Robust Double-Encoder Network for RGB-D Panoptic Segmentation

Matteo Sodano Federico Magistri Tiziano Guadagnino Jens Behley Cyrill Stachniss

***Abstract*— Perception is crucial for robots that act in real-world environments, as autonomous systems need to see and understand the world around them to act properly. Panoptic segmentation provides an interpretation of the scene by computing a pixelwise semantic label together with instance IDs. In this paper, we address panoptic segmentation using RGB-D data of indoor scenes. We propose a novel encoder-decoder neural network that processes RGB and depth separately through two encoders. The features of the individual encoders are progressively merged at different resolutions, such that the RGB features are enhanced using complementary depth information. We propose a novel merging approach called ResidualExcite, which reweighs each entry of the feature map according to its importance. With our double-encoder architecture, we are robust to missing cues. In particular, the same model can train and infer on RGB-D, RGB-only, and depth-only input data, without the need to train specialized models. We evaluate our method on publicly available datasets and show that our approach achieves superior results compared to other common approaches for panoptic segmentation.**

I. Introduction

Holistic scene understanding is crucial in several robotics applications. The ability of recognizing objects and obtaining a semantic interpretation of the surrounding environment is one of the key capabilities of truly autonomous systems. Semantic scene perception and understanding supports several robotics tasks such as mapping [5] [28], place recognition [9], and manipulation [36]. Panoptic segmentation [20] unifies semantic and instance segmentation, and solves both jointly. Its goal is to assign a semantic label and an instance ID to each pixel of an image. The content of an image is typically divided into two sets: *things* and *stuff*. Thing classes are composed of countable objects (such as person, car, table), while stuff classes are amorphous regions of space without individual instances (such as sky, street, floor).

In this paper, we target panoptic segmentation using RGB-D sensors. This data is especially interesting in indoor environments where the geometric information provided by the depth can help dealing with challenging scenarios such as cluttered scenes and dynamic objects. Additionally, we address the problem of being robust to missing cues, i.e., when either the RGB or the depth image is missing. This is a practical issue, as robots can be equipped with both, RGB-D and RGB cameras, and sometimes have to operate in poor lighting conditions in which RGB data is not reliable.

All authors are with the University of Bonn, Germany. C. Stachniss is also with the Department of Engineering Science at the University of Oxford, UK, and with the Lamarr Institute for Machine Learning and Artificial Intelligence, Germany.

This work has partially been funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017008 (Harmony).

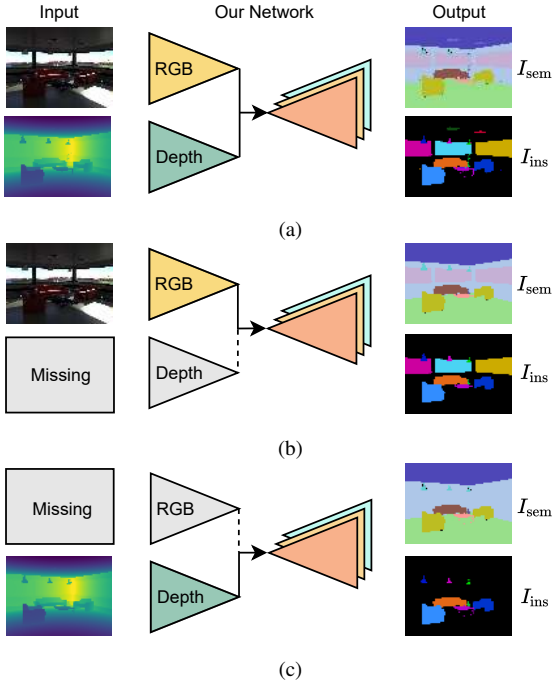


Fig. 1: Our double-encoder network for RGB-D panoptic segmentation is able to provide predictions dealing with full RGB-D images (a), RGB-only (b) or depth-only (c). Dashed lines indicates a detached encoder.

Thus, a single model for handling RGB-D, RGB, and depth data is helpful in practical applications. We investigate how an encoder-decoder architecture with two encoders for the RGB and depth cues can provide compelling results in indoor scenes. Previous efforts showed how double-encoder architectures are effective in processing RGB-D data [29] [37], but they target only semantic segmentation.

The main contribution of this paper is a novel approach for RGB-D panoptic segmentation based on a double-encoder architecture. We propose a novel feature merging strategy, called ResidualExcite, and a double-encoder structure robust to missing cues that allows training and inference with RGB-D, RGB-only, and depth-only data at the same time, without the need to re-train the model (see Fig. 1). We show that (i) our fusion mechanism performs better with respect to other state-of-the-art fusion modules, and (ii) our architecture allows training and inference on RGB-D, RGB-only and depth-only data without the need of a dedicated model for each modality. To back up these claims, we report extensive experiments on the ScanNet [3] and HyperSim [32] datasets. To support reproducibility, our code and dataset splits used in this paper are published at `https://github.com/PRBonn/PS-res-excite`.

