End-to-End Holistic 3D Scene Understanding with Attention from a Single Image

Kıray, Mert
mert.kiray@tum.de

Koçaş, Halil Eralp
halileralp.kocas@tum.de

Abstract

In this work, we tackle the challenging problem of 3D scene understanding. We achieve this by extending the Holistic3D model's Graph Convolutional Network by utilizing attention mechanisms. We further extend this attention mechanism by utilizing transformers. Transformers enabled us to consider this problem as a translation task from 2D bounding boxes to 3D objects. We evaluate our methods on Pix3D and SUN RGB-D. Our models show competitive results in terms of both object shape and scene layout estimation.

1. Introduction

Holistic 3D reconstruction of indoor scenes from single image is a challenging problem due to the complexity and variety of real-life objects. These challenges make the reconstruction difficult. To reconstruct the scene in 3D, a model should estimate a room layout, objects with their semantic labels, and the shapes of these objects as you can see an example in Figure 1. Most successful models utilize deep implicit representation jointly with separate modules for object detection and room layout estimation. The current state-of-the-art model, Holistic3D [12], has a graph network to exploit more information from neighbors of objects to refine the room layout, object poses, and orientations given to the graph. In this work, we propose to use Graph Attention Network and transformer instead of the Graph Convolutional Network in Holistic3D to exploit the relationship between objects and geometric features. Furthermore, we consider Holistic 3D Reconstruction from a single image as a translation problem where we translate initial 2D bounding box estimations to 3D object estimations. Thus, on top of the proposed Graph Attention Network, we further extend our approach via transformer-based models instead of Graph Convolutional Network.

2. Related Work

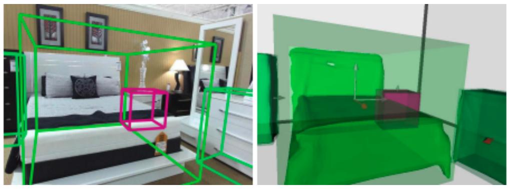


Figure 1: Our model estimates 3D object bounding boxes and poses with room layout given single image. Later, it reconstructs the object meshes and room in 3D.

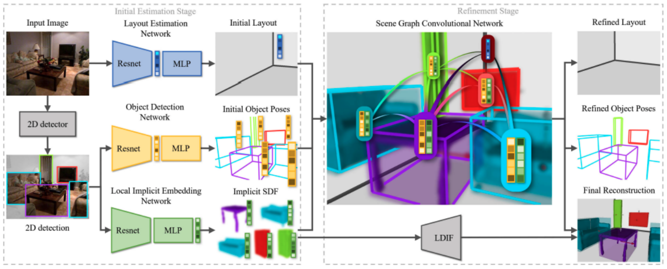


Figure 2: Model of Holistic3D. They estimate room layout and camera pose from LEN. ODN estimates 3D object poses. LIEN encodes the implicit shape representations of 2D detections. SGCN refines the estimation of LEN and ODN in the second stage. It utilizes the initial estimations and implicit shape representations. LDIF decodes the encoded shape representations to recover 3D object meshes for reconstruction.

3D scene understanding from a single image is an ill-posed problem. Indoor scene complexity increases with the diversity of objects and occlusion between objects. 3D scene understanding can be divided into layout estimation, object detection, pose estimation, and 3D object reconstruction.

