360-MLC: Multi-view Layout Consistency for Self-training and Hyper-parameter Tuning

Anonymous Author(s)
Affiliation
Address
email

Abstract

We present 360-MLC, a self-training method based on multi-view layout consistency for finetuning monocular room-layout models using unlabeled 360-images only. This comes in handy in practical scenarios where a pre-trained model needs to be adapted to a new data domain without using any ground truth annotations. Our simple yet effective assumption is that multiple layout estimations in the same scene must define a consistent geometry regardless of their camera positions. Based on this idea, we leverage a pre-trained model to project estimated layout boundaries from several camera views into the 3D world coordinate. Then, we re-project them back to the spherical coordinate and build a probability function, from which we sample the pseudo-labels for self-training. To handle unconfident pseudo-labels, we evaluate the variance in the re-projected boundaries as an uncertainty value to weight each pseudo-label in our loss function during training. In addition, since ground truth annotations are not available during training nor in testing, we leverage the entropy information in multiple layout estimations as a quantitative metric to measure the geometry consistency of the scene, allowing us to evaluate any layout estimator for hyper-parameter tuning, including model selection without ground truth annotations. Experimental results show that our solution achieves favorable performance against state-of-the-art methods when self-training from three publicly available source datasets to a unique, newly labeled dataset consisting of multi-view images of the same scenes.

1 Introduction

Room-layout geometry is one of the fundamental geometry representations for an indoor scene, which can be parameterized with points and lines describing corners and wall boundaries. Therefore, this geometry has been largely used as a primary stage for challenging tasks like robot localization [5, 42], scene understanding [24], floor plan estimation [29], etc. Several methods have been proposed for estimating layout geometry from imagery [12, 13, 45], while the current state-of-the-art methods [19, 10, 17, 47, 15] leverage deep learning approaches to regress the wall-ceiling and floor boundaries directly from monocular 360-images in a supervised manner.

However, deploying a room-layout model using 360-images in a new target domain remains a challenging problem. For instance, a pre-trained layout model may estimate inconsistent geometry, due to novel view positions, different lighting conditions, or severe object occlusions in the scene, especially for large and complex rooms. To handle these issues, ground truth annotations in the new target domain are usually required to finetune the model, which involves a cumbersome data labeling process. Moreover, considering the large variety of indoor scene styles, a room-layout model may demand more new labeled data to adapt to another target domain. In this paper, we aim to make a pre-trained room-layout model self-trainable in the new target domain without using any annotated

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

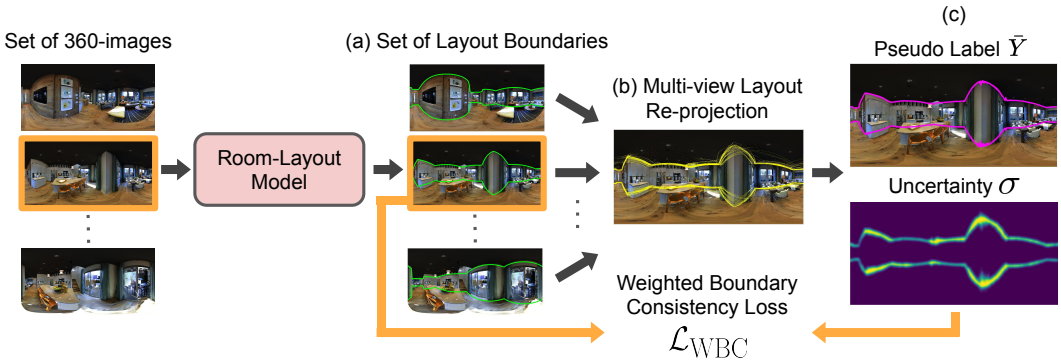


Figure 1: **360-MLC pipeline.** (a) Our method uses a pre-trained room-layout model and collects the layout estimations from multiple views in the same scenes; (b) we re-project multi-view estimates to the target view (the image with the orange bounding box); (c) 360-MLC then generates the pseudo-label and the uncertainty to compute the consistency loss as self-supervisions without requiring ground truth labels in the target scene.

data during model finetuning, in which such setting is practical but has not been studied widely for room-layout estimation.

To this end, we present 360-MLC, a self-training method based on Multi-view Layout Consistency (MLC), capable of adapting a pre-trained layout model into a new data domain using solely registered 360-images. Our main assumption is that multiple layout estimates must define together a consistent scene geometry regardless of their camera positions. Based on this idea, we project several estimated layout boundaries into the world coordinate with the Euclidean space and re-project them back to a target view (spherical coordinate) for building a probability function, from where we sample the most likely points as pseudo-labels. In addition, we evaluate the variance in the re-projected boundaries as an uncertainty measure to weight our pseudo-labels for unconfident boundary regions. We combine both our pseudo-label and its uncertainty into the proposed *weighted boundary consistency loss* ($\mathcal{L}_{\text{WBC}}$), which allows us to finetune a layout model in a self-training manner without requiring any ground truth annotations. Therefore, pseudo-labels generated via our MLC have higher quality than pseudo-labels predicted from a single view, which is crucial for achieving satisfactory performance when no annotations are available in the new domain.

