DFormer: Rethinking RGBD Representation Learning for Semantic Segmentation

Bowen Yin[1] **Xuying Zhang**[1] **Zhongyu Li**[1] **Li Liu**[2] **Ming-Ming Cheng**[1,3] **Qibin Hou**[1,3*]

[1] VCIP, School of Computer Science, Nankai University
[2] National University of Defense Technology
[3] Nankai International Advanced Research Institute (Shenzhen Futian)
bowenyin@mail.nankai.edu.cn, andrewhoux@gmail.com

Abstract

We present DFormer, a novel RGB-D pretraining framework to learn transferable representations for RGB-D segmentation tasks. DFormer has two new key innovations: 1) Unlike previous works that encode RGB-D information with RGB pretrained backbone, we pretrain the backbone using image-depth pairs from ImageNet-1K, and hence the DFormer is endowed with the capacity to encode RGB-D representations; 2) DFormer comprises a sequence of RGB-D blocks, which are tailored for encoding both RGB and depth information through a novel building block design. DFormer avoids the mismatched encoding of the 3D geometry relationships in depth maps by RGB pretrained backbones, which widely lies in existing methods but has not been resolved. We finetune the pretrained DFormer on two popular RGB-D tasks, *i.e.,* RGB-D semantic segmentation and RGB-D salient object detection, with a lightweight decoder head. Experimental results show that our DFormer achieves new state-of-the-art performance on these two tasks with less than half of the computational cost of the current best methods on two RGB-D semantic segmentation datasets and five RGB-D salient object detection datasets. Our code is available at: https://github.com/VCIP-RGBD/DFormer.

1 Introduction

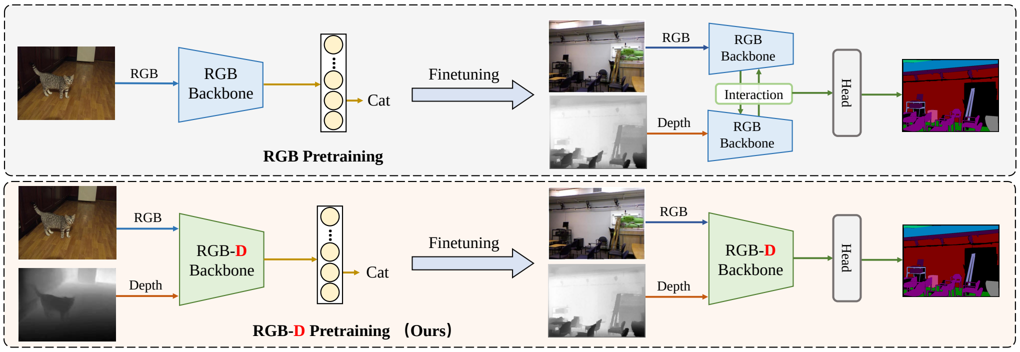


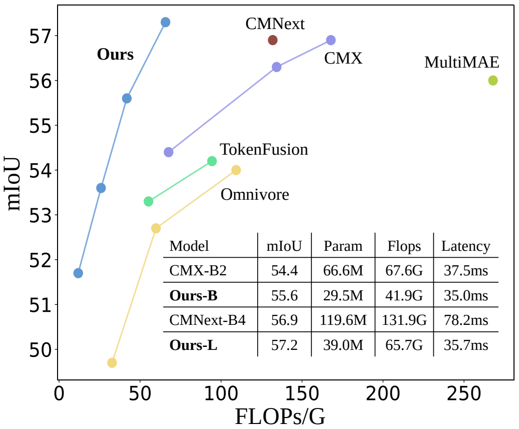
Figure 1: Comparisons between the existing popular training pipeline and ours for RGB-D segmentation. **RGB pretraining**: Recent mainstream methods adopt two RGB pretrained backbones to separately encode RGB and depth information and fuse them at each stage. **RGB-D pretraining**: The RGB-D backbone in DFormer learns transferable RGB-D representations during pretraining and then is finetuned for segmentation.

With the widespread use of 3D sensors, RGB-D data is becoming increasingly available to access. By incorporating 3D geometric information, it would be easier to distinguish instances and context, facilitating the RGB-D research for high-level scene understanding. Meanwhile, RGB-D data also presents considerable potential in a large number of applications, *e.g.*, SLAM (Wang et al., 2023), automatic driving (Huang et al., 2022), and robotics (Marchal et al., 2020). Therefore, RGB-D research has attracted great attention over the past few years.

[*] Qibin Hou is the corresponding author.

Fig. 1(top) shows the pipeline of current mainstream RGB-D methods. As can be observed, the features of the RGB images and depth maps are respectively extracted from two individual RGB pretrained backbones. The interactions between the information of these two modalities are performed during this process. Although the existing methods (Wang et al., 2022; Zhang et al., 2023b) have achieved excellent performance on several benchmark datasets, there are three issues that cannot be ignored: i) The backbones in the RGB-D tasks take an image-depth pair as input, which is inconsistent with the input of an image in RGB pretraining, causing a huge representation distribution shift; ii) The interactions are densely performed between the RGB branch and depth branch during finetuning, which may destroy the representation distribution within the pretrained RGB backbones; iii) The dual backbones in RGB-D networks bring more computational cost compared to standard RGB methods, which is not efficient. We argue that an important reason leading to these issues is the pretraining manner. The depth information is not considered during pretraining.

Taking the above issues into account, a straightforward question arises: Is it possible to specifically design an RGB-D pretraining framework to eliminate this gap? This motivates us to present a novel RGB-D pretraining framework, termed DFormer, as shown in Fig. 1(bottom). During pretraining, we consider taking image-depth pairs[1], not just RGB images, as input and propose to build interactions between RGB and depth features within the building blocks of the encoder. Therefore, the inconsistency between the inputs of pretraining and finetuning can be naturally avoided. In addition, during pretraining, the RGB and depth features can efficiently interact with each other in each building block, avoiding the heavy interaction modules outside the backbones, which is mostly adopted in current dominant methods. Furthermore, we also observe that the depth information only needs a small portion of channels to encode. There is no need to use a whole RGB pretrained backbone to extract depth features as done in previous works. As the interaction starts from the pretraining stage, the interaction efficiency can be largely improved compared to previous works as shown in Fig. 2.



Model	mIoU	Param	Flops	Latency
CMX-B2	54.4	66.6M	67.6G	37.5ms
Ours-B	55.6	29.5M	41.9G	35.0ms
CMNext-B4	56.9	119.6M	131.9G	78.2ms
Ours-L	57.2	39.0M	65.7G	35.7ms

Figure 2: Performance *vs.* computational cost on the NYUDepthv2 dataset (Silberman et al., 2012). DFormer achieves the state-of-the-art 57.2% mIoU and the best trade-off compared to other methods.

