IndoLayout: Leveraging Attention for Extended Indoor Layout Estimation from an RGB Image

Shantanu Singh[1], Jaidev Shriram[1], Shaantanu Kulkarni[1], Brojeshwar Bhowmick[2], and K. Madhava Krishna[1]

Fig. 1: We present **IndoLayout**, a novel attention-based network that generates *amodal* layouts for indoor scenes. Unlike typical layout estimation methods that only predict occupancy values for regions visible in the RGB image, *IndoLayout* predicts the occupancy of occluded areas (as pictured) using learnt priors. Further, our method surpasses prior work by over 10% on several challenging indoor datasets. Such improvements can help indoor robots plan better trajectories using our real-time method.

***Abstract*— In this work, we propose *IndoLayout*, a novel real-time approach for generating high-quality occupancy maps from an RGB image for indoor scenes. Such occupancy maps are often crucial for path-planning and mapping in indoor environments but are often built using only information contained in the ego view. In contrast, our approach also predicts occupancy values *beyond* immediately visible regions from just a monocular image, leveraging learnt priors from indoor scenes. Hence, our proposed network can produce a hallucinated, amodal scene layout that includes areas occluded in the RGB image, such as a navigable floor behind a desk. Specifically, we propose a novel architecture that uses self-attention and adversarial learning to vastly improve the quality of the predicted layout. We evaluate our model on several photorealistic indoor datasets and outperform previous relevant work on all metrics that measure layout quality, including newly adopted ones. Finally, we demonstrate the effectiveness of our method by showing significant improvements on the PointNav task over similar approaches using *IndoLayout*. For more details, please refer to the project page: https://indolayout.github.io/.**

I. Introduction

Humans have a remarkable ability to navigate new indoor spaces based on knowledge acquired from traversing similar scenes in the past. As a result, one can easily infer multiple properties about a given location by leveraging these priors, such as the semantic configuration of a room (the various objects present) [2], proximity to adjacent spaces (a kitchen may occur near the dining room), and more relevant to our task, the layout of the scene.

Indoor layout estimation is the problem of estimating the occupancy map of a scene, its navigable and non-navigable areas. In recent years, this field has gained traction, motivated primarily by its applications in several robotics tasks, such as SLAM, Exploration, and Indoor Navigation. Layouts that are typically used for such tasks are limited to information contained in the ego-view and do not consider priors that humans can easily apply to a scene. Alternatively, humans can predict the *extended or amodal* layout of a scene and guess the presence of free space or obstacles behind occluding surfaces like furniture. While there has been sufficient traction for amodal layout estimation for on-road scenes in the context of Autonomous Driving [29], [32], there have been very limited efforts for indoor scenes like offices and home spaces. Further, this task is far from trivial as indoor layouts are arguably more complex and diverse in nature compared to typical outdoor layouts, where the shape of a vehicle and the surrounding environment are largely consistent. Further, in indoor scenes, the layout of a single room itself can change drastically depending on the viewing angle, obstructions present, and the position of the robot. Hence, in this paper, we propose a learning based approach, *IndoLayout* , that uses attention to *amodally* predict the layout of indoor scenes given just an RGB image in such challenging environments. (Fig. 1)

The main contributions of this paper are:

1) We present *IndoLayout*, a lightweight architecture that beats existing state-of-the-art on amodal occupancy map representation estimation by effectively leveraging attention and adversarial learning. (Fig. 2)
2) We demonstrate significant improvements over state-of-the-art on three large challenging indoor datasets - Gibson [41], Matterport 3D [3], and HM3D [28]. (Table. I, Section VI-A)
3) We demonstrate the importance of analysing layout quality by adopting two new metrics for this task and also surpass prior work. (Section VI-A)
4) We generate an *IndoScene Layout* dataset to evaluate performance of indoor scene layout estimation methods.
5) Lastly, we apply *IndoLayout* to the PointNav [31] task

[1] Authors are affiliated to Robotics Research Center, IIIT-Hyderabad, [2] TCS Research, Kolkata, India

and show superior results over comparable methods. (Table. IV, Section VI-D)

II. Related Work

Indoor Layout Estimation: Layout is a catchall phrase that simultaneously refers to floorplans [20], [21], [25], Manhattan-world 3D room layouts [13], [42], [45], and occupancy maps [5]. Recent work on floorplans [20], [25] for instance, use a 3D point cloud as input to produce a polygonized floorplan of the indoor scene. [21] instead uses floorplans as a prior along with RGB images to predict the 3D room layout of a scene by exploiting the geometry of a scene. However, these representations often fail to capture the presence of obstacles in the scene, which is essential for downstream tasks such as robot navigation. Further, they often require the use of floorplans or 3D scans of the scene at inference, which is not easy to obtain. Instead, we focus on the occupancy map prediction from a monocular RGB image, which is easier to use on real robots.

Occupancy maps have been extensively used in robotics, particularly for mapping [5], navigation [18], [40], and planning [26]. Early approaches for mapping used LiDAR, and sensor fusion [14], [24] to build occupancy maps, but recently, deep learning approaches have shown great success using just RGB images, particularly in outdoor scenes [22], [23], [29]. Learning based approaches for indoor layouts are however, relatively new [5], [10], [27], [34] and are often used as a proxy for other tasks such as PointNav [17] and ObjectNav [1]. Further, the best approaches to these tasks use depth sensors at inference time, which incurs an additional computational expense and payload weight. Hence, we focus on improving layout predictions using just RGB images and surpassing relevant current state of the art, while also showing improvements on the PointGoal navigation task [17].

