PanoMixSwap – Panorama Mixing via Structural Swapping for Indoor Scene Understanding

Yu-Cheng Hsieh
sphinx5912@gapp.nthu.edu.tw
Cheng Sun
chengsun@gapp.nthu.edu.tw
Suraj Dengale
surajdengale@gapp.nthu.edu.tw
Min Sun
sunmin@ee.nthu.edu.tw

Vision Science Lab
National Tsing Hua University
Hsinchu, Taiwan

Abstract

The volume and diversity of training data are critical for modern deep learning-based methods. Compared to the massive amount of labeled perspective images, 360° panoramic images fall short in both volume and diversity. In this paper, we propose PanoMixSwap, a novel data augmentation technique specifically designed for indoor panoramic images. PanoMixSwap explicitly mixes various background styles, foreground furniture, and room layouts from the existing indoor panorama datasets and generates a diverse set of new panoramic images to enrich the datasets. We first decompose each panoramic image into its constituent parts: background style, foreground furniture, and room layout. Then, we generate an augmented image by mixing these three parts from three different images, such as the foreground furniture from one image, the background style from another image, and the room structure from the third image. Our method yields high diversity since there is a cubical increase in image combinations. We also evaluate the effectiveness of PanoMixSwap on two indoor scene understanding tasks: semantic segmentation and layout estimation. Our experiments demonstrate that state-of-the-art methods trained with PanoMixSwap outperform their original setting on both tasks consistently. The website for this paper can be found at *https://yuchenghsieh.github.io/PanoMixSwap*.

1 Introduction

Panoramic images have become increasingly popular in indoor scene understanding tasks because they provide a comprehensive 360° view of a specific room. With the widespread availability of 360° cameras, generating panoramic images has become more convenient. This inspired the development of various indoor panoramic datasets such as Stanford2D3D [2], Matterport3D [3], PanoContext [33] and Structured3D [34], as well as the emergence of related tasks such as semantic segmentation, layout estimation, and depth estimation. These

tasks leverage the unique characteristics of indoor panoramic images to enable a more holistic and immersive understanding of indoor environments.

Despite the availability of indoor panoramic datasets, these images are limited in volume and diversity compared to perspective images. For example, even Stanford2D3D [4], one of the largest real-world indoor panoramic datasets, contains only 1,413 panoramic images across 270 scene layouts. This scarcity of data presents difficulties in training models that require both robustness and accuracy. To address this issue, data augmentation techniques are often employed to artificially expand the dataset and enhance the diversity of training samples, thereby mitigating the effects of limited data availability.

Data augmentation in panoramic images poses unique challenges compared to traditional image data augmentation methods since the inherent structure and layout of panoramic images must be preserved during augmentation (*e.g.* for indoor panoramic images, ceilings must be on top of walls and floors). Some traditional data augmentation techniques, such as random cropping and free-angle rotation, may not be suitable for panoramic images as they can disrupt the intrinsic structure. This underscores the importance of developing novel and specialized data augmentation techniques for panoramic images.

Current panoramic augmentations are either traditional methods that can preserve the panoramic formats, such as horizontal rotation and flipping, or methods specifically designed for panoramic images like PanoStretch proposed by Sun *et al*. [22]. However, these methods only work on a single image, which prevents them from combining the variability in different panoramic images as explored by other augmentation methods for perspective images (*e.g.* MixUp [32]). Therefore, present panoramic augmentation methods have limited capability to generate more diverse images.

To address the limited diversity issue in current panoramic augmentations, we propose a novel panoramic augmentation technique called PanoMixSwap, which utilizes multiple panoramic views to augment data and take advantage of variations in different samples. By using two or more panoramic images, semantic masks, and room layouts, we can generate numerous combinations to diversify our training data. PanoMixSwap, as shown in Fig. 1, is inspired by the observation that every indoor panoramic image typically consists of three main parts: the room structure (*i.e.*, layout), style of the background (including the ceiling, floor, and each wall), and the foreground furniture. We use these three main parts from three different indoor panoramic views to create a diverse set of augmented samples. Our method leverages a two-stage network to sequentially fuse the background style and foreground furniture into the chosen room layout. The resulting augmented images exhibit a wide range of diverse outputs while preserving the structure of the original panoramic images. We evaluate the effectiveness of our augmentation on two scene understanding tasks: semantic segmentation and layout estimation. By incorporating PanoMixSwap during training, we observe significantly improved performance compared to the original settings.

Our key contributions to PanoMixSwap are summarized below.