We further propose the *multi-view layout consistency metric* based on entropy information ($\mathcal{H}_{\text{MLC}}$) that quantitatively evaluates any layout predictor without requiring ground truth labels. This metric can be used to monitor the quality of the predicted layouts in both training and testing. As a result, our $\mathcal{H}_{\text{MLC}}$ metric can be used for hyper-parameter tuning and model selection, which is practical when no ground truths are available in the new domain. Our key idea behind $\mathcal{H}_{\text{MLC}}$ is to evaluate the entropy information from multiple layouts projected into a discrete grid as a 2D density function, where a better geometry alignment of layouts would yield a lower entropy evaluation. We leverage our $\mathcal{H}_{\text{MLC}}$ metric across different experiments and show its versatility and reliability for the multi-view layout setting.

In experiments, we leverage the MP3D-FPE [29] multi-view dataset as our unlabeled new domain. Since layout ground truths are not available in this dataset, we manually annotate 7 scenes with 700 frames for only the performance evaluation purpose. For the dataset used in model pre-training, we validate our framework on three real-world room-layout estimation datasets: MatterportLayout [36, 48], Zillow Indoor Dataset (ZInD) [9], and the dataset used in LayoutNet [47]. We show that our method is able to handle two practical settings during training: 1) having both the labeled pre-trained data and the unlabeled data in the target domain, 2) only providing the unlabeled data with the pre-trained model. Moreover, we demonstrate the usefulness of the proposed modules, including the weighted boundary consistency loss with pseudo-labels, and the multi-view layout consistency metric that facilitates the hyper-parameter tuning without the need for ground truth labels.

Our contributions based on the idea of Multi-view Layout Consistency are summarized as follows:

1. Based on multi-view geometry consistency, we propose a self-training framework for room-layout estimation, requiring only registered 360-images as the input.

2. We propose the *weighted boundary consistency loss function* ($\mathcal{L}_{\text{WBC}}$) that uses multiple estimated layout boundaries with their uncertainty to improve the self-training process using pseudo-labels.

3. We introduce the *multi-view layout consistency metric* ($\mathcal{H}_{\text{MLC}}$) for measuring multi-view layout geometry consistency based on entropy information. This metric allows us to evaluate any layout estimator quantitatively without requiring ground truth labels, which enables hyper-parameter tuning and model selection.

2 Related Work

Indoor room layout estimation. Estimating the layout structure of a room for cluttered indoor environments is a challenging task. Early methods estimate plane surfaces and their orientations based on points or edge features to construct the spatial layouts for perspective images [12, 13, 35] or panorama images (i.e. 360-images) [45, 41]. On the other hand, some approaches [3, 14, 7, 40] leverage semantic cues in the scene, such as objects or humans, to improve the layout estimation.

Deep learning approaches [19, 10, 17, 47, 15] leverage convolutional neural networks (CNN) to extract the geometric cues (e.g., corners or edges) and semantic cues (e.g., pixel-wise segmentation), which largely enhance the performance. Recent state-of-the-art methods [43, 30, 31, 37] are able to robustly estimate the layout of the whole room from a single panorama image. CFL [11] and AtlantaNet [21] avoid the commonly-used Manhattan world assumption [8], enabling the ability to handle complex room shapes. In addition to monocular approaches, MVLayoutNet [16] and PSMNet [38] explore the usage of multi-view 360-images as inputs to further improve the layout estimations. However, training deep neural networks to perform layout estimation requires large-scale datasets with manual annotations. In this paper, we aim to mitigate this problem by learning from unlabeled data.

Self-training. To incorporate unlabeled data during training, self-training [27, 44, 23] uses a pre-trained teacher model to generate pseudo-labels for the data without ground truth annotations. Then, both the labeled data and the unlabeled data with pseudo-labels are used to train a better model (i.e., student model). Due to its simplicity, many attempts have been made for the field of semi-supervised learning [18, 2, 4, 28] and unsupervised domain adaptation [49, 50]. Moreover, recent methods [39, 46] show that self-training can surpass state-of-the-art fully-supervised models on large-scale datasets such as ImageNet.

However, most of these works focus on self-training for classification or object detection tasks, while self-training for tasks considering geometric predictions (e.g., room-layout estimation studied in this paper) is rarely explored. SSLayout360 [34] trains a layout estimation model with pseudo-labels produced by the Mean Teacher [33]. Yet, the simple extension from classification to layout estimation considers each image independently, ignoring the geometry information coming from other camera views. In addition, another challenge is that pseudo-labels are usually noisy. Previous methods attempt to reduce the noise by ensembling multiple predictions for an image under different augmentations [4, 22] or by selecting only the pseudo-labels with high confidence [28]. In this paper, we leverage multi-view consistency from the layout estimations and measure their uncertainty to construct reliable pseudo-labels.

Unsupervised model validation. Without ground truth annotations, how to evaluate a machine learning model remains an open issue. Traditional unsupervised learning algorithms (e.g., clustering) can be evaluated without external labels through computing cohesion and separation [32]. For unsupervised domain adaptations, where the problem assumes no labeled data is available in the target domain, unsupervised validation is more practical for hyper-parameter tuning [20, 25]. [20] evaluates the confidence of the predictions of the classifier using entropy. [25] proposes the soft neighborhood density, measuring the local similarity between data samples (the higher, the better). However, these metrics are designed for classification or segmentation tasks and cannot easily be adopted by our tasks. Therefore, in this paper, we study unsupervised validation from another perspective: evaluating the geometry consistency between multiple views, requiring no ground truth annotations.

3 Our Approach: 360-MLC

Our primary goal is to deploy a model pre-trained on source domain into a new dataset (target domain), where the data distribution may differs from the one used in the pre-trained model. We assume that several images in the new scene are captured and registered by their camera poses, but their layout ground truth annotations are not available. Under this practical scenario, we present 360-MLC, a self-training method that is based on multi-view layout consistency.

