Semantic Scene Completion from a Single Depth Image with Coarse-Grained Segmentation

Jiun Yen Ching*, Lai-Kuan Wong† and Fabian Wai-Lee Kung†
*,† Center for Image and Visual Computing, Multimedia University, Malaysia
* E-mail: 1132700451@student.mmu.edu.my
† E-mail: {lkwong,wlkung}@mmu.edu.my

***Abstract*—Semantic Scene Completion (SSC) plays an important role in computer vision applications such as mobile robots navigation. SSC aims to reconstruct a complete 3D volumetric scene from an incomplete 3D scene in the form of a single depth image, by assigning each voxel in the camera frustum a class label. The main challenge lies in directly predicting the fine-grained classes in a complex and class-diverse indoor environment. To resolve this challenge, we first introduce a novel class taxonomy for SUNCG scene labels based on common shared characteristics, whereby the hierarchy in the taxonomy describes the relationship between coarse-grained and fine-grained classes. We then designed a lightweight Coarse-Grained Segmentation (CGS) module, an adaptable decoder, to exploit the structured class relations and explicitly model the high- and mid-level features. We implemented and validated the CGS module with three existing SSC models, i.e., SSCNet, EdgeNet-D and Real-Time. Experimental results demonstrate that the incorporation of the CGS module improved the performance of fine-grained semantic segmentation of existing SSC models on NYUv2 and NYUCAD datasets, demonstrating the effectiveness of our approach.**

I. Introduction

We live in a 3D world where our surroundings are occupied by physical entities. Therefore, the ability to navigate and interact with the physical world is crucial for intelligent robotic systems. Sensors like RGB-D cameras and LiDAR are capable of capturing high-quality 3D images but are invariably limited by occlusion. For autonomous robots to function effectively, it is crucial that robotic systems are able to infer the semantics of the occluded regions given the visible information.

In recent years, research on semantic scene completion (SSC), which simultaneously predicts a complete 3D scene and classifies each voxel with a class label from a single depth image of a scene with occluded regions, is gaining increasing interest. The seminal work, Song *et al.* [1], utilized only depth data and proposed Flipped Truncated SDF (F-TSDF) which provides a higher gradient across surfaces and therefore improves accuracy. Follow-up works reasoned that depth information can be supplemented with RGB and/or HHA images to provide textural information that cannot be achieved with depth alone. SATNet [2] utilized the results of 2D semantic segmentation based on RGB and depth images by transforming them into 3D representation, and completed them using 3D CNNs. While performance of SSC is steadily improving, models are becoming increasingly complex and computationally expensive. With the aim to improve efficiency, VVNet [3] replaced part of the 3D CNN architecture with

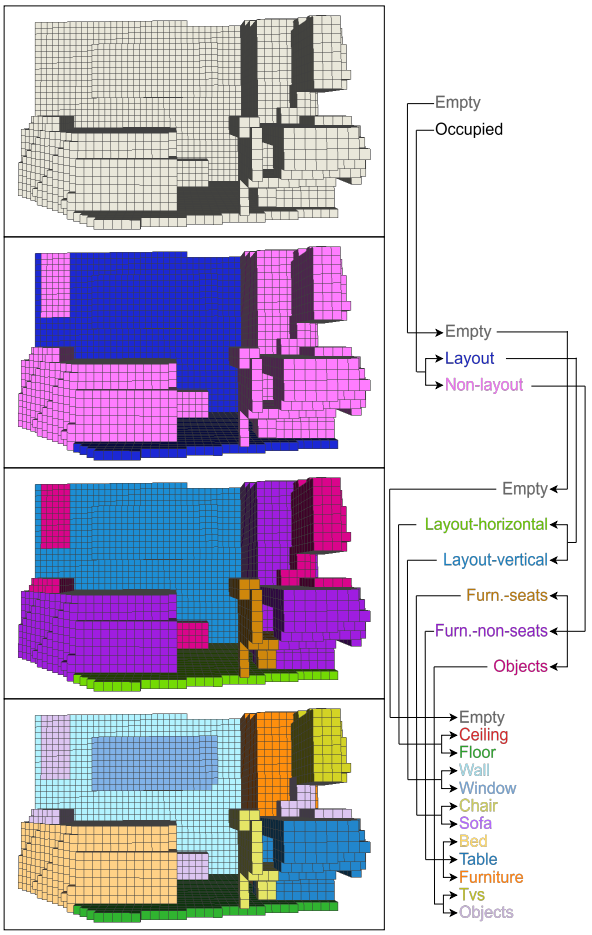

Fig. 1: Class taxonomy for fine- and coarse-grained labels (left) with the corresponding class relations (right) for SUNCG dataset.