- We propose a novel data augmentation method PanoMixSwap for indoor panoramic images. PanoMixSwap generates cubical increased diverse images by mixing three source images while maintaining the structural integrity (*i.e.*, layout). This approach addresses the issue of limited availability in the training data and enhances the variability of the augmented images.
- We apply PanoMixSwap to two scene understanding tasks, semantic segmentation and layout estimation. PanoMixSwap consistently improves results compared to the original training setting.

2 Related Works

Data Augmentations. In the field of computer vision, the size of the dataset plays a crucial role in determining the final performance of the model; hence data augmentation is an important technique for expanding training datasets. Existing data augmentation methods can be categorized into two types: (1) those that use only one training sample to derive one augmented sample and (2) those that use two or more training samples to derive one augmented sample, also called mixup. The first type of augmentation consists of a considerable amount of work, with traditional methods such as random cropping, image mirroring, and color jittering [16] commonly used for 2D images, as well as more advanced approaches like AutoAugment [7, 8] and GAN-based methods [19, 36]. Similarly, for panoramic images, horizontal rotation and flipping techniques and Panostretch [22] introduced in Section 1 are widely used in panoramic-related tasks. On the other hand, the second type of augmentation, *i.e.*, mixup, has been widely studied in 2D image processing, with several works proposing techniques for linearly interpolating two input data points along with their corresponding one-hot labels [11, 14, 26, 28, 29, 31, 32]. For example, Zhang *et al.* [31] generate virtual training examples using mixup by linearly interpolating data points and their one-hot labels. Yun *et al.* [29] introduce a random-cut rectangular region technique, where a portion of the image is removed and replaced with a patch obtained from another image. Mixup techniques have also been applied in the field of 3D point clouds [4, 25]. However, to the best of our knowledge, no existing work currently applies the concept of mixup to panoramic images, which serves as a key factor motivating our proposed approach, PanoMixSwap.

360° perception. The popularity of 360° cameras has recently surged, leading to an increased interest in vision tasks related to panoramic images [1, 37]. Equirectangular projections (ERPs) are commonly used to represent and manipulate the wide field of view captured by these cameras. ERPs allow all captured information to be preserved in a single image. However, they also introduce distortion that can impede the performance of traditional convolution layers designed for perspective images. There has been extensive research on spherical convolution layers [6, 10, 20, 21, 24] that are aware of these distortions. To use 360° panoramic images with conventional convolutional neural networks (CNNs) that have a wide range of available pre-trained models, multiple perspective projections are employed to project the image onto multiple planar images. However, this method results in a loss of information due to the projection process, which limits the field of view (FOV). Furthermore, generating planar images from 360° panoramic images requires additional computational resources and time, which increases exponentially with higher-resolution images. To address the problems associated with projection-related works, several newer methods propose different ways of padding [5, 27] and sampling [9] image boundaries to remove inconsistencies in panoramic images. The icosahedron mesh [17, 30] provides a versatile and effective method for representing 3D shapes and scenes in computer vision, particularly for tasks that involve spherical or panoramic data.

3 PanoMixSwap

The commonly-used panoramic data augmentations mostly take only one sample as input. However, the diversity of this kind of one-to-one mapping is rather limited. We propose PanoMixSwap to mix three panoramic views into one, which is as clean and high-fidelity as the source views. Thus, we can generate more diverse training samples which are beyond the conventional panoramic augmentation.

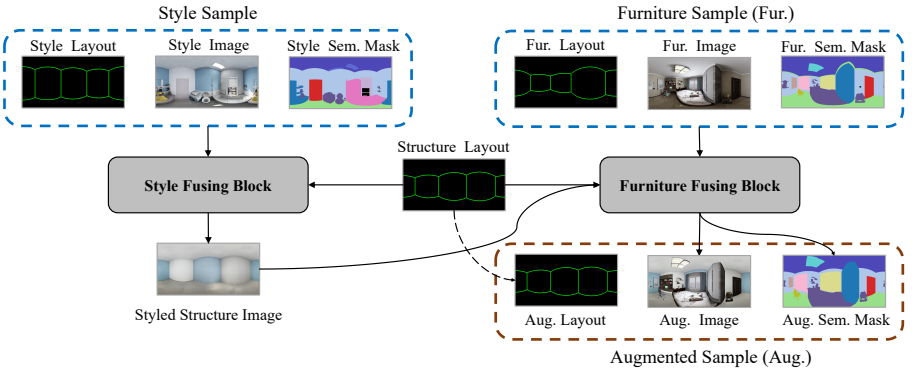

Figure 1: **Pipeline of PanoMixSwap.** PanoMixSwap involves three major inputs: style sample, structure layout, and furniture sample. PanoMixSwap is composed of two blocks: Style Fusing Block and Furniture Fusing Block. The Style Fusing Block generates a foreground-free styled structure image that fuses the background style from the style image and the room layout from structure layout. Furniture Fusing Block transforms furniture from the furniture image onto the styled structure image to produce the final augmented image and semantic mask.

