2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

From Semi-supervised to Omni-supervised Room Layout Estimation Using Point Clouds

Huan-ang Gao[1,2], Beiwen Tian[1,2], Pengfei Li[1,2], Xiaoxue Chen[1,2]
Hao Zhao[2✉], Guyue Zhou[2], Yurong Chen[4] and Hongbin Zha[3]

***Abstract*— Room layout estimation is a long-existing robotic vision task that benefits both environment sensing and motion planning. However, layout estimation using point clouds (PCs) still suffers from data scarcity due to annotation difficulty. As such, we address the semi-supervised setting of this task based upon the idea of model exponential moving averaging. But adapting this scheme to the state-of-the-art (SOTA) solution for PC-based layout estimation is not straightforward. To this end, we define a quad set matching strategy and several consistency losses based upon metrics tailored for layout quads. Besides, we propose a new online pseudo-label harvesting algorithm that decomposes the distribution of a hybrid distance measure between quads and PC into two components. This technique does not need manual threshold selection and intuitively encourages quads to align with reliable layout points. Surprisingly, this framework also works for the fully-supervised setting, achieving a new SOTA on the ScanNet benchmark. Last but not least, we also push the semi-supervised setting to the realistic omni-supervised setting, demonstrating significantly promoted performance on a newly annotated ARKitScenes testing set. Our codes, data and models are made publicly available[*].**

I. Introduction

Over the past decade, room layout estimation has drawn a lot of attention from the robotics community [1]–[6] since it marks a crucial step towards understanding indoor scenes and might help robot agents make better decisions in challenging environments [7]–[10]. However, the majority of earlier efforts exploit perspective or panoramic RGB images as input [11]–[28], whereas the promising paradigm of layout estimation using point clouds (PCs) [29] still suffers from the lack of annotated data. It is due to the difficulty of annotating the boundaries of 3D indoor scenes manually, particularly rooms containing non-cuboid shapes and many corners.

We envision an omni-supervised setting [30] where intelligent robots all over the world can exploit enormous unannotated raw point clouds to continuously improve the collective intelligence (i.e., layout estimation accuracy in this study). To this end, we start from the semi-supervised setting in which we assume a large portion of ScanNet [31] annotations is not available and push it finally to the omni-supervised setting using the recent ARKitScenes [32] dataset.

[1]Department of Computer Science and Technology, Tsinghua University, China, {gha20,tbw18,li-pf22,chenxx21}@mails.tsinghua.edu.cn.
[2]Institute for AI Industry Research (AIR), Tsinghua University, China, {zhaohao, zhouguyue}@air.tsinghua.edu.cn.
[3]Peking University, China, zha@cis.pku.edu.cn.
[4]Intel Labs, China, yurong.chen@intel.com.
[*]Code: https://github.com/AIR-DISCOVER/Omni-PQ
[*]We thank Didi Chuxing Technology Co. for supporting this project.
[*]✉: Corresponding author.

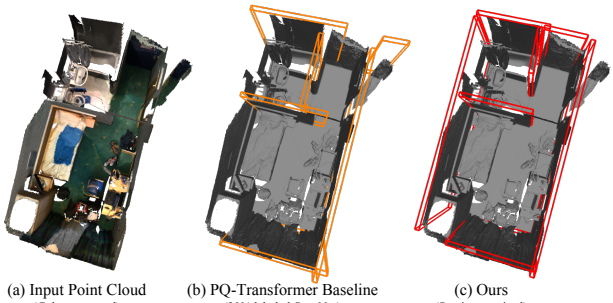

Fig. 1. *(a)* The input is a 3D point cloud whose colors are only for visualization. *(b)* We train the former SOTA method PQ-Transformer with only 20% labeled data of ScanNet training set and use it as the baseline. *(c)* We adopt our method on the whole ScanNet training set with only 20% annotations, resulting in a more accurate layout prediction.

Actually, semi-supervised room layout estimation has already been studied in a recent work SSLayout360 [33]. However, it still relies upon hand-crafted post-processing and only exploits the model exponential moving averaging (EMA) technique to learn representations from many unannotated panoramic images. Note that this paradigm does not apply to the state-of-the-art (SOTA) PC-based layout estimator [29], which directly predicts quads end-to-end.

To this end, we propose the first semi-supervised room layout estimation method using point cloud inputs. Our method builds upon the SOTA counterpart PQ-Transformer [29], which takes the 3D point cloud of a scene as input (see Fig. 1(a)) and predicts a set of quadrilateral (referred to as *quads*) equations representing layout elements (wall, floor and ceiling). As observed in Fig. 1(b), it performs poorly on unseen scenes if only 20% annotations are used for training. By contrast, our model is able to predict a more accurate layout by making use of the unlabeled data (see Fig. 1(c)).

Specifically, the success of our method is credited to two techniques. The **first** is a consistency based training framework inspired by the Mean Teacher [34] method. We design a quad matching strategy and three consistency regularization losses that are tailored for the layout estimation problem. We also identify a simple but effective add-on that capitalizes on the confidence of the teacher model. The **second** is a pseudo label generation module that decomposes the distribution of a new hybrid metric into two components, based upon gamma mixture. It intuitively aligns quad predictions to reliable layout point clouds. Through ablation experiments, both techniques are proven effective, and combining them brings larger improvements.

Experimental results highlight four notable messages: (1)

our solution with different percentages (e.g., 5% to 40%) of annotations available consistently and greatly outperforms supervised baselines on the ScanNet dataset. (2) with only 40% of labeled data we are able to surpass prior fully-supervised SOTA. (3) in the fully-supervised setting, our method can also improve strong baselines by +4.11%. (4) we further extend the method into a more realistic omni-supervised [30] setting, where we leverage all ScanNet training data and unlabeled ARKitScenes [32] training data. On a newly crowd-sourced ARKitScenes testing set, a significant performance gain is achieved, with F1-score going from 10.66% to 25.85%. Our contributions are as follows:

- We propose the first semi-supervised framework for room layout estimation using point clouds, with tailored designs including a quad set matching strategy and three confidence-guided consistency losses.
- We propose a threshold-free pseudo-label harvesting technique based upon a newly-proposed hybrid distance metric and gamma mixture decomposition.
- We achieve significant results in semi-supervised, fully-supervised and omni-supervised settings. We contribute a new crowd-sourced testing set and release our codes.