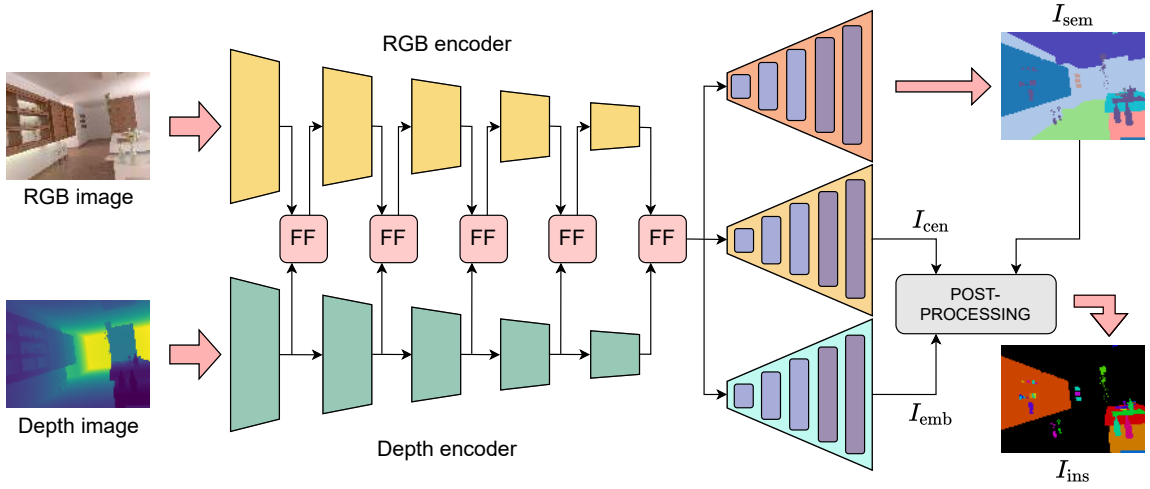


Fig. 2: Our double-encoder network for RGB-D panoptic segmentation. RGB and depth images are separately processed, and their features are merged at different output strides by the feature fusion modules (FF).

II. Related Work

With the advent of deep learning, we witnessed a tremendous progress in the capabilities to provide scene interpretation for autonomous robots. Kirillov et al. [21] define the task of panoptic segmentation as the combination between semantic and instance segmentation. The goal of this task is to assign a class label to every pixel and to additionally segment objects instances. Most of the approaches targeting panoptic segmentation on images tackle it top-down, as they rely on bounding box-based object proposals [15] [20]. Their goal is to extract a number of candidate object regions [11] [17], and then evaluate them independently. These methods are effective but they can lead to overlapping segments in the instance prediction. In this work, we follow bottom-up approaches [2] [8] [33], not relying on bounding boxes but operating directly at a pixel level.

The works mentioned so far use RGB images. Panoptic segmentation is common also for LiDAR data, both in form of range images [24] and point clouds [10]. However, when considering RGB-D data, semantic segmentation [4] [31] and instance segmentation [6] [18] are common, while panoptic segmentation has received less attention so far [26] [42]. The most common ways of elaborating RGB-D data rely on 3D representations via truncated signed distance functions [18] or voxel grids [13]. Few works go in the direction of using directly RGB-D images. In our approach, we target panoptic segmentation directly on RGB-D frames.

Double-encoder architectures are the most successful way for processing 2D representations of RGB-D frames. They allow to process RGB and depth cues separately with individual encoders and rely on feature fusion for combining the outputs of the encoders [30] [37]. An alternative to the direct exploitation of RGB and depth, proposed by Gupta et al. [12], consist in a pre-processing of the depth to encode it with three channels for each pixel, describing horizontal disparity, height above ground and angle between the pixel's surface normal and the gravity direction. The core idea of all these works, however, is that RGB and depth are processed separately and fusion happens only at a later point in the network, after the encoding part (late fusion). Hazirbas et al. [14], however, show that feature merging at different feature resolutions can enhance performance (early-mid fusion). In contrast, we propose to use multi-resolution merging at every downsampling step of the encoder.

Different merging strategies for features of data streams are available. Summation [14] and concatenation [22] are the earliest strategies, which have the limit of considering all features without weighing them according to their effective usefulness. Newest efforts go in the direction of Squeeze-and-Excitation modules [37] and gated fusion [43], which are two different channel-attention mechanisms that aim to increase the focus on features that are more relevant. Other works exploit correlations between modalities to recalibrate feature maps based on the most informative features [38] [40]. In our work, we build on top of channel-attention mechanisms. We propose a new merging mechanism called ResidualExcite, inspired by Squeeze-and-Excitation and residual networks [16], that aims to measure the importance of features at a more fine-grained scale.

Additionally, we leverage the double-encoder structure to have a single model capable of training and inferring on different modalities (RGB-D, RGB-only, depth-only). Multi-modal models have been investigated in the past, but mostly exploiting multiple "expert models" whose outputs are fused in a single prediction, as in the work by Blum et al. [1].