For illustration purposes, Fig. 1 presents the overview of our method. First, we begin with a set of 360-images and a pre-trained room-layout model that can generate a set of estimated layout boundaries (see green lines in Fig. 1-(a)). Then, by leveraging the related camera poses of every image, all layout boundaries are projected into the world coordinate and then re-projected them back into a target view (see yellow lines in Fig. 1-(b)). Details of these projections are described in §3.1.

Assuming that the projected boundaries from all the views describe the same scene geometry, we can compute the most likely layout boundary positions as pseudo-labels for self-training along with their uncertainty based on the variance (see Fig. 1-(c)). Upon the estimated pseudo-labels, we define our weighted boundary consistency loss $\mathcal{L}_{\text{WBC}}$, allowing us to define a reliable regularization used for self-training. More details are presented in §3.2.

A challenging step towards our goal is the lack of a metric for evaluating a layout estimator when no ground truth annotations are available. To tackle this issue, §3.3 describes our multi-view layout consistency metric $\mathcal{H}_{\text{MLC}}$, which allows us to evaluate multiple layout estimates without requiring any ground truth. We leverage the proposed metric throughout all the experiments described in §4 as the quantitative metric for model selection and hyper-parameter tuning.

3.1 Multi-view Layout Re-projection

In this section, we describe the multi-view layout re-projection process of all estimated layout boundaries from different cameras positions in the scene. To this end, we define the set of 360-images and boundaries in the scene as follows:

$$\{(I_i, Y_i)\}_{i=1:N}, \quad I_i \in \mathbb{R}^{H\times W}, \quad Y_i \in \mathbb{R}^{W} \ , \tag{1}$$

where I_i is the i-th 360-image with the size of W columns times H rows pixels, Y_i is the layout boundary of image I_i, and N is the number of images. $Y_i \in \mathbb{R}^W$ is a vector of boundaries at all columns, where $Y_i(\theta) = \phi$ specifies that the layout boundary is at row ϕ for column θ in the pixel coordinate.

For a supervised training, Y_i is given as a ground truth label. However, in our proposed self-training framework, we aim to ensemble pseudo-labels through geometry re-projection of multiple views. To begin with, we describe the process to project the boundary Y_i into the j-th target view as the re-projected boundary $Y_{i\to j}$. The projection of Y_i into the world coordinate using the Euclidean space is described as follows:

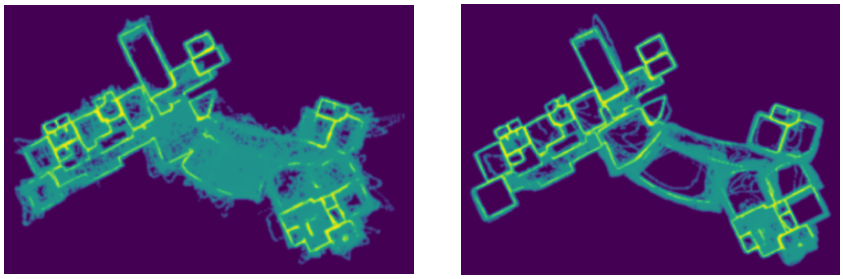
$$X_i = \text{Proj}(Y_i, T_i, h_i) \ , \tag{2}$$

where X_i is the projected layout boundary in the world coordinate; h_i is the camera height; $T_i \in \text{SE}(3)$ is the camera pose with respect to the world coordinate; $\text{Proj}(\cdot)$ is the layout projection for spherical cameras that maps spherical coordinates $[\theta, \phi]^\top \in \mathbb{R}^2$ into the 3D world coordinate $[x, y, z]^\top \in \mathbb{R}^3$. Details of this projection function $\text{Proj}(\cdot)$ and how we handle unknown camera heights with estimated poses are presented in the supplementary material.

Then, X_i in the world coordinate can be re-projected into a j-th target view ($Y_{i\to j}$) as follows:

$$Y_{i\to j} = \text{Proj}^{-1}(T_j, X_i) \ , \tag{3}$$

where $\text{Proj}^{-1}(\cdot)$ is the inverse of the projection function presented in Eq. (2). We collect all re-projected boundaries into $\mathbf{Y}_j = [Y_{1\to j}; \ldots; Y_{N\to j}] \in \mathbb{R}^{W\times N}$, where $\mathbf{Y}_j(\theta) \in \mathbb{R}^N$ is a vector of row positions on boundaries from N views at column θ. As shown in the yellow lines in Fig. 1-(b), the visualization of $\mathbf{Y}_j$ reveals the underlying geometry of the scene. More visualization results are presented in the supplementary material.