3.1 Overview

Let $\mathbf{S}$ be a training sample consisting of an RGB image $I \in \mathbb{R}^{H\times W\times 3}$, a semantic mask $M \in [0,1]^{H\times W\times C}$ in the form of one-hot vector with C classes, and layout coordinates $L \in \mathbb{R}^{T\times 2\times 2}$ recording the T-walls room corner junctions on floor and ceiling. An output *augmented sample* by PanoMixSwap is the combination of three main parts from three samples— room layout structure of *structure sample* $\mathbf{S}_{\text{rs}}$, background style of *style sample* $\mathbf{S}_{\text{bs}}$, and foreground furniture setups of *furniture sample* $\mathbf{S}_{\text{fs}}$. An overview pipeline of PanoMixSwap is illustrated in Fig. 1. We first generate a *styled structure image* I_{ss} by mixing the background appearance from $\mathbf{S}_{\text{bs}}$ and the room layout L_{rs} from $\mathbf{S}_{\text{rs}}$:

$$I_{\text{ss}} = \textbf{StyleFusingBlock}(\mathbf{S}_{\text{bs}}, L_{\text{rs}}) \ , \tag{1}$$

where the **StyleFusingBlock** is detailed in Sec. 3.2. We finally can generate the augmented sample $\mathbf{S}_{\text{aug}}$ by aligning the furniture setup of $\mathbf{S}_{\text{fs}}$ with the room layout L_{rs} and then changing the background style using I_{ss}:

$$\mathbf{S}_{\text{aug}} = \textbf{FurnitureFusingBlock}(\mathbf{S}_{\text{fs}}, L_{\text{rs}}, I_{\text{ss}}) \ , \tag{2}$$

where **FurnitureFusingBlock** is detailed in Sec. 3.3.

3.2 Style Fusing Block

There are two requirements about the generated styled structure image I_{ss}: *i)* the layout structure should be the same as the room layout L_{rs} from structure sample $\mathbf{S}_{\text{rs}}$ and *ii)* the background appearance should be similar to the style sample $\mathbf{S}_{\text{bs}}$ with all the furniture removed. To achieve this, we employ a semantic conditioned generative model, SEAN [65].

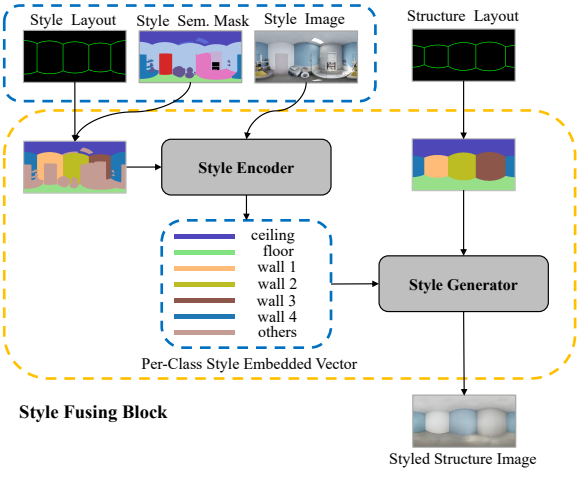

Figure 2: **Style Fusing Block.** Style Fusing Block is mainly composed of Style Encoder and Style Generator. The Style Encoder is responsible for extracting the embedded style vector for each semantic region of the style image. The Style Generator creates a foreground-free styled structure image by generating the appearance of each semantic region based on its corresponding style embedded vector.

Specifically, given a content semantic mask, SEAN generates the appearance of each semantic region based on the corresponding semantic region from a reference image. We use L_{rs} to generate the content semantic mask consisting of floor, ceiling, and walls where each wall is assigned a unique class. The reference semantic mask is generated in the same way using L_{bs}. To prevent generating the foreground, the reference semantic mask is further covered by an additional 'others' class from the furniture and objects classes in M_{bs}. We assume the number of walls is the same in L_{rs} and L_{bs}, so the walls can be one-to-one corresponding. An overview of the Style Fusing Block is illustrated in (Fig. 2).

