RGB-D Cube R-CNN: 3D Object Detection with Selective Modality Dropout

Jens Piekenbrinck[1], Alexander Hermans[1], Narunas Vaskevicius[2], Timm Linder[2], Bastian Leibe[1]
[1]RWTH Aachen University [2]Robert Bosch GmbH
piekenbrinck,hermans,leibe@vision.rwth-aachen.de
narunas.vaskevicius,timm.linder@de.bosch.com

Abstract

In this paper we create an RGB-D 3D object detector targeted at indoor robotics use cases where one modality may be unavailable due to a specific sensor setup or a sensor failure. We incorporate RGB and depth fusion into the recent Cube R-CNN framework with support for selective modality dropout. To train this model, we augment the Omni3D$_{IN}$ dataset with depth information, leading to a diverse dataset for 3D object detection in indoor scenes. In order to leverage strong pretrained networks, we investigate the viability of Transformer-based backbones (Swin, ViT) as an alternative to the currently popular CNN-based DLA backbone. We show that these Transformer-based image models work well based on our early-fusion approach and propose a modality dropout scheme to avoid the disregard of any modality during training, facilitating selective modality dropout during inference. In extensive experiments, our proposed RGB-D Cube R-CNN outperforms an RGB-only Cube R-CNN baseline by a significant margin on the task of indoor object detection. Additionally, we observe a slight performance boost from the RGB-D training when inferring on only one modality, which could for example be valuable in robotics applications with a reduced or unreliable sensor set. Code and scripts to recreate the RGB-D dataset can be found at: `https://github.com/VisualComputingInstitute/omni3d-rgbd`

1. Introduction

The ability to robustly detect 3D objects and to estimate their poses and extents is an important capability for embodied agents, *e.g.* robots that navigate and interact with a 3D scene. This is a challenging perception task due to the large variety of objects encountered, *e.g.* in typical household or industrial scenarios, varying camera intrinsics, and limited amounts of training data due to costly 3D labels. Recently, Brazil *et al.* [4] introduced Omni3D, a large diverse RGB 3D object detection dataset for indoor and outdoor scenes. Accompanying the dataset, they propose the Cube R-CNN approach, which shows promising results for the task of 3D object detection from RGB image, extending a Faster R-CNN detector with a simple 3D cube head. One key contribution of Cube R-CNN is the use of a virtual depth, allowing it to be trained and evaluated on a wide range of images that do not necessarily share the same camera intrinsics. Moreover, the method predicts bounding boxes with 9 degrees of freedom, *i.e.* full rotation, compared to most other existing approaches that assume axis-aligned objects [25–27, 31]. Especially when objects of interest can be tilted (*e.g.* leaning against a wall), they are being held by a human, or the camera pose is dynamic, such 9 degrees of freedom detection is relevant. Motivated by robotics scenarios with diverse sensor setups that often contain an additional depth modality, we propose a multi-modal extension of the Cube R-CNN detection framework to RGB-D inputs. While specialized approaches exist that use point cloud architectures [25, 26, 32], we are interested in extending a generic RGB backbone such as DLA [39], Swin Transformers [22] or ViTs [8] to RGB-D inputs. This has two key advantages: 1) Modalities, including depth, can be dropped out during training to increase robustness to missing modalities during inference, which is not possible for methods directly working on point clouds. 2) We can utilize strong pretrained image models, avoiding the need to train an RGB-D model from scratch on datasets with less variability. Realizing this extension to RGB-D inputs brings with it a number of interesting questions and design choices, which we thoroughly examine in this paper. 1) *Architecture*: Which common architectures can be used in such a setting without creating highly specific fusion modules or novel multi-modal architectures altogether? 2) *Modality fusion*: At what point in the architecture should the information from RGB and D channels be combined? We experimentally compare early and late fusion strategies based on state-of-the-art Transformer architectures. 3) *Model training*: How should we best train the resulting model? While using networks pretrained on large datasets has become the de facto standard for initializing RGB-based backbone networks, it is not obvious how to best re-use these initializations when training with an additional depth modality. We also employ a modality dropout scheme during training, ensuring the model cannot simply learn to ignore a modality and additionally making it more robust to missing modalities during inference. Here, we systematically evaluate different dropout strategies. In our experiments we focus on the indoor part of the Omni3D dataset, (Omni3D$_{\text{IN}}$), which merges subsets of SUN RGB-D [33], Hypersim [30], and ARKitScenes [3]. While all of these datasets consist of RGB-D images, Omni3D$_{\text{IN}}$ solely provides the RGB images. We retrofit the missing depth images, creating Omni3D$_{\text{IN}}$ RGB-D, enabling us to study RGB-D-based 3D object detectors on Omni3D$_{\text{IN}}$. We are, to our best knowledge, the first to conduct systematic RGB-D experiments on such a diverse and large-scale indoor 3D dataset. We show that our proposed RGB-D Cube R-CNN outperforms an RGB-only Cube R-CNN baseline by a significant margin on the task of indoor object detection. Depending on the exact setup of modality dropout, our model gracefully reverts to the performance of an RGB-only version and in some settings even outperforms models trained on a single modality.