2D CNNs since 2D convolution operations require less memory, and connected 2D features with 3D convolutions via a differentiable projection layer. ESSCNet [4] divided the input voxels into different groups and performed spatial group convolution (SGC) to achieve more efficient computations at the slight cost of accuracy. CCPNet [5] made use of multiscale spatial features and successively aggregated them in a self-cascading manner to produce full resolution scenes in a coarse-to-fine manner. On the other hand, to achieve practical inference speeds in real time, [6] used a low-resolution input F-TSDF and leveraged conditioned prediction to ease the

979-8-3315-7206-8/25/$31.00

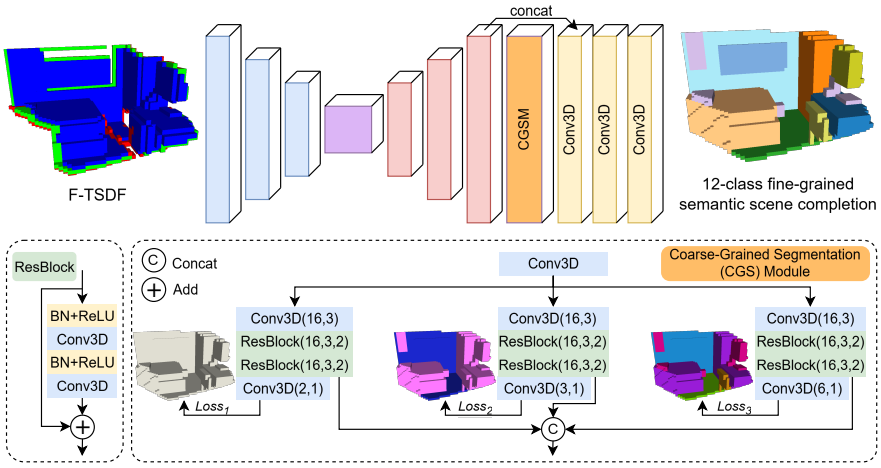

Fig. 2: Overview of the proposed method. Typical encoder-decoder architecture with final convolutions to produce final 3D scene segmentation (top). The CGS module is composed of three identical streams apart from the final convolution layer - each for predicting one class rank (bottom). The parameters for convolution are (channel number, kernel size) and ResBlock are shown as (channel number, kernel size, dilation rate).

task of predicting semantic labels. To reduce computational complexity, DDRNet [7] proposed factorizing 3D convolution layers. PCANet [8] pursued this approach further by proposing Planar Convolution.

Notably, real-world visual scenes possess an inherent hierarchical structure. Leveraging this hierarchical structure, researchers have proposed hierarchical semantic segmentation approaches [9], [10] that incorporate multi-level contextual information and structured reasoning into the segmentation process to successfully improve segmentation performance in the 2D domain. Despite its potential, explicitly modeling and leveraging a semantic class hierarchy for SSC remains largely unexplored. Several works have built upon Gestalt principles and hierarchical design concepts in their network architectures. SATNet [2] utilized a 2D segmentation network to extract 2D features as semantic priors to improve 3D SSC accuracy. To effectively fuse color and depth features, [11] proposed an integration and refinement module that operates top-down. CCPNet [5] designed the encoder as a stack of self-cascaded convolution layers. Ref. [6], dubbed as Real-Time in this paper, and [12] approached this principle from a geometry point of view by proposing that a structural basis can improve the accuracy of the final segmentation. To our best knowledge, no existing SSC method has explicitly referenced the class hierarchy in their model architectures.