3.3 Furniture Fusing Block

The purpose of the Furniture Fusing Block is to fuse the furniture sample $\mathbf{S}_{\text{fs}}$ with the room layout L_{rs} and the styled structure image I_{ss}. To this end, we first align the image I_{fs} and the semantic mask M_{fs} from their original layout L_{fs} to the target layout L_{rs}. The aligned image and mask are denoted as $I_{\text{fs}\to\text{rs}}$ and $M_{\text{fs}\to\text{rs}}$. The background pixels of $I_{\text{fs}\to\text{rs}}$ are then replaced by I_{ss} to change the background style. The final *augmented sample* is:

$$\mathbf{S}_{\text{aug}} = \{mI_{\text{fs}\to\text{rs}} + (1-m)I_{\text{ss}},\ M_{\text{fs}\to\text{rs}},\ L_{\text{rs}}\}\ , \tag{3}$$

where m is the foreground mask computed from $M_{\text{fs}\to\text{rs}}$. Below are the details of the alignment process.

Recap that we assume the number of walls is the same in L_{fs} and L_{rs}, and they are one-to-one corresponding. We depict the overall process in Fig. 3. We first use the wall-wall boundary annotated in L_{fs} to split the image columns of I_{fs} into multiple *image column groups*. Each image column group is then processed by Horizontal Alignment Block and Vertical Alignment Block sequentially. In the Horizontal Alignment Block, we use PanoStretch [22] to stretch each image column group from its original width to the corresponding wall width in L_{rs}. In the Vertical Alignment Block, we apply backward warping to each image column to align with the ceiling-wall and floor-wall intersection in L_{rs}. The source and destination coordinates for the backward warping are computed as follows. Let r be the destination row

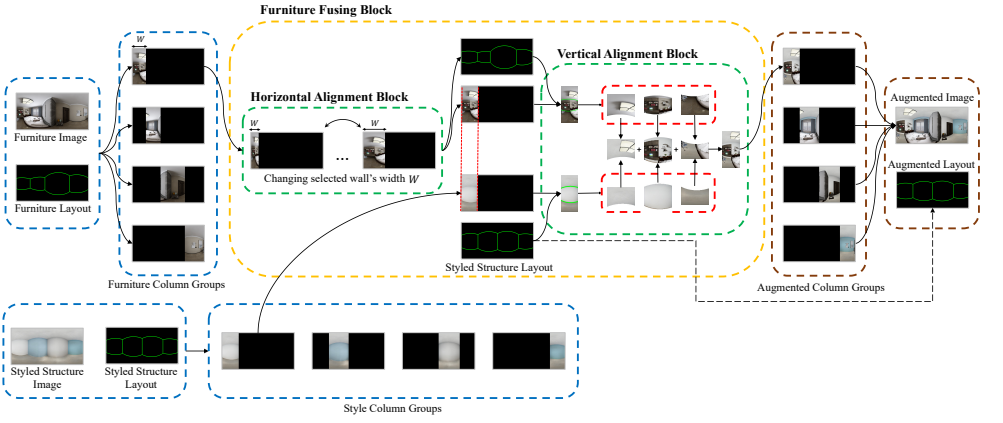

Figure 3: **Furniture Fusing Block.** Horizontal Alignment Block takes each furniture column group to produce the width-aligned column group that matches the wall width of the corresponding styled structure column group using PanoStretch [22]. Vertical Alignment Block views both the width-aligned furniture column group and the styled structure column group into ceiling, wall, and floor parts. Then generate the final augmented column group by back warping three parts of the width-aligned furniture column group (denoted aligned furniture column group) to match the same height of three parts from the styled structure column group and replacing background pixel of the styled structure column onto the aligned furniture column group. We repeat the process for T times to get the final augmented image.

index of an image column; the source index is computed as

$$\text{Source}(r) = \begin{cases} a_{\text{src}} - \alpha(a_{\text{dst}} - r), & \text{if } r < a_{\text{dst}} \\ b_{\text{src}} + \beta(r - b_{\text{dst}}), & \text{if } r > b_{\text{dst}} \\ a_{\text{src}} + (b_{\text{src}} - a_{\text{src}})\dfrac{(r - a_{\text{dst}})}{(b_{\text{dst}} - a_{\text{dst}})}, & \text{otherwise} \end{cases}, \tag{4}$$

where a, b are the index of the ceiling-wall and floor-wall intersection, α, β are hyperparameters. The equations in Eq. 4 correspond to the warping regions of ceiling, floor, and wall between source and destination. The image column groups are concatenated to form the aligned image $I_{\text{fs}\to\text{rs}}$. Semantic mask M_{fs} is processed in the same way to get $M_{\text{fs}\to\text{rs}}$.