Some of the early works only focus on layout estimation [2,5,7]. Later, CooP [3] proposes a solution to estimate object poses beyond the layout by using CNNs. Total3D [6] proposes an end-to-end solution to estimate the layout box and object poses. Also, it reconstructs object meshes using a mesh generation network. Holistic3D [12] builds on Total3D by adding Local Implicit Embedding Network (LIEN) to extract the implicit local shape information from

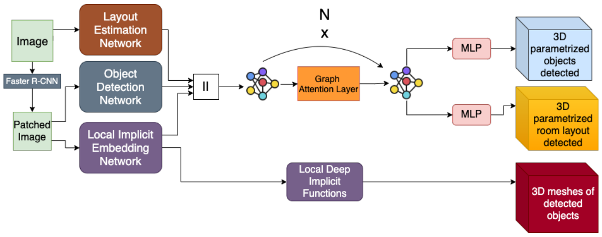


Image
Faster R-CNN
Patched Image
Layout Estimation Network
Object Detection Network
Local Implicit Embedding Network
||
N x
Graph Attention Layer
MLP
MLP
3D parametrized objects detected
3D parametrized room layout detected
Local Deep Implicit Functions
3D meshes of detected objects

Figure 3: Architecture of our Graph Attention Network. A graph is constructed with object and geometric features. Graph Attention Layers exploits the relationship between object and geometric features via attention mechanism.

the image that can be decoded by Local Deep Implicit Functions (LDIF) to infer the 3D geometry to solve the occlusion between objects. Holistic3D also proposes a Scene Graph Convolutional Network (SGCN) to refine the initial predictions with the scene context.

3. Method

This section structured as follows, first we explain our baseline model. Then, we introduce our Graph Attention Network, Decoder-only Transformer architecture, and Complete Transformer architecture.

3.1. Baseline Model: Holistic3D

Holistic3D consists of two stages as shown in Figure 2: i) initial estimation and ii) refinement stages. In the first stage, the input image is fed into the Layout Estimation Network (LEN) to predict a 3D layout bounding box and relative camera pose. In addition, a Faster R-CNN model extracts 2D bounding boxes from the input image. Then, the 2D bounding boxes are fed into the Object Detection Network (ODN) to predict object poses as 3D bounding boxes and into the Local Implicit Embedding Network (LIEN) to encode implicit local shape information.

In the second stage, a Scene Graph Convolutional Network (SGCN) represents the scene as a graph with nodes representing the room layout, objects, and the relationship between them. The encoding of layout appearance, parametrized outputs of LEN, and image-size normalized camera intrinsic parameters are concatenated and passed through a 2-layer MLP layer before they are fed into the node representing room layout. The same processes are valid for the object nodes but appearance-relationship features and the parametrized outputs of ODN are fed to the nodes. Lastly, the geometrical feature of 2D object bounding boxes and the box corner coordinates are encoded to relationship nodes. After encoding features to the nodes of the graph, several message-passing steps are performed where an attention-like feature aggregation mechanism from neighboring nodes is applied.

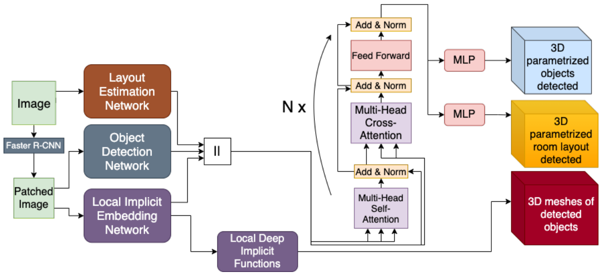


Image
Faster R-CNN
Patched Image
Layout Estimation Network
Object Detection Network
Local Implicit Embedding Network
||
Local Deep Implicit Functions
N x
Add & Norm
Feed Forward
Add & Norm
Multi-Head Cross-Attention
Add & Norm
Multi-Head Self-Attention
MLP
MLP
3D parametrized objects detected
3D parametrized room layout detected
3D meshes of detected objects

Figure 4: Architecture of our decoder-only Transformer. Decoder module of a Transformer model refines the initial estimations of LEN and ODN. It utilizes initial estimations as self-attention inputs in the first layer. 2D geometry features are given as keys and values through multi-layer cross-attention layer to learn directly from 2D encoded features.