Amodal Layout Prediction: Occupancy maps generated from single views are often incomplete, lacking any information beyond what is immediately visible in the RGB image. Humans, however, can hallucinate beyond this and use prior information to reason about the occluded areas as well, predicting the *amodal* layout. To this end, [23], [32], [43] show how learning-based approaches can reasonably predict the presence of cars and roads beyond visible regions using just RGB images in outdoor scenes. Similarly, in indoor environments, [27], [33], [40] do amodal layout estimation, predicting occluded regions and, in some cases, semantic classes. Of these, [27] is perhaps, the closest to our approach, and we adopt their work as our primary baseline. [33] hallucinates occupancy for a few selected semantic classes, while we do not discriminate between any, and predict occupancy for all objects. [40] attempt this as an inpainting problem and train a network to recreate the visible layout seen from a higher vantage point using an RGB image and the visible layout from a lower height, obtained using a depth sensor. In contrast, we only use a monocular image and predict the true bird's eye view.

Transformers for Image Synthesis: Generating bird's eye view images from a monocular camera is fundamentally ill-posed as RGB images lack concrete information about the depth of the scene. Hence, our work is more aligned with problems in the image translation domain where a new image is generated given a guiding image as input. Recent approaches that use attention [36], [44] are of particular relevance to us. [36] use attention to guide their generative network that translates one view to another, given a semantic map of the target view as guidance. Inspired by this body of work, we propose an attention-driven network to predict the bird's eye view map. However, unlike [36], [37], we do not use any secondary image to guide our generation and directly predict the bird's eye view using just a monocular image.

III. Problem Formulation

Our proposed method not only tackles general layout prediction, but also attempts to reason beyond visible areas. This is a difficult problem in general due to the limited information present in an RGB image but even more so in the indoor scenario due to the varied spatial arrangement of objects and the complexity of layouts. Hence, we attempt to solve this problem by training a network that takes an RGB image as input and produces a three class image (corresponding to unknown, occupied, and free/navigable) after training on several large-scale photorealistic indoor datasets.

IV. Approach

The goal of our network is, given an input image, generate the corresponding top-view occupancy layout in metric scale. Given the nature of this task, we adopt the GAN [12] framework, and the attention [38] module to leverage the benefits of each in our proposed model architecture as follows (Figure 2):

A. Feature Extraction

We use the first 4 blocks of ResNet-18 (pre-trained on ImageNet [30]) as our encoder, followed by convolution and max-pooling layers to reduce the input map from a resolution of $3 \times 512 \times 512$ to $128 \times 8 \times 8$. We use a convolution-based encoder as the backbone for our model, instead of a transformer-based encoder such as ViT [8], to generate patch embeddings since they are more efficient for finetuning with small datasets due to their inductive biases. They also reduce the computation overhead by reducing the number of patches for the subsequent attention module, which allows for faster training and inference times.

B. Feature Encoding

The features extracted in the previous step carry the spatial nature of the perspective view and need to be transformed into a space more relevant to the top view. To this end, we propose using a self-attention module to project and aggregate these features across different patches in a context-dependent manner. We experimented with various configurations for the self-attention implementation based on ViT

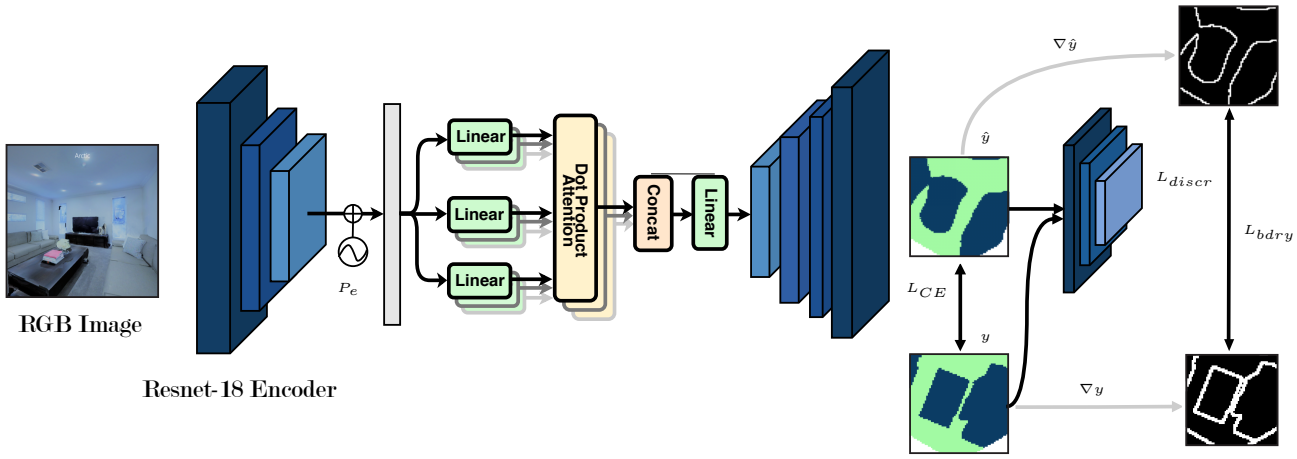

Fig. 2: *IndoLayout* consists of a Resnet-18 encoder that extracts features from an RGB image, a self-attention module acting on these features, and a decoder to predict the bird's eye view. We also use a discriminator during training to improve output quality.