In this work, our aim is to improve existing semantic scene segmentation models by introducing class hierarchy to existing hierarchy-agnostic models. To accomplish this, we must tackle two key challenges. First, the SUNCG dataset [1] lacks an established class hierarchy. Second, we need an efficient way to explicitly model high- and mid-level semantic features without high computational cost. To tackle these challenges, we first establish a taxonomy of class relationships for the SUNCG dataset as illustrated in Figure 1—for example, fine-grained categories like *"wall"* and *"window"* are aggregated under broader labels such as *"Layout-vertical"* while *"ceiling"* and *"floor"* are grouped as *"Layout-horizontal"*. This approach aligns with human intuition, as people naturally simplify objects based on shared characteristics. Next, we proposed a lightweight and scalable module to predict coarse-grained classes, which we termed as Coarse-Grained Segmentation (CGS) module. This module extracts contextual features for the prediction of coarse-grained segmentation at each hierarchical level, which are then combined with the primary features to enhance fine-grained class prediction.

We evaluated our CGS module on two public indoor benchmark datasets, i.e., NYUv2 [13] and NYUCAD [14]. Extensive experimental results with different depth-based SSC architectures, i.e., SSCNet [1], EdgeNet-D [15], and Real-Time [6] validates the generalization and effectiveness of our proposed method. In summary, our contributions are threefold:

- We propose a novel class taxonomy for the SUNCG dataset, a large-scale synthetic indoor dataset. The class hierarchy is divided into four levels with different semantic granularity.
- We design a lightweight, scalable decoder module for the explicit modeling of coarse-grained semantic features that are then used to refine fine-grained semantic scene labels.
- Extensive experiments on public real and synthetic benchmark datasets show that our proposed CGS module consistently improves the semantic scene segmentation performance on several depth-based SSC architectures.

II. Methodology

Inspired by the success of employing hierarchical scene structure to improve semantic segmentation in the 2D domain, we proposed a Coarse-Grained Segmentation module that can be incorporated into a 3D SSC network with an encoder-decoder structure. Specifically, CGS predicts several intermediate coarse-grained segmentations and fuse the feature embeddings from each hierarchical level to improve the final fine-grained semantic segmentation labels.

Figure 2 illustrates the overall structure of our proposed method. Given an input F-TSDF, our model assigns each voxel

a semantic label $c \in \{c_0, c_1, ..., c_N\}$, where N is the number of semantic categories, and c_0 stands for empty voxels. To explicitly model high-level semantic features of coarse-grained object categories, we first introduce a novel rank taxonomy for the SUNCG categories [1] as illustrated in Figure 1. We then design a lightweight, parallel-branch module to extract the coarse features from each hierarchical level, which are then used to refine the fine-grained prediction results.

A. Rank Taxonomy

The current 12-class SUNCG labels was inherited from the original 894-class NYUv2 [13] label set. Since then, the label set has been simplified through many-to-one relations, e.g., $window$, $blinds$, $curtain$ are mapped to $window$. We follow this mapping convention to create a class hierarchy with 2/3/6/12 relations. The *empty* class is always assigned to itself throughout all levels of hierarchy. We consider SUNCG categories as *fine-grained* and class labels from all other levels of hierarchy as *coarse-grained*. In addition, categories in higher ranks are dubbed as *parent* classes, relative to their subclasses, which are defined as *child* classes. The rank taxonomy can be formally defined as:

$$T = \{R_1, R_2, ..., R_M\} \tag{1}$$

where R_M is the rank and M denotes the rank level. We select $M = 4$ where R_4 denotes the SUNCG label set. To obtain coarse-grained labels, we progressively cast fine-grained SUNCG labels as such $12 \rightarrow 6 \rightarrow 3 \rightarrow 2$ classes. We avoid preprocessing the ground truth offline to save disk space. Instead, we cast the labels in real time during model training and testing. The lightweight operation introduces little to no computational costs in data loading.

B. 3D Volumetric Network

Generally, a typical SSC network is a 3D volumetric network that consists of an encoder, compressed latent features, a decoder, and final reconstructive convolution layers. As shown in overall architecture Figure 2, our proposed CGS module can be injected between the decoder and the final reconstructive layers of an existing SSC network.