4 Experiments

We present the implementation details and visualizations of our PanoMixSwap in Section 4.1. We showcase the effectiveness of our novel data augmentation technique on indoor 360° semantic segmentation task in Section 4.2 and layout estimation task in Section 4.4.

4.1 PanoMixSwap

Implementation Detail. We focus on four-wall indoor panoramic images for simplicity. To train the encoder-generator model discussed in Sec. 3.2, we adopt a similar pipeline as

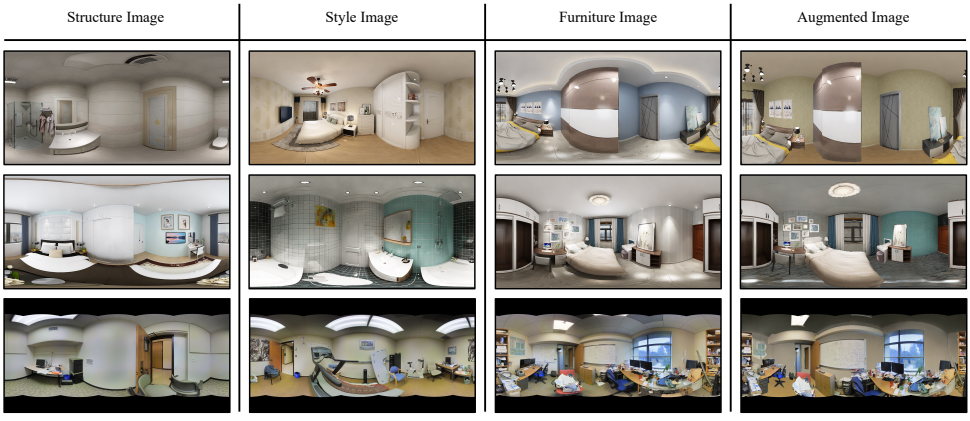

Figure 4: **Visualization of the results from our PanoMixSwap.** The augmented image (4th column) by our novel PanoMixSwap is a fusion of the room layout from the structure image (1st column), the background style from the style image (2nd column), and the furniture from the furniture image (3rd column). The images in the 1st and 2nd rows are from Structured3D [34] while the images from the 3rd row are from Stanford2D3D [2].

proposed in SEAN [35] for training on both the Structured3D and Stanford2D3D datasets. Specifically, we set the input image size to $H = 256$ and $W = 512$, use the Adam optimizer with hyperparameters $\beta_1 = 0.5$ and $\beta_2 = 0.999$, and set the learning rate to 2e-4. We use a batch size of 2 and train the model for 60 epochs on a single NVIDIA GTX 1080 Ti GPU. The inference run-time for an image is about 2 seconds, so we apply our augmentation in an offline manner for efficiency.

Visualizations. We illustrate the inputs and outputs of PanoMixSwap in Fig. 4. Our method can generate a high-quality image by incorporating the background style, room layout structure, and furniture information from three different input samples. We use high-quality augmented images to enrich the training set of different tasks. For instance, semantic segmentation training data can now be augmented to different room structures and background styles; we can also synthesize different room styles and furniture setups for a given ground-truth room layout.

4.2 Semantic Segmentation

Model, Dataset and Evaluation. In the semantic segmentation task, we use HoHoNet [23] and PanoFormer [18], which are two state-of-the-art 360 semantic segmentator. We evaluate PanoMixSwap's ability to handle real-world and synthetic data by conducting experiments on two datasets: Stanford2D3D [2] and Structured3D [34], whichttps://www.overleaf.com/projec respectively represent real-world and virtual-world environments. For Stanford2D3D [2], we use fold 5a and fold 5b for validation and the remaining folds for training following prior works. As for Structured3D [34], we follow the official training, testing, and validation setting, where there are 3,000 scenes for training and 250 scenes for validation, and 250 scenes for testing. We employ the class-wise mean intersection of union (mIoU) and mean accuracy (mACC) for semantic segmentation evaluation.

Implementation Detail. In accordance with the original HoHoNet's setting [23], we adopt

similar implementation settings. For low resolution input, a shallow U-Net with planar CNN is chosen, and the network is trained for 60 epochs on Structured3D [34] and 300 epochs on Stanford2D3D [2], using a batch size of 16 and a learning rate of 1e-3 with polynomial decay of factor 0.9. For high resolution input, ResNet-101 [12] is used as the backbone, and the network is trained for 60 epochs on both Structured3D [34] and Stanford2D3D [2], with a batch size of 4 and a learning rate of 1e-4 with polynomial decay of factor 0.9. For both low resolution and high resolution images, Adam [15] is employed as the optimizer for cross entropy loss.