III. Approach to RGB-D Panoptic Segmentation

Our panoptic segmentation network is an encoder-decoder architecture that operates on RGB-D images and processes RGB and depth data by means of two different encoders. Encoders features are merged at different output strides, and are sent to three decoders that restore the backbone features to the original image resolution. The first decoder targets semantic segmentation. The second decoder predicts the location of object centers in the form of a probability heatmap. The third decoder predicts an embedding vector for

each pixel of the image. Finally, a post-processing module aggregates information coming from the last two decoders to obtain instance segmentation in a bottom-up fashion. Fig. 2 illustrates our proposed network architecture. The next sections explain the individual parts of our method.

A. Encoders

Our panoptic segmentation network is based on two ResNet34 encoders [16], which are fed with the RGB image $I_{\text{rgb}} \in \mathbb{R}^{3\times H\times W}$ and the depth image $I_{\text{depth}} \in \mathbb{R}^{1\times H\times W}$, respectively. In both encoders, the basic ResNet block is replaced by the Non-Bottleneck-1D block [34], which allows a more lightweight architecture than the vanilla ResNet, since all 3×3 convolutions are replaced by a sequence of 3×1 and 1×3 convolutions with a ReLU in between, while increasing segmentation performance [37]. We merge features from the two encoders at different output strides and project them into the RGB encoder. We provide more details about our merging strategy in Sec. III-B. After the last merging, the resulting feature is processed by an adaptive pyramid pooling module [44], which has the role of increasing the receptive field of the network. From the RGB encoder, we extract features at different output strides and use them in the decoders by means of skip connections [35].

B. Feature Fusion

We perform feature fusion in the encoders at different output strides. We merge features from the two encoders at every downsampling step, and then send them to the RGB encoder. The depth encoder processes depth features only, to avoid processing the same features with both encoders.

We propose a novel way of merging features, inspired by the Squeeze-and-Excitation module [19]. This module produces a channel descriptor (squeezing operation), and assigns to each channel a modulation weight that is finally applied to the feature map (excitation). Our goal is to obtain a global modulation weight rather than a channelwise weight, as we believe that a more fine-grained reweighing of features is crucial for effective segmentation results. Thus, we remove the squeezing operation, and we add a residual connection. This module, called ResidualExcite (see Fig. 3), is given by

$$\mathbf{X}_{\text{rgb}} = \mathbf{X}_{\text{rgb}} + \lambda \left(E(\mathbf{X}_{\text{rgb}}) \, \mathbf{X}_{\text{rgb}} + E(\mathbf{X}_{\text{depth}}) \, \mathbf{X}_{\text{depth}} \right), \quad (1)$$

where $\mathbf{X}_i \in \mathbb{R}^{C_d \times H_d \times W_d}$, $i \in \{\text{rgb}, \text{depth}\}$ is the feature coming from the respective branch, $E(\mathbf{X}_i) \in \mathbb{R}^{C_d \times H_d \times W_d}$ is the excitation module, which is a sequence of 1×1 convolutions followed by a sigmoid activation function, λ is a (non-trained) parameter for weighing the excitation module over the residual connection, and the subscript d refers to the dimension of the features at the specific output stride in which the merging happens. The RGB and the depth features are both individually excited (meaning both excitation and elementwise multiplication) and then summed, so that each of them can be used separately in case the other cue is missing. Finally, a residual connection adds $\mathbf{X}_{\text{rgb}}$ again.

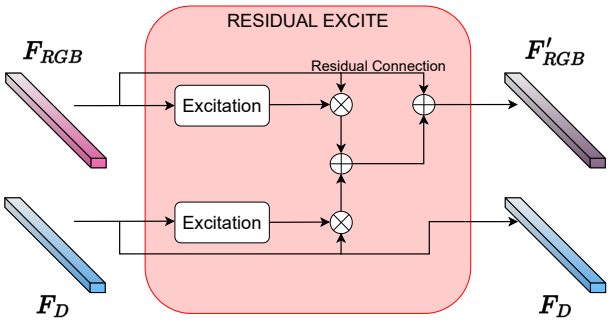


Fig. 3: Detail of the ResidualExcite module. It elaborates the feature maps and produces a novel one that encodes information from both RGB and depth. Symbols $\bigoplus$ and $\bigotimes$ stand for elementwise addition and multiplication, respectively.

C. Decoders

The decoders are composed of three SwiftNet-like modules [27], where we incorporate Non-Bottleneck-1D blocks, and we extend the feature channel to 512 in the first module and then we reduce it as the resolution increases. Finally, two upsampling modules based on nearest-neighbor and depthwise convolutions, that are less computationally expensive than transposed convolutions [37], restore the original resolution. Our model is composed of three decoders, for semantic segmentation, center prediction, and embedding prediction.