Coarse-Grained Segmentation (CGS) Module. Input feature maps to the CGS module are first reshaped by a single convolution layer to standardize and reduce the channel size. Then, three identical convolution branches process the feature maps, where each branch predicts the labels of one hierarchy level. The design of each branch is kept simple to avoid introducing a large number of parameters. It comprises a top and bottom convolution layer with two dilated pre-activation Residual Block in the middle. The identical design of each branch enables scalability according to the number of ranks used in the module. The output features of each branch are then aggregated by concatenation as shown in Figure 2. Therefore, the output of the CGS module is formally defined as

$$X_{ac} = \begin{cases} [X_1, X_2, ..., X_K], \\ c_1 < c_2 < ... < c_K. \end{cases} \tag{2}$$

where X_{ac} denotes the final aggregated context, X_K is the K-rank taxonomy and c_K is the number of semantic categories. $[\cdots]$ is the channel-wise concatenation operation. We use all levels of the class hierarchy to extract multi-level semantic features to supplement the fine-grained semantic features.

Data Balancing. In most scenes, the number of occluded-empty voxels is much larger than the number of occupied voxels. To address this imbalance, we follow Song *et al.* [1] to randomly sample the occluded voxels to balance the N occupied and $2N$ empty-occluded voxels selected for training. The voxel sampling locations are kept consistent for both the fine- and coarse-grained label volumes.

Loss Function. The weighted softmax cross entropy (WCE) loss is employed to train the proposed end-to-end network:

$$L_K = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} w_c y_{nc} log\left(\frac{e^{\hat{y}_{nc}}}{\sum_{c'=1}^{C} e^{\hat{y}_{nc'}}}\right) \tag{3}$$

where y_{nc} is the one-hot ground truth vector, i.e., y_{nc} = 1 if voxel n is labeled by class c, otherwise y_{nc} = 0. C and N are the number of classes and voxels respectively. w_c is the class weight. K is the taxonomic rank where $K \in \{1, 2, 3, 4\}$. For training, the total loss is the summation of fine- and coarse-grained semantic completion losses as shown in 4:

$$L = \alpha \cdot L_1 + \beta \cdot L_2 + \gamma \cdot L_3 + \delta \cdot L_4 \tag{4}$$

In our experiments, we set $\alpha = \beta = \gamma = \delta = 1$.

C. Post-Training Refinement

Directly predicting 12 fine-grained classes is challenging, especially when the shape and size of objects are class-ambiguous. We take advantage of the fact that predicting fewer classes is more effective than predicting the 12 target classes. Thus, we refine the fine-grained label predictions with results obtained from coarse-grained segmentation. First, we convert the fine-grained output logits into initial class probabilities using Softmax, and keep the coarse-grained raw logits unmodified. We avoid performing Softmax on raw logits during inference to save computation. Instead, let $x_n^{R_K}$ be the raw logit for voxel that belongs to class c in rank K. Suppose $x_n^{R_K}$ has the values $x_0, x_1, \ldots, x_C$. We binarize the $x_n^{R_K}$, where $x_n^{R_K}$ is assigned a value of 1 if it is equivalent to the maximum logit value and 0 otherwise, as shown in 5:

$$B(x_n^{R_K}) = \begin{cases} 1, & \text{if } x_n^{R_K} = \max\{x_0, x_1, \ldots, x_C\} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where B indicates the binary operation. The results are then distributed to their respective child classes and summed to determine the final prediction. The final fine-grained class prediction, c^{R_4} is formally formulated as

$$x_n^{R_4} = S(x_n^{R_4}) + \sum_{K=1}^{3} B(x_n^{R_K}) \tag{6}$$

$$c^{R_4} = \arg\max_n x_n^{R_4} \tag{7}$$

where S is the Softmax operation, and $x_n^{R_4}$ denotes the final value of the voxel for class c.