[8] and decided to use a single transformer block with multi-head attention, as empirical results with more number of blocks gave marginal improvements despite higher training/inference costs. We also add a learned positional embedding to the extracted features to provide additional context for feature aggregation in the self-attention module.

C. Occupancy Decoder

Our decoder takes the features computed by the transformer block and iteratively applies a series of convolutions, followed by BatchNorm [15], ReLU activation, and upsampling layers to produce a final output of shape $3 \times 128 \times 128$, where each channel corresponds to the probability of being unexplored, occupied, or free respectively, after applying the Softmax function.

D. Discriminator

Training a network with just a per-pixel loss such as binary cross entropy may not always produce outputs that are structurally coherent, as we typically perceive objects and layouts in groups of pixels or patches. Further, the shape of objects in the predicted layout may be irregular without including any priors about their typical shape. Motivated by this, we employ a patch-based discriminator [16] to distinguish between our predictions and the ground truth layouts, as such approaches have shown success in outdoor scenarios [23], [32], [43]. The discriminator takes the $3 \times 128 \times 128$ generated layout as input and outputs a label for various patches, corresponding to *real* or *fake*. Due to the high variance in indoor layouts, we do not compute this adversarial loss using a distribution like [23] and [43]; instead we use the corresponding ground truth.

E. Loss Functions

For training, we use a combination of the following terms:

1. Weighted Cross Entropy computed over the three classes of our output using ground truth supervision.

2. Boundary Loss that penalises misclassification around the boundary of objects in particular. We calculate this by applying a L1 loss with the ground truth boundary and the spatial gradient of our output.

$$L_{bdry} = |\|\nabla \hat{y}\|_2 - y_{bdry}| \; ; L_{CE} = -\sum_{j=1}^{3} y_j \log(\hat{y_j})) \quad (1)$$

Here, y_{bdry} is the contours/boundary of the ground truth layouts, and $\hat{y}$ is the predicted layout. The spatial gradient ($\nabla \hat{y}$) computes the gradient in the x and y directions separately, which we then combine by calculating its norm.

3. GAN Loss that trains our patch-based discriminator and provides additional supervision to the generator.

$$\begin{aligned} L_{discr} &= \mathbb{E}_{y \sim p_{true}}[D(y)] + \mathbb{E}_{y \sim p_{fake}}[D(\hat{y})] \\ L_{gen} &= \mathbb{E}_{x \sim p_{true}}[D(\hat{y})] \end{aligned} \quad (2)$$

Here, $\hat{x}$ is the generated layout, corresponding to the fake distribution, and x is the ground truth layout, corresponding to the true distribution. The final loss can be expressed as:

$$L = L_{CE} + \lambda_{bdry} L_{bdry} + \lambda_{gen} L_{gen}$$

where λ_x is the weight assigned to the loss term. We find that $\lambda_{bdry} = 0.001$ and $\lambda_{gen} = 0.01$ gives us the best results.

V. Experiments

A. IndoScene Layout Dataset

We evaluate our approach on three different datasets - Gibson [41], Matterport3D [3], and HM3D [28] using the Habitat [31] simulator. For the Gibson dataset, we separately report results on the Gibson Tiny split and Gibson 4+ [31], which is a filtered version of Gibson [41], consisting of scenes rated 4 or above by human evaluators, based on the texture and mesh quality of the scene.

Trajectory Generation: Since there has been little prior work that comprehensively evaluates layout on indoor scenes, we generate the training and validation splits for the aforementioned datasets ourselves by programming an agent in Habitat [31] simulator. Specifically, we spawn an agent one meter above the ground and capture a continuous