In the case of PanoFormer [18], we use a batch size of 4 and an input resolution of 256*512 to train for 60 epochs. Additionally, Adam [15] is employed as the optimizer for optimizing the cross entropy loss. To apply PanoMixSwap, we first generate an augmented dataset with the same quantity as the original training data and combine the augmented dataset and original training data into a single data set.

Quantitative Results. The results of experiments on Stanford2D3D [2], as shown in the upper section of Table 1, reveal that the inclusion of our augmentation technique during training leads to significantly higher mIoU and mACC scores on both HoHoNet [23] and PanoFormer [18] compared to the original work without PanoMixSwap, across all models and resolutions. Notably, in high resolution settings, training with PanoMixSwap yields a remarkable improvement of 4.02% in mIoU and 2.43% in mACC for HoHoNet [23]. Based on these compelling results, it is evident that PanoMixSwap technique consistently enhances the mIoU and mACC in real-world indoor panoramic scenarios across different models and resolutions. In addition to real-world scenarios, we also evaluate our augmentation in virtual environment settings using Structured3D dataset [34], as presented in the lower part of Table 1. The results demonstrate that training with PanoMixSwap leads to higher mIoU and mACC scores in both low and high resolution settings, further substantiating the effectiveness of our technique in virtual indoor panoramic scenarios.

Dataset	Model	Image Size	PanoMixSwap	mIoU(%)	mACC(%)
Stanford2D3D	HoHoNet	64×128	-	31.67	46.27
			✓	**34.60**	**47.76**
		256×512	-	36.13	50.25
			✓	**41.25**	**52.50**
		1024×2048	-	52.00	65.00
			✓	**56.02**	**67.43**
	PanoFormer	256×512	-	42.20	61.03
			✓	**42.94**	**62.14**
Structured3D	HoHoNet	64×128	-	61.11	71.94
			✓	**62.50**	**73.64**
		256×512	-	70.07	78.91
			✓	**72.40**	**81.00**
		512×1024	-	80.80	87.98
			✓	**81.96**	**88.52**

Table 1: **Quantitative comparison on semantic segmentation.** Our novel PanoMixSwap significantly improves two state-of-the-art semantic segmentators, HoHoNet [23] and PanoFormer [18], on Stanford2D3D [2] and Structured3D [34].

4.3 Layout Estimation

Model, Dataset and Evaluation. We utilize HorizonNet [22] and LGT-Net [13] to test the effectiveness of PanoMixSwap on cuboid layout estimation task, and use the dataset introduced in LayoutNet by Zou *et al*. [37] to estimate cuboid layout. This dataset comprises 514 annotated cuboid room layouts from PanoContext [33] and 552 annotated cuboid room layouts from Stanford2D3D [2]. We follow train/valid/test split in layoutNet [37]. For evaluation, we use standard evaluation metrics proposed by Zou *et al*. [37] in cuboid layout estimation, including intersection of union of 3D room layout (3DIoU), corner error (CE), and pixel error (PE).

Implementation Detail. We follow all of the training settings in HorizonNet [22], which employs a learning rate of 3e-4, and a batch-size of 24 for 300 epochs. In addition, we utilize the training split of Stanford2D3D [2] and PanoContext [33] as training data. As for LGT-net [13] we train for 1,000 epochs with a learning rate of 1e-4 and a batch-size of 6. We follow the combined dataset scheme suggested by Zou *et al*. [38], which involved using the entire PanoContext [33] and the training split of Stanford2D3D [2] as the training data in LGT-net [13]. For both HorizonNet [22] and LGT-net [13], we employ Adam optimizer [15] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and PanoStretch [22] during training. In training with PanoMixSwap, We apply image augmentation only to the images in the Stanford2D3D [2], and allocate half of the batch size to augmented data and the other half to training data.

Quantitative Results. Table 4.3 presents a comparison between the performance of using PanoMixSwap during training and the original setting on Stanford2D3D [2]. The results show that utilizing PanoMixSwap during training outperforms the original setting in 3DIoU, CE on HorizonNet [22] and 3DIoU, PE on LGT-Net [13]. Especially on HorizonNet [22], training with PanoMixSwap yields a significant improvement of 3.1% in 3DIoU. This signifies that PanoMixSwap has the capability to diversify the training room style and furniture setup, thereby enhancing the overall performance.