TABLE I: **Quantitative results on NYUv2 test set.** * denotes our implementation of the original model.

	scene completion			semantic scene completion											
Methods	prec.	recall	IoU	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
SSCNet[1]	57.0	**94.5**	55.1	15.1	**94.7**	24.4	0.0	12.6	32.1	35.0	13.0	**7.8**	27.1	10.1	24.7
SSCNet*	**83.2**	75.9	65.6	42.1	94.3	36.8	0.0	19.7	**62.8**	44.0	19.6	0.0	42.1	17.2	34.4
CGS-SSCNet	77.5	84.3	**67.6**	**47.3**	94.2	**37.9**	**2.3**	**24.6**	60.0	**53.3**	**24.6**	0.0	**42.9**	**23.3**	**37.3**
EdgeNet-D*[15]	**84.8**	74.0	65.0	40.1	**94.2**	**37.6**	0.3	12.7	**63.8**	**48.9**	21.3	0.0	**42.5**	16.5	34.4
CGS-EdgeNet-D	79.5	**80.8**	**66.8**	**46.1**	94.1	37.4	**10.8**	**24.3**	57.1	48.2	**24.4**	**5.2**	39.8	**24.1**	**37.4**
Real-Time[6]	-	-	**73.4**	-	-	-	-	-	-	-	-	-	-	-	**34.4**
Real-Time*	76.2	**84.0**	66.3	34.0	**94.1**	29.1	0.0	14.0	**60.0**	44.0	18.2	0.0	37.1	15.9	31.5
CGS-Real-Time	**76.4**	78.4	62.8	**42.7**	93.9	**39.7**	**0.6**	**20.1**	57.4	**45.1**	**22.6**	**0.1**	**38.7**	**17.5**	**34.4**

TABLE II: **Quantitative results on NYUCAD test set.** * denotes our implementation of the original model.

	scene completion			semantic scene completion											
Methods	prec.	recall	IoU	ceil.	floor	wall	win.	chair	bed	sofa	table	tvs	furn.	objs.	avg.
SSCNet [1]	75.4	**96.3**	73.2	32.5	92.6	40.2	8.9	33.9	57.0	59.5	28.3	**8.1**	44.8	25.1	40.0
SSCNet*	**91.5**	86.0	79.5	68.4	**95.0**	64.0	1.8	47.0	62.1	60.9	37.9	0.1	**55.5**	30.2	47.5
CGS-SSCNet	90.0	89.0	**81.1**	**69.5**	94.9	**65.8**	**13.4**	**50.6**	**64.9**	**64.7**	**38.2**	3.7	55.0	**31.0**	**50.2**
EdgeNet-D* [15]	**90.9**	85.7	78.9	67.1	**95.1**	64.1	5.3	38.7	63.1	62.1	34.0	**0.0**	**56.3**	28.2	46.7
CGS-EdgeNet-D	89.6	**89.9**	**81.5**	**71.5**	95.0	**66.7**	**9.0**	**48.2**	**68.0**	**67.8**	**40.9**	**0.0**	56.2	**31.2**	**50.4**
Real-Time [6]	-	-	**82.2**	-	-	-	-	-	-	-	-	-	-	-	44.5
Real-Time*	**83.0**	**85.0**	72.2	62.4	94.5	51.7	3.4	35.4	**58.9**	51.0	28.8	0.6	47.4	21.8	41.4
CGS-Real-Time	82.0	75.3	64.4	**66.4**	**94.6**	**58.6**	**5.9**	**41.8**	54.8	**54.9**	**34.7**	**3.2**	**50.9**	**26.8**	**44.8**

III. Experimental Results

A. Datasets and Implementation Details

Datasets. We evaluate the proposed method on the NYUv2 [13] and NYUCAD [14] datasets. **NYUv2** is a real-life RGB-D indoor dataset consisting of 1,449 aligned pairs of RGB and depth images with a resolution of 480×640 pixels and train/test split of 795/645 samples, respectively. The manually labeled volumes and their corresponding depth data have poor alignment and regions of missing depth. **NYUCAD** is a semi-synthetic dataset, created to alleviate the misalignment in NYUv2. Ref. [14] provided the NYUCAD dataset, in which the depth data is rendered from the same 3D meshes that were voxelized to produce the ground truth.