2. Related Work

2.1. 3D Object Detection

The field of 3D object detection is diverse in terms of input representation used (RGB, RGB-D or point clouds), and degrees of freedom (DoF) of the output (translation, scale, rotation). Most current methods for 3D object detection are point-cloud-based [9, 25–27, 32, 36], which directly utilize the geometric structure of the scene, either given by a point cloud or by backprojecting RGB-D images to 3D. However, our focus in this paper is on 3D object detection from multi-modal RGB-D data even when one modality is absent, making point cloud methods not applicable to our research due to their dependency on depth information. Meanwhile, 2D-based methods are inherently relevant for us. ImVoxelNet [31] operates only on RGB images and uses a dense voxel representation of intermediate features to predict 3D boxes with 7DoF. [18] is an example of an RGB-based 9DoF categorical 3D object detection and pose estimation approach. Cube R-CNN [4] is a recently proposed method for image-based 3D object detection. It was introduced as a baseline on a new dataset called Omni3D which covers outdoor and indoor scenarios. Cube R-CNN extends the Faster R-CNN [29] architecture with a 3D cube head to predict 3D bounding boxes in a camera agnostic virtual depth space. For more details about the architecture, see Sec. 3.2. However, none of the aforementioned methods investigate how additional depth information could improve the performance. We chose Cube R-CNN as our base architecture, leveraging the extensible 2D architecture capable of predicting 9DoF 3D bounding boxes and create an RGB-D extension. To train our model, we utilize the associated diverse Omni3D dataset, which we extend to a multi-modal RGB-D dataset.

2.2. Image-based RGB-D Fusion

Recent work demonstrates that leveraging multiple visual modalities like RGB and depth jointly improves performance in downstream tasks including classification [10], 2D segmentation [10, 41], or 3D object detection [20]. Many fusion strategies exist to fuse the color and depth modalities, such as early, mid, or late fusion [11, 19, 27, 41]. Some of these methods introduce fairly complicated new modules and architecture changes. Recently, work has been done on joint training on multi-modal data using a single Transformer-based architecture [10, 35]. Omnivore [10], a Swin Transformer-based [22] approach, is trained for classification on multiple visual modalities, namely images, videos, and RGB-D data. Instead of developing a specialized architecture, the modalities are simply integrated by extending the input token space of the Transformer in such a way that the model can operate given only one modality during inference, but can still implicitly learn from the synergy of multiple modalities. They use an early fusion technique for the RGB-D modalities, by embedding both modalities with separate networks and fusing them by addition. This is of particular interest to us since it allows us to use a standard network with only few modifications. Furthermore, we also drop the RGB image randomly to encourage the model to also use depth cues, which addresses the weak modality problem discussed by Liu *et al*. [20].

2.3. Modality Dropout

Recent developments have yielded unified Transformer-based architectures, that can cope with all visual modalities by simply extending the input embedding space [10, 15]. These unified architectures allow modalities to be optional, for example through some sort of modality dropout [7, 10, 15]. This could enable the model to be jointly trained on multiple modalities, without a hard requirement for all modalities to be available during inference, which can be useful in robotic setups where sensor setups can vary and might be more limited on certain platforms [24].

3. RGB-D Cube R-CNN

In this section, we introduce our proposed method, RGB-D Cube R-CNN. Our contributions are twofold: We extend the Omni3D$_{\text{IN}}$ dataset to RGB-D, and show that by leveraging a Transformer-based encoder paired with modality dropout,

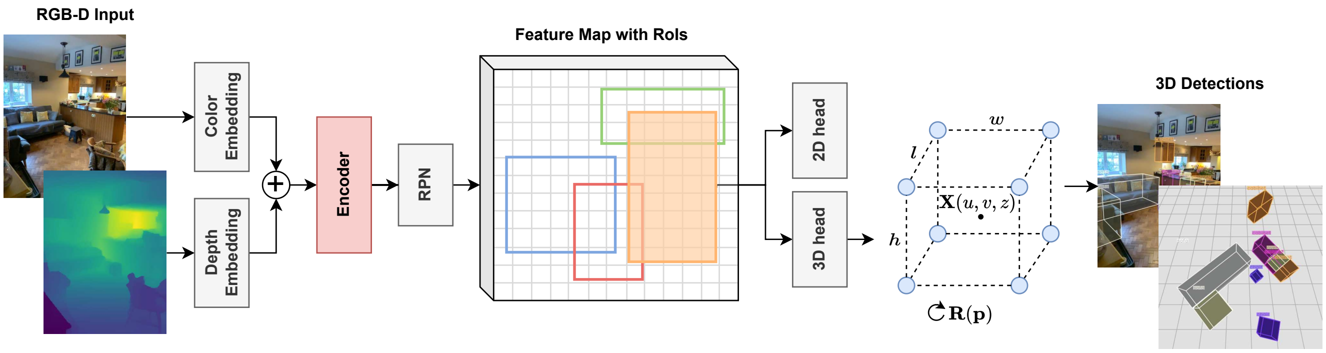


Figure 1. Overview of our RGB-D Cube R-CNN. The overall architecture follows the original Cube R-CNN [4], however, we use RGB-D inputs and experiment with different backbones.

we can boost AP$_{3D}$ performance and outperform the Cube R-CNN baseline with a DLA34 backbone.

3.1. Dataset

The Omni3D dataset encompasses two distinct subsets: Omni3D$_{\text{IN}}$ and Omni3D$_{\text{OUT}}$, representing indoor and outdoor environments, respectively. Our primary focus in this work lies on indoor scene understanding, making Omni3D$_{\text{IN}}$ particularly pertinent. Omni3D$_{\text{IN}}$ is itself a superset of subsets from SUN RGB-D [33], Hypersim [30] and ARKitScenes [3]. However, while these all are RGB-D datasets, Omni3D consists only of their RGB modalities. We extended Omni3D$_{\text{IN}}$ to RGB-D by re-adding the depth modality of the original datasets. Each of the different subsets uses a different depth format though, so we first have to convert them to one common format. For SUN RGB-D, we used a Python-based version of the official Matlab code to convert their depth data to metric depth. For ARKitScenes, depth is already provided in millimeters. Hypersim provides Euclidean distances to the camera origin, which we convert to depth in meters using the given camera intrinsics. As input to our network, instead of using the raw depth in meters, we use inverse depth given by

$$inv(x) = \frac{1}{1 + c \cdot x} \in [0, 1]$$

where x is the depth in meters, which is always positive, and c is a scale factor. This maps the depth into a more well-behaved interval between $[0, 1]$. In all our experiments we set c to 0.5.