(a) Pre-trained model (b) After our self-training

Figure 2: **2D density function of multi-view layouts.** We project multiple layout boundaries into a top-view 2D density map that reveals the geometry consistency in the scene. (a) We present the projection of multi-view layout using a pre-trained model before self-training. (b) We observe a better layout consistency of the scene after our self-training process. In §3.3, we use such a top-view map for calculating our layout consistency metric $\mathcal{H}_{\text{MLC}}$

3.2 Weighted Boundary Consistency Loss

In this section, we describe how multiple re-projected boundaries can be used to estimate pseudo-label boundaries and its uncertainty as a reliable supervision for our self-training formulation. First, we obtain a set of predicted boundaries $\{Y_i = \mathcal{M}(I_i)\}_{i\in 1:N}$ for all views using the pre-trained model $\mathcal{M}$. Following the process in §3.1, we re-project the estimated boundaries into $\mathbf{Y}_j$ for each view. We define the pseudo-label ($\bar{Y}_j$) and its uncertainty (σ_j) as follows:

$$\bar{Y}_j = [\text{Median}(\mathbf{Y}_j(\theta))]_{\theta=1:W} \in \mathbb{R}^W \ , \quad \sigma_j = [\text{STD}(\mathbf{Y}_j(\theta))]_{\theta=1:W} \in \mathbb{R}^W \ , \tag{4}$$

where Median$(\cdot)$ and STD$(\cdot)$ are the median and standard deviation both applied to $\mathbf{Y}_j(\theta)$ (i.e., N re-projections of row positions on boundary at column θ). The pseudo-labels are computed for each target view. By leveraging Eq. (4), we define our geometry loss as follows:

$$\mathcal{L}_{\text{WBC}} = \sum_{j=1}^{N} \sum_{\theta=1}^{W} \frac{||Y_j(\theta) - \bar{Y}_j(\theta)||_1}{\sigma_j^2(\theta)} \ . \tag{5}$$

$\mathcal{L}_{\text{WBC}}$ is our proposed weighted boundary consistency loss for finetuning the room-layout model $\mathcal{M}$. Note that the denominator weights the pseudo-label accordingly as the uncertainty at each column θ. This design aims to reduce the effect of unstable predictions in the scene (e.g., drastic occlusions) that generally incur a larger variance in the re-projection due to the inconsistency between multiple views.

3.3 Multi-view Layout Consistency Metric

In practice, one challenge for adapting a pre-trained model into a completely unlabeled new data domain is that there is no labelled hold-out dataset for tuning hyper-parameters such as learning rate, etc. Hence, we need a metric to measure the performance based on unlabeled data. To this end, we propose a the $\mathcal{H}_{\text{MLC}}$ metric to measure the multi-view layout consistency that does not rely on ground truth labels but on model outputs only.

First, we collect all estimated layout boundaries in the scene projected by Eq. (2) to build a top-view 2D density map. This can be described as follows:

$$\mathbf{X} = \{X_i\}_{i=1:N}, \quad \Phi(\mathbf{X}) \in \mathbb{R}^{U\times V}, \tag{6}$$

where $\mathbf{X}$ is the set of all estimated layouts in the scene, and $\Phi(\cdot)$ is a top-down projection function that maps $\mathbf{X}$ into a discrete 2D-grid with the size $U \times V$ as a normalized histogram. Our key idea is to evaluate the entropy in this discrete grid as follows:

$$\mathcal{H}_{\text{MLC}} = \sum_{u,v} -\Phi_{u,v}(\mathbf{X}) \cdot \log \Phi_{u,v}(\mathbf{X}). \tag{7}$$

The intuition behind this evaluation comes from the fact that better alignment of layout boundaries yield in a lower entropy evaluation, while a higher entropy value would reflect the poor alignment of

layout geometry between multiple views. As a result, we can use $\mathcal{H}_{\text{MLC}}$ to select hyper-parameters and early stop the model training, without using any ground truth annotations. For illustration purposes, Fig. 2 presents two projected scenes that show how the geometry consistency correlates with a less disordered 2D projection. More details are presented in the supplementary material.

4 Experiments

4.1 Experimental Setup

Datasets. We conduct extensive experiments using publicly available 360-image layout datasets: Matterport3D Floor Plan Estimation (MP3D-FPE) [29] as the target dataset, and three real-world datasets as the pre-training datasets, including MatterportLayout [36, 48], Zillow Indoor Dataset (ZInD) [9], and the dataset used in LayoutNet [47] that combines PanoContext [45] and Stanford2D3D [1] (referred to as the LayoutNet dataset for simplicity). MP3D-FPE is collected using the MINOS simulator [26] to render sequences of 360-images within each scene/room from the Matterport3D dataset [6]. Note that it is the only dataset containing multi-view 360-images and thus we consider it as the target dataset.

In experiments, we aim to pre-train the layout model using each of the pre-trained dataset, and then we utilize the target MP3D-FPE dataset for self-training and performing evaluation. We follow the standard training split released in each pre-trained dataset. For the target dataset MP3D-FPE, we use 2094 frames as the training set and 700 frames as the testing set, from 32 and 7 scenes, respectively, where those scenes are not included in the training set of the Matterport3D dataset. On average, there are 10.46 views within each room. Note that the target MP3D-FPE dataset is a challenging dataset since it includes many complex scenes that do not follow the Manhattan assumption [8], and we carefully label the ground truth 2D and 3D layout for the testing set. As a result, compared to other datasets, the performance scores on MP3D-FPE are lower when the model is evaluated using state-of-the-art methods.

Evaluation metrics. We follow Zou *et al.* [48] to construct the four standard protocols for evaluation. To evaluate layout boundary, we use 2D and 3D intersection-of-union (IoU). To evaluate layout depth, we use root-mean-square error (RMSE) by setting the camera height as 1.6 meters, and δ_1, which describes the percentage of pixels where the ratio between the estimation and the ground truth depth is within the threshold of 1.25. In addition, we introduce a new metric $\mathcal{H}_{\text{MLC}}$ outlined in §3.3 to analyze the consistency between multiple layout estimations. We show in §4.3 that $\mathcal{H}_{\text{MLC}}$ is highly co-related to 2D/3D IoUs on the hold-out dataset. Hence, $\mathcal{H}_{\text{MLC}}$ is suitable for hyper-parameter tuning when ground truths are not available.