TABLE III: Experimental settings for each model.

Models	Grid size (m)	Trunc. value (m)	In, Out	Batch size	Epochs	LR Policy
CGS-SSCNet	0.02	0.24	(240,60)	4	90	OneCycle
CGS-EdgeNet-D	0.02	0.24	(240,60)	4	90	OneCycle
CGS-Real-Time	0.08	0.06	(60,60)	16	500	Polynomial

Implementation Details. We implement SSCNet, EdgeNet-D, and Real-Time with PyTorch and train these models from scratch. For models injected with our CGS module, we add a prefix "CGS" to the model's name, e.g. CGS-SSCNet. The batch size, number of training epochs, and learning rate policies are listed in Table III. All models are optimized with Weighted Cross Entropy (WCE) Loss using Stochastic Gradient Descent (SGD) with a momentum of 0.9 and a weight decay of 0.0005.

B. Evaluation Metric

For the fine-grained semantic segmentation (SS) task, we report the IoU for each class for both observed and occluded voxels and the averaged result over all classes (mIoU) except for the *empty* class. For scene completion (SC), we treat all voxels as binary predictions, i.e., non-empty voxels are labeled as '1', while empty voxels are labeled as '0'. We report the precision, recall, and Intersection over Union (IoU) between the predicted label and ground truth label on only the occluded region. Voxels outside the view frustum or the room are not considered. Since the inception of SSC, there have been differences between researchers in defining the SC evaluation region. Ref. [8] takes into account all non-empty voxels while others [1], [6], [15] ignore surface voxels in the evaluation of SC. In our work, we follow the latter and report the scores obtained along with the scores of the corresponding papers for a fair comparison. Note that semantic segmentation is a more crucial aspect of SSC [16].

C. Quantitative Evaluation

Quantitative performance comparison of the base models and the corresponding models injected with our proposed CGS module are shown in Table I and Table II. The models are evaluated on the NYUv2 and NYUCAD datasets.

Table I shows the results on real-life images. For SSCNet, the results show that our method improved the original performance by 2.9%. For EdgeNet-D, the results show the corresponding model with CGS module, CGS-EdgeNet-D improved the original performance by 3.0%. For Real-Time, we improved the mean-IoU by 2.9% when compared to our re-implementation of the model. For predicting SC, we use the

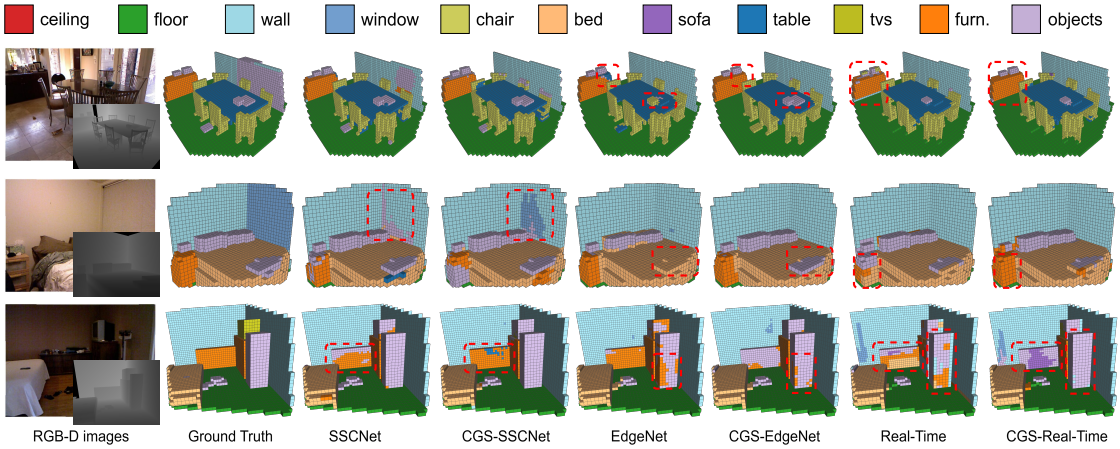

Fig. 3: **Comparison results between vanilla SOTA and our modified models on NYUCAD.** Results show that our proposed method is able to more precisely predict classes when compared to its vanilla counterpart. (Best viewed in color).

model's original stream. A hybrid of both original stream and CGS degrades SC. Across all three models, our method scores higher in majority classes' IoU and mIoU.

