` and of proposals `n_pred`. To generate realistic cost matrices, each proposed instance has a nonzero random cost for at most 3 true objects. We then measure the time taken by the Hungarian algorithm to solve the assignment.

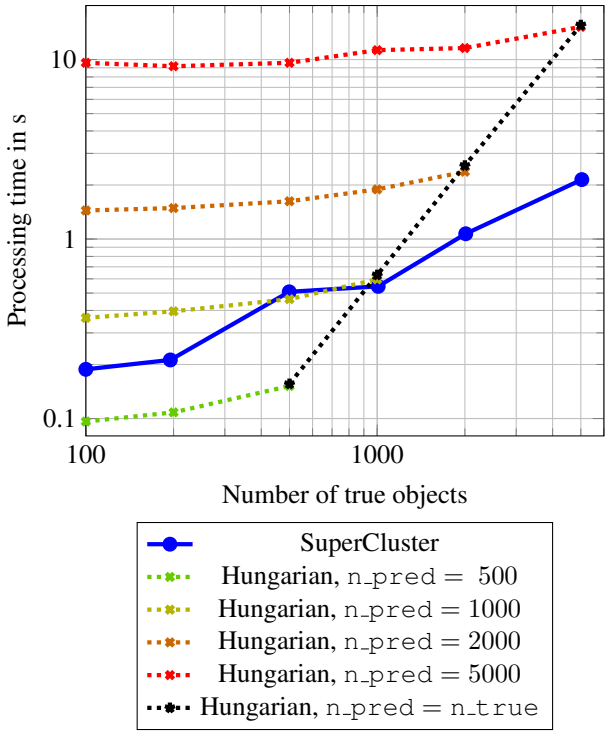


Figure A-2. **Cost of Matching Step.** We plot the time taken by our method to perform the panoptic segmentation step for scenes with various numbers of true objects. We also show the time necessary for the Hungarian algorithm to perform the matching for different numbers of maximum instance proposals `n_pred` and true instances `n_true`.

Analysis. We report the results of this experiment in Figure A-2. One significant advantage of our approach, denoted SuperCluster, is that it does not require a predefined maximum number of detected objects. In contrast, the processing time of the Hungarian algorithm is strongly affected by this parameter. Attempting to predict too many instances can lead to significantly prolonged training times, regardless of the number of true objects. Even in the idealized setting of `n_pred` equals to `n_true`, the Hungarian algorithm becomes much slower than our method when the number of true objects exceeds 1000. We also remind the reader that approaches relying on matching-based supervision must perform this step at each training iteration, while SuperCluster only solves Equation 1 once during inference and never during training.

A-7. Large-scale Inference

Our method can process large 3D point clouds with just one inference. In this section, we represent the largest portion of each dataset that SuperCluster can handle in one inference with an A40 GPU (48G of VRAM). Results for each dataset are presented in Figure A-3, Figure A-4, Figure A-5, and Figure A-6.

A-8. Detailed Results

We report in Table A-2, Table A-3, Table A-4, and Table A-5 the average and per-class performances of SuperCluster on each dataset.