We demonstrate the effectiveness of DFormer on two popular RGB-D downstream tasks, *i.e.*, semantic segmentation and salient object detection. By adding a lightweight decoder on top of our pretrained RGB-D backbone, DFormer sets new state-of-the-art records with less computational cost compared to previous methods. Remarkably, our largest model, DFormer-L, achieves a new state-of-the-art result, *i.e.,* 57.2% mIoU on NYU Depthv2 with less than half of the computations of the second-best method CMNext (Zhang et al., 2023b). Meanwhile, our lightweight model DFormer-T is able to achieve 51.8% mIoU on NYU Depthv2 with only 6.0M parameters and 11.8G Flops. Compared to other recent models, our approach achieves the best trade-off between segmentation performance and computations.

To sum up, our main contributions can be summarized as follows:

- We present a novel RGB-D pretraining framework, termed DFormer, with a new interaction method to fuse the RGB and depth features to provide transferable representations for RGB-D downstream tasks.
- We find that in our framework it is enough to use a small portion of channels to encode the depth information compared to RGB features, an effective way to reduce model size.
- Our DFormer achieves new state-of-the-art performance with less than half of computational cost of the current best methods on two RGB-D segmentation datasets and five RGB-D salient object detection datasets.

[1] For depth maps, we employ a widely used depth estimation model (Bhat et al., 2021) to predict depth map for each RGB image,which we found works well.

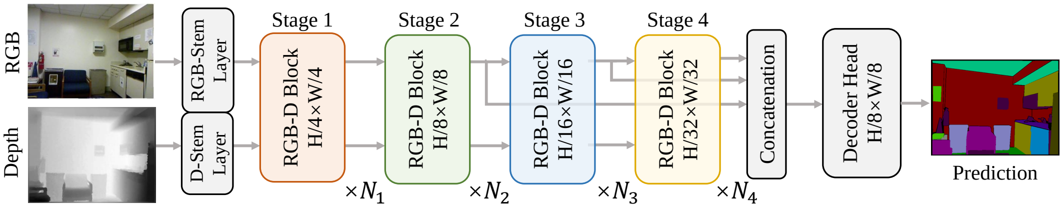


Figure 3: Overall architecture of the proposed DFormer. First, we use the pretrained DFormer to encode the RGB-D data. Then, the features from the last three stages are concatenated and delivered to a lightweight decoder head for final prediction. Note that only the RGB features from the encoder are used in the decoder.

2 PROPOSED DFORMER

Fig. 3 illustrates the overall architecture of our DFormer, which follows the popular encoder-decoder framework. Given an RGB image and the corresponding depth map with spatial size $H \times W$, they are first separately processed by two parallel stem layers consisting of two convolutions with kernel size 3×3 and stride 2. Then, the RGB features and depth features are fed into the hierarchical encoder to encode multi-scale features at $\{1/4, 1/8, 1/16, 1/32\}$ of the original image resolution. Next, we pretrain this encoder using the image-depth pairs from ImageNet-1K using the classification objective to generate the transferable RGB-D representations. Finally, we send the visual features from the pretrained RGB-D encoder to the decoder to produce predictions, *e.g.*, segmentation maps with a spatial size of $H \times W$. In the rest of this section, we will describe the encoder, RGB-D pretraining framework, and task-specific decoder in detail.

2.1 HIERARCHICAL ENCODER

As shown in Fig. 3, our hierarchical encoder is composed of four stages, which are utilized to generate multi-scale RGB-D features. Each stage contains a stack of RGB-D blocks. Two convolutions with kernel size 3×3 and stride 2 are used to down-sample RGB and depth features, respectively, between two consecutive stages.

Building Block. Our building block is mainly composed of the global awareness attention (GAA) module and the local enhancement attention (LEA) module and builds interaction between the RGB and depth modalities. GAA incorporates depth information and aims to enhance the capability of object localization from a global perspective, while LEA adopts a large-kernel convolution to capture the local clues from the depth features, which can refine the details of the RGB representations. The details of the interaction modules are shown in Fig. 4.

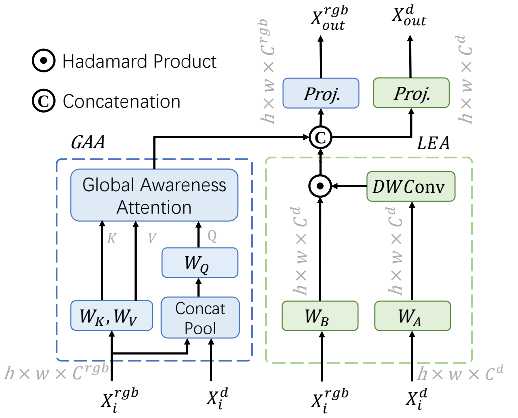

⊙ Hadamard Product
Ⓒ Concatenation
GAA
LEA
Global Awareness Attention
K
V
Q
W_Q
W_K, W_V
Concat Pool
W_B
W_A
DWConv
Proj.
Proj.
X_{out}^{rgb}
X_{out}^{d}
$h \times w \times C^{rgb}$
$h \times w \times C^{d}$
$h \times w \times C^{d}$
$h \times w \times C^{d}$
$h \times w \times C^{rgb}$
$h \times w \times C^{d}$
X_i^{rgb}
X_i^{d}
X_i^{rgb}
X_i^{d}

Figure 4: Diagrammatic details on how to conduct interactions between RGB and depth features.

Our GAA fuses depth and RGB features to build relationships across the whole scene, enhancing 3D awareness and further helping capture semantic objects. Different from the self-attention mechanism (Vaswani et al., 2017) that introduces quadratic computation growth as the pixels or tokens increase, the Query (Q) in GAA is down-sampled to a fixed size and hence the computational complexity can be reduced. Tab. 7 illustrates that fusing depth features with Q is adequate and there is no need to combine them with K or V, which brings computation increment but no performance improvement. So, Q comes from the concatenation of the RGB features and depth features, while key (K) and value (V) are extracted from RGB features. Given the RGB features X_i^{rgb} and depth features X_i^{d}, the above process can be formulated as:

$$Q = \text{Linear}(\text{Pool}_{k \times k}([X_i^{rgb}, X_i^{d}])), K = \text{Linear}(X_i^{rgb}), V = \text{Linear}(X_i^{rgb}), \tag{1}$$

where $[\cdot, \cdot]$ denotes the concatenation operation along the channel dimension, $\text{Pool}_{k \times k}(\cdot)$ performs adaptively average pooling operation across the spatial dimensions to $k \times k$ size, and Linear is linear transformation. Based on the generated $Q \in \mathbb{R}^{k \times k \times C^d}$, $K \in \mathbb{R}^{h \times w \times C^d}$, and $V \in \mathbb{R}^{h \times w \times C^d}$, where h and w are the height and width of features in the current stage, we formulate the GAA as follows:

$$X_{GAA} = \mathrm{UP}(V \cdot \mathrm{Softmax}(\frac{Q^\top K}{\sqrt{C^d}})), \tag{2}$$

where $\mathrm{UP}(\cdot)$ is a bilinear upsampling operation that converts the spatial size from $k \times k$ to $h \times w$. In practical use, Eqn. 2 can also be extended to a multi-head version, as done in the original self-attention (Vaswani et al., 2017), to augment the feature representations.