In this work, we propose three different approaches. First, we change the feature aggregation in SGCN. We implement a Graph Attention Network. Attention mechanisms bring the capability of focusing specific parts of an input to a model. Therefore, we propose that implementing an attention mechanism would increase the performance. In addition, we replace SGCN with two different Transformer models in our two other approaches because transformer models can improve several shortcomings of attention mechanisms.

3.2. Graph Attention Network:

With the help of the attention mechanism, the model can learn which objects in the scene is more important for the scene reconstruction. Hence, we propose using attention layers in the SGCN instead of standard message-passing steps. You can see the overview of our model in Figure 3. We adopt the same graph structure as Holistic3D as we connect all object nodes together and added self loops. We also connect the object node pairs with the corresponding relationship nodes. After the graph construction, the graph is fed into several attention convolution layers in the message-passing step where we adopt the attention mechanism from GAT [1, 11].

3.3. Transformer Decoder Architecture:

We implement a decoder-only Transformer model in our second approach to overcome the shortcomings of the Attention mechanism. Firstly, random initialization of attention weights may lead to destabilized dot-product attention. Multi-head attention in Transformers helps to overcome this issue. Secondly, the features may scale in different magnitudes after the dot-product attention. Therefore, the model may have very sharp or distributed attention weights in some cases. Layer normalization in Transformers may eliminate this problem. Also, Transformers scale

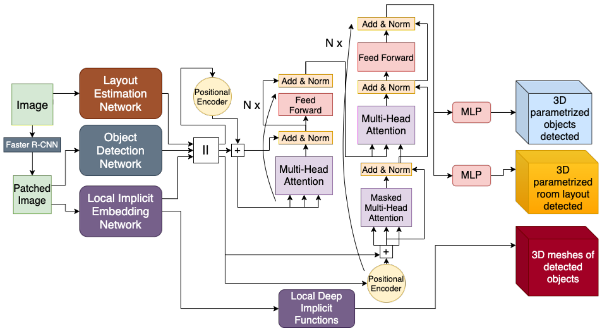


Figure 5: Architecture of our Transformer. Transformer model refines the initial estimations of LEN and ODN by utilizing the 2D geometry features and the initial estimations. Since it is considered as a translation problem, 2D geometry features are given into the encoder module of transformer. Later, the initial estimations are given into the decoder module since they are learning targets. Furthermore, sine and cosine positional encodings are utilized as proposed in [10] over encoder and decoder inputs.

Method	Layout IoU	Cam Pitch	Cam Roll	L_g(Chamfer Distance)
Holistic3D-Paper	64.4	2.98	2.11	1.11
Holistic3D-Baseline	63.7	3.04	2.26	1.09
Ours-GAT	64.0	**2.97**	**2.19**	**1.06**
Ours-Decoder-Only Transformer	63.97	3.06	2.23	1.27
Ours-Transformer	**64.11**	3.08	**2.19**	1.13

Table 1: 3D Layout and Camera Pose Evaluation.

the dot-product attention with the square root of the number of dimensions of input features. In addition, the Feed-Forward neural network in Transformers helps in the scaling of features as well.

We adopt our decoder-only Transformer model from [4]. We consider our LEN, LIEN, and ODN modules as input encoders as you can see in Figure 4. Encoded features of LEN and ODN are given through the Multi-Head Self-Attention layer as decoding targets. Then, encoded features from 2D detections are given through the Multi-Head Cross-Attention layer where the contribution of each object is directly learned from 2D encoded features.

3.4. Complete Transformer Architecture:

We implement another Transformer model consisting of both encoder and decoder modules in our third approach as you can see our implementation in Figure 5. We adopt our model from [10] since we consider the Holistic 3D reconstruction problem as a translation from 2D to the 3D problem. Hence, we utilize the geometry feature of bounding boxes and box corner coordinates from objects in 2D as encoder inputs. In addition, we utilize sine and cosine position embeddings presented in Attention is All You Need [10] in both encoder and decoder. Also, we utilize initial room layout and 3D bounding box estimations as decoder input for our transformer since they are our targets to refine.