II. Related Works

Recently, semi-supervised and weakly-supervised learning are hot topics in the robotics community, with many methods proposed for various tasks including point cloud semantic parsing [35]–[38] and representation learning [39], 3D object detection [40] [41], articulation understanding [42], single-view reconstruction [43] and intrinsic decomposition [44]. This line of research envisions an exciting future scheme that robots all over the world exploit unlimited unlabeled data to continuously improve the collective intelligence. [45]–[48] Our study is the first semi-supervised framework for room layout estimation from point clouds, which contributes to this robotic vision trend.

From the perspective of methodology, we briefly review two kinds of semi-supervised learning (SSL) paradigms.

The consistency based SSL methods rely on the assumption that near samples from the low-dimensional input data manifold result in near outputs in the feature space [49] [50]. Thus, they enforce the model to stay in agreement with itself despite perturbations. Under this scope, multiple perturbation strategies are explored. The Π model [51] [52] penalizes the difference of hidden features of the same input with different data transformations and dropout. Temporal Ensembling training [52] regularizes consistency on current and former predictions. The Mean Teacher method [34] uses exponential moving average of student network parameters.

The pseudo-label based SSL methods, on the other hand, are more general as they don't require domain-specific data transformations. By equipping them with a few necessary designs, they can be as proficient as consistency based ones. For example, in 3D object detection task, [53] proposes two post-processing modules to improve the recall rate and the precision rate of the pseudo labels. In image classification task, [54] sets a constant confidence threshold τ for determining whether to discard a pseudo-label, and [55] upgrades that constant to a set of per-class learnable variables. In 2D object detection task, Noisy Pseudo-box Learning strategy is proposed by [56], which only considers N proposals of top-quality as pseudo labels and the rest ones as false positives.

III. Method

We aim to develop a learning framework that allows robot agents to leverage enormous unlabeled data to infer room layouts $\mathbf{Y}$ from indoor scene point clouds $\mathbf{X}$. Following [29], we denote a layout wall (represented by a quad) $\mathbf{y} = \{\mathbf{c}, \mathbf{n}, \mathbf{s}, p\} \in \mathbf{Y}$ by its center coordinate $\mathbf{c}$, unit normal vector $\mathbf{n}$, size $\mathbf{s} = (w, h)$, and predicted quadness score p. The quadness scores of ground truths are fixed to 1.0.

To start with, we formally describe three training settings. Suppose we have a 3D point cloud (PC) dataset $\mathcal{D}_L$ with layout annotations in conjunction with a much larger unlabeled PC dataset $\mathcal{D}_U$. In the **fully-supervised setting**, $\mathcal{D}_L$ is the whole training set of ScanNet with quad annotations whereas $\mathcal{D}_U$ is a null set. In the **semi-supervised setting**, $\mathcal{D}_L$ is part of the ScanNet training set along with quad annotations whereas $\mathcal{D}_U$ is the complementary set whose annotations are assumed unknown. In the **omni-supervised setting** [30] which is a real-world generalization of the semi-supervised setting, $\mathcal{D}_L$ is the whole training set of ScanNet with annotations whereas $\mathcal{D}_U$ is the ARKitScenes training set without annotations.

We introduce our method in the three settings using unified notations $\mathcal{D}_L$ and $\mathcal{D}_U$. As depicted in Fig. 2, we adapt the Mean Teacher [34] training framework (see Sec. III-A) to end-to-end room layout estimation with a tailored quad matching strategy and three consistency losses. We also integrate a novel pseudo-label refinement module (Sec. III-B) for quads, which is based upon gamma mixture decomposition. In Sec. III-C, we describe the loss terms to optimize.

A. Quad Mean Teacher (QMT)

Mean Teacher [34] is a successful framework for semi-supervised learning with a student model and a teacher model of the same architecture. The general idea is to feed two models with the same input samples transformed differently and enforce the predictions of the two models to be consistent. The student model is updated by gradient descent while the teacher model is updated by exponential moving average (EMA) of the weights of the student model.

Inspired by the idea of Mean Teacher, we first sample $\mathbf{X}^U$ from $\mathcal{D}_U$ and $(\mathbf{X}^L, \mathbf{Y}^L)$ from $\mathcal{D}_L$ to form a batch $\mathbf{X} = \{\mathbf{X}^L, \mathbf{X}^U\}$. $\mathbf{X}$ is transformed with stochastic transformation T before feeding into the student model to yield $\tilde{\mathbf{Y}}_S = \{\tilde{\mathbf{Y}}_S^L, \tilde{\mathbf{Y}}_S^U\}$. $\mathbf{Y}^L$ is transformed into $\tilde{\mathbf{Y}}^L$ with the same transformation. Meanwhile, $\mathbf{X}$ is also fed into the teacher model and then applied the same transformation T to yield $\tilde{\mathbf{Y}}_T = \{\tilde{\mathbf{Y}}_T^L, \tilde{\mathbf{Y}}_T^U\}$. Following the same loss design in [29], we impose a supervised loss $\mathcal{L}_{\text{sup}}$ between $\tilde{\mathbf{Y}}_S^L$ and $\tilde{\mathbf{Y}}^L$.

The success of Mean Teacher based methods relies on domain-specific data transformation and carefully designed