Semantic Segmentation. The semantic segmentation decoder has an output depth equal to the number of semantic classes C, $I_{\text{sem}} \in \mathbb{R}^{C\times H\times W}$, and a softmax activation function. It is trained with the usual cross-entropy loss $\mathcal{L}_{\text{sem}}$ for one-hot encoded multi-label classification.

Center Prediction. The center prediction decoder has an output depth of 1, $I_{\text{cen}} \in \mathbb{R}^{1\times H\times W}$, and a sigmoid activation function to predict pixelwise probabilities of being a center. It is optimized with a binary focal loss [25]:

$$\mathcal{L}_{\text{cen}} = \begin{cases} -\alpha \, (1-\hat{y})^{\tau} \, \log(\hat{y}) & \text{, if } y = 1, \\ -(1-\alpha) \, \hat{y}^{\tau} \, \log(1-\hat{y}) & \text{, otherwise,} \end{cases} \quad (2)$$

where α and τ are design parameters and are fixed in all experiments to 0.1 and 2, respectively.

Embedding Prediction. The third decoder of the network predicts a D_{emb}-dimensional embedding vector $I_{\text{emb}} \in \mathbb{R}^{D_{\text{emb}}\times H\times W}$ for each pixel in the image, and is optimized with a composed hinged loss. The first term $\mathcal{L}_{\text{att}}$ attracts embedding vectors of pixels belonging to the same instance, the second term $\mathcal{L}_{\text{rep}}$ repel embedding vectors of pixels belonging to different instances, and the third term $\mathcal{L}_{\text{reg}}$ is a regularization term that avoids exploding entries:

$$\mathcal{L}_{\text{emb}} = \beta_1 \, \mathcal{L}_{\text{att}} + \beta_2 \, \mathcal{L}_{\text{rep}} + \beta_3 \, \mathcal{L}_{\text{reg}}, \quad (3)$$

$$\mathcal{L}_{\text{att}} = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{P_k} \sum_{p=1}^{P_k} \left[\|\hat{\boldsymbol{e}}_k - \hat{\boldsymbol{e}}_p\| - \delta_{\text{a}} \right]^{+}, \quad (4)$$

$$\mathcal{L}_{\text{rep}} = \frac{1}{K(K-1)} \sum_{k_1=1}^{K} \sum_{\substack{k_2=1 \\ k_1 \neq k_2}}^{K-1} \left[\delta_{\text{r}} - \|\hat{\boldsymbol{e}}_{k_1} - \hat{\boldsymbol{e}}_{k_2}\| \right]^{+}, \quad (5)$$

$$\mathcal{L}_{\text{reg}} = \frac{1}{K} \sum_{k=1}^{K} \|\hat{\boldsymbol{e}}_k\|, \tag{6}$$

where $\hat{\boldsymbol{e}}_i \in \mathbb{R}^{D_{\text{emb}}}$ is the unbounded logit predicted by the decoder, K is the number of instances in the image, P_k is the number of pixels of the specific instance, $[\cdot]^+$ corresponds to $\max(0, \cdot)$, and δ_{a} and δ_{r} are thresholds for attracting and repelling the embedding vectors, respectively. To speed up computations, we compute $\mathcal{L}_{\text{att}}$ only between pixels belonging to an instance and their corresponding center, and $\mathcal{L}_{\text{rep}}$ only among centers of different instances. Similarly, we regularize only the vectors of the centers.

We optimize the network with a panoptic loss that is a weighted sum of the previously-defined terms:

$$\mathcal{L}_{\text{pan}} = w_1 \, \mathcal{L}_{\text{sem}} + w_2 \, \mathcal{L}_{\text{cen}} + w_3 \, \mathcal{L}_{\text{emb}}. \tag{7}$$

D. Post-processing

Our post-processing module computes the instance mask based on the output of the three decoders. Since the center prediction decoder usually outputs blobs around the desired center, we perform a non-maximum suppression operation in order to reduce each blob to a single pixel, filtered by the semantic prediction to ensure consistency.

In particular, centers are first filtered by the semantic prediction I_{sem} to avoid having centers belonging to stuff classes, which do not have any instance. Then, pixels that have a probability of being a center lower than a threshold δ_{cen} are discarded. Next, for each blob, we extract the pixel with the highest probability of being a center. A blob $\mathcal{B}$ is defined as the set of pixels belonging to the same semantic class and having a similar embedding vector. Referring to Ω as the set of pixels who are predicted as centers in I_{cen}, i.e., $\Omega = \{p \mid I_{\text{cen}}(p) \geq \delta_{\text{cen}}\}$, a blob is defined as

$$\begin{aligned} \mathcal{B} = \big\{ p_i,\, p_j \in \Omega \mid I_{\text{sem}}(p_i) = c \wedge I_{\text{sem}}(p_j) = c \\ \wedge \, \|\hat{\boldsymbol{e}}_{p_i} - \hat{\boldsymbol{e}}_{p_j}\| < \delta_{\text{emb}},\ i \neq j \big\}, \end{aligned} \tag{8}$$

where c is a specific semantic class, δ_{emb} is a threshold for aggregating embedding vectors, and p_i, p_j are generic pixels.