We also design the LEA module to capture more local details, which can be regarded as a supplement to the GAA module. Unlike most previous works that use addition and concatenation to fuse the RGB features and depth features. We conduct a depth-wise convolution with a large kernel on the depth features and use the resulting features as attention weights to reweigh the RGB features via a simple Hadamard product inspired by (Hou et al., 2022). This is reasonable in that adjacent pixels with similar depth values often belong to the same object and the 3D geometry information thereby can be easily embedded into the RGB features. To be specific, the calculation process of LEA can be defined as follows:

$$X_{LEA} = \mathrm{DConv}_{k \times k}(\mathrm{Linear}(X_i^d)) \odot \mathrm{Linear}(X_i^{rgb}), \tag{3}$$

where $\mathrm{DConv}_{k \times k}$ is a depth-wise convolution with kernel size $k \times k$ and $\odot$ is the Hadamard product.

To preserve the diverse appearance information, we also build a base module to transform the RGB features X_i^{rgb} to X_{Base}, which has the same spatial size as X_{GAA} and X_{LEA}. The calculation process of X_{Base} can be defined as follows:

$$X_{Base} = \mathrm{DConv}_{k \times k}(\mathrm{Linear}(X_i^{rgb})) \odot \mathrm{Linear}(X_i^{rgb}). \tag{4}$$

Finally, the features, *i.e.*, $X_{GAA} \in \mathbb{R}^{h_i \times w_i \times C_i^d}$, $X_{LEA} \in \mathbb{R}^{h_i \times w_i \times C_i^d}$, $X_{Base} \in \mathbb{R}^{h_i \times w_i \times C_i^{rgb}}$, are fused together by concatenation and linear projection to update the RGB features X_{out}^{rgb} and depth features X_{out}^d.

Overall Architecture. We empirically observe that encoding depth features requires fewer parameters compared to the RGB ones due to their less semantic information, which is verified in Fig. 6 and illustrated in detail in the experimental part. To reduce model complexity in our RGB-D block, we use a small portion of channels to encode the depth information. Based on the configurations of the RGB-D blocks in each stage, we design a series of DFormer encoder variants, termed DFormer-T, DFormer-S, DFormer-B, and DFormer-L, respectively, with the same architecture but different model sizes. DFormer-T is a lightweight encoder for fast inference, while DFormer-L is the largest one for attaining better performance. For detailed configurations, readers can refer to Tab. 16.

2.2 RGB-D PRETRAINING

The purpose of RGB-D pretraining is to endow the backbone with the ability to achieve the interaction between RGB and depth modalities and generate transferable representations with rich semantic and spatial information. To this end, we first apply a depth estimator, *e.g.*, Adabin (Bhat et al., 2021), on the ImageNet-1K dataset (Russakovsky et al., 2015) to generate a large number of image-depth pairs. Then, we add a classification head on the top of the RGB-D encoder to build the classification network for pretraining. Particularly, the RGB features from the last stage are flattened along the spatial dimension and fed into the classifier head. The standard cross-entropy loss is employed as our optimization objective, and the network is pretrained on RGB-D data for 300 epochs, like ConvNeXt (Liu et al., 2022). Following previous works (Liu et al., 2022; Guo et al., 2022b), the AdamW optimizer (Loshchilov & Hutter, 2019) with learning rate 1e-3 and weight decay 5e-2 is employed and the batch size is set to 1024. More specific settings for each variant of DFormer are described in the appendix.

2.3 TASK-SPECIFIC DECODER.

For the applications of our DFormer to downstream tasks, we just add a lightweight decoder on the top of the pretrained RGB-D backbone to build the task-specific network. After being finetuned on

Table 1: Results on NYU Depth V2 (Silberman et al., 2012) and SUN-RGBD (Song et al., 2015). Some methods do not report the results or settings on the SUN-RGBD datasets, so we reproduce them with the same training config. † indicates our implemented results. All the backbones are pre-trained on ImageNet-1K.

Model	**Backbone**	**Params**	NYUDepthv2			SUN-RGBD			**Code**
			Input size	**Flops**	**mIoU**	**Input size**	**Flops**	**mIoU**	
ACNet$_{19}$ (Hu et al.)	ResNet-50	116.6M	480 × 640	126.7G	48.3	530 × 730	163.9G	48.1	Link
SGNet$_{20}$ (Chen et al.)	ResNet-101	64.7M	480 × 640	108.5G	51.1	530 × 730	151.5G	48.6	Link
SA-Gate$_{20}$ (Chen et al.)	ResNet-101	110.9M	480 × 640	193.7G	52.4	530 × 730	250.1G	49.4	Link
CEN$_{20}$ (Wang et al.)	ResNet-101	118.2M	480 × 640	618.7G	51.7	530 × 730	790.3G	50.2	Link
CEN$_{20}$ (Wang et al.)	ResNet-152	133.9M	480 × 640	664.4G	52.5	530 × 730	849.7G	51.1	Link
ShapeConv$_{21}$ (Cao et al.)	ResNext-101	86.8M	480 × 640	124.6G	51.3	530 × 730	161.8G	48.6	Link
ESANet$_{21}$ (Seichter et al.)	ResNet-34	31.2M	480 × 640	34.9G	50.3	480 × 640	34.9G	48.2	Link
FRNet$_{22}$ (Zhou et al.)	ResNet-34	85.5M	480 × 640	115.6G	53.6	530 × 730	150.0G	51.8	Link
PGDENet$_{22}$ (Zhou et al.)	ResNet-34	100.7M	480 × 640	178.8G	53.7	530 × 730	229.1G	51.0	Link
EMSANet$_{22}$ (Seichter et al.)	ResNet-34	46.9M	480 × 640	45.4G	51.0	530 × 730	58.6G	48.4	Link
TokenFusion$_{22}$ (Wang et al.)	MiT-B2	26.0M	480 × 640	55.2G	53.3	530 × 730	71.1G	50.3†	Link
TokenFusion$_{22}$ (Wang et al.)	MiT-B3	45.9M	480 × 640	94.4G	54.2	530 × 730	122.1G	51.0†	Link
MultiMAE$_{22}$ (Bachmann et al.)	ViT-B	95.2M	640 × 640	267.9G	56.0	640 × 640	267.9G	51.1†	Link
Omnivore$_{22}$ (Girdhar et al.)	Swin-T	29.1M	480 × 640	32.7G	49.7	530 × 730	—	—	Link
Omnivore$_{22}$ (Girdhar et al.)	Swin-S	51.3M	480 × 640	59.8G	52.7	530 × 730	—	—	Link
Omnivore$_{22}$ (Girdhar et al.)	Swin-B	95.7M	480 × 640	109.3G	54.0	530 × 730	—	—	Link
CMX$_{22}$ (Zhang et al.)	MiT-B2	66.6M	480 × 640	67.6G	54.4	530 × 730	86.3G	49.7	Link
CMX$_{22}$ (Zhang et al.)	MiT-B4	139.9M	480 × 640	134.3G	56.3	530 × 730	173.8G	52.1	Link
CMX$_{22}$ (Zhang et al.)	MiT-B5	181.1M	480 × 640	167.8G	56.9	530 × 730	217.6G	52.4	Link
CMNext$_{23}$ (Zhang et al.)	MiT-B4	119.6M	480 × 640	131.9G	56.9	530 × 730	170.3G	51.9†	Link
DFormer-T	Ours-T	6.0M	480 × 640	11.8G	51.8	530 × 730	15.1G	48.8	Link
DFormer-S	Ours-S	18.7M	480 × 640	25.6G	53.6	530 × 730	33.0G	50.0	Link
DFormer-B	Ours-B	29.5M	480 × 640	41.9G	55.6	530 × 730	54.1G	51.2	Link
DFormer-L	Ours-L	39.0M	480 × 640	65.7G	**57.2**	530 × 730	83.3G	**52.5**	Link

corresponding benchmark datasets, the task-specific network is able to generate great predictions, without using extra designs like fusion modules (Chen et al., 2020a; Zhang et al., 2023a).