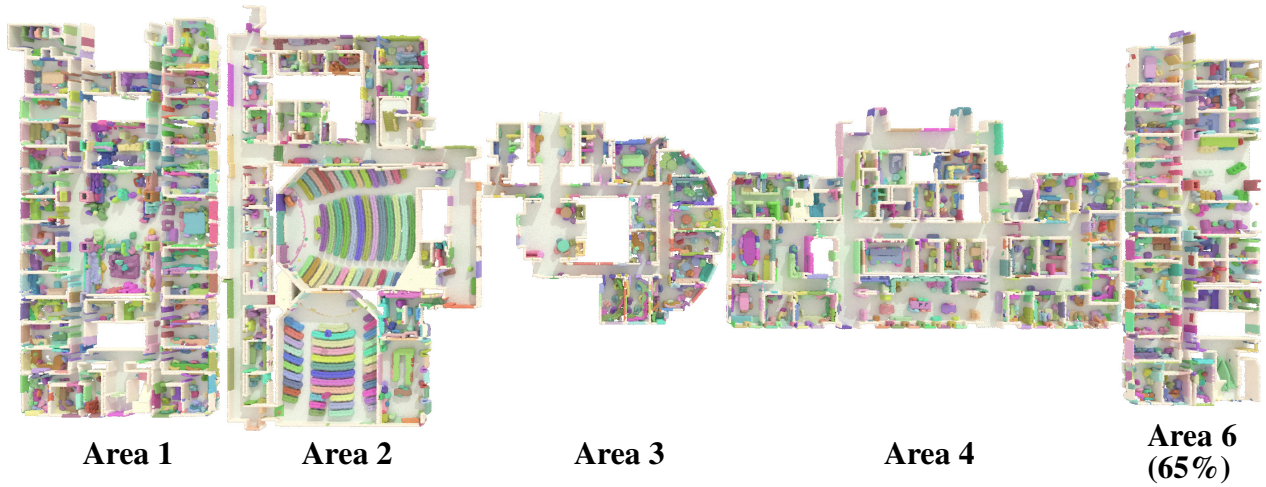


Figure A-3. **Large-Scale Inference on S3DIS.** Largest scan that SuperCluster can segment in one inference on an A40 GPU: **4.6** areas, **21.3M** points, **646k** superpoints, **5298** target objects, and **4565** predicted objects. Inference takes **7.4 seconds**.

Figure A-4. **Large-Scale Inference on ScanNet.** Largest number of scans that SuperCluster can segment in one inference on an A40 GPU: **105** scans , **10.9M** points, **398k** superpoints , **1683** target objects, and **2148** predicted objects. The inference takes **6.8 seconds**.

Figure A-5. **Large-Scale Inference on DALES.** Largest scan that SuperCluster can segment in one inference on an A40 GPU: **15.3** tiles, **7.8 km^2**, **18.0M** points, **589k** superpoints, **1727** target objects, and **1559** predicted objects. Inference takes **10.1 seconds**.

Figure A-6. **Large-Scale Inference on KITTI-360.** Largest scan that SuperCluster can segment in one inference on an A40 GPU: $\mathbf{7.5}$ tiles, $\mathbf{11.0M}$ points, $\mathbf{414k}$ superpoints, $\mathbf{602}$ target objects, and $\mathbf{1947}$ predicted objects. Inference takes $\mathbf{6.6}$ **seconds**.

Table A-2. **S3DIS Class-wise Performance.** We report the average and per-class panoptic quality (PQ), recognition quality (RQ), segmentation quality (SQ), precision (Prec), and recall (Rec) performance of SuperCluster on S3DIS. We indicate "stuff" classes with †.