After the center extraction, we perform an agglomerative clustering operation to group pixels to centers according to the Euclidean distance in the embedding space and semantic class. For each center, we compute its distance in the embedding space from all pixels of the same semantic class. This operation is less computationally intensive than the similarity matrix between all pixels of the image, and motivates the use of object centers. Finally, we assign the pixel to a center if their distance in the embedding space is below a threshold θ. The use of the semantic segmentation prediction enforces consistency and avoids grouping pixels belonging to different semantic classes in the same object.

E. Robustness to Missing Inputs

Since we process RGB and depth with two separate encoders, it is possible to feed the network with partial information, i.e., without either RGB or depth, and freeze the part corresponding to the missing data. This can be done also at training time, with a switching mechanism that freezes gradients if no input is provided to one branch. In this way, the frozen encoder does not contribute to the forward and backward pass, and the network can train at the same time with complete RGB-D, RGB-only, or depth-only images. Furthermore, the network is able to infer on different data without the need for re-training. Feature merging with partial data is not an issue, since the remaining cue can still be excited (or squeezed and excited) and processed.

We train the full model with a probability of dropping data (RGB or depth), equal to p_{drop}. This means that the network can train either with the full RGB-D data or not. If data is dropped, then no input is sent to the corresponding encoder, which we freeze. Additionally, we use an adaptive sampling mechanism to choose what needs to be dropped: in particular, if one cue has been dropped more times than the other, its probability of being dropped in the next iteration is reduced. This helps having a more balanced dropping mechanism and alleviates the problem of dropping always the same modality.

IV. Experimental Evaluation

We present our experiments to show the capabilities of our method and compare it with other fusion methods common in the literature. Furthermore, we show performance of models trained with partial data.

A. Experimental Setup

Datasets and Metrics. We test our method on the validation sets of two datasets: ScanNet [3] and HyperSim [32]. ScanNet is composed of 2.5M real-world images organized in 1,513 scenes. HyperSim is a photorealistic synthetic dataset of indoor scenes, and it is composed of 77.4K images organised in 461 scenes. For both datasets, we do not consider stuff classes (wall, floor) for instance segmentation.

For the center prediction, we pre-process the instance masks of both datasets to extract a center ground truth that is inside the object mask. We consider this to be more effective than computing the center of the associated bounding box, which can fall outside the object mask and the segmentation mask, for example in the case of an isolated concave object.

We evaluate our method by means of the panoptic quality (PQ) [20] and the mean intersection over union (mIoU) [7] over all classes for semantic segmentation.

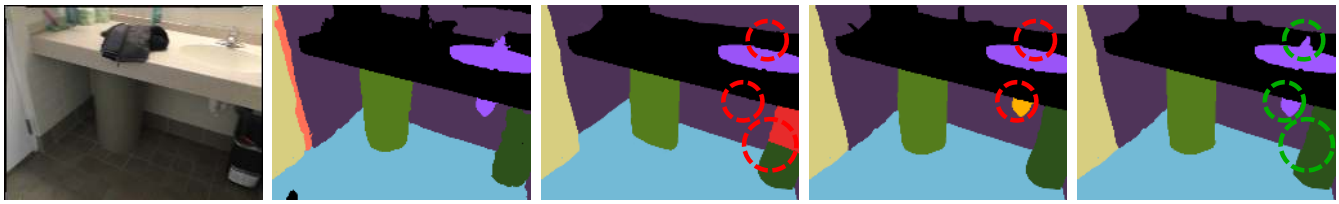
Training details and parameters. In all experiments, except when explicitly specified, we use the one-cycle learning rate policy [39] with an initial learning rate of 0.004. We perform random scale, crop, and flip data augmentation, and optimize with AdamW [23], for 200 epochs. The batch size is set to 32. Additionally, we set $D_{\text{emb}} = 32$ as embedding dimension, $\delta_a = 0.1$, $\delta_r = 1$, $\delta_{\text{emb}} = 0.5$, $\delta_{\text{cen}} = 0.5$, $\theta = 0.5$, and $\lambda = 1.5$. Loss weights are set to $w_1 = 1$, $w_2 = 0.1$, $w_3 = 10$, $\beta_1 = 1$, $\beta_2 = 1$, $\beta_3 = 0.001$.

B. Panoptic Segmentation on RGB-D Images

The first set of experiments evaluates the performance of our proposed method, and offers comparisons to other architectures common in the literature. We base our work on ESANet [37], which is a double-encoder network for RGB-D

(a) Input RGB (b) Ground truth (c) Panoptic DeepLab (d) ESANet + ins. seg. (e) Ours

Fig. 4: Experimental results on ScanNet. Our approach achieves superior segmentation results when compared to the baselines.