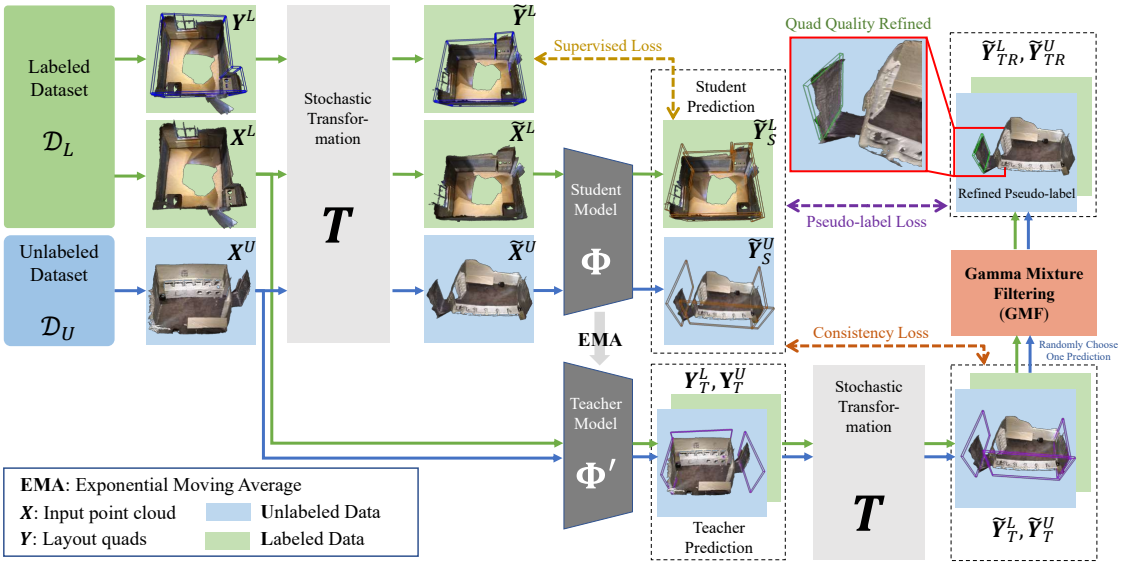

Fig. 2. **Method Overview**. In each training iteration, we sample $(\mathbf{X}^L, \mathbf{Y}^L)$ from labeled dataset and $\mathbf{X}^U$ from unlabeled dataset to form a batch. The input batch is first stochastically transformed then fed into the student model to produce predictions $\tilde{\mathbf{Y}}^L_S$ and $\tilde{\mathbf{Y}}^U_S$. Meanwhile, the input batch is also fed into the teacher model then transformed to produce predictions $\tilde{\mathbf{Y}}^L_T$ and $\tilde{\mathbf{Y}}^U_T$. In the two adopted transformations, FPS sampling uses different seeds whereas rotation, flipping and scaling are identical. We impose three losses in total: (1) a supervised loss between the transformed label and predictions of student model. (2) a consistency loss that minimizes the difference between student predictions and teacher predictions. (3) a pseudo-label loss that encourages quads to align with reliable layout points. The student parameters are updated by gradient descent according to the sum of three losses, whereas the teacher parameters are updated by exponential moving average (EMA) of student parameters.

consistency losses between two sets of predictions, without which the method could suffer from degeneration. Based upon this observation, we design the transformation domain and consistency losses for room layout estimation as follows.

Data transformation We adopt four kinds of transformations: Farthest Point Sampling (FPS) [57], flipping along horizontal axes, rotating along vertical axes and coordinates scaling. FPS [57] downsamples the point cloud by repeatedly choosing the point farthest from the chosen ones, discarding only redundant points. Also, flipping, rotating and scaling in constrained ways mimic the natural viewpoint changes of humans. Among them, layout annotations are invariant to FPS [57] as subsampling does not change the layout geometries and equivariant to the other three transformations with which the geometries should be transformed accordingly. Hence, when applying the same transformation, for invariant transformation (i.e., FPS [57]) we use different seeds and for the other three we apply the same transformation before the student model and after the teacher model.

Quad Set Matching To encourage consistency between the predicted quad sets of two models, the difference between two quads should be defined first. Given two quad predictions, $\tilde{\mathbf{y}}_1 = \{\tilde{\mathbf{c}}_1, \tilde{\mathbf{n}}_1, \tilde{\mathbf{s}}_1, p_1\}$ and $\tilde{\mathbf{y}}_2 = \{\tilde{\mathbf{c}}_2, \tilde{\mathbf{n}}_2, \tilde{\mathbf{s}}_2, p_2\}$, the differences of three geometrical characteristics (quad center location $\tilde{\mathbf{c}}$, quad normal $\tilde{\mathbf{n}}$, quad size $\tilde{\mathbf{s}}$) should all be considered. Thus, as illustrated in Fig. 3(b), we define the distance between two quads as: ($\|\cdot\|_k$ denotes k-norm)

$$d(\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2) = \|\tilde{\mathbf{c}}_1 - \tilde{\mathbf{c}}_2\|_2 + |1 - \tilde{\mathbf{n}}_1 \cdot \tilde{\mathbf{n}}_2| + \|\tilde{\mathbf{s}}_1 - \tilde{\mathbf{s}}_2\|_2^2 \quad (1)$$

Based on the distance metric between quads, we calculate the difference of two predicted quad sets by first finding the correspondences between the two quad sets and then summing up the distances between corresponding quads. To establish the correspondences, we find the nearest student-predicted quad $\tilde{\mathbf{y}}_S = \{\tilde{\mathbf{c}}_S, \tilde{\mathbf{n}}_S, \tilde{\mathbf{s}}_S, p_S\}$ for each teacher-predicted quad $\tilde{\mathbf{y}}_T = \{\tilde{\mathbf{c}}_T, \tilde{\mathbf{n}}_T, \tilde{\mathbf{s}}_T, p_T\}$:

$$\mathcal{P}_{\tilde{\mathbf{Y}}_S}(\tilde{\mathbf{y}}_T) = \underset{\tilde{\mathbf{y}}_S \in \tilde{\mathbf{Y}}_S}{\operatorname{argmin}} \|\tilde{\mathbf{c}}_S - \tilde{\mathbf{c}}_T\|_2 \quad (2)$$

We use $\mathcal{P}(\cdot)$ to represent this injective mapping from the teacher model prediction to the student model prediction.

Consistency Loss Design Although the quad geometries (i.e., $\tilde{\mathbf{c}}, \tilde{\mathbf{n}}, \tilde{\mathbf{s}}$) predicted by teacher are not adequately precise, the predicted quadness score p could measure the correctness of the predictions. Considering that the teacher-predicted quads are generally more reliable than the student-predicted quads, we use teacher-predicted quadness scores p_T as the confidence and define the consistency loss $\mathcal{L}_{\text{QMT}}$ as:

$$\mathcal{L}_{\text{QMT}} = \frac{1}{|\tilde{\mathbf{Y}}_T|} \sum_{\tilde{\mathbf{y}}_T \in \mathbf{Y}_T} d(\mathcal{P}_{\tilde{\mathbf{Y}}_S}(\tilde{\mathbf{y}}_T), \tilde{\mathbf{y}}_T) \cdot p_T \quad (3)$$

Remark A similar idea to evaluate the closeness of two sets is the Chamfer Distance, which establishes two injective mappings from each of the two sets to its counterpart. On the contrary, our method only establishes a one-way mapping from the teacher model predictions to the student model predictions since the latter is less reliable than the former. As depicted in Fig. 3(a), finding the nearest teacher prediction around S and penalizing the quad distance in between would wrongly push S to the unreliable prediction T_1. By contrast, as S is the nearest student prediction around unreliable T_1 and reliable T_2, optimizing the weighted quad distance sum would push S to the reliable prediction T_2.