Model	PanoMixSwap	3DIoU(%)	CE(%)	PE(%)
HorizonNet	-	83.51	0.62	**1.97**
	✓	**86.61**	**0.61**	1.99
LGT-Net	-	86.03	0.63	2.11
	✓	**86.96**	0.63	**2.04**

Table 2: **Quantitative comparison on cuboid room layout estimation.** Our PanoMixSwap can improve HorizonNet [22] and LGT-Net [13] on LayoutNet dataset [37].

4.4 Comparison Between SOTA Augmentation

This section provides a comprehensive comparison between PanoMixSwap and 360 state-of-the-art data augmentation – PanoStretch proposed by Sun *et al*. [22] on semantic segmentation task and layout estimation task. The comparison results of semantic segmentation and layout estimation are shown in Table. 3 and Table. 4, respectively. The results of the above two tables show that utilizing PanoMixSwap outperforms PanoStretch in above two tasks.

PanoStretch	PanoMixSwap	mIoU(%)	mACC(%)
-	-	52.00	65.00
✓	-	53.63	65.06
-	✓	**56.02**	**67.43**
✓	✓	55.91	67.03

Table 3: **Quantitative comparison between PanoMixSwap and PanoStretch on semantic segmentation task**. We use HoHoNet [23] on the Stanford2D3D dataset [2] for comparison.

Model	PanoStretch	PanoMixSwap	3DIoU(%)	CE(%)	PE(%)
HorizonNet	✓	-	83.88	0.63	2.00
	-	✓	85.15	0.62	1.98
	✓	✓	**86.59**	**0.62**	**1.94**
LGT-Net	✓	-	85.98	0.65	2.11
	-	✓	86.60	**0.62**	2.06
	✓	✓	**86.96**	0.63	**2.04**

Table 4: **Quantitative comparison between PanoMixSwap and 360 PanoStretch across the LayoutNet dataset [37] on layout estimation task.**

4.5 Qualitative Comparison on Downstream Tasks

Fig. 5 presents a qualitative comparison of layout estimation and semantic segmentation. We use HoHoNet [23] as semantic segmentator and HorizonNet [22] as layout estimator. More qualitative results can be found in the supplementary materials.

5 Conclusion

We present PanoMixSwap, a novel data augmentation method for 360° indoor panoramic images. PanoMixSwap aims to mix multiple panoramic images to address the issue of data scarcity in panoramic image datasets. Moreover, PanoMixSwap introduces an intuitive idea by decomposing a single indoor panoramic image into three distinct parts: foreground furniture, background style, and room layout parts. Then, it mixes multiple panoramic images by swapping these structural parts to generate diverse images. Finally, comprehensive experiments demonstrate that PanoMixSwap consistently improves state-of-the-art models on multiple 360° indoor scene understanding tasks.

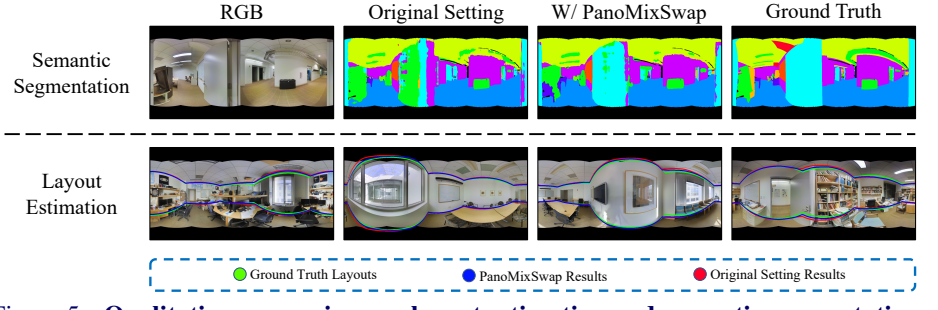

Figure 5: **Qualitative comparison on layout estimation and semantic segmentation**

Acknowledgement

This work is supported in part by Ministry of Science and Technology of Taiwan (NSTC 111-2634-F-002-022). We thank National Center for High-performance Computing (NCHC) for computational and storage resource. We especially thank Chun-Che Wu for providing invaluable guidance for our paper.

References

[1] 360sd-net: 360° stereo depth estimation with learnable cost volume. In *2020 IEEE International Conference on Robotics and Automation, ICRA 2020*, Proceedings - IEEE International Conference on Robotics and Automation, pages 582–588, United States, May 2020. Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/ICRA40945.2020.9196975.

[2] 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.

[3] 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. *arXiv preprint arXiv:1709.06158*, 2017.