Take RGB-D semantic segmentation as an example. Following SegNeXt (Guo et al., 2022a), we adopt a lightweight Hamburger head (Geng et al., 2021) to aggregate the multi-scale RGB features from the last three stages of our pretrained encoder. Note that, our decoder only uses the X^{rgb} features, while other methods (Zhang et al., 2023a; Wang et al., 2022; Zhang et al., 2023b) mostly design modules that fuse both modalities features X^{rgb} and X^{d} for final predictions. We will show in our experiments that our X^{rgb} features can efficiently extract the 3D geometry clues from the depth modality thanks to our powerful RGB-D pretrained encoder. Delivering the depth features X^{d} to the decoder is not necessary.

3 Experiments

3.1 RGB-D Semantic Segmentation

Datasets & implementation details. Following the common experiment settings of RGB-D semantic segmentation methods (Xie et al., 2021; Guo et al., 2022a), we finetune and evaluate our DFormer on two widely used datasets, *i.e.,* NYUDepthv2 (Silberman et al., 2012) and SUN-RGBD (Song et al., 2015). During finetuning, we only adopt two common data augmentation strategies, *i.e.,* random horizontal flipping and random scaling (from 0.5 to 1.75). The training images are cropped and resized to 480 × 640 and 480 × 480 respectively for NYU Depthv2 and SUN-RGBD benchmarks. Cross-entropy loss is utilized as the optimization objective. We use AdamW (Kingma & Ba, 2015) as our optimizer with an initial learning rate of 6e-5 and the poly decay schedule. Weight decay is set to 1e-2. During test, we employ mean Intersection over Union (mIoU), which is averaged across semantic categories, as the primary evaluation metric to measure the segmentation performance. Following recent works (Zhang et al., 2023a; Wang et al., 2022; Zhang et al., 2023b), we adopt multi-scale (MS) flip inference strategies with scales $\{0.5, 0.75, 1, 1.25, 1.5\}$.

Comparisons with state-of-the-art methods. We compare our DFormer with 13 recent RGB-D semantic segmentation methods on the NYUDepthv2 (Silberman et al., 2012) and SUN-RGBD (Song et al., 2015) datasets. These methods are chosen according to three criteria: a) recently published, b) representative, and c) with open-source code. As shown in Tab. 1, our DFormer achieves new state-of-the-art performance across these two benchmark datasets. We also plot the performance-efficiency curves of different methods on the validation set of the NYUDepthv2 (Silberman et al., 2012) dataset in Fig. 2. It is clear that DFormer achieves much better performance and computation

Table 2: Quantitative comparisons on RGB-D SOD benchmarks. The best results are **highlighted**.

Dataset	Param	Flops	DES(135)				NLPR(300)				NJU2K(500)				STERE(1,000)				SIP(929)			
Metric	(M)	(G)	M	F	S	E	M	F	S	E	M	F	S	E	M	F	S	E	M	F	S	E
BBSNet$_{21}$ (Zhai et al.)	49.8	31.3	.021	.942	.934	.955	.023	.927	.930	.953	.035	.931	.920	.941	.041	.919	.908	.931	.055	.902	.879	.910
DCF$_{21}$ (Ji et al.)	108.5	54.3	.024	.910	.905	.941	.022	.918	.924	.958	.036	.922	.912	.946	.039	.911	.902	.940	.052	.899	.876	.916
DSA2F$_{21}$ (Sun et al.)	36.5	172.7	.021	.896	.920	.962	.024	.897	.918	.950	.039	.901	.903	.923	.036	.898	.904	.933	-	-	-	-
CMINet$_{21}$ (Zhang et al.)	-	-	.016	.944	.940	.975	.020	.931	.932	.959	.028	.940	.929	.954	.032	.925	.918	.946	.040	.923	.898	.934
DSNet$_{21}$ (Wen et al.)	172.4	141.2	.021	.939	.928	.956	.024	.925	.926	.951	.034	.929	.921	.946	.036	.922	.914	.941	.052	.899	.876	.910
UTANet$_{21}$ (Zhao et al.)	48.6	27.4	.026	.921	.900	.932	.020	.928	.932	.964	.037	.915	.902	.945	.033	.921	.910	.948	.048	.897	.873	.925
BIANet$_{21}$ (Zhang et al.)	49.6	59.9	.017	.948	.942	.972	.022	.926	.928	.957	.034	.932	.923	.945	.038	.916	.908	.935	.046	.908	.889	.922
SPNet$_{21}$ (Zhou et al.)	150.3	68.1	.014	.950	.945	.980	.021	.925	.927	.959	.028	.935	.925	.954	.037	.915	.907	.944	.043	.916	.894	.930
VST$_{21}$ (Liu et al.)	83.3	31.0	.017	.940	.943	.978	.024	.920	.932	.962	.035	.920	.922	.951	.038	.907	.913	.951	.040	.915	.904	.944
RD3D+$_{22}$ (Chen et al.)	28.9	43.3	.017	.946	.950	**.982**	.022	.921	.933	.964	.033	.928	.928	.955	.037	.905	.914	.946	.046	.900	.892	.928
BPGNet$_{22}$ (Yang et al.)	84.3	138.6	.020	.932	.937	.973	.024	.914	.927	.959	.034	.926	.923	.953	.040	.904	.907	.944	-	-	-	-
C2DFNet$_{22}$ (Zhang et al.)	47.5	21.0	.020	.937	.922	.948	.021	.926	.928	.956	-	-	-	-	.038	.911	.902	.938	.053	.894	.782	.911
MVSalNet$_{22}$ (Zhou et al.)	-	-	.019	.942	.937	.973	.022	.931	.930	.960	.036	.923	.912	.944	.036	.921	.913	.944	-	-	-	-
SPSN$_{22}$ (Lee et al.)	37.0	100.3	.017	.942	.937	.973	.023	.917	.923	.956	.032	.927	.918	.949	.035	.909	.906	.941	.043	.910	.891	.932
HiDANet$_{23}$ (Wu et al.)	130.6	71.5	**.013**	.952	.946	.980	.021	.929	.930	.961	.029	.939	.926	.954	.035	.921	.911	.946	.043	.919	.892	.927
DFormer-T	5.9	4.5	.016	.947	.941	.975	.021	.931	.932	.960	.028	.937	.927	.953	.033	.921	.915	.945	.039	.922	.900	.935
DFormer-S	18.5	10.1	.016	.950	.939	.970	.020	.937	.936	.965	.026	.941	.931	.960	.031	.928	.920	.951	.041	.921	.898	.931
DFormer-B	29.3	16.7	**.013**	**.957**	**.948**	**.982**	.019	.933	.936	.965	.025	.941	.933	.960	**.029**	**.931**	**.925**	.951	.035	.932	.908	.943
DFormer-L	38.8	26.2	**.013**	.956	**.948**	.980	**.016**	**.939**	**.942**	**.971**	**.023**	**.946**	**.937**	**.964**	.030	.929	.923	**.952**	**.032**	**.938**	**.915**	**.950**