4. Experiments

4.1. Datasets:

Following [12] and [6], we train each module individually and jointly using two datasets: SUN RGB-D [8] and Pix3D [9]. Pix3D contains 10,069 images 395 furniture models of 9 categories with pixel-level image-shape pairs. Pix3D is used in the training of LIEN with LDIF decoder since it is a shape-related dataset. SUN RGB-D contains 10,335 RGB-D indoor images containing 3D room layout, 3D object bounding box annotations, object orientations, and semantic labels.

4.2. Metrics:

We utilize various metrics to measure how the separate modules of our network performs. Following [12] and [6], average 3D Intersection over Union (IoU) for layout estimation, mean absolute error for camera pose, average precision for object detection, and Chamfer Distance(L_g) for single-object mesh generation from single image are computed during evaluation of our models.

4.3. Experiment Results:

We compare our Graph Attention Network, Transformer Decoder, and Complete Transformer Architecture results with the Holistic3D results we reproduce on our local machines. On 3D Layout and Camera Pose Evaluation, Graph Attention Network manages to get better results than the Holistic3D on every metric as you can see in Table 1. Both Transformer architectures get competitive results with the Holistic3D and managed to get better results than Holistic3D on Layout IoU and Cam Roll error. Experiments shows that LEN, ODN, LIEN, and LDIF modules have greater effect on the performance of both Holistic3D and our models. Another reason why the results do not differ much is both Pix3D and SUN RGB-D datasets are not representative enough for many categories they include. Therefore, imbalanced datasets also hurt the performance since our models tend to learn overwhelming categories better while losing performance of less representative categories.

In 3D Object Detection Evaluation results, Holistic3D manages to get better results than all of our architectures on almost all of the categories as you can see in Table 2. As ODN is the main module used for these results and our architectures does not interfere with ODN directly, these results needs further investigating and the initial reasoning for this can be the randomness of the models we trained.

We also visualize the attention weights in Figure 6, 7 and 8 for Graph Attention, Transformer Decoder, and Complete Transformer Architecture results.

Method	Bed	Chair	Sofa	Table	Desk	Dresser	Nightstand	Sink	Cabinet	Lamp	mAP
Holistic3D-Paper	89.32	35.14	69.10	57.37	49.03	29.27	41.34	33.81	33.93	11.90	45.21
Holistic3D-Baseline	**87.03**	**34.08**	**70.39**	**55.75**	**46.70**	**33.29**	**34.84**	29.15	**28.03**	**9.32**	**42.85**
Ours-GAT	86.07	32.68	62.49	52.83	44.24	25.81	26.30	**33.19**	24.06	7.90	39.55
Ours-Decoder-Only Transformer	83.00	29.36	67.56	51.11	43.31	27.05	32.14	30.69	23.18	6.66	39.41
Ours-Transformer	84.02	31.91	68.66	51.84	44.44	27.22	33.19	30.39	22.31	07.32	40.13

Table 2: 3D Object Detection Evaluation results.

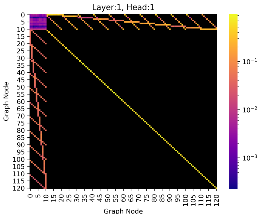


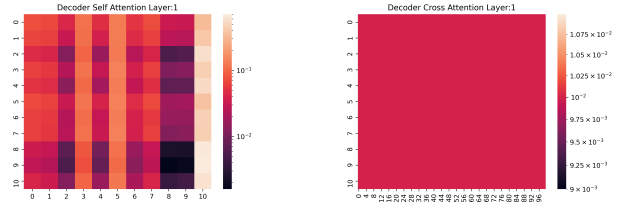
Figure 6: Attention weights of Layer 1 and Head 1 in GAT

For Graph Attention Network, as we adopt the graph construction method from Holistic3D, the objects in the graph are connected to each other and the geometric feature nodes connect pairs of object nodes. We can see that different objects has different attention scores for each object node connection and geometric features in Figure 6. Also, geometric feature nodes give the highest score on self attention.