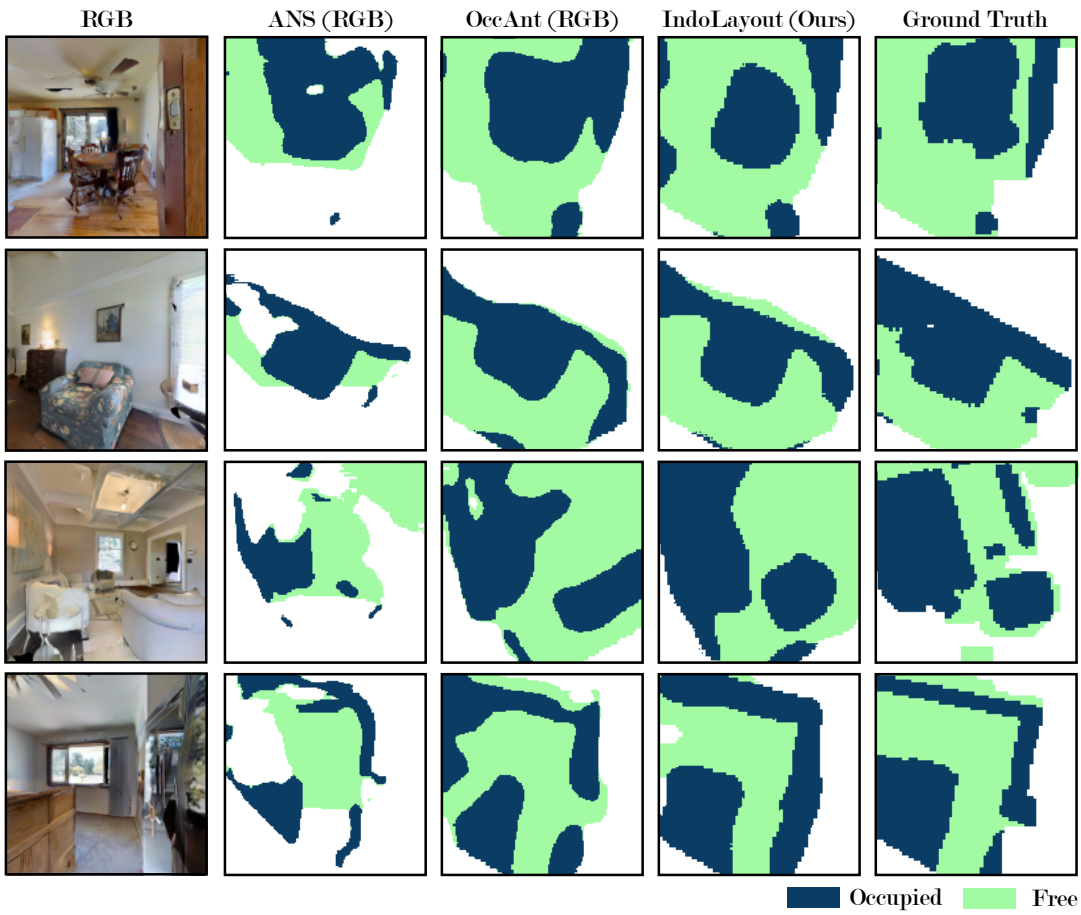

Fig. 3: **Qualitative Results**: ANS (RGB) [5] predicts only visible occupancy. When compared against OccAnt (RGB) [27], *IndoLayout* captures the shape of objects more accurately and preserves free narrow regions for indoor navigation in tight environments. All examples are from the Gibson 4+ [31] validation set. The final row corresponds to a challenging instance for all models, where even *IndoLayout* is unable to produce sharp boundaries, likely due to the complexity of the scene.

trajectory as the agent maps the entire scene. The agent maximises coverage by choosing nearby areas yet to be mapped while avoiding movement near walls and obstacles. Our mapping objective ensures that our dataset includes a variety of different semantic classes and rooms, as well as diverse layouts that a typical robot may see during navigation. The agent is equipped with an RGB sensor of 512×512 resolution and a local map sensor that extracts a 128×128 occupancy map at a scale of 0.025 metres per pixel, or $3.2m \times 3.2m$.

Layout Generation: The raw output of the map sensor is ill-suited for our task as it includes areas that are outside the field of view. For instance, it is not feasible to guess the layout of rooms behind a closed door, which is included in the raw sensor output. Hence, we mask out such regions from the raw occupancy map using the technique proposed in [27] - we project rays from the agent until it hits a wall and exclude areas not covered by the ray. Note that we only use walls or other tall view-obstructing objects for masking purposes, due to which our layouts still include occlusions induced by furniture and other objects. We shoot rays with a 120° FOV, as opposed to the camera's 90° FOV, which further increases the amount of hallucinated area. Finally, we dilate the mask to additionally increase coverage. All areas outside the mask are marked as *unknown*, and pixels within the mask are limited to either *occupied* or *free*.

After generating the trajectories and layouts for all scenes, we filter out potentially noisy samples by removing images with up-close obstacles based on the maximum depth visible. We also remove images where the unknown region is more than 90% of the image, similar to [35], [40]. Here are some statistics for the final dataset used:

1) **Gibson 4+** [31], [41]: It consists of 72 training and 12 validation scenes, (18,435 training images, 2955 validation images). 2) **Gibson Tiny** [41]: It consists of 25 training and 5 validation scenes, (8,176 training images, 1360 validation images). 3) **Matterport** [3]: It consists of 11 large and varying validation scenes, (7,885 validation images). 4) **HM3D** [28]: It consists of 100 validation scenes, (32,470 validation images). We only report out-of-the-box validation scores on Matterport and HM3D for all models to evaluate the robustness of our model.

			mIoU %			mAP %			F1 %			SSIM	Boundary IOU %
Dataset	Method	Only RGB?	Occ	Free	Mean	Occ	Free	Mean	Occ	Free	Mean		
Gibson 4+	ANS (RGB) [5]	✓	33.04	32.67	32.85	**77.81**	69.72	73.76	48.23	45.84	47.03	49.23	14.65
	OccAnt (RGB) [27]	✓	52.87	61.32	57.09	70.02	72.33	71.17	68.07	**74.08**	71.07	66.19	36.42
	***IndoLayout* (Ours)**	✓	**59.06**	**67.84**	**63.45**	71.92	**83.01**	**77.46**	**72.96**	74.02	**73.49**	**69.37**	**39.06**
	OccAnt (RGBD) [27]	✗	69.63	71.54	70.58	83.01	81.02	82.01	81.5	82.15	81.82	74.75	54.02
Gibson Tiny	ANS (RGB) [5]	✓	29.21	36	32.60	**70.27**	72.27	**71.27**	43.84	49.6	46.72	47.99	14.46
	OccAnt (RGB) [27]	✓	47.6	61.1	54.35	63.13	72.56	67.84	63.22	73.84	68.53	64.16	33.16
	***IndoLayout* (Ours)**	✓	**52.3**	**64.49**	**58.39**	64.13	**77.8**	70.96	**67.53**	**76.94**	**72.23**	**66.45**	**34.96**
	OccAnt (RGBD) [27]	✗	70.8	71.96	71.38	85.45	80.5	82.975	82.36	82.24	82.3	73.2	51.25
Matterport	ANS (RGB) [5]	✓	24.1	34.32	29.21	**66.29**	77.06	71.67	37.24	48.06	42.65	42.62	12.11
	OccAnt (RGB) [27]	✓	43.02	63.3	53.16	63.53	76.45	69.99	58.08	75.65	66.86	63.79	33.44
	***IndoLayout* (Ours)**	✓	**49.48**	**66.39**	**57.93**	64.61	**81.62**	**73.11**	**64.55**	**78.12**	**71.33**	**67.34**	**36.44**
	OccAnt(RGBD) [27]	✗	67.53	74.68	71.10	82.04	83.25	82.64	79.69	84.12	81.90	74.92	53.33
HM3D	ANS (RGB) [5]	✓	31.53	35.61	33.57	**80.67**	70.84	**75.755**	46.65	48.88	47.765	49.73	13.52
	OccAnt (RGB) [27]	✓	53.17	62.85	58.01	71.9	71.68	71.79	68.26	75.15	71.705	66.61	35.79
	***IndoLayout* (Ours)**	✓	**57.02**	**66.23**	**61.625**	71.64	**76.19**	73.915	**71.57**	**77.86**	**74.715**	**69.22**	**37.6**
	OccAnt (RGBD) [27]	✗	71.55	71.68	71.615	85.91	78.84	82.375	82.84	81.9	82.37	74.98	53.13