S3DIS Area 5														
Metric	Avg.	ceiling †	floor †	wall †	beam	column	window	door	chair	table	bookcase	sofa	board	clutter
PQ	58.4	93.8	96.2	84.0	0.0	48.5	64.7	45.3	64.3	40.1	47.7	62.9	70.5	40.6
RQ	68.4	100.0	100.0	100.0	0.0	61.0	77.2	57.4	76.4	52.6	55.8	78.3	81.1	49.7
SQ	77.8	93.8	96.2	84.0	0.0	79.5	83.9	78.9	84.1	76.2	85.4	80.4	86.9	81.7
Prec.	71.4	100.0	100.0	100.0	0.0	64.2	79.6	59.8	75.7	53.3	66.0	75.0	93.8	60.5
Rec.	66.2	100.0	100.0	100.0	0.0	58.1	75.0	55.1	77.1	52.0	48.4	81.8	71.4	42.2
S3DIS 6-FOLD														
PQ	62.7	93.8	95.2	84.1	58.9	64.7	70.2	41.1	48.0	45.5	45.8	55.7	64.3	47.2
RQ	73.2	100.0	100.0	100.0	67.9	78.2	81.8	55.3	57.6	58.6	54.9	65.3	74.7	56.8
SQ	84.7	93.8	95.2	84.1	86.8	82.7	85.8	74.4	83.4	77.7	83.4	85.3	86.0	83.2
Prec.	77.8	100.0	100.0	100.0	67.9	80.2	84.7	61.7	69.0	55.9	66.2	74.4	80.0	71.1
Rec.	69.8	100.0	100.0	100.0	67.9	76.4	79.2	50.1	49.4	61.5	46.9	58.2	70.1	47.2
S3DIS Area 5 - no "stuff"														
Metric	Avg.	ceiling	floor	wall	beam	column	window	door	chair	table	bookcase	sofa	board	clutter
PQ	50.1	46.9	69.5	39.0	0.0	45.7	68.1	47.9	64.2	41.1	48.6	66.2	74.8	39.1
RQ	60.1	52.0	78.4	49.3	0.0	58.6	80.8	60.2	75.9	53.5	57.6	81.8	85.7	47.8
SQ	76.6	90.3	88.6	79.1	0.0	78.0	84.2	79.6	84.5	76.8	84.3	80.9	87.3	81.8
Prec.	63.6	45.5	70.6	43.7	0.0	62.1	90.5	68.7	76.7	58.5	74.4	81.8	94.3	59.9
Rec.	58.4	60.5	88.2	56.6	0.0	55.4	73.1	53.5	75.2	49.3	47.0	81.8	78.6	39.8
S3DIS 6-FOLD - no "stuff"														
PQ	55.9	68.6	64.1	40.0	65.6	64.0	70.1	42.7	48.0	48.3	43.7	55.4	69.4	46.7
RQ	66.3	74.8	72.6	50.8	74.3	76.7	81.8	57.1	57.3	62.8	52.5	64.6	80.5	56.0
SQ	83.8	91.8	88.2	78.6	88.3	83.4	85.8	74.7	83.7	76.9	83.2	85.7	86.2	83.4
Prec.	72.8	76.4	69.8	50.2	77.9	78.3	86.7	68.8	70.7	63.9	67.0	72.7	90.8	73.1
Rec.	61.7	73.3	75.7	51.4	71.1	75.2	77.4	48.8	48.2	61.8	43.1	58.2	72.3	45.4

Table A-3. **ScanNetv2 Val. Class-wise Performance.** We report the average and per-class panoptic quality (PQ), recognition quality (RQ), segmentation quality (SQ), precision (Prec), and recall (Rec) performance of SuperCluster on ScanNet. We indicate "stuff" classes with †.

Metric	Avg.	wall †	floor †	cabinet	bed	chair	sofa	table	door	window	bookshelf	picture	counter	desk	curtain	refrigerator	shower	toilet	sink	bathtub	otherfurniture
PQ	58.7	73.3	91.8	50.5	70.3	61.3	69.0	58.9	42.5	44.5	65.9	27.7	42.9	49.6	40.9	64.1	72.0	88.6	51.0	61.7	46.7
RQ	69.1	88.7	99.3	61.9	77.9	70.7	79.2	68.9	52.9	54.6	72.1	34.7	58.4	62.6	49.6	71.7	84.2	99.2	64.5	75.4	56.0
SQ	84.1	82.6	92.4	81.6	90.2	86.8	87.2	85.5	80.3	81.5	91.4	79.7	73.5	79.1	82.6	89.4	85.5	89.3	79.1	81.8	83.4
Prec.	76.7	93.1	100.0	69.7	81.1	79.0	80.0	68.0	65.9	63.4	75.7	64.9	68.4	60.1	58.0	94.3	82.8	98.3	86.0	76.7	69.5
Rec.	64.3	84.6	98.7	55.7	75.0	63.9	78.4	69.9	44.2	48.0	68.8	23.7	51.0	65.4	43.3	57.9	85.7	100.0	51.6	74.2	46.8

Table A-4. **KITTI-360 Val. Class-wise Performance.** We report the average and per-class panoptic quality (PQ), recognition quality (RQ), segmentation quality (SQ), precision (Prec), and recall (Rec) performance of SuperCluster on the KITTI-360 Validation set. We indicate "stuff" classes with †.