3.2. Method

Our method is built upon the recently introduced Cube R-CNN by Brazil *et al.* [4], which itself is based on the Faster R-CNN [29] architecture. Faster R-CNN employs a vision backbone, such as a ResNet [12] with a feature pyramid network [17], that encodes an RGB image into a spatial feature map. The features are further processed by a region proposal network (RPN) that predicts region of interests (RoIs) based on predefined anchor boxes with different scales and aspect ratios. Subsequently, a box prediction head is applied to the features of each RoI to predict a class category and a refined 2D bounding box. The contributions of Cube R-CNN [4] are manifold. The RPN objective is redefined by adopting a new *IoU* based approach instead of the *objectness*. A novel cube head is introduced for 3D bounding box regression. Furthermore, the training objective is based on a *virtual depth* which makes it camera agnostic. We use the same architectural design as Cube R-CNN for the RPN and the RoI heads. Moreover, our method employs the same training objectives and optimization procedure.

A 3D bounding box is defined by its eight corner points and parameterized by 13 parameters for translation, scale and rotation that are predicted by the cube head. Given the focal length f and principal point p of the camera's intrinsics, the 2D bounding box r and the predictions of the projected 3D center on the image plane relative to the 2D RoI, $[u, v]$, and the depth of the center point z, the object center in world coordinates is computed as:

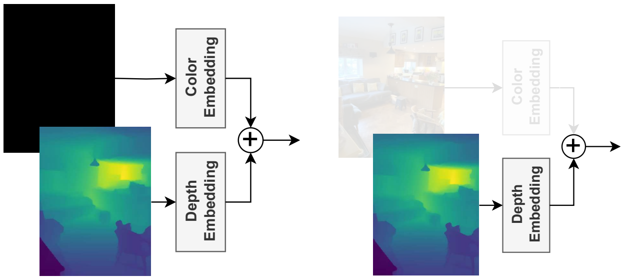
$$\mathbf{X}(u, v, z) = (\frac{z}{f_x}(r_x + ur_w - p_x), \frac{z}{f_y}(r_y + vr_h - p_y), z).$$

The box dimensions $\mathbf{d}$ are computed as a diagonal matrix in form of:

$$\mathbf{d}(\bar{w}, \bar{h}, \bar{l}) = \text{diag}(\exp(\bar{w})w_0, \exp(\bar{h})h_0, \exp(\bar{l})l_0),$$

where $\bar{w}, \bar{h}, \bar{l}$ are the log-normalized width, height and length, respectively, and w_0, h_0, l_0 category-specific precomputed means. The pose of an object is parameterized by a 6D continuous vector $\mathbf{p}$ that represents the allocentric rotation and is converted to the egocentric rotation matrix $\mathbf{R}(\mathbf{p})$. The eight corners of the final 3D bounding box are computed by

$$B_{\text{3D}}(u, v, z, \bar{w}, \bar{h}, \bar{l}, \mathbf{p}) = \mathbf{R}(\mathbf{p})\mathbf{d}(\bar{w}, \bar{h}, \bar{l})B_{\text{unit}} + \mathbf{X}(u, v, z)$$



(a) Pre-embedding dropout.

(b) Post-embedding dropout.

Figure 2. Different RGB dropout approaches. 2a Omnivore's pre-embedding dropout. 2b Our post-embedding dropout.

where B_{unit} is an axis-aligned and origin-centered unit box represented by eight corners. An overview of the architecture and the 3D box parametrization is illustrated in Fig. 1.

Backbones Cube R-CNN adopts the DLA34 [39] backbone paired with a standard FPN. DLA is a common choice within the monocular 3D object detection community [6, 18, 21, 42, 43], for other tasks, backbones such as ResNets [12] or Swin Transformers [22] are more common though. We investigate whether the DLA34 backbone can be replaced with modern and popular Transformer-based architectures, specifically Swin or Vision Transformer [8] (ViT). Swin possesses an inherent feature hierarchy akin to that found in ResNet, making it seamlessly compatible with the conventional FPN. On the other hand, ViT lacks a built-in hierarchy. To address this limitation, we rely on the SimpleFeaturePyramid (SFP) proposed by Li *et al.* [16]. SFP generates a synthetic hierarchy by sampling from the features of the last ViT layer.

Modality Fusion Common modality fusion approaches are early and late fusion. Inspired by Girdhar *et al.* [10], we use a simple but effective early fusion method. After each modality is embedded by a separate single convolutional layer, their embeddings are fused by summation before they are further processed by the encoder. We incorporate normalization for the embedding, considering the original embedding implementation from the corresponding backbone. We additionally examined late fusion using two different approaches. In the first method, the modality embeddings are processed by a shared encoder, while in the second, these embeddings are individually processed using separate encoders $\text{enc}_{\text{color}}$, $\text{enc}_{\text{depth}}$ for each each modality respectively. In both cases, the outputs of the modalities are combined through summation.