Implementation details. We adopt HorizonNet [30] as our layout estimation backbone due to its state-of-the-art performance. Note that, different from the original HorizonNet, our model is trained without the supervision on the corner channel since our pseudo-label contains only the boundary information. Common data augmentation techniques for 360-images are also applied during training, including left-right flipping, panoramic horizontal rotation, and luminance augmentation. We use the Adam optimizer to train the model for 300 epochs by setting the learning rate as 0.0001 and the batch size as 4. We save the model every 5 epochs and the early-stopped model is selected based on the lowest $\mathcal{H}_{\text{MLC}}$ score on the hold-out test dataset for all methods. All models are trained on a single NVIDIA TITAN X GPU with 12 GB of memory. We will make our models, codes, and dataset available to the public.

4.2 Experimental Results

Baselines. The proposed framework is compared against our re-implementation of SSLayout360 [34]. We ensure that our re-implemented SSLayout360* has similar performance compared to the reported results in [34]. Note that we use the same backbone, i.e., HorizonNet, for both our 360-MLC and SSLayout360*, and we train models using only the boundary supervision as stated in §4.1. We implement an extended self-training version based on SSLayout360*, namely SSLayout360-ST*, in which we disable the supervised loss and train without any ground truth annotations. We initialize

Table 1: Dataset statistics.

Source Dataset	Number of Frames	Target Dataset	Number of Frames
MatterportLayout [48]	2094	MP3D-FPE [29]	2094
ZInD [9]	2094	MP3D-FPE	2094
LayoutNet [47]	817	MP3D-FPE	817

Table 2: Evaluation results of Setting 1 on MP3D-FPE [29].

Pre-trained Dataset	Method	2D IoU (%) ↑	3D IoU (%) ↑	RMSE ↓	δ_1 ↑	$\mathcal{H}_{\text{MLC}}$ ↓
MatterportLayout [48]	Pre-trained	65.38	62.28	0.58	0.78	8.18
	SSLayout360* [34]	70.53	66.74	0.48	0.82	8.15
	Ours	**71.50**	**67.70**	**0.46**	**0.82**	**8.10**
ZInD [9]	Pre-trained	45.43	42.17	1.02	0.61	8.32
	SSLayout360*	62.62	58.27	0.66	0.74	8.34
	Ours	**65.60**	**60.70**	**0.56**	**0.75**	**8.19**
LayoutNet [47]	Pre-trained	64.34	58.92	0.61	0.70	8.50
	SSLayout360*	67.48	62.89	0.56	**0.77**	**8.29**
	Ours	**69.40**	**65.22**	**0.55**	0.72	8.32

all models with the same pre-trained weights from the official HorizonNet[1] release. In addition, different from our approach, both SSLayout360* and SSLayout360-ST* are trained on two NVIDIA TITAN X GPUs due to the requirement of larger memory for their teacher-student architecture.

Setting 1: labeled pre-training data + unlabeled target data. In this setting, we sample the same amount of training data from both the pre-training dataset and the target dataset, as shown in Table 1. We start from the pre-trained model, and then generate pseudo-labels for unlabeled data. Then, during model finetuning, we include both labeled pre-trained data and pseudo-labeled data from MP3D-FPE. In Table 2, we show that our method consistently performs favorably against SSLayout360* across most metrics, in which SSLayout360* does not consider the usage of multi-view layout consistency as our approach does.

Setting 2: pre-trained model + unlabeled target data. A more practical and challenging setting is that only the pre-trained model and unlabeled data in the target domain are available during model finetuning. In Table 3, similar to Setting 1, our method has consistent performance improvement against SSLayout360-ST*. Note that we observe that our performance gains are larger (especially on ZInD and LayoutNet) compared to Setting 1. For MatterportLayout, since its data distribution is closer to the MP3D-FPE (both are from Matterport3D), the performance difference is less.

Moreover, when we remove the labeled pre-trained data from Setting 1, we do not observe a significant performance drop in Setting 2 on all dataset settings, while SSLayout360-ST* is more sensitive to the labeled pre-trained data, e.g., on ZInD, 2D/3D IoU is decreased by 6.1%/6.15%. This demonstates the effectiveness of considering multi-view layout consistency that generates more reliable pseudo-labels, even when no labeled data is provided during training.

Qualitative results. Fig. 4 shows qualitative results of multiple layouts projected into the 3D world coordinate for two test scenes evaluated on MP3D-FPE [29], following the Setting 2. It can be observed that our proposed 360-MLC presents sharper and clearer layout boundaries for those scenes, demonstrating a better performance compared with the baselines.

4.3 Ablation Study

In this section, we present our ablation study to validate the effectiveness of the proposed components. We conduct experiments using the ZInD pre-trained weights under the Setting 2 mentioned in §4.2.

[1] https://github.com/sunset1995/HorizonNet

Table 3: Evaluation results of Setting 2 on MP3D-FPE [29].