(a) Input RGB (b) Ground truth (c) Addition (d) Squeeze and Excitation (e) Ours

Fig. 5: Experimental results on ScanNet. Our approach achieves superior segmentation results when compared to other fusion modules.

(a) Input RGB (b) Ground truth (c) $p_{\text{drop}} = 0.25$ (d) $p_{\text{drop}} = 0.5$ (e) $p_{\text{drop}} = 0.75$

Fig. 6: Results when doing inference on RGB-only after training with missing inputs. The bigger p_{drop}, the better the performance.

Method	Dataset	PQ	mIoU
RGB Panoptic DeepLab [2]	ScanNet	30.11	43.12
RGB-D Panoptic DeepLab	ScanNet	31.43	45.45
ESANet [37] with Addition	ScanNet	35.65	51.78
ESANet [37] with SE [19]	ScanNet	37.09	54.01
Ours with CBAM [41]	ScanNet	39.11	58.11
Ours with ResidualExcite	ScanNet	**40.87**	**58.98**
RGB Panoptic DeepLab [2]	HyperSim	26.10	40.45
RGB-D Panoptic DeepLab	HyperSim	28.56	41.08
ESANet [37] with Addition	HyperSim	32.18	50.74
ESANet [37] with SE [19]	HyperSim	35.87	54.07
Ours with CBAM [41]	HyperSim	37.02	54.21
Ours with ResidualExcite	HyperSim	**38.67**	**55.14**

TABLE I: Performance of the different panoptic segmentation methods. Best result in bold.

semantic segmentation on images. To use it as a baseline for panoptic segmentation, we expand ESANet with the decoders for the center prediction and embedding prediction. Notice that ESANet leverages Squeeze-and-Excitation as a fusion strategy, but reports in the paper also fusion by addition that simply sums up features coming from the two encoders and projects them into the RGB encoder. Here, we use both variants. Additionally, we use another fusion module as baseline, CBAM [41], which infers attention maps along two separate dimensions, channel, and spatial. Furthermore, we also compare against a single-encoder architecture that process the RGB-D image as a four-channel input signal. For that, we adapted Panoptic DeepLab [2] to process images with four channels, and we fed the model with a 4D tensor that is the concatenation of the RGB and the depth images.

We compare our approach to such methods since we focus on image-like data, without relying on 3D representations such as truncated signed distance fields or point clouds. Results are reported in Tab. I, qualitative results are shown in Fig. 4 and Fig. 5. Our reimplementation of Panoptic DeepLab shows inferior performance when compared to ESANet and our approach. We also report numbers from the vanilla implementation of Panoptic DeepLab (called RGB Panoptic DeepLab in the Table) that does not make use of the depth. Interestingly, the performance of the RGB Panoptic DeepLab is close to the one of the RGB-D re-implementation, that simply processes an input with four channels rather than three. This suggests that processing depth as an additional input channel does not add much information, while a separate processing via a second encoder is more effective for such a task. The ResidualExcite module helps segmentation performance, and outperforms other merging strategies such as CBAM and Squeeze-and-Excitation (ESANet + instance segmentation). Fusion by addition shows inferior performance, which is an expected result as it processes all features without weighing them according to their effective usefulness. This experiment indicates that our more fine-grained weighing mechanism, which has effect on each single entry of the encoder feature rather than each channel, enhances performance of the downstream task. Additionally, our network provides end-to-end predictions at 10 Hz, that need to be post-processed to obtain the final instance segmentation mask.

Method	Dataset	mIoU
AdapNet++ [40]	ScanNet	54.61
FuseNet [14]	ScanNet	56.65
SSMA [40]	ScanNet	66.13
Ours (full model)	ScanNet	58.98
Ours (semantic)	ScanNet	**69.78**

TABLE II: Performance of different semantic segmentation models on the ScanNet dataset. Best result in bold.

	RGB-D		RGB-only		Depth-only	
p_{drop}	PQ	mIoU	PQ	mIoU	PQ	mIoU
0	**40.87**	**58.98**	9.12	21.13	12.54	24.57
0.25	31.12	44.55	20.12	34.83	21.18	35.14
0.5	30.73	42.86	25.61	39.18	26.74	38.87
0.75	26.81	40.07	**27.48**	**39.56**	**28.18**	**39.45**

TABLE III: Performance of the model when dropping either RGB or depth with different probability. Best result in bold.

To empirically validate our architecture design, we compare it with some state-of-the-art models from the ScanNet benchmark for semantic segmentation [14] [40]. We use both our full model and its task-specific reduction, in which the decoders for center and embedding prediction are cut out in order to do semantic segmentation only. Table II shows that even if our full model has weaker performance, our task specific model outperforms the baselines. Note that some methods higher in the benchmark rely on multiple frames as input, and thus cannot be directly compared to our approach.