TABLE I: We compare the performance of *IndoLayout* against prior work on several challenging indoor datasets. We train all models on Gibson 4+ [31] and Gibson Tiny [41] for evaluation, and report out of the box performance on Matterport [3] and HM3D [27]. *IndoLayout* outperforms prior RGB baselines on all datasets by a significant margin, including those which it is not fine-tuned on.
.

B. Approaches Evaluated

To evaluate the effectiveness of our proposed method, we compare *IndoLayout* against the current state-of-the-art models:

- ANS RGB: The monocular indoor layout estimation method proposed in [5].
- OccAnt RGB: The amodal monocular indoor layout estimation method proposed in [27].

We additionally report the scores of OccAnt RGBD [27], the state-of-the-art on indoor layout estimation, as a benchmark. Note that this approach uses both RGB and depth information during training and at inference time. While this is an unfair comparison, we include this to report the best-known performance for this task.

C. Evaluation Metrics

We quantitatively evaluate the performance of all approaches against the ground truth layouts. In line with prior work on layout estimation, we report the mean Intersection-over-Union (mIoU) and mean Average Precision (mAP) metrics. The mIoU metric effectively captures the minimum of precision and recall; hence we additionally report the mean F1 score. We report each metric for the occupied and navigable/free class in the layouts.

While these metrics capture the efficacy of our method, they may not capture the visual quality of the layouts well. Optimising for IoU or F1 in particular, can have the effect of producing rounded edges, when the layout of objects in bird's eye view are typically sharper. This is particularly true for large objects, where small errors along the boundary will have a minimal contribution to the loss function. Therefore, we report the Boundary IoU [6] for each class, a more sensitive metric that focuses on boundary quality alone. [11] also proposed a boundary metric for segmentation that measured the difference in tangent angles between contours on the predicted and ground truth segmentation, but such a metric is more suited for building segmentation than our scenario due to the polygonal nature of buildings. Additionally, we report the SSIM [39] score, which considers the similarity in structure between two images.

VI. Results and Discussion

A. Layout estimation

1) ***Performance on Evaluation Metrics****:* To evaluate the performance of the models on the task of layout estimation, we compare the predicted local occupancy maps against the ground truths and quantify the correctness of these predictions using IoU and F1 score metrics described in section V-C. In Table I, we compare the performance of our models on the Gibson4+ and Gibson Tiny datasets. All the models are trained on the training split for both datasets and then evaluated on a separate validation split. As observed, among the RGB-only models, *IndoLayout* model is substantially better than the other baselines in its prediction for both the occupied as well as the free space. Our model reduces the gap in the performance compared to the RGBD model, thus alleviating the penalty incurred for tasks where only a monocular setup can be used.

In Table I, we report the out-of-the-box performance on the validation splits for Matterport and HM3D datasets. Again, we observe that our model outperforms the other RGB models and generalizes better to novel scenes.

To better understand the reason why our model does better, we inspect the saliency maps computed using GradCAM for all the models, as well as the attention map for our model in Figure 4. This gives an insight into which regions the models focus on, for a given image while generating the corresponding local occupancy maps. From the figure, it is clear that due to attention, our model is able to focus on relevant surfaces while predicting the target classes.

*2) **Quality of Generated Outputs**: IndoLayout* produces vastly better layouts qualitatively compared to prior art, as shown in Figure 3. We find that [27] often produces blurry outputs with more rounded edges, as compared to our model. This analysis is confirmed by the reported SSIM and Boundary IoU metrics, on which we also show significant improvements over [27] and [5]. We attribute this gain to the use of self-attention and a discriminator that operate at the patch level while attending to the global context.

B. Amodal estimation

We try to investigate the reason behind the performance boost by evaluating the hallucinated and visible occupancy regions separately. We find that the percentage of pixels hallucinated by our approach is 4% less than OccAnt RGB [27] but 6% more accurate within this area. Since the percentage of hallucinated pixels was calculated using the model's output, the higher accuracy obtained by our approach shows that we perform better within the hallucinated area. This improvement is significant for planners that may use the amodal layout predicted. Further, we find that our model is 4% more accurate within the visible occupancy region, which is also critical for navigation purposes.