B. Gamma Mixture Filtering (GMF)

In this stage, we introduce the Gamma Mixture Filtering module which makes further use of the unlabeled data and

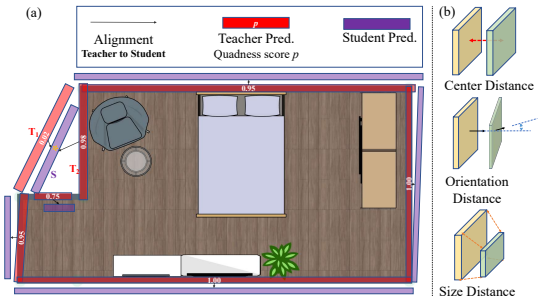

Fig. 3. **Illustration on Teacher Student Alignment**. *(a)* For every teacher-predicted quad, we find the nearest student-predicted quad. Although teacher predictions are noisy, the quadness scores demonstrate how accurate the predictions are. *(b)* These three figure illustrate the three components of the defined distance between two quads.

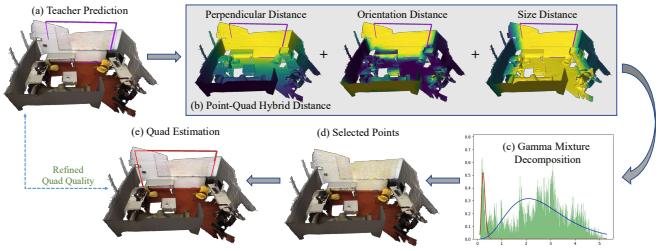

Fig. 4. **Illustration on Gamma Mixture Filtering**. We calculate the proposed hybrid metrics between points and quads in (b), where warmer colors indicate shorter distances. Then we decompose the distribution of metrics into two components, corresponding to points that belong to the quad and those don't, respectively. We filter out redundant points using the mixture distribution model (depicted in (c)), and re-estimate quads with higher accuracy for the student model to learn.

re-estimates a more accurate quad prediction $\tilde{\mathbf{y}}_{TR}$ from the noisy prediction $\tilde{\mathbf{y}}_T$. A naive approach to do this is to select points whose perpendicular distance to the quad is below a manually chosen distance threshold ϵ_D and use these points to estimate a more accurate quad. However, it is inevitable to manually tune the hyper-parameter ϵ_D, which is time-consuming and ineffective as a fixed threshold is usually not applicable to all scenes. Besides, using perpendicular distance solely as the metric may erroneously select points in the room corners which belong to other quads.

To address these issues, we introduce 1) hybrid distance between point and quad as an improved metric and 2) the gamma mixture decomposition filtering strategy to automatically select the threshold for filtering.

Hybrid Point-Quad Metric We propose a hybrid metric to measure the distance between a point and a quad. Instead of using the perpendicular distance alone, we also leverage normals and quad sizes. Consider a point $\mathbf{p}$ with coordinate $\mathbf{c}_p$ and normal $\mathbf{n}_p$ estimated with adjacent points in the PC, and a quad $\tilde{\mathbf{y}}_T$ whose plane equation is $\tilde{\mathbf{n}}_T \cdot (\mathbf{c} - \tilde{\mathbf{c}}_T) = 0$, $\mathbf{c} \in \mathbb{R}^3$. Then the perpendicular distance can be written as:

$$\mathcal{M}_{\mathrm{p}}(\mathbf{p}, \tilde{\mathbf{y}}_T) = |(\mathbf{c}_p - \tilde{\mathbf{c}}_T) \cdot \tilde{\mathbf{n}}_T| \tag{4}$$

Note that $\tilde{\mathbf{n}}_T$ is of unit length. In some corner cases where points are close but differ greatly in normals (e.g. in wall corners), using this measure solely would erroneously include points on other quads. Therefore, we also define a cosine similarity metric for the normals:

$$\mathcal{M}_{\mathrm{o}}(\mathbf{p}, \tilde{\mathbf{y}}_T) = |1 - \mathbf{n}_p \cdot \tilde{\mathbf{n}}_T| \tag{5}$$

Furthermore, as the size of quads is not considered in the proposed two measures, we consider the extent to which the projections of points lay outside the quad. Since the vertical edges of predicted quads are parallel to $\hat{\mathbf{z}} = (0, 0, 1)^T$, the horizontal edges should be parallel to $\hat{\mathbf{x}} = \dfrac{\tilde{\mathbf{n}}_T \times \hat{\mathbf{z}}}{\|\tilde{\mathbf{n}}_T \times \hat{\mathbf{z}}\|_2}$ ($\times$ denotes cross product). The horizontal and vertical distances between the quad center and the projection of $\mathbf{p}$ on the quad are then given by $w_p = |(\mathbf{c}_p - \tilde{\mathbf{c}}_T) \cdot \hat{\mathbf{x}}|$ and $h_p = |(\mathbf{c}_p - \tilde{\mathbf{c}}_T) \cdot \hat{\mathbf{z}}|$, respectively. Thus, we define the out-of-quad metric as:

$$\mathcal{M}_{\mathrm{s}}(\mathbf{p}, \tilde{\mathbf{y}}_T) = \|\mathrm{ReLU}((w_p, h_p)^T - \tilde{\mathbf{s}}_T)\|_1 \tag{6}$$

Finally, the hybrid point-quad distance is defined as:

$$\mathcal{M} = \mathcal{M}_p + \mathcal{M}_o + \mathcal{M}_s \tag{7}$$

In Fig. 4(b) we illustrate the three proposed metrics between the highlighted quad and points.