For Transformer Decoder Network, we can see in Figure 7 different objects has different weights for self attention. However, in cross attention we see that different geometric features have the same attention weights for different objects. We think this may be because all geometric features contribute equally for 2D bounding box to 3D bounding box translation and this results needs further experiments to say a specific reasoning which can be a future work.

For Complete Transformer Network, we can see similar results as Transformer Decoder Network in Figure 8. However, there is a difference between training the Transformer stand-alone and in joint training. In Transformer stand-alone training different geometric features affect objects differently but in joint training all geometric features has same attention weights.

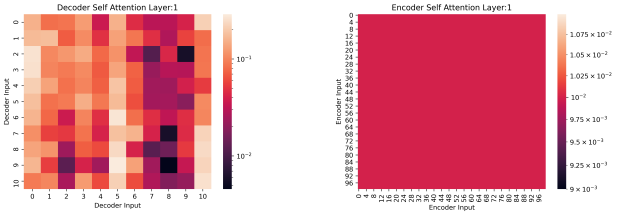
Both for Transformer Decoder and Complete Transformer we can also say in joint training Holistic3D physical loss dominates other loss metrics and geometric features does not contribute adequate information for the objects to update the physical loss as an initial observation.



(a) Self attention

(b) Cross attention

Figure 7: Attention weight of Layer 1 and Head 1 in Transformer Decoder-only



(a) Decoder self attention

(b) Encoder self attention

Figure 8: Attention weight of Layer 1 and Head 1 in Complete Transformer

5. Conclusion

In this work, we bring a new perspective to the holistic 3D scene reconstruction problem where we consider it as a translation problem of 2D input to 3D outputs. In order to do so, we extract 2D features from the input image and also we estimate 3D room layout and object bounding boxes as well as shape representation of objects. Later, we refine initial 3D estimations with the help of 2D features.

Our main contributions are Graph Attention Network, Decoder-Only Transformer, and Transformer models to replace Scene Graph Convolutional Network. We had sufficient results with GAT and Transformer models, even competitive results in some metrics. The limiting factor for reaching a greater performance for both Holistic3D and our approaches is the representation of categories in Pix3D and SUN RGB-D. Thus, creating a more comprehensive dataset in terms of number of categories and number of objects in each category would carry the field forward. Lastly, we think that building a fully-differentiable model with joint optimization of all modules would be a challenging future research direction.

References

[1] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks?, 2021. 2

[2] Saumitro Dasgupta, Kuan Fang, Kevin Chen, and Silvio Savarese. Delay: Robust spatial layout estimation for cluttered indoor scenes. pages 616–624, 06 2016. 1

[3] Siyuan Huang, Siyuan Qi, Yinxue Xiao, Yixin Zhu, Ying Nian Wu, and Song-Chun Zhu. Cooperative holistic scene understanding: Unifying 3d object, layout, and camera pose estimation, 2019. 1

[4] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3d object detection via transformers, 2021. 3

[5] Arun Mallya and Svetlana Lazebnik. Learning informative edge maps for indoor scene layout prediction. pages 936–944, 12 2015. 1

[6] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image, 2020. 1, 3

[7] Yuzhuo Ren, Chen Chen, Shangwen Li, and C. C. Jay Kuo. A coarse-to-fine indoor layout estimation (cfile) method, 2016. 1

[8] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, pages 567–576. IEEE Computer Society, 2015. 3

[9] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B. Tenenbaum, and William T. Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling, 2018. 3