C. Ablation Studies

*1) **Importance of Attention**:* We find that adding self-attention to our network significantly improves performance, as shown in Table II, with the results most pronounced on the Gibson 4+ dataset. We attribute this gain to the expressive power of self-attention, which has been well-established for vision-related tasks in recent years [7] [19].

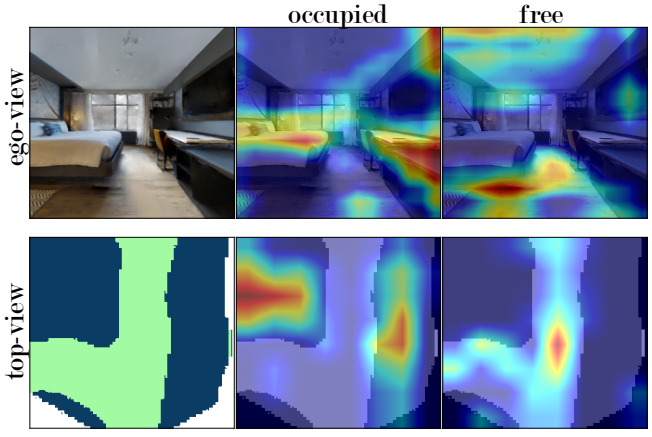

Fig. 4: We visualise the saliency maps from various layers of our network and notice that *IndoLayout* focuses on relevant objects for the class being predicted.

	mIoU %			mAP %		
Method	Occ	Free	Mean	Occ	Free	Mean
Base	52.59	64.09	58.34	70.84	73.13	71.98
w/ Boundary Loss	54.2	63.3	58.75	69.9	74.8	72.35
w/ Discriminator	54.4	64.4	59.4	**72.5**	75.3	73.9
w/ Self-Attention	56.79	64.24	60.51	68.97	77.85	73.41
IndoLayout (All)	**59.06**	**67.84**	**63.45**	71.92	**83.01**	**77.46**

TABLE II: We examine the role of each component in *IndoLayout* by comparing the performance gain over the base model (encoder-decoder architecture) for both IOU and mAP scores. We observe that self-attention has a substantial impact on the model's overall performance, followed by the boundary loss and the discriminator.

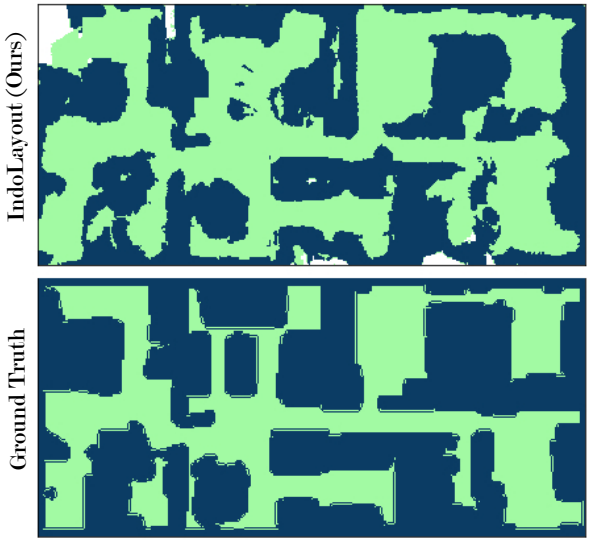

Fig. 5: We register the map for an entire scene using our predicted occupancy maps to evaluate its utility in mapping tasks.

Using self-attention, our model can learn the global context in addition to local features, which is critical for such a task as distant pixels may help *contextualise* local patches. As mentioned earlier, the saliency maps visualised in Figure 4 further support our hypothesis that attention enables our model to focus on relevant regions in the input image more effectively.

*2) **Effect of Adversarial Learning**:* By using a patch-based discriminator [16], we observe a 2% improvement in IoU for the occupied class. In addition, we also notice an improvement in the visual quality for several images, as shown in figure 3. However, the visual improvements are not as pronounced as in outdoor scenarios [23] [43], such as the KITTI [9] and Argoverse [4] datasets. We attribute this to the typical shape of vehicles and road layouts, which belong to a smaller distribution than indoor layouts. Upon inspection, we find that objects belonging to the same semantic class, such as a dining table, can have vastly varying layouts, in contrast to the outdoor scenario, where vehicle shapes are largely similar. Further, the shape of navigable areas in indoor scenes is highly dependent on furniture placements and room size, making it more challenging to regularise the output using existing layouts, as originally proposed in [32].

D. Application: PointNav

To establish the significance of our improvements, we apply *IndoLayout* to the PointGoal navigation task from the Habitat Challenge 2020 [31] where an agent in Habitat simulator [31] has to pathfind and move to a target location without any prior global map. [27] showed that hallucination could significantly help in path planning and navigation for this task, and successfully beat prior work [5] that only used the visible occupancy. Hence, we simply replace the layout module used in the RL pipeline of [27] with *IndoLayout*, trained on Gibson4+, and evaluate the performance of our model. We only compare against the RGB variants of [5], [27] to be fair.