Pre-trained Dataset	Method	2D IoU (%) ↑	3D IoU (%) ↑	RMSE ↓	δ_1 ↑	$\mathcal{H}_{\text{MLC}}$ ↓
MatterportLayout [48]	SSLayout360-ST*	70.58	66.64	0.52	**0.81**	**8.10**
	Ours	**71.50**	**67.20**	**0.49**	0.78	8.14
ZInD [9]	SSLayout360-ST*	56.52	52.12	0.72	0.74	8.20
	Ours	**64.45**	**59.53**	**0.59**	**0.75**	**8.17**
LayoutNet [47]	SSLayout360-ST*	66.14	61.51	0.57	**0.76**	**8.30**
	Ours	**69.12**	**64.62**	0.57	0.69	8.31

Table 4: Ablation study with Setting 2 on MP3D-FPE [29], pre-trained on ZInD [9].

	Frames (%)	Median	Mean	$\mathcal{L}_{\text{WBC}}$	2D IoU (%) ↑	3D IoU (%) ↑	RMSE ↓	δ_1 ↑	$\mathcal{H}_{\text{MLC}}$ ↓
(a)	10	✓	-	✓	57.29	53.44	0.71	0.68	8.23
(b)	50	✓	-	✓	59.55	55.46	0.65	0.69	8.22
(c)	100	-	✓	✓	46.17	42.63	0.82	0.63	8.22
(d)	100	✓	-	-	62.81	58.05	**0.58**	**0.76**	8.18
(e)	100	✓	-	✓	**64.45**	**59.53**	0.59	0.75	**8.17**

Results are presented in Table 4, describing four main experiments. First, we investigate how the number of views used to generate our pseudo-labels may affect the performance of our proposed solution. Second, we replace the Median function in Eq. (4) with Mean, aiming to demonstrate the effects of outliers in the quality of our pseudo-labels. Then, we analyze the proposed $\mathcal{L}_{\text{WBC}}$ with the vanilla $\mathcal{L}_1$ loss function. Lastly, we show how $\mathcal{H}_{\text{MLC}}$ could be used in hyper-parameter tuning, such as selecting learning rates and models.

Multi-view pseudo-labeling. In this experiment, we aim to verify whether the more views of estimations we use, the better quality of the pseudo-labels we can acquire. Therefore, we experiment three models trained with the same amount of data but using pseudo-labels created by 10%, 50% and 100% of frames, as shown in row (a), (b), and (e), respectively in Table 4. The result demonstrates the contribution of our proposed method: using the layout estimations from more views for self-training can help ensemble more robust training signals.

Median and mean in Eq. (4). In this experiment, we investigate the impact of the median operator for pseudo-label generation. We compare the mean and median functions in row (c) and (e) in Table 4. It can be observed that the median function has a better performance since it is capable of ignoring outliers in the re-projected boundaries from multiple views, increasing the robustness of pseudo-labels.

$\mathcal{L}_{\text{WBC}}$ and $\mathcal{L}_1$. The results in row (d) and (e) of Table 4 show that adding the uncertainty σ in $\mathcal{L}_{\text{WBC}}$ performs better than the simple $\mathcal{L}_1$ loss. This is because our proposed loss function down-weights the part of the layout boundaries that are noisy (i.e., high variance), avoiding the effect of unreliable pseudo-labels during self-training.

$\mathcal{H}_{\text{MLC}}$ for hyper-parameter tuning. We present an example of using our proposed metric $\mathcal{H}_{\text{MLC}}$ for hyper-parameter tuning. In Fig. 3, we test three different settings using the same model training process. Among them, using learning rate 1×10^{-5} and $1K$ training samples (green lines) performs the worst, while adopting learning rate 1×10^{-5} and $2K$ training samples (blue lines) performs the best. We show that the trend of our proposed unsupervised metric (Fig. 3-(c)) is consistent with the two supervised metrics, 2D/3D IoU (Fig. 3-(a) and (b)). Therefore, even without ground truth labels, our metric can be served as a robust indication for validation.

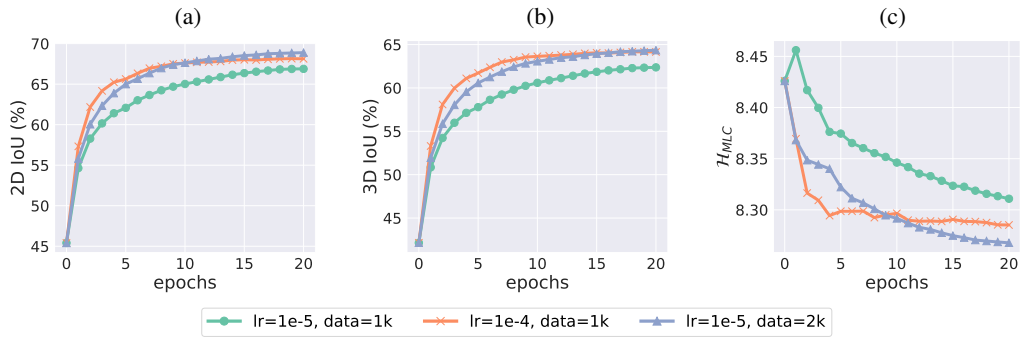


Figure 3: **An example of hyper-parameter tuning.** We show that the layout evaluation metrics, (a) 2D IoU and (b) 3D IoU, evaluated using ground truth labels, are inversely correlated with the proposed metric (c) $\mathcal{H}_{\text{MLC}}$ under three different settings. In this way, we are able to select better models with the lower value of $\mathcal{H}_{\text{MLC}}$. We conduct the experiment under two conditions: different learning rates and different amount of training data.