Table 3: RGB-D pretraining. 'RGB+RGB' pretraining replaces depth maps with RGB images during pretraining. Input channel of the stem layer is modified from 1 to 3. The depth map is duplicated three times during finetuning.

pretrain	**Finetune**	**mIoU** (%)
RGB+RGB	RGB+D	53.3
RGB+D (**Ours**)	RGB+D	55.6

Table 4: Different inputs of the decoder head for DFormer-B. 'X^{rgb}+X^{d}' means simultaneously uses RGB and depth features. Specifically, both features from the last three stages are used as the input of the decoder head.

Decoder input	**#Params**	**FLOPs**	**mIoU** (%)
X^{rgb} (**Ours**)	29.5	41.9	55.6
$X^{rgb} + X^{d}$	30.8	44.8	55.5

trade-off compared to other methods. Particularly, DFormer-L yields 57.2% mIoU with 39.0M parameters and 65.7G Flops, while the recent best RGB-D semantic segmentation method, *i.e.,* CMX (MiT-B2), only achieves 54.4% mIoU using 66.6M parameters and 67.6G Flops. It is noteworthy that our DFormer-B can outperform CMX (MIT-B2) by 1.2% mIoU with half of the parameters (29.5M, 41.9G vs 66.6M, 67.6G). Moreover, the qualitative comparisons between the semantic segmentation results of our DFormer and CMNext (Zhang et al., 2023b) in Fig. 14 of the appendix further demonstrate the advantage of our method. In addition, the experiments on SUN-RGBD (Song et al., 2015) also present similar advantages of our DFormer over other methods. These consistent improvements indicate that our RGB-D backbone can more efficiently build interactions between the RGB and depth features, and hence yields better performance with even lower computational cost.

3.2 RGB-D Salient Object Detction

Datasets & implementation details. We finetune and test DFormer on five popular RGB-D salient object detection datasets. The finetuning dataset consists of 2,195 samples, where 1,485 are from NJU2K-train (Ju et al., 2014) and the other 700 samples are from NLPR-train (Peng et al., 2014). The model is evaluated on five datasets, *i.e.,* DES (Cheng et al., 2014) (135), NLPR-test (Peng et al., 2014) (300), NJU2K-test (Ju et al., 2014) (500), STERE (Niu et al., 2012) (1,000), and SIP (Fan et al., 2020) (929). For performance evaluation, we adopt four golden metrics of this task, *i.e.,* Structure-measure (S) (Fan et al., 2017), mean absolute error (M) (Perazzi et al., 2012), max F-measure (F) (Margolin et al., 2014), and max E-measure (E) (Fan et al., 2018).

Comparisons with state-of-the-art methods. We compare our DFormer with 11 recent RGB-D salient object detection methods on the five popular test datasets. As shown in Tab. 2, our DFormer is able to surpass all competitors with the least computational cost. More importantly, our DFormer-T yields comparable performance to the recent state-of-the-art method SPNet (Zhou et al., 2021a) with less than 10% computational cost (5.9M, 4.5G vs 150.3M, 68.1G). The significant improvement further illustrates the strong performance of DFormer.

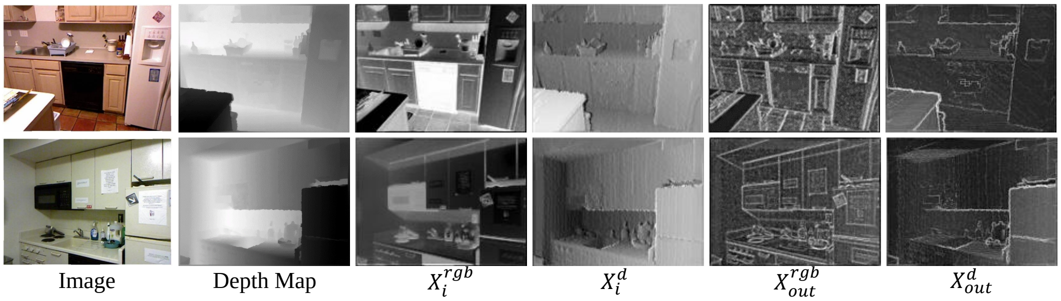


Figure 5: Visualizations of the feature maps around the last RGB-D block of the first stage.

Table 5: Ablation results on the components of the RGB-D block in DFormer-S.

Model	#Params	FLOPs	mIoU (%)
DFormer-S	18.7M	25.6G	53.6
w/o Base	14.6M	19.6G	52.1
w/o GAA	16.5M	21.7G	52.3
w/o LEA	16.3M	23.3G	52.6

Table 6: Ablations on GAA in the DFormer-S. '$k \times k$' means the adaptive pooling size of Q.

Kernel Size	#Params	FLOPs	mIoU (%)
3×3	18.7M	22.0G	52.7
5×5	18.7M	23.3G	53.1
7×7	18.7M	25.6G	53.6
9×9	18.7M	28.8G	53.6

Table 7: Different fusion methods for Q, K, and V. We can see that there is no need to fuse RGB and depth features for K and V.

Fusion variable	#Params	FLOPs	mIoU (%)
None	18.5M	25.4G	53.2
only Q (ours)	18.7M	25.6G	53.6
Q, K, V	19.3M	28.5G	53.6

Table 8: Different fusion manners in the LEA module. Fusion manner refers to the operation to fuse the RGB and depth information in LEA.

Fusion manner	#Params	FLOPs	mIoU(%)
Addition	18.6M	25.1G	53.1
Concatenation	19.0M	26.4G	53.3
Hadamard	18.7M	25.6G	53.6

3.3 Ablation Study and Analysis

We investigate the effectiveness of each component here. All experiments here are conducted under the RGB-D segmentation setting on NYU DepthV2 (Silberman et al., 2012).

RGB-D *vs.* RGB pretraining. To explain the necessity of the RGB-D pretraining, we attempt to replace the depth maps with RGB images during pretraining, dubbed as RGB pretraining. To be specific, RGB pretraining modifies the input channel of the depth stem layer from 1 to 3. Note that for the finetuning setting, the modalities of the input data and the model structure are the same. As shown in Tab. 3, our RGB-D pretraining brings 2.3% improvement for DFormer-B compared to the RGB pretraining in terms of mIoU on NYU DepthV2. We argue that this is because our RGB-D pretraining avoids the mismatch encoding of the 3D geometry features of depth maps caused by the use of pretrained RGB backbones and enhances the interaction efficiency between the two modalities. Tab. 12 and Fig. 7 in the appendix also demonstrate this. These experimental results indicate that the RGB-D representation capacity learned during the RGB-D pretraining is crucial for segmentation accuracy.