We find that by simply replacing the layout module, we observe a 5% improvement in success rate and SPL, standard metrics used for this task. Further, we do significantly better for the *medium* and *hard* episodes in the validation set, with a 7% higher success rate than OccAnt RGB on *medium* episodes and 9% higher success rate on *hard* episodes. We also observe similar trends for other reported metrics, as shown in Table IV. Since these episodes involve navigation across longer distances, the performance improvements suggest that our layouts are more suited for planning and navigation purposes. Once again, we note that we did not train the policy from scratch alongside our model, and expect even greater improvements if trained with *IndoLayout* in an end-to-end manner.

E. Application: Mapping

Since our dataset consists of a continuous trajectory per scene, we use the predicted layouts for the validation split and register the maps using the raw probability values predicted by our model. We filter out low-confidence predictions using a threshold and aggregate information using a moving average. We find that the results, as shown in figure 5 closely resemble the ground truth, despite never being trained on it. While there is room for improvement, this shows the efficacy of *IndoLayout* for potential mapping tasks despite being an RGB only model.

F. Timing Analysis

In addition to the above, we also report additional model statistics like model parameter count (in millions) and inference speed (frames-per-second) in table III. We evaluate all the models with an input size of 3 x 512 x 512 and an output size of 3 x 128 x 128 on an NVIDIA GeForce GTX 1080Ti GPU. *IndoLayout* layout is twice as fast, with a lower memory footprint, while showing superior performance.

Method	Parameters (M)	FPS
Occant (RGB)	19.86	33.1
IndoLayout (Ours)	**14.35**	**61.02**

TABLE III: Timing analysis of *IndoLayout* against state-of-the-art

Difficulty	Method	Success Rate ↑	SPL ↑	Time ↓
Easy	ANS RGB [5]	0.851	0.676	154.248
	Occant RGB [27]	0.888	0.715	135.498
	***IndoLayout* (Ours)**	**0.913**	**0.731**	**127.105**
Medium	ANS RGB	0.626	0.488	283.329
	Occant RGB	0.698	0.532	261.636
	***IndoLayout* (Ours)**	**0.763**	**0.566**	**233.803**
Hard	ANS RGB	0.303	0.239	429.541
	Occant RGB	0.339	0.248	417.312
	***IndoLayout* (Ours)**	**0.431**	**0.337**	**383.33**
Overall	ANS RGB	0.7	0.552	236.51
	Occant RGB	0.752	0.59	217.288
	***IndoLayout* (Ours)**	**0.8**	**0.621**	**198.246**

TABLE IV: Using *IndoLayout* as the mapping module in previous layout based state of the art [27] on the PointNav task [31], we show considerable improvement over all baselines, showing the significance and utility of our approach.

G. Failure Cases

While our model shows improvements on several fronts, we also notice certain instances where the network either incorrectly hallucinates regions correctly or fails to produce sharp outputs. We display one such instance in Figure 3. Further, in some instances, multiple objects are combined together, suggesting that the small gaps between objects can confuse the network.

H. Limitations and Future Work

IndoLayout shows considerable improvements over prior art across multiple datasets, but more work is required to reach the high benchmark set by RGBD models. Future work could focus on further improving the quality of predicted outputs using monocular depth estimation techniques and LiDAR. Such techniques are not trivial either and can introduce new problems vis-à-vis scale and sparsity. Subsequent work can also focus on further improving the shapes of objects using novel shape constraints on predicted layouts. *IndoLayout* shows improvements on multiple metrics here, but the best scores are still far from perfect. Another substantial extension to our work would be to report the uncertainty associated with the predicted occupancy map, which can be used for the downstream tasks in navigation and mapping. Exploration of the above ideas can help improve the overall utility of indoor robots by making their navigation capabilities more efficient and robust to novel environments.

VII. Conclusion

We propose *IndoLayout*, a real-time network that predicts amodal layouts in indoor scenes using just an RGB image. Our analysis across multiple photorealistic indoor datasets shows the efficacy of our proposed network, which leverages attention and surpasses prior work by a significant margin in quantitative and qualitative studies. Finally, we show the utility of our approach by presenting improvements on the challenging PointNav task.

REFERENCES

[1] D. Batra, A. Gokaslan, *et al.*, "ObjectNav Revisited: On Evaluation of Embodied Agents Navigating to Objects," *Computing Research Repository (CoRR)*, vol. abs/2006.13171, 2020.

[2] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic slam," in *International Conference in Robotics and Automation (ICRA)*, 2017, pp. 1722–1729.

[3] A. Chang, A. Dai, *et al.*, "Matterport3d: Learning from rgb-d data in indoor environments," *International Conference on 3D Vision (3DV)*, 2017.

[4] M.-F. Chang, J. Lambert, *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8740–8749.

[5] D. S. Chaplot, D. Gandhi, *et al.*, "Learning to explore using active neural slam," in *International Conference on Learning Representations (ICLR)*, 2020.

[6] B. Cheng, R. Girshick, *et al.*, "Boundary iou: Improving object-centric image segmentation evaluation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 15 334–15 342.

[7] J.-B. Cordonnier, A. Loukas, and M. Jaggi, "On the relationship between self-attention and convolutional layers," in *International Conference on Learning Representations (ICLR)*, 2019.

[8] A. Dosovitskiy, L. Beyer, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations (ICLR)*, 2021.

[9] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3354–3361.

[10] G. Georgakis, B. Bucher, *et al.*, "Learning to map for active semantic goal navigation," in *International Conference on Learning Representations (ICLR)*, 2022.