C. Experiments on Robustness to Missing Inputs

The second set of experiments backs up our claim that our approach can train and infer on partial data, such that the network learns to deal with missing RGB- or depth-frames. We test different values for p_{drop}: 0.25, 0.5, and 0.75. This means that the network will drop either the RGB frame or the depth frame according to the specified probability. If dropping happens, we choose which cue to drop according to the adaptive sampling mechanism mentioned in Sec. III-E. This strategy gives a better performance than random sampling, which is therefore not reported here. Tab. III shows performance for inference on RGB-D, RGB-only, and depth-only data. All models produce inferior segmentation results than the model that does not drop any frame (same model discussed above), when doing inference on full RGB-D frames. However, its performance drops substantially when doing inference on partial data, as the network was never trained with missing cues. Additionally, we notice how dropping frames more often makes the model better for doing inference on partial data, since the network trained more with missing cues. On the contrary, low values of p_{drop} bring poor performance when handling partial data, because the network was mainly trained with both, RGB and depth. The model trained with $p_{\text{drop}} = 0.5$ is the best compromise to achieve satisfactory results both on RGB-D, RGB-only, and depth-only, even without reaching the performance of the RGB-D model. Qualitative results are shown in Fig. 6.

All experiments described in Sec. IV-C are done with a batch size of 4 and an initial learning rate of 0.001. Due to the missing inputs, the training procedure is less stable and thus required a smaller learning rate. We use ResidualExcite for merging; experiments are performed on ScanNet only.

RGB	D	SE	E	RE	PQ	mIoU
✓					25.63	38.91
	✓				28.89	41.01
✓	✓				35.65	51.78
✓	✓	✓			37.09	54.01
✓	✓		✓		38.73	55.57
✓	✓			✓	38.80	56.67
✓	✓			✓	**40.87**	**58.98**

TABLE IV: The first two lines refer to RGB- and depth-only. Then, we show double-encoder networks with addition (RGB + D), Squeeze-and-Excitation (SE), ExciteOnly (E) and ResidualExcite (RE). We use ✓ to indicate which branch processes fused features.

D. Ablation Studies

In this last section, we provide ablations to show the improvements provided by the fusion strategy. We perform all ablations on the ScanNet dataset only.

First, we analyze the ResidualExcite and investigate the effect of the residual connection. Without it, the excitation module (ExciteOnly) still provides an entrywise reweighing of the feature. Experiments show that this is already enough to ensure superior performance with respect to other baselines, but the residual connection gives further improvements, see Tab. IV. Additionally, in our case, fusing in the RGB encoder is more effective than fusing in the depth encoder.

In the same table, we compare the performance of the full model reductions in which a single encoder is used. We test panoptic segmentation on RGB-only and depth-only data. The results are clearly inferior to the double-encoder models. Interestingly, depth-only gives better results than RGB-only. This is probably due to the fact that some scenes have challenging lighting conditions, and some objects are hard to recognize in the RGB image. Such information is not lost in the depth image. Also, this suggests that geometric cues may be more relevant than color information when it comes to object recognition for segmentation.

V. Conclusion

In this paper, we presented a novel approach for panoptic segmentation on RGB-D images based on a double encoder architecture with intermediate feature merging. Our method exploits the inner structure of the neural network to enable training and inference when cues are missing using the same model and without the need for retraining. We implemented and evaluated our approach on different datasets and provided comparisons with other existing models and supported all claims made in this paper. The experiments suggest that our more fine-grained reweighing of features is crucial for effective segmentation results. Additionally, models trained with partial data achieve inferior performances on RGB-D segmentation when compared with full models, but they work better when inferring on partial data.

Acknowledgments

We thank Andres Milioto and Xieyuanli Chen for their constructive feedback and useful discussions.

REFERENCES

[1] H. Blum, A. Gawel, R. Siegwart, and C. Lerma. Modular Sensor Fusion for Semantic Segmentation. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2018.

[2] B. Cheng, M.D. Collins, Y. Zhu, T. Liu, T.S. Huang, H. Adam, and L. Chen. Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation. In *Proc. of the CVF/IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[3] A. Dai, A. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[4] A. Dai and M. Niessner. 3DMV: Joint 3D-Multi-View Prediction for 3D Semantic Scene Segmentation. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2018.

[5] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Lerma. SegMatch: Segment Based Place Recognition in 3D Point Clouds. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2017.

[6] F. Engelmann, M. Bokeloh, A. Fathi, B. Leibe, and M. Niessner. 3D-MPA: Multi-Proposal Aggregation for 3D Semantic Instance Segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[7] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *Intl. Journal of Computer Vision (IJCV)*, 88(2):303–338, 2010.

[8] N. Gao, Y. Shan, Y. Wang, X. Zhao, Y. Yu, M. Yang, and K. Huang. SSAP: Single-Shot Instance Segmentation With Affinity Pyramid. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2019.

[9] S. Garg, N. Süderhauf, and M. Milford. Don't look back: Robustifying place categorization for viewpoint and condition-invariant place recognition. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018.