Input features of the decoder. Benefiting from the powerful RGB-D pretraining, the features of the RGB branch can efficiently fuse the information of two modalities. Thus, our decoder only uses the RGB features X^{rgb}, which contains expressive clues, instead of using both X^{rgb} and X^{d}. As shown in Tab. 4, using only X^{rgb} can save computational cost without performance drop, while other methods usually need to use both X^{rgb} and X^{d}. This difference also demonstrates that our proposed RGB-D pretraining pipeline and block are more suitable for RGB-D segmentation.

Components in our RGB-D block. Our RGB-D block is composed of a base module, GAA module, and LEA module. We take out these three components from DFormer respectively, where the results are shown in Tab. 5. It is clear that all of them are essential for our DFormer. Moreover, we visualize the features around the RGB-D block in Fig. 5. It can be seen the output features can capture more comprehensive details. As shown in Tab. 7, in the GAA module, we find that only fusing the depth features into Q is adequate and further fusing depth features to K and V brings negligible improvement but extra computational burden. Moreover, we find that the performance of DFormer initially rises as the fixed pooling size of the GAA increases, and it achieves the best when the fixed pooling size is set to 7×7 in Tab. 6. We also use two other fusion manners to replace that

Table 9: Comparisons under the RGB-only pretraining. 'NYU' and 'SUN' means the performance on the NYU DepthV2 and SUNRGBD.

Model	**Params**	**FLOPs**	NYU	SUN
CMX (MiT-B2)	66.6M	67.6G	54.4	49.7
DFormer-B	29.5M	41.9G	53.3	49.5
DFormer-L	39.0M	65.7G	55.4	50.6

Table 10: Comparisons under the RGB-D pretraining. 'NYU' and 'SUN' means the performance on NYU DepthV2 and SUNRGBD.

Model	**Params**	**FLOPs**	NYU	SUN
CMX (MiT-B2)	66.6M	67.6G	55.8	51.1
DFormer-B	29.5M	41.9G	55.6	51.2
DFormer-L	39.0M	65.7G	57.2	52.5

in LEA, *i.e.,* concatenation, addition. As shown in Tab. 8, using the depth features that processed by a large kernel depth-wise convolution as attention weights to reweigh the RGB features via a simple Hadamard product achieves the best performance.

Channel ratio between RGB and depth. RGB images contain information pertaining to object color, texture, shape, and its surroundings. In contrast, depth images typically convey the distance information from each pixel to the camera. Here, we investigate the channel ratio that is used to encode the depth information. In Fig. 6, we present the performance of DFormer-S with different channel ratios, *i.e.,* C^d/C^{rgb}. We can see that when the channel ratio excceds '1/2', the improvement is trivial while the computational burden is significantly increased. Therefore, we set the ratio to 1/2 by default.

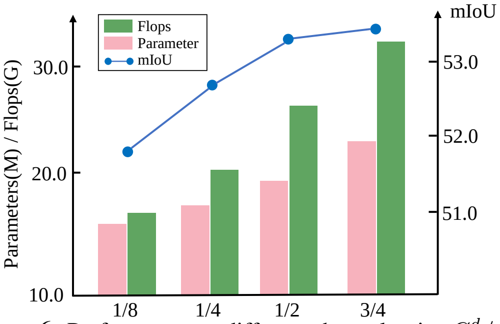


Figure 6: Performance on different channel ratios C^d/C^{rgb} based on DFormer-S. C^{rgb} is fixed and we adjust C^d to get different ratios.

Applying the RGB-D pretraining manner to CMX. To verify the effect of the RGB-D pretraining on other methods and make the comparison fairer, we pretrain CMX (MiT-B2) on the RGB-D data of ImageNet and it obtains about 1.4% mIoU improvement, as shown in Tab. 9 and Tab. 10. Under the RGB-D pretraining, DFormer-L still outperforms CMX (MiT-B2) by a large margin, which should be attributed to that the pretrained fusion weight within DFormer can achieve better and efficient fusion between RGB-D data. Besides, we provide the RGB pretrained DFormers to provide more insights in Tab. 9. A similar phenomenon appears under the RGB-only pretraining.

Discussions on the generalization ability to other modalities. Through RGB-D pretraining, our DFormer is endowed with the capacity to interact the RGB and depth features during pretraining. To verify whether the interaction method still works when replacing depth with another modality, we apply our DFormer to some benchmarks with other modalities, *i.e.,* RGB-T on MFNet (Ha et al., 2017) and RGB-L on KITTI-360 (Liao et al., 2021). As shown in Tab. 11 (Comparisons to more methods are shown in Tab. 14 and Tab. 15), our pretraining framework still improves the performance. However, the improvement is limited compared to that on RGB-D scenes. To address this issue, a foreseeable solution is to conduct pretraining of DFormer on other modalities. We will view this as our future work.

Table 11: Results on the RGB-T semantic segmentation benchmark MFNet (Ha et al., 2017) and RGB-L semantic segmentation benchmark KITTI-360 (Liao et al., 2021). '(RGB)' and '(RGBD)' mean RGB-only and RGB-D pretraining, respectively. '*' represents our implementations.

Model	**Params**	**FLOPs**	MFNet	KITTI
CMX-B2	66.6M	67.6G	58.2	64.3
CMX-B4	139.9M	134.3G	59.7	65.5*
CMNeXt-B2	65.1M	65.5G	58.4*	65.3
CMNeXt-B4	135.6M	132.6G	59.9	65.6*
Ours-L (RGB)	39.0M	65.7G	59.5	65.2
Ours-L (RGBD)	39.0M	65.7G	60.3	66.1

4 Related Work

RGB-D scene parsing. In recent years, with the rise of deep learning technologies, *e.g.*, CNNs (He et al., 2016), and Transformers (Vaswani et al., 2017; Li et al., 2023a), significant progress has been made in scene parsing (Xie et al., 2021; Yin et al., 2022; Chen et al., 2023; Zhang et al., 2023d), one of the core pursuits of computer vision. However, most methods still struggle to cope with some challenging scenes in the real world (Li et al., 2023b; Sun et al., 2020), as they only focus on RGB images that provide them with distinct colors and textures but no 3D geometric information. To overcome these challenges, researchers combine images with depth maps for a comprehensive understanding of scenes.

Semantic segmentation and salient object detection are two active areas in RGB-D scene parsing. Particularly, the former aims to produce per-pixel category prediction across a given scene, and the latter one attempts to capture the most attention-grabbing objects. To achieve the interaction and alignment between RGB-D modalities, the prevalent methods investigate a lot of efforts in building fusion modules to bridge the RGB and depth features extracted by two parallel pretrained backbones. For example, methods like CMX (Zhang et al., 2023a), TokenFusion (Wang et al., 2022), and HiDANet (Wu et al., 2023) dynamically fuse the RGB-D representations from RGB and depth encoders and aggregate them in the decoder. The evolution of fusion manners has dramatically pushed the performance boundary in these applications of RGB-D scene parsing. Nevertheless, the three common issues, as discussed in Sec. 1, are still left unresolved.