Modality Dropout One problem in multimodal fusion is that the model can ignore one of the modalities [10, 15]. To avoid the network from ignoring one modality and to direct it towards exploiting the synergy of both modalities, we add modality dropout. Besides forcing the model to consider both modalities, it allows us to omit one modality during inference without retraining, *e.g.* when switching from an RGB-D to an RGB-only or depth-only sensor (at the cost of performance degradation). The former motivation was also followed by Omnivore for their RGB-D embedding by dropping color 50% of the time, but our implementation differs decisively: Our dropout is implemented after the embedding computation, but before the fusion operation, as depicted in Fig. 2, whereas Omnivore dropped the color channels by setting the color input to zero before the embedding. During training time, we either drop color or depth by sampling probabilities p_c, p_d for dropping color or depth from a multinomial distribution defined by the probabilities $[p_c, p_d, 1-(p_c+p_d)]$.

4. Experiments

4.1. Implementation Details

We build upon the official Cube R-CNN implementation by Brazil *et al.* [4] based upon PyTorch3D [28] and Detectron2 [38]. The original Cube R-CNN was trained using 48 16GB V100 GPUs, however, since we have fewer hardware resources available, we first aimed at creating a reduced training schedule, while still achieving representative results. We trained all our models on 8 V100 32GB GPUs with a batch size of 8 images per GPU, thus reducing the total batch size of 128 to 64 images per batch. We linearly scale the standard Cube R-CNN learning rate, the learning rate schedule, and the number of iterations accordingly. For our experiments and ablations we only use 50% of the iterations while observing only a small performance degradation (see the first two lines of Tab. 1). Our model is optimized using stochastic gradient descent with a learning rate of 0.04 and a momentum of 0.9. During a warm-up phase, the learning rate is linearly increased from an initial value of 0.01% of the final learning rate for the first 3.125% of iterations. Subsequently, a step-scheduling is applied at 60% and 80% of iterations with a factor of 0.1. Additionally, we use weight decay with a coefficient of 0.0001. We augment the data by random horizontal flipping and scaling during training, as done by [4]. We experimented with other reduced learning rate schedules, as well as optimizers (*e.g.* AdamW [23]), however, we found this simple scaling to be the most effective. Using this setup, we can train an (Omni) Cube R-CNN within roughly $\sim$2 days while using significantly fewer GPUs. We rely on the Cube R-CNN implementation of the DLA34 model which is initialized with official ImageNet 1k weights. For both ViT-S [8] and Swin-T [22], we use the implementations of Detectron2. Unless specified otherwise, we use ImageNet 1k checkpoints provided by the timm library [37] for initializing encoders and color embeddings, whereas additional depth input layers, as well as Cube R-CNN specific layers are initialized from scratch. The latter applies also to our RGB-D version of DLA34. Notably, the timm checkpoints employed for ViT were trained utilizing the AugReg techniques as detailed in [34]. All RGB-D backbones use the same sum-embedding layer to produce fused tokens from RGB and depth modalities.

Swin In our experiment we use a Swin-T architecture with a patch size of 4, a window size of 7 and a path drop rate of 0.2. The outputs of stages $[0, 1, 2, 3]$ are used to form a hierarchical feature pyramid that is processed by a standard FPN [17].

Vision Transformer (ViT) We use a standard ViT-S architecture with a patch size of 16 and a path drop rate of 0.1. The input images are rescaled and padded to 512×512. The outputs of layers $[3, 5, 8, 11]$ are processed by a SimpleFeaturePyramid (SFP) [16] to form a feature hierarchy.

4.2. Evaluation Metrics

We evaluate using the standard 2D/3D average precision metrics as defined by Brazil *et al*. [4].

AP$_{3D}$ Our focus lies on the 3D average precision (AP$_{3D}$). We compute the intersection-over-union (IoU$_{3D}$) with full 9 degrees of freedom (3D position, 3D rotation, 3D object extents) using the the exact Fast IoU$_{3D}$ [4]. The mean AP is computed over all 38 classes of Omni3D$_{IN}$, and several different 3D IoU thresholds $\tau \in [0.05, 0.10, \dots, 0.50]$. Here we also ignore objects with high occlusion/truncation, or objects that appear tiny after projection to the image plane.

AP$_{2D}$ Although our main focus is on 3D detection, we additionally report 2D metrics. The AP is based on the IoU between the 2D box head predictions and the 2D bounding box of the 3D ground truth cube projected to the image plane. It is computed for IoU thresholds $\tau \in [0.50, 0.55, \dots, 0.95]$. Note the AP$_{2D}$ uses stricter thresholds since the overlap of 2D boxes in general is bigger than the overlapping volume of 3D cubes. Both the 2D box and the 3D cube prediction heads process RoIs independently, hence the 2D and 3D performance are not directly tied to each other and the AP$_{2D}$ should not be considered as a proxy for the 3D performance.

4.3. Backbones

In Tab. 1 we compare the effect of different backbones. The first and second line show that our reduced training setup obtains slightly lower scores than the original Cube R-CNN model by Brazil *et al*. [4], however, we require significantly fewer GPUs to train this model. We train all other models with the same reduced setup, performances might thus increase slightly with a longer schedule, however, in the following we will compare to our DLA34 baseline. Looking at the different backbones, Swin-T actually outperforms DLA34, both compared to our shorter schedule, as well as the original results. The ViT-S backbone on the other hand performs worse. We found the main reason for this to be more severe overfitting of the model, since the ViT backbone actually converged to a better train set performance. We also briefly experimented with a ViT-B model, but this only exacerbated the problem further. Now turning to models trained on RGB-D inputs, we can see some interesting effects on the right side of Tab. 1. All three backbone architectures see significant gains in both AP$_{2D}$ and AP$_{3D}$. DLA34 gets the biggest boost in AP$_{2D}$ (about 6%, getting the absolute best performance in this metric), but both Transformer-based methods get a significantly larger performance boost in AP$_{3D}$, where Swin-T gets a major boost of almost 11%. Further analyzing these differences, Fig. 3 shows that almost all classes benefit from the added depth modality, where some classes, such as *pillow* or *television*, were almost not detected based on the RGB input alone. Since the DLA34 backbone sees a significantly smaller AP$_{3D}$ improvement from the added depth modality, in the remainder of the paper we will focus on the two Transformer-based backbones.