Mixture Decomposition Filtering In this stage we use the hybrid metric $x = \mathcal{M}(\cdot, \cdot)$ to select points from the PC for each quad. We first collect the metrics between the quad and all points, and then use the metrics to fit a probabilistic mixture model. The possibility density function (PDF) of the probability model is defined as

$$P(x|\theta_0, \theta_1) = w_0 P(x|\theta_0) + w_1 P(x|\theta_1) \tag{8}$$

where $P(x|\theta_0)$ and $P(x|\theta_1)$ are PDFs of individual components and w_0, w_1 denote weights of them, with $w_0 + w_1 = 1$.

The two individual components correspond to points that belong to the quad and those don't, respectively. We empirically choose gamma distribution for the two components:

$$P(x|\theta_i) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1}, \theta_i = \{a, b\} \tag{9}$$

To fit this mixture distribution, we follow [58] to decide the parameters θ_0, θ_1, w_0 and w_1. By using the expectation maximization (EM) algorithm, we take the parameters when $\sum_{p \in P} \log P(x_p|\theta_0, \theta_1)$ is maximized. The fitting result is illustrated in Fig. 4(c), where the blue curve represents $P(x|\theta_0)$ and the red curve represents $P(x|\theta_1)$.

Finally, with this mixture model, we examine the probabilities that an unlabeled point belongs and not belongs to the quad. When the former is larger than the latter, we keep this point during filtering. In other words, for each quad $\mathbf{y}_T$ we keep points $\mathbf{p}_i$ that satisfy $w_0 P(x_i|\theta_0) \leq w_1 P(x_i|\theta_1)$ where $x_i = \mathcal{M}(\mathbf{p}_i, \mathbf{y}_T)$, as shown in Fig. 4(d). It is unnecessary to manually tune a threshold, as the intersection point of the two component PDFs works as a per-quad threshold obtained by statistics of the unlabeled points around that quad.

Quad estimation With the set of selected points P', we reconstruct a more accurate quad $\tilde{\mathbf{y}}_{TR}$ for each predicted quad $\tilde{\mathbf{y}}_T$. We refine the quad center and quad normal to $\mathbf{c}' = \frac{1}{|P'|} \sum_{p \in P'} \mathbf{c}_p$ and $\mathbf{n}' = \sum_{p \in P'} \mathbf{n}_p \;/\; \|\sum_{p \in P'} \mathbf{n}_p\|_2$. To estimate the quad size, we randomly take K_s samples $\{\tau_i\}_{i=1}^{K_s}$ from $[0, 1]$. Under the assumption that the point

collection P' is uniformly sampled from the refined quad, we refine the quad size to $\mathbf{s}' = \frac{1}{K_s}\sum_{i=1}^{K_s}\frac{1}{\tau_i}\cdot \text{quantile}(\tau_i)$. Here quantile$(\tau_i)$ is defined as τ_i-th quantiles of $\{\mathbf{s}_p | p \in P'\}$ computed on $\hat{\mathbf{x}}$ axis and $\hat{\mathbf{z}}$ axis, respectively.

In each scene of each training step, due to tractability concerns, we choose one of all teacher predicted quads to refine, as illustrated in Fig. 2. Based on the refined quad $\tilde{\mathbf{y}}_{TR} = \{\mathbf{c}', \mathbf{n}', \mathbf{s}', 1.0\}$, we propose the pseudo-label loss:

$$\mathcal{L}_{\text{GMF}} = d(\mathcal{P}(\tilde{\mathbf{y}}_T), \tilde{\mathbf{y}}_{TR}) \tag{10}$$

C. Loss

The loss term we aim to optimize during training is:

$$\mathcal{L} = \mathcal{L}_{\text{sup}} + \lambda_{\text{QMT}}\mathcal{L}_{\text{QMT}} + \lambda_{\text{GMF}}\mathcal{L}_{\text{GMF}} \tag{11}$$

where λ_{QMT} and λ_{GMF} are loss weights.

IV. Experiment

A. Datasets and Implementation Details

Datasets In the semi-supervised setting, our methods are evaluated on the ScanNet dataset. ScanNet [31] is a large-scale RGB-D video dataset with 3D reconstructions of indoor scenes, including 1513 scans reconstructed from around 2.5 million views. On top of the ScanNet, SceneCAD [59] provides scene layout annotations containing 8.4K polygons. In our experiments, we use the 3D reconstructions from ScanNet [31] as the input point clouds and use the scene layouts from SceneCAD [59] as the ground truth labels.

Furthermore, we extend our methods to the omni-supervised setting and employ ARKitScenes dataset [32]. ARKitScenes is another large-scale RGB-D dataset containing 4493 training scans and 549 validation scans. In our experiments, the training scans are leveraged as the unlabeled input. The validation scans are used for testing, whose ground-truth layouts are annotated by crowd-sourcing.

Implementation Details In the transformation stage, the point cloud is first downsampled to 40,000 points with FPS and rotated along the z-axis by $\theta = \theta_1 + \theta_2$, with θ_1 randomly chosen from $\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$ and θ_2 uniformly sampled from $[-5°, 5°]$. Next, the point cloud is flipped along the x-axis and the y-axis with the probability of 0.5 and scaled by a ratio uniformly sampled from $[0.85, 1.15]$.

We implement the teacher and student models in the proposed Quad Mean Teacher framework with PQ-Transformer [29], while the framework also works with other layout estimators. The preprocessing of quad annotations and the evaluation metrics are the same as [29]. The consistency loss weight is set to $\lambda_{\text{QMT}} = 0.05$, using the same warm-up strategy as [60]. The pseudo-label loss weight is set as $\lambda_{\text{GMF}} = 5 \times 10^{-4}$. Our experiments run on a single NVIDIA RTX A4000 GPU with batch size of 6. Half of the samples in a batch have quad annotations.

B. Results

To the best of our knowledge, our methods are the first to perform the PC-based layout estimation task in the semi-supervised and the omni-supervised setting. Hence we compare our method with fully-supervised methods including SceneCAD [59] and PQ-Transformer [29].