Another line of works focuses on the design of operators (Wang & Neumann, 2018; Wu et al., 2020; Cao et al., 2021; Chen et al., 2021a) to extract complementary information from RGB-D modalities. For instance, methods like ShapeConv (Cao et al., 2021), SGNet (Chen et al., 2021a), and Z-ACN (Wu et al., 2020) propose depth-aware convolutions, which enable efficient RGB features and 3D spatial information integration to largely enhance the capability of perceiving geometry. Although these methods are efficient, the improvements brought by them are usually limited due to the insufficient extraction and utilization of the 3D geometry information involved in the depth modal.

Multi-modal learning. The great success of the pretrain-and-finetune paradigm in natural language processing and computer vision has been expanded to the multi-modal domain, and the learned transferable representations have exhibited remarkable performance on a wide variety of downstream tasks. Existing multi-modal learning methods cover a large number of modalities, *e.g.*, image and text (Castrejon et al., 2016; Chen et al., 2020b; Radford et al., 2021; Zhang et al., 2021b; Wu et al., 2022), text and video (Akbari et al., 2021), text and 3D mesh (Zhang et al., 2023c), image, depth, and video (Girdhar et al., 2022). These methods can be categorized into two groups, *i.e.*, multi- and joint-encoder ones. Specifically, the multi-encoder methods exploit multiple encoders to independently project the inputs in different modalities into a common space and minimize the distance or perform representation fusion between them. For example, methods like CLIP (Radford et al., 2021) and VATT (Akbari et al., 2021) employ several individual encoders to embed the representations in different modalities and align them via a contrastive learning strategy. In contrast, the joint-encoder methods simultaneously input the different modalities and use a multi-modal encoder based on attention mechanisms to model joint representations. For instance, MultiMAE (Bachmann et al., 2022) adopts a unified transformer to encode the tokens with a fixed dimension that are linearly projected from a small subset of randomly sampled multi-modal patches and multiple task-specific decoders to reconstruct their corresponding masked patches by attention mechanisms separately.

In this paper, we propose DFormer, a novel framework that achieves RGB-D representation learning in a pretraining manner. To the best of our knowledge, this is the first attempt to encourage the semantic cues from RGB and depth modalities to align together by the explicit supervision signals of classification, yielding transferable representations for RGB-D downstream tasks.

5 Conclusions

In this paper, we propose a novel RGB-D pretraining framework to learn transferable representations for RGB-D downstream tasks. The core of our model is a tailored RGB-D block. Thanks to the RGB-D block, our method is able to achieve better interactions between the RGB and depth modalities during pretraining. This avoids the use of another branch for encoding depth information. Our experiments show that DFormer can achieve new state-of-the-art performance in RGB-D downstream tasks, *e.g.,* semantic segmentation and salient object detection, with far less computational cost compared to existing methods. We also show that our pretraining framework works also well for other modalities, like RGB-T and RGB-L. We hope this work could bring the multi-modal segmentation community new insights for advanced model design.

Acknowledgments

This research was supported by National Key Research and Development Program of China (No. 2021YFB3100800), NSFC (NO. 62225604, No. 62276145, and No. 62376283), the Fundamental Research Funds for the Central Universities (Nankai University, 070-63223049), CAST through Young Elite Scientist Sponsorship Program (No. YESS20210377). Computations were supported by the Supercomputing Center of Nankai University (NKSC).

References

Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *NeurIPS*, 2021. 9

Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. MultiMAE: Multi-modal multi-task masked autoencoders. In *ECCV*, 2022. 5, 9

Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *CVPR*, 2021. 2, 4, 16

Tim Broedermann, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. HRFuser: A multi-resolution sensor fusion architecture for 2D object detection. *arXiv preprint arXiv:2206.15157*, 2022. 18

Jinming Cao, Hanchao Leng, Dani Lischinski, Danny Cohen-Or, Changhe Tu, and Yangyan Li. ShapeConv: Shape-aware convolutional layer for indoor RGB-D semantic segmentation. In *ICCV*, 2021. 5, 9

Lluis Castrejon, Yusuf Aytar, Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Learning aligned cross-modal representations from weakly aligned data. In *CVPR*, 2016. 9

Lin-Zhuo Chen, Zheng Lin, Ziqin Wang, Yong-Liang Yang, and Ming-Ming Cheng. Spatial information guided convolution for real-time RGBD semantic segmentation. *TIP*, 30:2313–2324, 2021a. 5, 9

Qian Chen, Zhenxi Zhang, Yanye Lu, Keren Fu, and Qijun Zhao. 3-d convolutional neural networks for rgb-d salient object detection and beyond. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 6

Xiaokang Chen et al. Bi-directional cross-modality feature propagation with separation-and-aggregation gate for RGB-D semantic segmentation. In *ECCV*, 2020a. 5, 18

Xinghao Chen, Chang Xu, Minjing Dong, Chunjing Xu, and Yunhe Wang. An empirical study of adder neural networks for object detection. *NeurIPS*, 2021b. 16

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020b. 9

Yuming Chen, Xinbin Yuan, Ruiqi Wu, Jiabao Wang, Qibin Hou, and Ming-Ming Cheng. Yolo-ms: Rethinking multi-scale representation learning for real-time object detection. *arXiv preprint arXiv:2308.05480*, 2023. 8

Yupeng Cheng, Huazhu Fu, Xingxing Wei, Jiangjian Xiao, and Xiaochun Cao. Depth enhanced saliency detection method. In *ICIMCS*, 2014. 6, 20

Fuqin Deng et al. FEANet: Feature-enhanced attention network for RGB-thermal real-time semantic segmentation. In *IROS*, 2021. 18

Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *ICCV*, 2021. 16

Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In *IEEE ICCV*, 2017. 6, 20

Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. Enhanced-alignment measure for binary foreground map evaluation. In *IJCAI*, 2018. 6, 20

Deng-Ping Fan, Zheng Lin, Zhao Zhang, Menglong Zhu, and Ming-Ming Cheng. Rethinking rgb-d salient object detection: Models, data sets, and large-scale benchmarks. *TNNLS*, 32(5):2075–2089, 2020. 6, 20

Oriel Frigo, Lucien Martin-Gaffe, and Catherine Wacongne. Doodlenet: Double deeplab enhanced feature fusion for thermal-color semantic segmentation. In *CVPR*, 2022. 18

Zhengyang Geng, Meng-Hao Guo, Hongxu Chen, Xia Li, Ke Wei, and Zhouchen Lin. Is attention better than matrix decomposition? *NeurIPS*, 2021. 5, 20

Anjith George and Sebastien Marcel. Cross modal focal loss for rgbd face anti-spoofing. In *CVPR*, 2021. 21

Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A single model for many visual modalities. In *CVPR*, 2022. 5, 9