Table II shows the results of the comparison on the NYU-CAD dataset on models with and without our proposed CGS module. The result we reported for the original SSCNet is trained on SUNCG and fine-tuned on NYUCAD as the original paper did not have results from training and testing on NYUCAD alone. The results show that our CGS module improved the majority classes by a 2.7%. EdgeNet-D [15] did not train nor test on NYUCAD, hence we implemented their model and performed training and testing on the synthetic dataset to establish a baseline score. CGS-EdgeNet-D shows improvement when implemented with our module, with most classes improving. The mean IoU for the SS task improved by a 3.7%. Real-Time [6] did not report individual class scores. It reports only the IoU and mean-IoU scores for SC and SS tasks, respectively. The result shows that CGS-Real-Time obtains a higher score for most classes when compared with its re-implementation. The average SS IoU is increased by a margin of 3.4%.

Visual Results. Visual results are presented in Figure 3. In the first row, EdgeNet-D* incorrectly predicts a portion of the cabinet and confuses the objects on the table as *chair* and *table* while our method is able to recover the correct class. Real-Time* also confuses the lower portion of the cabinet and the objects on the cabinet, while CGS-Real-Time correctly predicts them. In the second row, our method is able to recover some parts of the window, object on the bed, and furniture when compared to its corresponding model. In the third row, our method is mostly able to correctly predict the drawer (center) and the mechanical weighing scale (right) compared to its base model.

Efficiency Analysis. We analyze the effect on computational efficiency following Li *et al.* [7] on models with and without our implementation. The statistics are listed in Table V. The proposed method increases computational complexity by a fixed amount, with the exception of the convolution reshaping layer. Although the number of parameters is increased within the range of 111 K to 153 K parameters, we see an increase in SS mean-IoU within the range of 2.7% to 3.7%. Despite the significant performance gain for Real-Time*, we observe an increase in FLOPs and decrease in the FPS, which is a typical speed-performance trade-off.

TABLE IV: Ablation studies in the usage of different ranks for Real-Time* on NYUCAD

Rank 1				✓	✓	✓		✓
Rank 2			✓			✓	✓	✓
Rank 3		✓			✓		✓	✓
Rank 4	✓	✓	✓	✓	✓	✓	✓	✓
SC-IoU (%)	69.9	73.5	72.0	73.1	70.5	**74.5**	63.4	69.2
SS-mIoU (%)	42.3	43.0	42.1	42.0	43.1	43.3	42.6	**44.2**

TABLE V: Efficiency analysis on the NYUCAD test set.

Methods	**Params** (k)	**FLOPs** (G)	**Speed** (FPS)	**Memory** (M)	**SSC mIoU** (%)
SSCNet*[1]	930	327	74	1841	47.5
CGS-SSCNet	1117	363	67	2367	50.2
EdgeNet-D*[15]	3440	191	33	3952	46.7
CGS-EdgeNet-D	3593	231	27	4478	50.4
Real-Time[6]	65	1.6	110	655	44.5
Real-Time*	79	3.2	354	398	41.4
CGS-Real-Time	190	32	232	773	44.8

D. Ablation Studies

To evaluate the effectiveness of our proposed components, we performed ablation studies with Real-Time* on NYUCAD. Table IV shows the contribution of all possible taxonomic rank combinations. We also evaluate the contribution of each rank prediction result towards the final SS mean-IoU.

Does adding more ranks improve results? We experiment with different combinations of class hierarchy. Table IV shows the results of Real-Time* on the NYUCAD dataset with different class hierarchies. In this setup, we replace the original preconditioned prediction with R_1 stream. Training all ranks gives the best score for SS-mIoU.