TABLE I
LAYOUT ESTIMATION F1-SCORES ON SCANNET

Method	5%	10%	20%	30%	40%	100%
SceneCAD [59]	-	-	-	-	-	37.90
PQ-Transformer [29]	22.43	29.26	39.60	46.02	48.08	56.64
Ours (QMT only)	26.83	34.76	44.42	49.30	51.84	58.80
Ours (GMF only)	26.65	35.17	45.25	51.69	52.69	60.54
Ours (QMT & GMF)	**29.08**	**36.85**	**48.68**	**54.35**	**56.92**	**60.75**
Margin	+6.65	+7.59	+9.08	+8.33	+8.84	+4.11
Relative Improv. (%)	29.65↑	25.94↑	22.93↑	18.10↑	18.37↑	7.26↑

We evaluate our method and the baselines in various semi-supervised settings on ScanNet validation set and report the F1-scores of the predicted layouts in Tab. I. The size of labeled set $\mathcal{D}_L$ sampled from the ScanNet training split, or the amount of ground truth annotations in use, is denoted by percentages in the first row. And $\mathcal{D}_U$ is the complementary set whose annotations are assumed unknown.

It can be seen that either QMT or GMF can result in performance boost. And by combining these two techniques together, we see further improvement in performance. In all semi-supervised settings, the performances of our methods are better than baselines by large margins. With only 40% quad annotations available, our method achieves similar performance to that of the state-of-the-art method trained in fully supervised settings. Surprisingly, our method also performs better in fully supervised settings than former arts. We attribute the outperformance to the consistency regularization mechanism promoting the model's robustness to perturbations and the pseudo-label refinement module providing guidance on the geometrical information of layouts.

We further demonstrate the robustness of our method in the omni-supervised setting [30]. To be more specific, we train our method and the baselines with the whole labeled training split of ScanNet $\mathcal{D}_L$ and then evaluate the performance on the validation split of the ARKitScenes dataset with crowd-sourced layout annotations. Besides, in our method, the unlabeled training split of ARKitScenes serves as the unlabeled dataset $\mathcal{D}_U$. As shown in Tab. II, our method achieves a significant margin over former arts, showing the ability to generalize to more realistic omni-supervised settings.

In addition, we provide visualization of the quad predictions of our method on ScanNet and ARKitScenes in Fig. 5 and Fig. 6. These qualitative results show that exterior quads as well as the interior quads are predicted by our method accurately, compensating for the ineffectiveness of PQ-Transformer [29] w.r.t. interior wall quads.

TABLE II
LAYOUT ESTIMATION F1-SCORES ON ARKITSCENES

Method	Recall (%)	Precision (%)	F1-score (%)
PQ-Transformer [29]	6.72	25.81	10.66
Ours (QMT & GMF)	**23.00**	**29.50**	**25.85** (+15.19)

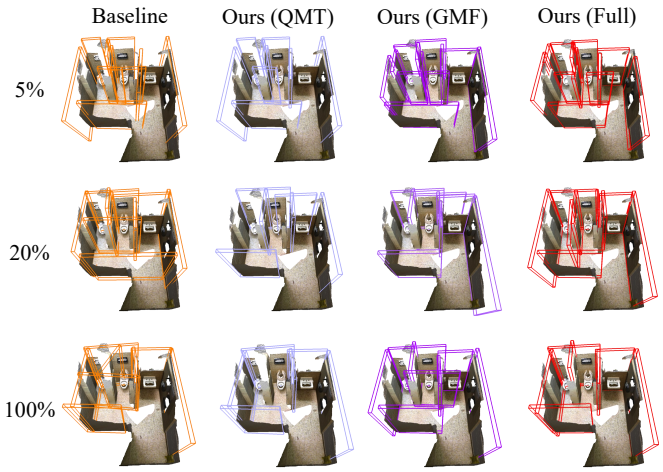

Fig. 5. **Qualitative results on ScanNet.** The ratio represents the proportion of annotated data in use.

C. Ablation Study

Data Transformation Strategies We run the 10%-supervised experiment on ScanNet with different data transformations. As shown in Tab. III, data transformation is crucial to our proposed method, as any of the transformations improves the performance, and in extreme cases without transformations the F1-score decreases by 6.31%.

Among the four transformations, rotation has the largest influence on the performances. One possible reason is that rotation brings the most changes to the coordinates of points whilst keeping the holistic layouts of the scenes unchanged.

TABLE III
ABLATIONS ON DATA TRANSFORMATION STRATEGIES

Downsample	Flipping	Rotation	Scaling	F1-score (%)
×	×	×	×	30.54
✓	×	×	×	31.59
×	✓	×	×	32.47
×	×	✓	×	32.95
×	×	×	✓	30.70
×	✓	✓	✓	35.61
✓	×	✓	✓	34.53
✓	✓	×	✓	33.42
✓	✓	✓	×	33.96
✓	✓	✓	✓	**36.85**

Quad Mean Teacher We compare Quad Mean Teacher and the basic Mean Teacher (MT) method in the 10%-supervised settings and report their performances on ScanNet in Tab. IV. MT assumes that all teacher predictions are equally reliable. Results show that the QMT achieves a large margin over MT on the precision of prediction. We believe this is because the confidence of predictions by the teacher model is exploited and erratic or incorrect predictions are neglected accordingly.

Gamma Mixture Filtering In the 10%-supervised settings, we compare our method using only pseudo-label loss with the naive ϵ_D approach introduced in Sec. III-B. We set the fixed threshold $\epsilon_D = 0.2$m. More specifically, in the alternative method, a point stays after filtering if its perpendicular distance to the plane is less than ϵ_D. Compared to ours, the alternative method achieves significantly lower performance, since no supervision is applied on the quad normals and sizes.

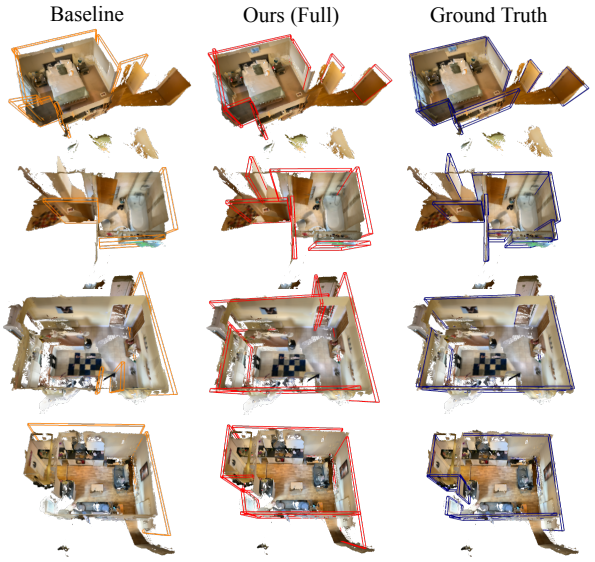

Fig. 6. **Qualitative results on ARKitScenes.** Ground truth layouts are annotated by crowd-sourcing.