	RGB		RGB-D	
Backbone	AP$_{2D}$	AP$_{3D}$	AP$_{2D}$	AP$_{3D}$
DLA34[4]	19.28	15.04	-	-
DLA34	18.28	14.24	**24.26**	17.22
Swin-T	**19.54**	**15.33**	22.25	**26.18**
ViT-S	13.90	12.45	16.19	21.34

Table 1. Results on the Omni3D$_{IN}$ datasets for Cube R-CNN trained with different backbones and with RGB or RGB-D inputs. The first line shows the original RGB Cube R-CNN results, with our reduced training schedule we achieve similar performance. All backbones benefit from the additional depth input.

Backbone	Fusion	Shared weights	AP$_{2D}$	AP$_{3D}$
Swin-T	Early	✓	**22.25**	**26.18**
Swin-T	Late	✓	19.95	24.25
Swin-T*	Late	✗	20.93	25.01
ViT-S	Early	✓	**16.19**	**21.34**
ViT-S	Late	✓	14.04	19.61
ViT-S	Late	✗	14.34	20.15

Table 2. Results on Omni3D$_{IN}$ RGB-D for different sum-fusion approaches. The Swin-T model marked with * was trained with a smaller learning rate as it would diverge otherwise.

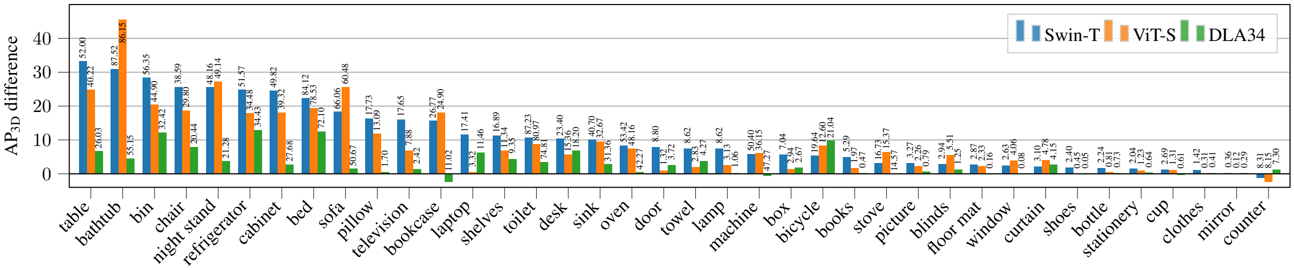


Figure 3. AP$_{3D}$ *differences* per category between models trained with RGB-D and RGB respectively. Both Swin and ViT see large improvements for certain categories, in most cases significantly bigger than the DLA34 backbone. In almost no cases the performance drops based on the added depth input. The numbers at the top of the bars indicate the resulting performance of the RGB-D model.

4.4. Modality Fusion Strategies

Inspired by the Omnivore [10] approach, we use an early input fusion by adding the RGB and depth embeddings in the first layer. While such an early fusion worked suboptimally for CNNs, it seems to empirically work well for Transformer-based approaches. However, in principle we can also pass these embeddings through separate networks and add their outputs, resulting in a late fusion scheme. In Tab. 2 we compare these two settings, where for the late fusion scheme we can additionally compare what happens when sharing the weights between the two separate networks. Here it should be noted that the first layer creating the embeddings is always specific to the RGB/depth input. While these experiments are by no means exhaustive, in our case the early fusion approach consistently performs best, although with additional tuning or other setups, it is likely that a late fusion approach could also perform well. The early fusion has the clear advantage that the compute increase is negligible though, whereas late fusion incurs a major runtime and memory increase, plus the additional weights when using individual encoders. For ViT additional settings could be interesting, where further tokens are simply computed for each modality separately and concatenated into a bigger set of tokens, but we decided not to pursue this here due to the quadratic increase in complexity.

4.5. Modality Dropout

When building upon a strong RGB pretrained network and simply adding an additional depth input, there is no real guarantee that the network learns to incorporate the new input in a meaningful way. To encourage it to learn from the depth, we examine different modality dropout strategies. Some approaches, such as [7, 10], suggest dropping the color modality before embedding by setting the color channels to zero, which we refer to as *pre-embedding* dropout. Omnivore [10] embeds RGB and RGB-D tokens separately and encodes them with the same encoder, which can be interpreted as dropping depth after embedding, resulting in similar behavior to our approach. Our dropout only takes place after the embedding and before the fusion, therefore we refer to it as *post-embedding* dropout.

Dropout Implementations We have ablated the two different dropout approaches, which are depicted in Fig. 2, with our Transformer backbones and listed the results in Tab. 4. The results highlight that for both backbones our *post-embedding* strategy performs best. While the difference in performance for ViT is not significant, a substantial difference is observed for Swin. We further analyzed our approach for different dropout settings, which are summarized in Tab. 3. For Swin the improvements based on different dropout settings is not that large and often actually decreases the scores, however, small gains can be obtained. Especially dropping the color and thus forcing the model to learn from the depth channel is somewhat effective to boost the scores. For ViT the possible improvements are actually a bit better and in general dropping more seems to help quite a bit. A possible reason could be that this prevents the model from overfitting too strongly which seems to be a problem in general with the ViT architecture. While the optimal dropout setting is different for the two backbones, it seems to be important that the networks are trained with some samples that contain both the color as well as the depth channels since the models trained only of either color or depth images perform significantly worse when evaluated on the RGB-D input.