Metric	Avg.	road †	sidewalk †	building	wall †	fence †	pole †	traffic lig. †	traffic sig. †	vegetation †	terrain †	person †	car	truck †	motorcycle †	bicycle †
PQ	48.3	94.3	75.6	44.8	31.5	20.0	43.4	0.0	30.5	89.5	49.5	19.4	84.2	81.5	55.5	5.4
RQ	58.4	100.0	95.3	53.4	51.4	33.3	65.6	0.0	40.8	100.0	68.5	20.7	89.1	83.3	66.7	7.1
SQ	75.1	94.3	79.3	83.9	61.2	60.0	66.2	0.0	74.6	89.5	72.3	93.6	94.4	97.8	83.3	75.5
Prec.	60.3	100.0	96.2	48.4	51.9	33.3	65.6	0.0	43.5	100.0	76.0	23.1	92.4	90.9	73.3	10.0
Rec.	56.9	100.0	94.4	59.5	50.9	33.3	65.6	0.0	38.5	100.0	62.3	18.8	86.1	76.9	61.1	5.6

Table A-5. **DALES Class-wise Performance.** We report the average and per-class panoptic quality (PQ), recognition quality (RQ), segmentation quality (SQ), precision (Prec), and recall (Rec) performance of SuperCluster on DALES. We indicate "stuff" classes with †.

Metric	Avg.	ground †	vegetation †	car	truck	power line	fence	pole	building
PQ	61.2	95.6	90.3	70.9	45.0	18.8	23.5	64.3	81.5
RQ	68.6	100.0	99.0	78.4	51.1	23.1	31.3	79.6	86.6
SQ	87.1	95.6	91.2	90.4	88.2	81.3	75.0	80.8	94.1
Prec.	68.5	100.0	99.0	87.3	55.1	16.3	24.2	81.5	84.7
Rec.	71.0	100.0	99.0	71.1	47.5	39.7	44.3	77.8	88.5

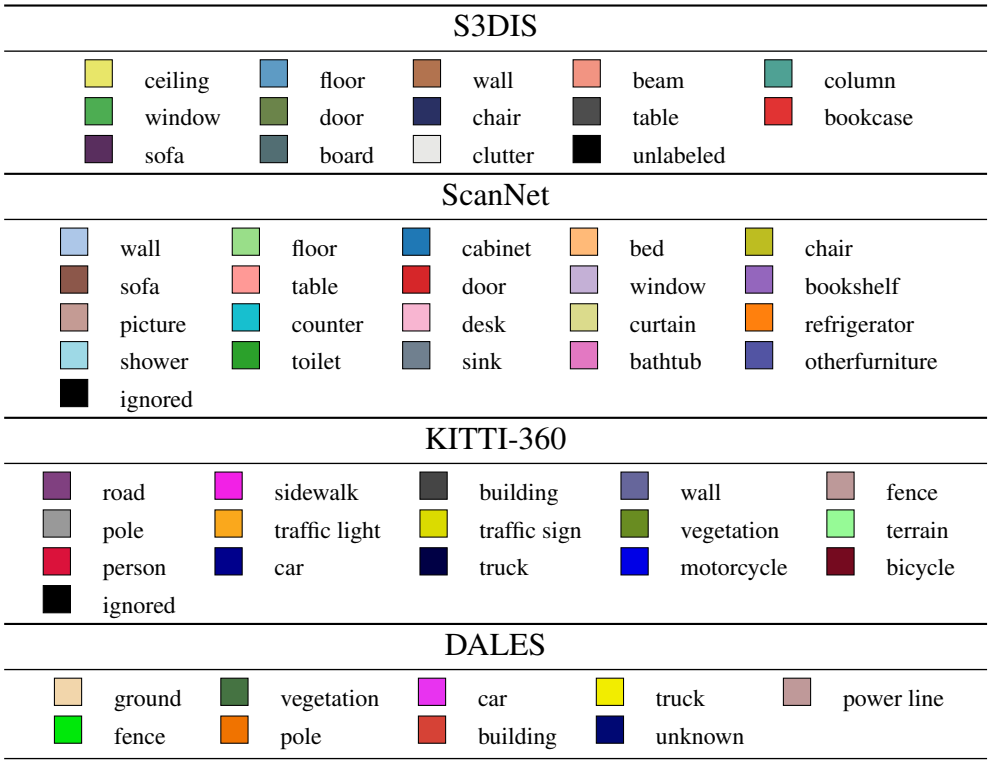


Figure A-7. **Colormaps.** Throughout all visualization in the main paper, the appendix, and the interactive visualization, we use this colormaps to represent the semantic of each point.