[11] N. Girard, D. Smirnov, J. Solomon, and Y. Tarabalka, "Polygonal building segmentation by frame field learning," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[12] I. Goodfellow, J. Pouget-Abadie, *et al.*, "Generative adversarial nets," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 27, 2014.

[13] V. Hedau, D. Hoiem, and D. A. Forsyth, "Recovering the spatial layout of cluttered rooms," *International Conference on Computer Vision (ICCV)*, pp. 1849–1856, 2009.

[14] F. Homm, N. Kaempchen, J. Ota, and D. Burschka, "Efficient occupancy grid computation on the gpu with lidar and radar for road boundary detection," in *2010 IEEE Intelligent Vehicles Symposium*, 2010, pp. 1006–1013.

[15] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning (ICML)*. PMLR, 2015, pp. 448–456.

[16] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1125–1134.

[17] A. Kadian, J. Truong, *et al.*, "Sim2Real Predictivity: Does Evaluation in Simulation Predict Real-World Performance?" *Robotics and Automation Letters (RA-L)*, vol. 5, pp. 6670–6677, 2020.

[18] K. D. Katyal, A. Polevoy, *et al.*, "High-speed robot navigation using predicted occupancy maps," *International Conference in Robotics and Automation (ICRA)*, pp. 5476–5482, 2021.

[19] S. Khan, M. Naseer, *et al.*, "Transformers in vision: A survey," *ACM Computing Surveys (CSUR)*.

[20] C. Liu, J. Wu, and Y. Furukawa, "Floornet: A unified framework for floorplan reconstruction from 3d scans," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 201–217.

[21] C. Liu, A. G. Schwing, *et al.*, "Rent3d: Floor-plan priors for monocular layout estimation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3413–3421.

[22] C. Lu, M. van de Molengraft, and G. Dubbelman, "Monocular semantic occupancy grid mapping with convolutional variational encoder–decoder networks," *Robotics and Automation Letters (RA-L)*, vol. 4, pp. 445–452, 2019.

[23] K. Mani, S. Daga, *et al.*, "Monolayout: Amodal scene layout from a single image," in *Winter Conference on Applications of Computer Vision WACV)*, 2020, pp. 1689–1697.

[24] P. Moghadam, W. S. Wijesoma, and D. J. Feng, "Improving path planning and mapping based on stereo vision and lidar," in *2008 10th International Conference on Control, Automation, Robotics and Vision*, 2008, pp. 384–389.

[25] C. Mura, O. Mattausch, *et al.*, "Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts," *Computers & Graphics*, vol. 44, pp. 20–32, 2014.

[26] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *European Conference on Computer Vision (ECCV)*, 2020.

[27] S. K. Ramakrishnan, Z. Al-Halah, and K. Grauman, "Occupancy anticipation for efficient exploration and navigation," in *European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 400–418.

[28] S. K. Ramakrishnan, A. Gokaslan, *et al.*, "Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2021.

[29] T. Roddick and R. Cipolla, "Predicting semantic map representations from images using pyramid occupancy networks," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 138–11 147.

[30] O. Russakovsky, J. Deng, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, pp. 211–252, 2015.

[31] M. Savva, A. Kadian, *et al.*, "Habitat: A Platform for Embodied AI Research," in *International Conference on Computer Vision (ICCV)*, 2019.

[32] S. Schulter, M. Zhai, N. Jacobs, and M. Chandraker, "Learning to look around objects for top-view representations of outdoor scenes," in *European Conference on Computer Vision (ECCV)*, 2018.

[33] Z. Seymour, K. Thopalli, *et al.*, "Maast: Map attention with semantic transformers for efficient visual navigation," *International Conference in Robotics and Automation (ICRA)*, pp. 13 223–13 230, 2021.

[34] Z. Shen, L. Kästner, and J. Lambrecht, "Spatial imagination with semantic cognition for mobile robots," *International Conference on Intelligent Robots and Systems (IROS)*, pp. 2174–2180, 2021.

[35] S. Song, F. Yu, *et al.*, "Semantic scene completion from a single depth image," *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[36] H. Tang, H. Liu, *et al.*, "Attentiongan: Unpaired image-to-image translation using attention-guided generative adversarial networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[37] H. Tang, D. Xu, *et al.*, "Multi-channel attention selection gan with cascaded semantic guidance for cross-view image translation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2417–2426.

[38] A. Vaswani, N. Shazeer, *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[39] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[40] M. Wei, D. Lee, V. Isler, and D. Lee, "Occupancy map inpainting for online robot navigation," in *International Conference in Robotics and Automation (ICRA)*, 2021, pp. 8551–8557.

[41] F. Xia, A. R. Zamir, *et al.*, "Gibson env: Real-world perception for embodied agents," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 9068–9079.

[42] S.-T. Yang, F.-E. Wang, *et al.*, "Dula-net: A dual-projection network for estimating room layouts from a single rgb panorama," *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3358–3367, 2019.

[43] W. Yang, Q. Li, *et al.*, "Projecting your view attentively: Monocular road scene layout estimation via cross-view transformation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[44] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *International Conference on Machine Learning (ICML)*. PMLR, 2019, pp. 7354–7363.

[45] Y. Zhao and S.-c. Zhu, "Image parsing with stochastic scene grammar," in *Advances in Neural Information Processing Systems (NeurIPS)*, J. Shawe-Taylor, R. Zemel, *et al.*, Eds., vol. 24. Curran Associates, Inc., 2011.