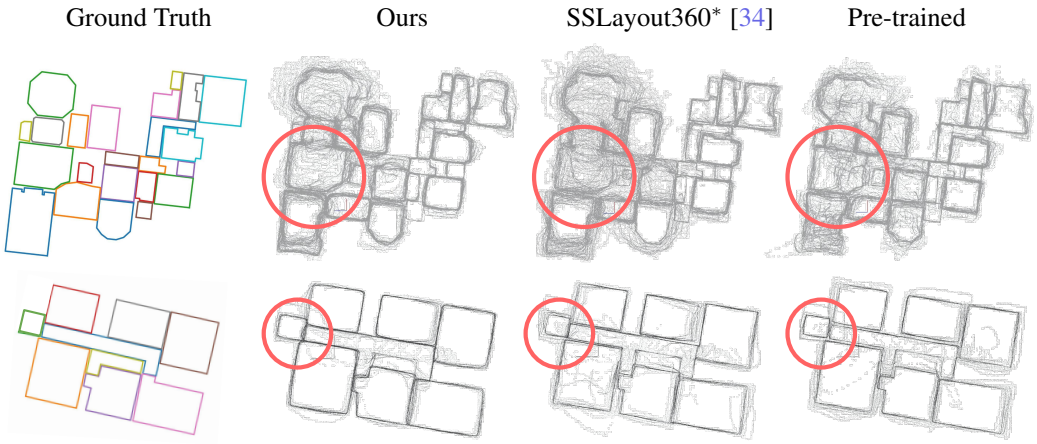


Figure 4: **Qualitative results in the 2D top-view.** We project the estimated layouts registered by the corresponding camera poses into 2D. Comparing to our baseline SSLayout360* [34] and the pre-trained model, our model produces more consistent layouts (highlighted with red circles).

5 Conclusions

We present 360-MLC, a self-training method based on multi-view layout consistency for finetuning monocular 360-layout models using unlabeled data only. Our method tackles a practical scenario where a pre-trained model needs to be adapted to a new data domain without using any ground truth annotations. In addition, we leverage the entropy information in multiple layout estimations as a quantitative metric to measure the geometry consistency of the scene, allowing us to evaluate any layout estimator for hyper-parameter tuning and model selection in an unsupervised fashion. Experimental results show that our self-training solution achieves favorable performance against state-of-the-art methods from three publicly available source datasets to the newly labeled multi-view MP3D-FPF dataset.

References

[1] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017. 6

[2] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. *Advances in neural information processing systems*, 27, 2014. 3

[3] Sid Yingze Bao, Min Sun, and Silvio Savarese. Toward coherent object detection and scene layout understanding. *Image and Vision Computing*, 29(9):569–579, 2011. 3

[4] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[5] Federico Boniardi, Abhinav Valada, Rohit Mohan, Tim Caselitz, and Wolfram Burgard. Robot localization in floor plans using a room layout edge extraction network. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5291–5297. IEEE, 2019. 1

[6] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 6

[7] Yu-Wei Chao, Wongun Choi, Caroline Pantofaru, and Silvio Savarese. Layout estimation of highly cluttered indoor scenes using geometric and semantic cues. In *International Conference on Image Analysis and Processing*, pages 489–499. Springer, 2013. 3

[8] James M Coughlan and Alan L Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 941–947. IEEE, 1999. 3, 6

[9] Steve Cruz, Will Hutchcroft, Yuguang Li, Naji Khosravan, Ivaylo Boyadzhiev, and Sing Bing Kang. Zillow indoor dataset: Annotated floor plans with 360deg panoramas and 3d room layouts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2133–2143, 2021. 2, 6, 7, 8

[10] Saumitro Dasgupta, Kuan Fang, Kevin Chen, and Silvio Savarese. Delay: Robust spatial layout estimation for cluttered indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 616–624, 2016. 1, 3

[11] Clara Fernandez-Labrador, Jose M Facil, Alejandro Perez-Yus, Cédric Demonceaux, Javier Civera, and Jose J Guerrero. Corners for layout: End-to-end layout recovery from 360 images. *IEEE Robotics and Automation Letters*, 5(2):1255–1262, 2020. 3

[12] Alex Flint, Christopher Mei, David Murray, and Ian Reid. A dynamic programming approach to reconstructing building interiors. In *European conference on computer vision*, pages 394–407. Springer, 2010. 1, 3

[13] Alex Flint, David Murray, and Ian Reid. Manhattan scene understanding using monocular, stereo, and 3d features. In *2011 International Conference on Computer Vision*, pages 2228–2235. IEEE, 2011. 1, 3

[14] Abhinav Gupta, Martial Hebert, Takeo Kanade, and David Blei. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. *Advances in neural information processing systems*, 23, 2010. 3

[15] Martin Hirzer, Vincent Lepetit, and PETER ROTH. Smart hypothesis generation for efficient and robust room layout estimation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2912–2920, 2020. 1, 3

[16] Zhihua Hu, Bo Duan, Yanfeng Zhang, Mingwei Sun, and Jingwei Huang. Mvlayoutnet: 3d layout reconstruction with multi-view panoramas. *arXiv preprint arXiv:2112.06133*, 2021. 3

[17] Chen-Yu Lee, Vijay Badrinarayanan, Tomasz Malisiewicz, and Andrew Rabinovich. Roomnet: End-to-end room layout estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 4865–4874, 2017. 1, 3

[18] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013. 3

[19] Arun Mallya and Svetlana Lazebnik. Learning informative edge maps for indoor scene layout prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 936–944, 2015. 1, 3