Test-time Dropout Tab. 5 and Tab. 6 show the performance of our RGB-D Cube R-CNN when evaluated only on either RGB or depth. Depending on the modality dropout settings used during training, when only providing color images, most models gracefully degrade to performances similar to a model trained only on RGB (see Tab. 1). One of the settings actually outperforms the model trained only on RGB, indicating that using depth as privileged information during training can still be valuable to obtain a stronger model, similar to for example the Mask3D approach [14]. The performances when only evaluating on depth are much higher though. While one might suspect that it is an easier task to predict accurate 3D bound boxes from a depth im-

Training Modality			Swin-T		ViT-S	
RGB	D	RGB-D	AP_{2D}	AP_{3D}	AP_{2D}	AP_{3D}
0%	0%	100%	22.25	26.18	16.19	21.34
25%	0%	75%	22.34	23.35	16.80	20.38
0%	25%	75%	✗	✗	16.39	22.01
50%	0%	50%	21.35	20.11	14.98	16.10
0%	50%	50%	22.77	**27.08**	17.03	23.08
25%	25%	50%	**23.71**	25.73	**18.68**	**23.97**
50%	25%	25%	23.01	22.57	17.17	19.24
50%	50%	0%	17.48	12.47	9.98	2.53
25%	50%	25%	22.86	25.52	18.34	23.96

Table 3. Results of different modality dropout probabilities for Swin-T and ViT-S on Omni3D$_{IN}$ RGB-D. The first three columns represent the percentage of training samples with a specific modality. While not all settings help, improvements are possible for both architectures. One Swin-T setup diverged (marked with ✗).

	Swin-T		ViT-S	
Dropout	AP_{2D}	AP_{3D}	AP_{2D}	AP_{3D}
Post-embedding	**22.77**	**27.08**	**17.03**	**23.08**
Pre-embedding	19.56	23.19	16.66	23.01

Table 4. Results on Omni3D$_{IN}$ RGB-D for different modality dropout approaches as illustrated in Fig. 2. For this ablation, only the color modality is dropped for 50% of the samples, since both methods drop the depth in the same way.

age, it is somewhat surprising that a model initialized with a strong RGB-based ImageNet pretraining can cope so well when the color information is no longer present.

4.6. ViT Initializations

In our experiments the ViT backbone suffered from overfitting the most. However, since Vision Transformers are interesting for all kinds of reasons, *e.g*. their very nice handling of additional inputs and their general widespread use for all kinds of applications, we investigated these in more detail. While modality dropout did indeed boost performance quite a bit, there is still a fairly large gap to the Swin Transformer performance. We also experimented with the window attention approach proposed by Li *et al*. [16] with little effect. Orthogonal to this, we also looked into using self-supervised learning on the same RGB-D data, as an additional pretraining step. For this, we considered DINO [5] and Barlow Twins [40]. Tab. 7 shows that pretraining with Barlow Twins gives a similar boost as modality dropout. Further adding modality dropout *during pretraining* additionally boosts the AP_{2D} score, and combining these two setups yields even better results, slowly approaching the Swin-T performance. DINO pretraining using the same setup on the other hand made performance worse, which we hypothesize is due to stability issues, likely caused by not being able to train with smaller batch sizes. As Barlow Twins does not require large batch sizes, we did not encounter similar issues here. We tried similar setups for the Swin Transformer, but here in all cases the plain pretrained ImageNet initialization performed best. In general this shows that some form of additional self-supervised learning is an interesting venue for improvements and other methods such as MAE [13] or data2vec [1, 2] could be interesting alternatives here.

Training Modality			Swin-T		ViT-S	
RGB	D	RGB-D	AP_{2D}	AP_{3D}	AP_{2D}	AP_{3D}
25%	0%	75%	19.07	14.24	13.93	12.24
50%	0%	50%	19.35	15.38	13.59	12.02
25%	25%	50%	18.53	13.85	14.18	12.55
50%	25%	25%	**20.05**	**15.49**	**14.58**	**12.83**
25%	50%	25%	17.99	13.15	13.64	12.05
50%	50%	0%	18.81	14.33	14.48	12.79

Table 5. Results of different modality dropout settings when evaluated on RGB inputs. We omit settings without RGB only training samples. While performances drop significantly, the models do not completely fail and for Swin-T some settings result in better performance than when only training on RGB.

Training Modality			Swin-T		ViT-S	
RGB	D	RGB-D	AP_{2D}	AP_{3D}	AP_{2D}	AP_{3D}
0%	25%	75%	✗	✗	14.08	19.71
0%	50%	50%	**21.44**	**25.65**	15.26	21.09
25%	25%	50%	20.64	23.21	15.96	21.75
50%	25%	25%	20.65	21.50	15.58	20.39
25%	50%	25%	21.19	24.23	16.07	**22.00**
50%	50%	0%	20.40	21.99	**16.04**	21.27

Table 6. Results of different modality dropout settings when evaluated on depth inputs. We omit settings without depth only training samples. Performances drop significantly less than in the RGB-only case indicating all models learned to utilize depth even though they were initialized with ImageNet weights (✗: diverged).

4.7. Qualitative Results

Fig. 4 shows qualitative results from a Cube R-CNN and an RGB-D Cube R-CNN, both using a Swin-T backbone. In general our RGB-D Cube R-CNN manages to detect additional objects, but especially predicts boxes with better alignments in 3D space. Additional qualitative results, including side-by-side backbone comparisons can be found in the supplementary material (Sec. 6).