TABLE IV
ABLATIONS ON QUAD MEAN TEACHER

Method	Recall (%)	Precision (%)	F1-score (%)
Mean Teacher	26.39	39.61	31.67
Ours (QMT)	**27.73**	**46.54**	**34.76**

TABLE V
ABLATIONS ON GAMMA MIXTURE FILTERING

Method	Recall (%)	Precision (%)	F1-score (%)
Simple ϵ_D Filtering	25.29	40.51	31.14
Ours (GMF)	**29.75**	**43.01**	**35.17**

V. CONCLUSION

Our research makes the first step towards omni-supervised layout estimation merely using point clouds, which has promising implications in robotics. Our training framework combines Quad Mean Teacher and Gamma Mixture Filtering to better exploit the unlabeled data. Experimental results demonstrate our method's effectiveness in semi-supervised, fully-supervised and omni-supervised settings.

Despite the effectiveness of our method, limitations still exist. The predictions of our method are unsatisfactory in incomplete scenes, in which insufficient points fail to form a layout wall. In the future, we will consider possible rectifications including ensembling online inference results, thanks to the quasi-real-time speed brought by the PQ-Transformer [29] implementation.

REFERENCES

[1] L. Ma, C. Kerl, J. Stückler, and D. Cremers, "Cpa-slam: Consistent plane-model alignment for direct rgb-d slam," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1285–1291.

[2] M. Hsiao, E. Westman, G. Zhang, and M. Kaess, "Keyframe-based dense planar slam," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. Ieee, 2017, pp. 5110–5117.

[3] M. Kaess, "Simultaneous localization and mapping with infinite planes," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4605–4611.

[4] S. Yang, Y. Song, M. Kaess, and S. Scherer, "Pop-up slam: Semantic monocular plane slam for low-texture environments," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1222–1229.

[5] S. Yang and S. Scherer, "Monocular object and plane slam in structured environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3145–3152, 2019.

[6] H. Zhao, R. Ranftl, Y. Chen, and H. Zha, "Transferable end-to-end room layout estimation via implicit encoding," *arXiv preprint arXiv:2112.11340*, 2021.

[7] Y. M. Chung, H. Youssef, and M. Roidl, "Distributed timed elastic band (dteb) planner: Trajectory sharing and collision prediction for multi-robot systems," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 10 702–10 708.

[8] B. Jin, B. Tian, H. Zhao, and G. Zhou, "Language-guided semantic style transfer of 3d indoor scenes," in *Proceedings of the 1st Workshop on Photorealistic Image and Environment Synthesis for Multimedia Experiments*, ser. PIES-ME '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 11–17.

[9] C. Pérez-D'Arpino, C. Liu, P. Goebel, R. Martín-Martín, and S. Savarese, "Robot navigation in constrained pedestrian environments using reinforcement learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 1140–1146.

[10] L. Yuan, X. Gao, Z. Zheng, M. Edmonds, Y. N. Wu, F. Rossano, H. Lu, Y. Zhu, and S.-C. Zhu, "In situ bidirectional human-robot value alignment," *Science robotics*, vol. 7, no. 68, p. eabm4183, 2022.

[11] S. Yang, D. Maturana, and S. Scherer, "Real-time 3d scene layout from a single image using convolutional neural networks," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 2183–2189.

[12] V. Hedau, D. Hoiem, and D. Forsyth, "Recovering the spatial layout of cluttered rooms," in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 1849–1856.

[13] M. Hosseinzadeh, Y. Latif, T. Pham, N. Suenderhauf, and I. Reid, "Structure aware slam using quadrics and planes," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 410–426.

[14] G. Pintore, M. Agus, and E. Gobbetti, "Atlantanet: Inferring the 3d indoor layout from a single 360° image beyond the manhattan world assumption," in *European Conference on Computer Vision*. Springer, 2020, pp. 432–448.

[15] C. Yan, B. Shao, H. Zhao, R. Ning, Y. Zhang, and F. Xu, "3d room layout estimation from a single rgb image," *IEEE Transactions on Multimedia*, vol. 22, no. 11, pp. 3014–3024, 2020.

[16] H. Zhao, M. Lu, A. Yao, Y. Guo, Y. Chen, and L. Zhang, "Physics inspired optimization on semantic transfer features: An alternative method for room layout estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 10–18.

[17] C. Zou, J.-W. Su, C.-H. Peng, A. Colburn, Q. Shan, P. Wonka, H.-K. Chu, and D. Hoiem, "Manhattan room layout reconstruction from a single 360° image: A comparative study of state-of-the-art methods," *International Journal of Computer Vision*, vol. 129, no. 5, pp. 1410–1431, 2021.

[18] A. G. Schwing and R. Urtasun, "Efficient exact inference for 3d indoor scene understanding," in *European conference on computer vision*. Springer, 2012, pp. 299–313.

[19] C. Liu, A. G. Schwing, K. Kundu, R. Urtasun, and S. Fidler, "Rent3d: Floor-plan priors for monocular layout estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3413–3421.

[20] W. Zhang, Y. Zhang, R. Song, Y. Liu, and W. Zhang, "3d layout estimation via weakly supervised learning of plane parameters from 2d segmentation," *IEEE Transactions on Image Processing*, vol. 31, pp. 868–879, 2021.

[21] S. Huang, S. Qi, Y. Zhu, Y. Xiao, Y. Xu, and S.-C. Zhu, "Holistic 3d scene parsing and reconstruction from a single rgb image," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 187–203.

[22] M. Hirzer, V. Lepetit, and P. ROTH, "Smart hypothesis generation for efficient and robust room layout estimation," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2020, pp. 2912–2920.

[23] W. Zhang, W. Zhang, and Y. Zhang, "Geolayout: Geometry driven room layout estimation based on depth maps of planes," in *European Conference on Computer Vision*. Springer, 2020, pp. 632–648.

[24] H. J. Lin and S.-H. Lai, "Deeproom: 3d room layout and pose estimation from a single image," in *Asian Conference on Pattern Recognition*. Springer, 2019, pp. 719–733.