[20] Pietro Morerio, Jacopo Cavazza, and Vittorio Murino. Minimal-entropy correlation alignment for unsupervised deep domain adaptation. *International Conference on Learning Representations*, 2018. 3

[21] Giovanni Pintore, Marco Agus, and Enrico Gobbetti. Atlantanet: Inferring the 3d indoor layout from a single 360° image beyond the manhattan world assumption. In *European Conference on Computer Vision*, pages 432–448. Springer, 2020. 3

[22] Ilija Radosavovic, Piotr Dollár, Ross Girshick, Georgia Gkioxari, and Kaiming He. Data distillation: Towards omni-supervised learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4119–4128, 2018. 3

[23] Ellen Riloff. Automatically generating extraction patterns from untagged text. In *Proceedings of the national conference on artificial intelligence*, pages 1044–1049, 1996. 3

[24] Antoni Rosinol, Arjun Gupta, Marcus Abate, Jingnan Shi, and Luca Carlone. 3D Dynamic Scene Graphs: Actionable Spatial Perception with Places, Objects, and Humans. In *Proceedings of Robotics: Science and Systems*, July 2020. 1

[25] Kuniaki Saito, Donghyun Kim, Piotr Teterwak, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Tune it the right way: Unsupervised validation of domain adaptation via soft neighborhood density. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9184–9193, 2021. 3

[26] Manolis Savva, Angel X Chang, Alexey Dosovitskiy, Thomas Funkhouser, and Vladlen Koltun. Minos: Multimodal indoor simulator for navigation in complex environments. *arXiv preprint arXiv:1712.03931*, 2017. 6

[27] Henry Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965. 3

[28] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33:596–608, 2020. 3

[29] Bolivar Solarte, Yueh-Cheng Liu, Chin-Hsuan Wu, Yi-Hsuan Tsai, and Min Sun. 360-dfpe: Leveraging monocular 360-layouts for direct floor plan estimation. *IEEE Robotics and Automation Letters*, pages 1–1, 2022. 1, 2, 6, 7, 8

[30] Cheng Sun, Chi-Wei Hsiao, Min Sun, and Hwann-Tzong Chen. Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1047–1056, 2019. 3, 6

[31] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Hohonet: 360 indoor holistic understanding with latent horizontal features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2573–2582, 2021. 3

[32] Pang-Ning Tan, Micahel Steinbach, and Vipin Kumar. Introduction to data mining, pearson education. *Inc., New Delhi*, 2006. 3

[33] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017. 3

[34] Phi Vu Tran. Sslayout360: Semi-supervised indoor layout estimation from 360° panorama. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15348–15357. IEEE Computer Society, 2021. 3, 6, 7, 9

[35] Grace Tsai, Changhai Xu, Jingen Liu, and Benjamin Kuipers. Real-time indoor scene understanding using bayesian filtering with motion cues. In *2011 International Conference on Computer Vision*, pages 121–128. IEEE, 2011. 3

[36] Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. Layoutmp3d: Layout annotation of matterport3d. *arXiv preprint arXiv:2003.13516*, 2020. 2, 6

[37] Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. Led2-net: Monocular 360deg layout estimation via differentiable depth rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12956–12965, 2021. 3

[38] Haiyan Wang, Will Hutchcroft, Yuguang Li, Zhiqiang Wan, Ivaylo Boyadzhiev, Yingli Tian, and Sing Bing Kang. Psmnet: Position-aware stereo merging network for room layout estimation. *arXiv preprint arXiv:2203.15965*, 2022. 3

[39] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020. 3

[40] Jiu Xu, Björn Stenger, Tommi Kerola, and Tony Tung. Pano2cad: Room layout from a single panorama image. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 354–362. IEEE, 2017. 3

[41] Hao Yang and Hui Zhang. Efficient 3d room shape recovery from a single panorama. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5422–5430, 2016. 3

[42] Shichao Yang and Sebastian Scherer. Monocular object and plane slam in structured environments. *IEEE Robotics and Automation Letters*, 4(4):3145–3152, 2019. 1

[43] Shang-Ta Yang, Fu-En Wang, Chi-Han Peng, Peter Wonka, Min Sun, and Hung-Kuo Chu. Dula-net: A dual-projection network for estimating room layouts from a single rgb panorama. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3363–3372, 2019. 3

[44] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196, 1995. 3

[45] Yinda Zhang, Shuran Song, Ping Tan, and Jianxiong Xiao. Panocontext: A whole-room 3d context model for panoramic scene understanding. In *European conference on computer vision*, pages 668–686. Springer, 2014. 1, 3, 6

[46] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. Rethinking pre-training and self-training. *Advances in neural information processing systems*, 33:3833–3845, 2020. 3

[47] Chuhang Zou, Alex Colburn, Qi Shan, and Derek Hoiem. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2051–2059, 2018. 1, 2, 3, 6, 7, 8

[48] Chuhang Zou, Jheng-Wei Su, Chi-Han Peng, Alex Colburn, Qi Shan, Peter Wonka, Hung-Kuo Chu, and Derek Hoiem. 3d manhattan room layout reconstruction from a single 360 image. 2019. 2, 6, 7, 8

[49] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305, 2018. 3

[50] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5982–5991, 2019. 3

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes]
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [No] It is clear from the problem definition.
   (c) Did you discuss any potential negative societal impacts of your work? [No] There is no such issue.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] will do after the publication.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No] please referred to the cited papers.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] will do after the publication.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] please referred to the cited papers.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] please referred to the cited papers.
5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]