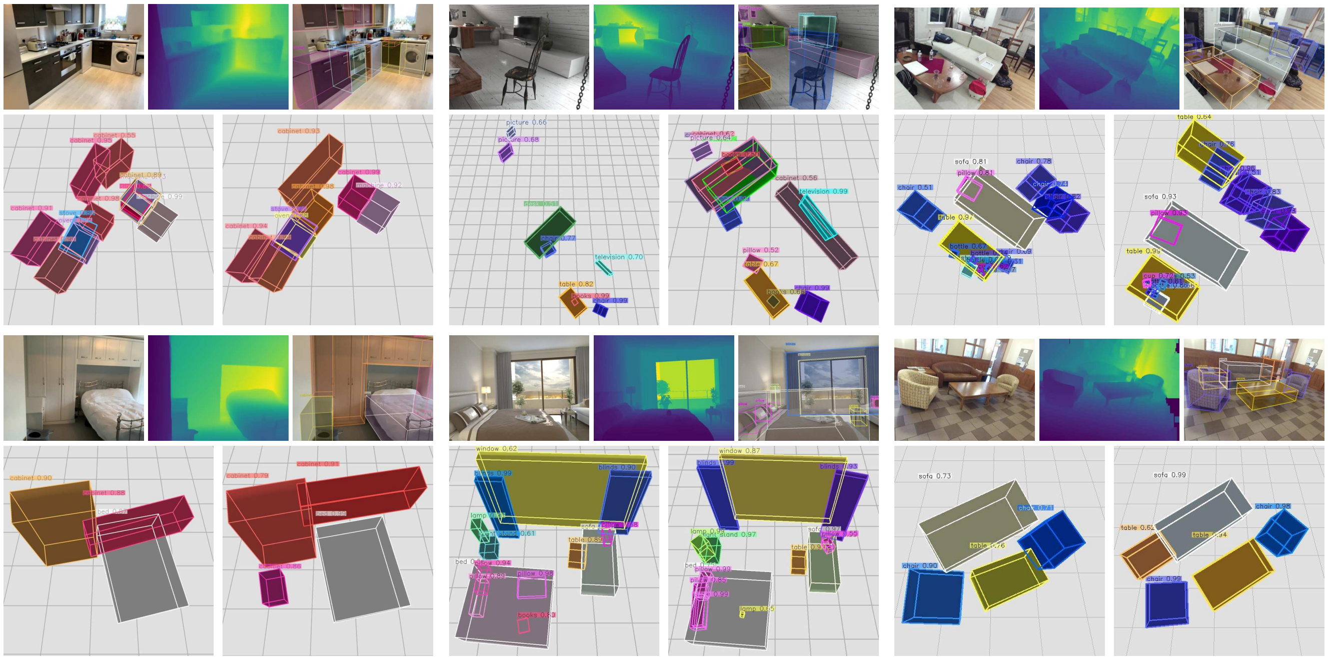


Figure 4. Qualitative results for our final Swin-T Cube R-CNN on samples from Omni3D$_{\text{IN}}$. Each set of five images shows the RGB image, depth image, the ground truth and the top view from predictions trained on RGB only (left) and RGB-D (right).

Init	SSL Pretraining	Modality	AP$_{2D}$	AP$_{3D}$
IN1k		RGB-D	16.19	21.34
IN1k		Drop	18.68	23.97
IN1k	Barlow Twins	RGB-D	18.35	<u>24.05</u>
IN1k	Barlow Twins	Drop	<u>19.43</u>	23.96
IN1k	Barlow Twins + Drop	Drop	**20.42**	**24.95**
IN1k	DINO + Drop	Drop	17.20	22.33

Table 7. Results of different initializations and self-supervised learning strategies for ViT-S trained on RGB-D. The modality column indicates if modality dropout was used during the final Cube R-CNN training. Some setups can boost the scores significantly.

5. Conclusion

We extended the Omni3D$_{\text{IN}}$ dataset by adding depth to form a new multi-modal RGB-D dataset. We integrated both RGB and depth modalities into Cube R-CNN, while exploring recent Transformer-based models, *i.e.* Swin and ViT, as alternatives to the DLA model. Our findings showcase that Swin matches the performance of DLA in the RGB domain, whereas ViT faces challenges with overfitting. However, by incorporating depth into the models based on an early fusion method, both Swin and ViT models surpass DLA in terms of 3D precision, with Swin exhibiting particularly strong results. In our experiments early fusion outperformed both late fusion approaches, while at the same time saving a significant amount of compute due to not having two separate backbones. We introduced a post-embedding modality dropout approach, which aims to avoid disregarding a modality during training and to allow selective modality dropout during inference. Furthermore, it acts as a regularization for ViTs. With our extensive ablations, we have shown that modality dropout can improve performance on RGB-D, while achieving similar or slightly better results when training on both modalities but utilizing only one modality during inference. This robustness to missing modalities during inference renders it especially interesting for indoor robotics applications where sensors might fail or be absent. While only inferring on RGB images most models gracefully revert to the performance of models trained on RGB only, whereas only using depth during inference performs significantly better, indicating that the pretrained models can deal well with the encountered modality shift. To further improve the performance of ViTs, we explored different initialization strategies motivated by recent advancements in self-supervised learning allowing us to partially bridge the initial gap between ViT- and Swin-based training. Altogether, our proposed approach presents a viable RGB-D based 3D object detector that achieves significantly improved quantitatively and qualitatively results compared to the RGB-only baseline.