TABLE VI: Ablation on the contribution of different ranks during post-training refinement on Real-Time* on NYUCAD

Rank 1				✓	✓	✓		✓
Rank 2			✓			✓	✓	✓
Rank 3		✓			✓		✓	✓
Rank 4	✓	✓	✓	✓	✓	✓	✓	✓
SS-mIoU (%)	43.3	43.5	43.9	44.0	43.8	**44.2**	44.1	**44.2**

Rank contribution in post-training refinement. In this ablation setup, all taxonomic ranks are used during training. We isolate and verify the contributions of each rank prediction during refinement as shown in Table VI. Results show that using Rank 1, compared to Ranks 2 and 3 in isolation, produces the highest SS refinement. This is because predicting a larger number of classes inherently increases the difficulty of making accurate predictions. Combining the result of Rank 1 and 2 achieves the same result as combining the result of all ranks.

IV. Conclusion

In this work, we introduced a novel class hierarchy for the indoor SUNCG dataset. To leverage the inherent class relations and explicitly model the coarse features, we proposed a lightweight, scalable decoder that is adaptable to most existing depth-based SSC models. To further exploit the coarse predictions, we propose a simple refinement step to improve the final segmentation. We conducted experiments on NYU and NYUCAD to validate our method. The experimental results on three models achieve higher scores than their vanilla counterpart.

V. Acknowledgment

This work is supported by the Malaysia MOHE FRGS Grant FRGS/1/2023/ICT02/MMU/02/3. Special thanks to Prof. Wen-Nung Lie, ISP Multimedia Lab 520 and TEEP Project, under the sponsorship of MOE, Taiwan.

References

[1] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1746–1754.

[2] S. Liu, Y. Hu, Y. Zeng, *et al.*, "See and think: Disentangling semantic scene completion," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[3] Y.-X. Guo and X. Tong, "View-volume network for semantic scene completion from a single depth image," *arXiv preprint arXiv:1806.05361*, 2018.

[4] J. Zhang, H. Zhao, A. Yao, Y. Chen, L. Zhang, and H. Liao, "Efficient semantic scene completion network with spatial group convolution," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 733–749.

[5] P. Zhang, W. Liu, Y. Lei, H. Lu, and X. Yang, "Cascaded context pyramid for full-resolution 3d semantic scene completion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7801–7810.

[6] X. Chen, Y. Xing, and G. Zeng, "Real-time semantic scene completion via feature aggregation and conditioned prediction," in *2020 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2020, pp. 2830–2834.

[7] J. Li, Y. Liu, D. Gong, *et al.*, "Rgbd based dimensional decomposition residual network for 3d semantic scene completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7693–7702.

[8] J. Li, Q. Song, X. Yan, Y. Chen, and R. Huang, "From front to rear: 3d semantic scene completion through planar convolution and attention-based network," *IEEE Transactions on Multimedia*, vol. 25, pp. 8294–8307, 2023.

[9] P. Meletis and G. Dubbelman, "Training of convolutional networks on multiple heterogeneous datasets for street scene semantic segmentation," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2018, pp. 1045–1050.

[10] L. Li, T. Zhou, W. Wang, J. Li, and Y. Yang, "Deep hierarchical semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1246–1257.

[11] R. Chen, Z. Huang, and Y. Yu, "Am 2 fnet: Attention-based multiscale & multi-modality fused network," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, IEEE, 2019, pp. 1192–1197.

[12] Y. Wang, D. J. Tan, N. Navab, and F. Tombari, "Forknet: Multi-branch volumetric semantic completion from a single depth image," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 8608–8617.

[13] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, Springer, 2012, pp. 746–760.

[14] M. Firman, O. Mac Aodha, S. Julier, and G. J. Brostow, "Structured prediction of unobserved voxels from a single depth image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5431–5440.

[15] A. Dourado, T. E. De Campos, H. Kim, and A. Hilton, "Edgenet: Semantic scene completion from a single rgb-d image," in *2020 25th international conference on pattern recognition (ICPR)*, IEEE, 2021, pp. 503–510.

[16] X. Liu, H. Xie, S. Zhang, *et al.*, "2d semantic-guided semantic scene completion," *International Journal of Computer Vision*, pp. 1–20, 2024.