[4] Yunlu Chen, Vincent Tao Hu, Efstratios Gavves, Thomas Mensink, Pascal Mettes, Pengwan Yang, and Cees GM Snoek. Pointmixup: Augmentation for point clouds. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 330–345. Springer, 2020.

[5] Hsien-Tzu Cheng, Chun-Hung Chao, Jin-Dong Dong, Hao-Kai Wen, Tyng-Luh Liu, and Min Sun. Cube padding for weakly-supervised saliency prediction in 360° videos. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1420–1429, 2018. doi: 10.1109/CVPR.2018.00154.

[6] Taco S. Cohen, Mario Geiger, Jonas Koehler, and Max Welling. Spherical cnns, 2018.

[7] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 113–123, 2019.

[8] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

[9] Marc Eder, Mykhailo Shvets, John Lim, and Jan-Michael Frahm. Tangent images for mitigating spherical distortion. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[10] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning so(3) equivariant representations with spherical cnns. *CoRR*, 2017.

[11] Hongyu Guo, Yongyi Mao, and Richong Zhang. Mixup as locally linear out-of-manifold regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3714–3722, 2019.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

[13] Zhigang Jiang, Zhongzheng Xiang, Jinhua Xu, and Ming Zhao. Lgt-net: Indoor panoramic room layout estimation with geometry-aware transformer network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1654–1663, 2022.

[14] Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *International Conference on Machine Learning*, pages 5275–5285. PMLR, 2020.

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[17] Yeonkun Lee, Jaeseok Jeong, Jongseob Yun, Wonjune Cho, and Kuk-Jin Yoon. Spherephd: Applying cnns on a spherical polyhedron representation of 360° images. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9173–9181, 2019. doi: 10.1109/CVPR.2019.00940.

[18] Zhijie Shen, Chunyu Lin, Kang Liao, Lang Nie, Zishuo Zheng, and Yao Zhao. Panoformer: Panorama transformer for indoor 360° depth estimation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I*, pages 195–211. Springer, 2022.

[19] Leon Sixt, Benjamin Wild, and Tim Landgraf. Rendergan: Generating realistic labeled data. *Frontiers in Robotics and AI*, 5:66, 2018.

[20] Yu-Chuan Su and Kristen Grauman. Learning spherical convolution for fast features from 360°imagery. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[21] Yu-Chuan Su and Kristen Grauman. Kernel transformer networks for compact spherical convolution. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9434–9443, 2019. doi: 10.1109/CVPR.2019.00967.

[22] Cheng Sun, Chi-Wei Hsiao, Min Sun, and Hwann-Tzong Chen. Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 1047–1056, 2019.

[23] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Hohonet: 360 indoor holistic understanding with latent horizontal features. In *CVPR*, 2021.

[24] Keisuke Tateno, Nassir Navab, and Federico Tombari. Distortion-aware convolutional filters for dense prediction in panoramic images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[25] Ardian Umam, Cheng-Kun Yang, Yung-Yu Chuang, Jen-Hui Chuang, and Yen-Yu Lin. Point mixswap: Attentional point cloud mixing via swapping matched structural divisions. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIX*, pages 596–611. Springer, 2022.

[26] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International conference on machine learning*, pages 6438–6447. PMLR, 2019.

[27] Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. Bifuse: Monocular 360 depth estimation via bi-projection fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[28] Jaejun Yoo, Namhyuk Ahn, and Kyung-Ah Sohn. Rethinking data augmentation for image super-resolution: A comprehensive analysis and a new strategy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8375–8384, 2020.

[29] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[30] Chao Zhang, Stephan Liwicki, William Smith, and Roberto Cipolla. Orientation-aware semantic segmentation on icosahedron spheres. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[31] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[32] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[33] Yinda Zhang, Shuran Song, Ping Tan, and Jianxiong Xiao. Panocontext: A whole-room 3d context model for panoramic scene understanding. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pages 668–686. Springer, 2014.

[34] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *Proceedings of The European Conference on Computer Vision (ECCV)*, 2020.

[35] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[36] Xinyue Zhu, Yifan Liu, Jiahong Li, Tao Wan, and Zengchang Qin. Emotion classification with data augmentation using generative adversarial networks. In *Advances in Knowledge Discovery and Data Mining: 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part III 22*, pages 349–360. Springer, 2018.

[37] 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.

[38] Chuhang Zou, Jheng-Wei Su, Chi-Han Peng, Alex Colburn, Qi Shan, Peter Wonka, Hung-Kuo Chu, and Derek Hoiem. Manhattan room layout reconstruction from a single 360° image: A comparative study of state-of-the-art methods. *International Journal of Computer Vision*, 129:1410–1431, 2021.