[25] Y. Zhao and S.-C. Zhu, "Scene parsing by integrating function, geometry and appearance models," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 3119–3126.

[26] C. Fernandez-Labrador, J. M. Facil, A. Perez-Yus, C. Demonceaux, J. Civera, and J. J. Guerrero, "Corners for layout: End-to-end layout recovery from 360 images," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1255–1262, 2020.

[27] B. Solarte, Y.-C. Liu, C.-H. Wu, Y.-H. Tsai, and M. Sun, "360-dfpe: Leveraging monocular 360-layouts for direct floor plan estimation," *IEEE Robotics and Automation Letters*, 2022.

[28] C. Fernandez-Labrador, A. Perez-Yus, G. Lopez-Nicolas, and J. J. Guerrero, "Layouts from panoramic images with geometry and deep learning," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3153–3160, 2018.

[29] X. Chen, H. Zhao, G. Zhou, and Y.-Q. Zhang, "Pq-transformer: Jointly parsing 3d objects and layouts from point clouds," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2519–2526, 2022.

[30] I. Radosavovic, P. Dollár, R. Girshick, G. Gkioxari, and K. He, "Data distillation: Towards omni-supervised learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4119–4128.

[31] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5828–5839.

[32] A. Dehghan, G. Baruch, Z. Chen, Y. Feigin, P. Fu, T. Gebauer, D. Kurz, T. Dimry, B. Joffe, A. Schwartz *et al.*, "Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data." in *NeurIPS Datasets and Benchmarks*, 2021.

[33] P. V. Tran, "Sslayout360: Semi-supervised indoor layout estimation from 360deg panorama," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 353–15 362.

[34] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *Advances in neural information processing systems*, vol. 30, 2017.

[35] K. Liu, Y. Zhao, Z. Gao, and B. M. Chen, "Weaklabel3d-net: A complete framework for real-scene lidar point clouds weakly supervised multi-tasks understanding," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 5108–5115.

[36] S. Deng, Q. Dong, B. Liu, and Z. Hu, "Superpoint-guided semi-supervised semantic segmentation of 3d point clouds," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 9214–9220.

[37] B. Tian, L. Luo, H. Zhao, and G. Zhou, "Vibus: Data-efficient 3d scene parsing with viewpoint bottleneck and uncertainty-spectrum modeling," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 194, pp. 302–318, 2022.

[38] L. Yi, B. Gong, and T. Funkhouser, "Complete & label: A domain adaptation approach to semantic segmentation of lidar point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 15 363–15 373.

[39] S. Huang, Y. Xie, S.-C. Zhu, and Y. Zhu, "Spatio-temporal self-supervised representation learning for 3d point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6535–6545.

[40] G. Ding, M. Zhang, E. Li, and Q. Hao, "Jst: Joint self-training for unsupervised domain adaptation on 2d&3d object detection," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 477–483.

[41] Z. Zhang, Y. Ji, W. Cui, Y. Wang, H. Li, X. Zhao, D. Li, S. Tang, M. Yang, W. Tan *et al.*, "Atf-3d: Semi-supervised 3d object detection with adaptive thresholds filtering based on confidence and distance," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 573–10 580, 2022.

[42] J. Lv, Q. Yu, L. Shao, W. Liu, W. Xu, and C. Lu, "Sagci-system: Towards sample-efficient, generalizable, compositional, and incremental robot learning," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 98–105.

[43] P. Li and H. Zhao, "Monocular 3d detection with geometric constraint embedding and semi-supervised training," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5565–5572, 2021.

[44] K. Wang and S. Shen, "Semi-supervised learning: Structure, reflectance and lighting estimation from a night image pair," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 976–983, 2021.

[45] T. Zhao, N. Zhang, X. Ning, H. Wang, L. Yi, and Y. Wang, "Codedvtr: Codebook-based sparse voxel transformer with geometric guidance," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1435–1444.

[46] Q. Li, Y. Wang, Y. Wang, and H. Zhao, "Hdmapnet: An online hd map construction and evaluation framework," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4628–4634.

[47] T. Zhang, X. Chen, Y. Wang, Y. Wang, and H. Zhao, "Mutr3d: A multi-camera tracking framework via 3d-to-2d queries," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4537–4546.

[48] Y. Liu, Y. Liu, C. Jiang, K. Lyu, W. Wan, H. Shen, B. Liang, Z. Fu, H. Wang, and L. Yi, "Hoi4d: A 4d egocentric dataset for category-level human-object interaction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 21 013–21 022.

[49] A. Ghosh and A. H. Thiery, "On data-augmentation and consistency-based semi-supervised learning," in *International Conference on Learning Representations*, 2020.

[50] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, pp. 373–440, 2020.

[51] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, "Semi-supervised learning with ladder networks," *Advances in neural information processing systems*, vol. 28, 2015.

[52] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," in *International Conference on Learning Representations*, 2017.

[53] J. Yin, J. Fang, D. Zhou, L. Zhang, C.-Z. Xu, J. Shen, and W. Wang, "Semi-supervised 3d object detection with proficient teachers," *arXiv preprint arXiv:2207.12655*, 2022.

[54] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *Advances in neural information processing systems*, vol. 33, pp. 596–608, 2020.

[55] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, and T. Shinozaki, "Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18 408–18 419, 2021.

[56] G. Li, X. Li, Y. Wang, S. Zhang, Y. Wu, and D. Liang, "Pseco: Pseudo labeling and consistency training for semi-supervised object detection," *arXiv preprint arXiv:2203.16317*, 2022.

[57] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[58] H. Zhao, M. Lu, A. Yao, Y. Guo, Y. Chen, and L. Zhang, "Pointly-supervised scene parsing with uncertainty mixture," *Computer Vision and Image Understanding*, vol. 200, p. 103040, 2020.

[59] A. Avetisyan, T. Khanova, C. Choy, D. Dash, A. Dai, and M. Nießner, "Scenecad: Predicting object alignments and layouts in rgb-d scans," in *European Conference on Computer Vision*. Springer, 2020, pp. 596–612.

[60] N. Zhao, T.-S. Chua, and G. H. Lee, "Sess: Self-ensembling semi-supervised 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 079–11 087.