Acknowledgments. This work was funded by Bosch Research as part of a Bosch-RWTH collaboration on “Context Understanding for Autonomous Systems”

References

[1] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. In *International Conference on Machine Learning*, 2022. 7

[2] Alexei Baevski, Arun Babu, Wei-Ning Hsu, and Michael Auli. Efficient self-supervised learning with contextualized target representations for vision, speech and language. In *International Conference on Machine Learning*, 2023. 7

[3] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. ARKitScenes: A Diverse Real-World Dataset For 3D Indoor Scene Understanding Using Mobile RGB-D Data. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. 2, 3

[4] Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. Omni3D: A Large Benchmark and Model for 3D Object Detection in the Wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 1, 2, 3, 4, 5

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *International Conference on Computer Vision*, 2021. 7

[6] Yongjian Chen, Lei Tai, Kai Sun, and Mingyang Li. Monopair: Monocular 3d object detection using pairwise spatial relationships. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 12093–12102, 2020. 4

[7] Sébastien de Blois, Mathieu Garon, Christian Gagné, and Jean-François Lalonde. Input dropout for spatially aligned modalities. In *International Conference on Image Processing*, 2020. 2, 6

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 1, 4

[9] Guofan Fan, Zekun Qi, Wenkai Shi, and Kaisheng Ma. Point-gcc : Universal self-supervised 3d scene pre-training via geometry-color contrast. *arXiv preprint arXiv:2305.19623*, 2023. 2

[10] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A Single Model for Many Visual Modalities. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 2, 4, 6

[11] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *European Conference on Computer Vision*, 2014. 2

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3, 4

[13] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 7

[14] Ji Hou, Xiaoliang Dai, Zijian He, Angela Dai, and Matthias Nießner. Mask3D: Pre-training 2D Vision Transformers by Learning Masked 3D Priors. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 6

[15] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General Perception with Iterative Attention. In *International Conference on Machine Learning*, 2021. 2, 4

[16] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring Plain Vision Transformer Backbones for Object Detection. In *European Conference on Computer Vision*, 2022. 4, 5, 7

[17] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3, 5

[18] Yunzhi Lin, Jonathan Tremblay, Stephen Tyree, Patricio A. Vela, and Stan Birchfield. Single-stage keypoint-based category-level object pose estimation from an RGB image. In *IEEE International Conference on Robotics and Automation*, 2022. 2, 4

[19] Timm Linder, Kilian Y Pfeiffer, Narunas Vaskevicius, Robert Schirmer, and Kai O Arras. Accurate detection and 3D localization of humans using a novel YOLO-based RGB-D fusion approach and synthetic training data. In *IEEE International Conference on Robotics and Automation*, 2020. 2

[20] Yunze Liu, Li Yi, Shanghang Zhang, Qingnan Fan, Thomas Funkhouser, and Hao Dong. P4Contrast: Contrastive Learning with Pairs of Point-Pixel Pairs for RGB-D Scene Understanding. *arXiv preprint arXiv:2012.13089*, 2020. 2

[21] Zechen Liu, Zizhang Wu, and Roland Tóth. Smoke: Single-stage monocular 3d object detection via keypoint estimation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 996–997, 2020. 4

[22] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *International Conference on Computer Vision*, 2021. 1, 2, 4

[23] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019. 4

[24] Johannes Meyer, Andreas Eitel, Thomas Brox, and Wolfram Burgard. Improving Unimodal Object Recognition with Multimodal Contrastive Learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020. 2

[25] Ishan Misra, Rohit Girdhar, and Armand Joulin. An End-to-End Transformer Model for 3D Object Detection. In *International Conference on Computer Vision*, 2021. 1, 2

[26] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *International Conference on Computer Vision*, 2019. 1

[27] Charles R Qi, Xinlei Chen, Or Litany, and Leonidas J Guibas. Imvotenet: Boosting 3d object detection in point clouds with image votes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2

[28] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3D Deep Learning with PyTorch3D. *arXiv:2007.08501*, 2020. 4

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Advances in Neural Information Processing Systems*, 2015. 2, 3

[30] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A Photorealistic Synthetic Dataset for Holistic Indoor Scene Understanding. In *International Conference on Computer Vision*, 2021. 2, 3

[31] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection. In *IEEE Winter Conference on Applications of Computer Vision*, 2022. 1, 2

[32] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Tr3d: Towards real-time indoor 3d object detection. In *International Conference on Image Processing*, pages 281–285. IEEE, 2023. 1, 2

[33] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 2, 3

[34] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021. 5

[35] Giorgos Tziafas and Hamidreza Kasaei. Early or Late Fusion Matters: Efficient RGB-D Fusion in Vision Transformers for 3D Object Recognition. *arXiv preprint arXiv:2210.00843*, 2023. 2

[36] Haiyang Wang, Shaocong Dong, Shaoshuai Shi, Aoxue Li, Jianan Li, Zhenguo Li, Liwei Wang, et al. Cagroup3d: Class-aware grouping for 3d object detection on point clouds. *Advances in Neural Information Processing Systems*, 35: 29975–29988, 2022. 2

[37] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 4

[38] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019. 4

[39] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2403–2412, 2018. 1, 4

[40] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow Twins: Self-Supervised Learning via Redundancy Reduction. In *International Conference on Machine Learning*, 2021. 7

[41] Jiaming Zhang, Huayao Liu, Kailun Yang, Xinxin Hu, Ruiping Liu, and Rainer Stiefelhagen. CMX: Cross-modal fusion for RGB-X semantic segmentation with transformers. *IEEE Transactions on Intelligent Transportation Systems*, 2023. 2

[42] Yunpeng Zhang, Jiwen Lu, and Jie Zhou. Objects are different: Flexible monocular 3d object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3289–3298, 2021. 4

[43] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 4