[10] S. Gasperini, M.N. Mahani, A. Marcos-Ramiro, N. Navab, and F. Tombari. Panoster: End-To-End Panoptic Segmentation of LiDAR Point Clouds. *IEEE Robotics and Automation Letters (RA-L)*, 6(2):3216–3223, 2021.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[12] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from RGB-D images for object detection and segmentation. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2014.

[13] L. Han, T. Zheng, L. Xu, and L. Fang. OccuSeg: Occupancy-Aware 3D Instance Segmentation. In *Proc. of the CVF/IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[14] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *Proc. of the Asian Conf. on Computer Vision (ACCV)*, 2016.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2017.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[17] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. What makes for effective detection proposals? *IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI)*, 38(4):814–830, 2015.

[18] J. Hou, A. Dai, and M. Niessner. 3D-SIS: 3D Semantic Instance Segmentation of RGB-D Scans. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[19] J. Hu, L. Shen, and G. Sun. Squeeze-and-Excitation Networks. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[20] A. Kirillov, R. Girshick, K. He, and P. Dollar. Panoptic Feature Pyramid Networks. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[21] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. Panoptic Segmentation. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[22] M. Loghmani, M. Planamente, B. Caputo, and M. Vincze. Recurrent Convolutional Fusion for RGB-D Object Recognition. *IEEE Robotics and Automation Letters (RA-L)*, 4(3):2878–2885, 2019.

[23] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint:1711.05101*, 2017.

[24] A. Milioto, J. Behley, C. McCool, and C. Stachniss. LiDAR Panoptic Segmentation for Autonomous Driving. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[25] A. Milioto, L. Mandtler, and C. Stachniss. Fast Instance and Semantic Segmentation Exploiting Local Connectivity, Metric Learning, and One-Shot Detection for Robotics. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019.

[26] G. Narita, T. Seno, T. Ishikawa, and Y. Kaji. PanopticFusion: Online Volumetric Semantic Mapping at the Level of Stuff and Things. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[27] M. Orsic, I. Kreso, P. Bevandic, and S. Segvic. In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[28] E. Palazzolo, J. Behley, P. Lottes, P. Giguere, and C. Stachniss. ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[29] J. Park, Q. Zhou, and V. Koltun. Colored Point Cloud Registration Revisited. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2017.

[30] S. Park, K. Hong, and S. Lee. RDFNet: RGB-D Multi-Level Residual Feature Fusion for Indoor Semantic Segmentation. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2017.

[31] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun. 3D Graph Neural Networks for RGBD Semantic Segmentation. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2017.

[32] M. Roberts, J. Ramapuram, A. Ranjan, A. Kumar, M.A. Bautista, N. Paczan, R. Webb, and J.M. Susskind. Hypersim: A Photorealistic Synthetic Dataset for Holistic Indoor Scene Understanding. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2021.

[33] G. Roggiolani, M. Sodano, T. Guadagnino, F. Magistri, J. Behley, and C. Stachniss. Hierarchical Approach for Joint Semantic, Plant Instance, and Leaf Instance Segmentation in the Agricultural Domain. *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2023.

[34] E. Romera, J.M. Alvarez, L.M. Bergasa, and R. Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Trans. on Intelligent Transportation Systems (ITS)*, 19(1):263–272, 2018.

[35] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Proc. of the Intl. Conf. on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2015.

[36] M. Schwarz, A. Milan, A. Periyasamy, and S. Behnke. Rgb-d object detection and semantic segmentation for autonomous manipulation in clutter. *Intl. Journal of Robotics Research*, 37(4-5):437–451, 2017.

[37] D. Seichter, M. Köhler, B. Lewandowski, T. Wengefeld, and H. Gross. Efficient RGB-D Semantic Segmentation for Indoor Scene Analysis. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[38] K. Sirohi, R. Mohan, D. Büscher, W. Burgard, and A. Valada. EfficientLPS: Efficient LiDAR Panoptic Segmentation. *IEEE Trans. on Robotics (TRO)*, 38(3):1894–1914, 2021.

[39] L.N. Smith and N. Topin. Super-convergence: Very fast training of neural networks using large learning rates. *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, 11006:369–386, 2019.

[40] A. Valada, R. Mohan, and W. Burgard. Self-supervised model adaptation for multimodal semantic segmentation. *Intl. Journal of Computer Vision (IJCV)*, 128(5):1239–1285, 2019.

[41] S. Woo, J. Park, J.Y. Lee, and I.S. Kweon. Cbam: Convolutional block attention module. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2018.

[42] S.C. Wu, J. Wald, K. Tateno, N. Navab, and F. Tombari. SceneGraphFusion: Incremental 3D Scene Graph Prediction From RGB-D Sequences. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[43] J. Xu, R. Zhang, J. Dou, Y. Zhu, J. Sun, and S. Pu. RPVNet: A Deep and Efficient Range-Point-Voxel Fusion Network for LiDAR Point Cloud Segmentation. In *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*, 2021.

[44] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid Scene Parsing Network. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.