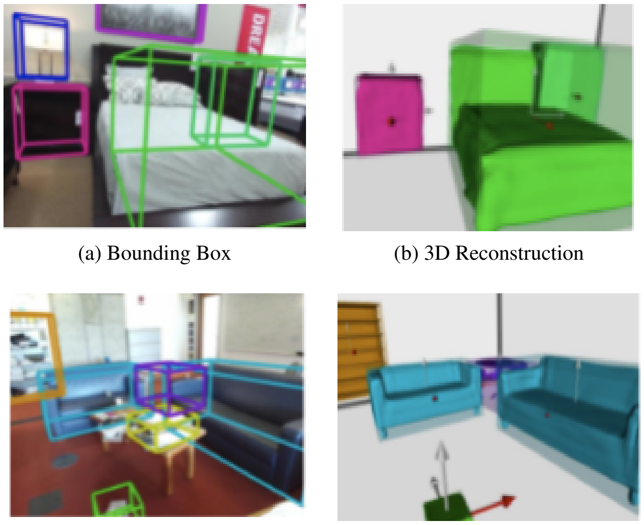
[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. 3

[11] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018. 2

[12] Cheng Zhang, Zhaopeng Cui, Yinda Zhang, Bing Zeng, Marc Pollefeys, and Shuaicheng Liu. Holistic 3d scene understanding from a single image with implicit representation, 2021. 1, 3

A. Visualizations

In this appendix we provide example visualizations from our models.



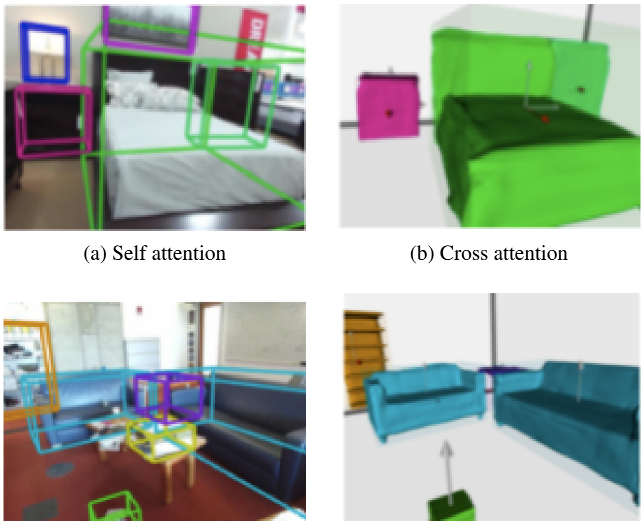
(a) Bounding Box

(b) 3D Reconstruction

(c) Bounding Box

(d) 3D Reconstruction

Figure 9: Visualization outputs from our Graph Attention Network



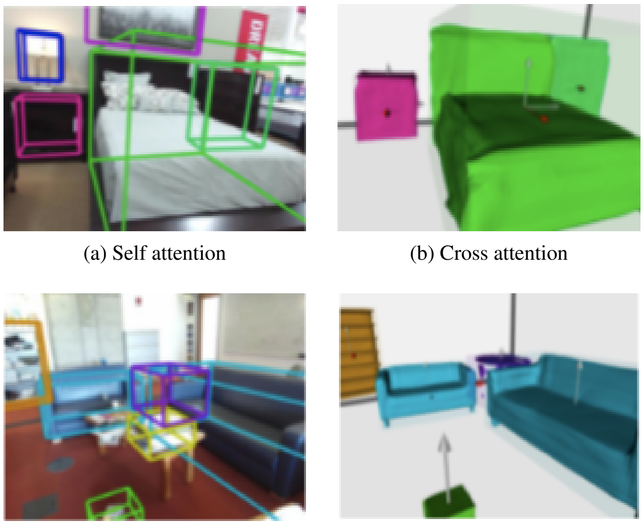
(a) Self attention

(b) Cross attention

(c) Self attention

(d) Cross attention

Figure 10: Visualization outputs from our Decoder-only Network



(a) Self attention

(b) Cross attention

(c) Self attention

(d) Cross attention

Figure 11: Visualization outputs from our Complete Transformer Network