Meng-Hao Guo, Cheng-Ze Lu, Qibin Hou, Zhengning Liu, Ming-Ming Cheng, and Shi-Min Hu. Segnext: Rethinking convolutional attention design for semantic segmentation. *NeurIPS*, 2022a. 5, 19, 20

Meng-Hao Guo, Cheng-Ze Lu, Zheng-Ning Liu, Ming-Ming Cheng, and Shi-Min Hu. Visual attention network. *arXiv preprint arXiv:2202.09741*, 2022b. 4

Qishen Ha, Kohei Watanabe, Takumi Karasawa, Yoshitaka Ushiku, and Tatsuya Harada. MFNet: Towards real-time semantic segmentation for autonomous vehicles with multi-spectral scenes. In *IROS*, 2017. 8, 18

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 8

Qibin Hou, Cheng-Ze Lu, Ming-Ming Cheng, and Jiashi Feng. Conv2former: A simple transformer-style convnet for visual recognition. *arXiv preprint arXiv:2211.11943*, 2022. 4

Xinxin Hu, Kailun Yang, Lei Fei, and Kaiwei Wang. ACNet: Attention based network to exploit complementary features for RGBD semantic segmentation. In *ICIP*, 2019. 5

Keli Huang, Botian Shi, Xiang Li, Xin Li, Siyuan Huang, and Yikang Li. Multi-modal sensor fusion for auto driving perception: A survey. *arXiv preprint arXiv:2202.02703*, 2022. 1

Wei Ji, Jingjing Li, Shuang Yu, Miao Zhang, Yongri Piao, Shunyu Yao, Qi Bi, Kai Ma, Yefeng Zheng, Huchuan Lu, et al. Calibrated rgb-d salient object detection. In *CVPR*, 2021. 6

Ran Ju, Ling Ge, Wenjing Geng, Tongwei Ren, and Gangshan Wu. Depth saliency based on anisotropic center-surround difference. In *ICIP*, 2014. 6, 20

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5, 20

Minhyeok Lee, Chaewon Park, Suhwan Cho, and Sangyoun Lee. Spsn: Superpixel prototype sampling network for rgb-d salient object detection. In *ECCV*, 2022. 6

Gongyang Li, Yike Wang, Zhi Liu, Xinpeng Zhang, and Dan Zeng. Rgb-t semantic segmentation with location, activation, and sharpening. *IEEE TCSVT*, 33(3):1223–1235, 2022. 18

Zhong-Yu Li, Shanghua Gao, and Ming-Ming Cheng. Sere: Exploring feature self-relation for self-supervised transformer. *IEEE TPAMI*, 2023a. 8

Zhong-Yu Li, Bo-Wen Yin, Shanghua Gao, Yongxiang Liu, Li Liu, and Ming-Ming Cheng. Enhancing representations through heterogeneous self-supervised learning. *arXiv preprint arXiv:2310.05108*, 2023b. 9

Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D. *arXiv preprint arXiv:2109.13410*, 2021. 8, 18

Haolin Liu, Anran Lin, Xiaoguang Han, Lei Yang, Yizhou Yu, and Shuguang Cui. Refer-it-in-rgbd: A bottom-up approach for 3d visual grounding in rgbd images. In *CVPR*, 2021a. 21

Nian Liu, Ni Zhang, Kaiyuan Wan, Ling Shao, and Junwei Han. Visual saliency transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4722–4732, 2021b. 6

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022. 4

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 4

Nicolas Marchal, Charlotte Moraldo, Hermann Blum, Roland Siegwart, Cesar Cadena, and Abel Gawel. Learning densities in feature space for reliable segmentation of indoor scenes. *RA-L*, 5 (2):1032–1038, 2020. 1

Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps? In *IEEE CVPR*, 2014. 6, 20

Yuzhen Niu, Yujie Geng, Xueqing Li, and Feng Liu. Leveraging stereopsis for saliency analysis. In *CVPR*, 2012. 6, 20

Houwen Peng, Bing Li, Weihua Xiong, Weiming Hu, and Rongrong Ji. Rgbd salient object detection: A benchmark and algorithms. In *ECCV*, 2014. 6, 20

Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *IEEE CVPR*, 2012. 6, 20

Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *CVPR*, 2021. 18

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 9

Olga Russakovsky et al. ImageNet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 4, 16

Daniel Seichter, Mona Köhler, Benjamin Lewandowski, Tim Wengefeld, and Horst-Michael Gross. Efficient rgb-d semantic segmentation for indoor scene analysis. In *ICRA*, 2021. 5

Daniel Seichter, Söhnke Benedikt Fischedick, Mona Köhler, and Horst-Michael Groß. Efficient multi-task rgb-d scene analysis for indoor environments. In *IJCNN*, 2022. 5

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *ECCV*, 2012. 2, 5, 7, 19, 20, 22

Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *CVPR*, 2015. 5, 6, 19, 20

Peng Sun, Wenhu Zhang, Huanyu Wang, Songyuan Li, and Xi Li. Deep rgb-d saliency detection with depth-sensitive attention and automatic multi-modal fusion. In *CVPR*, 2021a. 6

Xiaoshuai Sun, Xuying Zhang, Liujuan Cao, Yongjian Wu, Feiyue Huang, and Rongrong Ji. Exploring language prior for mode-sensitive visual attention modeling. In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 4199–4207, 2020. 9

Yuxiang Sun, Weixun Zuo, Peng Yun, Hengli Wang, and Ming Liu. FuseSeg: Semantic segmentation of urban scenes based on RGB and thermal data fusion. *T-ASE*, 18(3):1000–1011, 2021b. 18

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 3, 4, 8

Hengyi Wang, Jingwen Wang, and Lourdes Agapito. Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam. In *CVPR*, 2023. 1

Weiyue Wang and Ulrich Neumann. Depth-aware cnn for rgb-d segmentation. In *ECCV*, 2018. 9

Yikai Wang, Wenbing Huang, Fuchun Sun, Tingyang Xu, Yu Rong, and Junzhou Huang. Deep multimodal fusion by channel exchanging. *NeurIPS*, 2020. 5

Yikai Wang, Xinghao Chen, Lele Cao, Wenbing Huang, Fuchun Sun, and Yunhe Wang. Multimodal token fusion for vision transformers. In *CVPR*, 2022. 2, 5, 9, 15, 18, 20

Hongfa Wen, Chenggang Yan, Xiaofei Zhou, Runmin Cong, Yaoqi Sun, Bolun Zheng, Jiyong Zhang, Yongjun Bao, and Guiguang Ding. Dynamic selective network for rgb-d salient object detection. *IEEE TIP*, 30:9179–9192, 2021. 6

Mingrui Wu, Xuying Zhang, Xiaoshuai Sun, Yiyi Zhou, Chao Chen, Jiaxin Gu, Xing Sun, and Rongrong Ji. Difnet: Boosting visual information flow for image captioning. In *CVPR*, 2022. 9

Zongwei Wu, Guillaume Allibert, Christophe Stolz, and Cédric Demonceaux. Depth-adapted cnn for rgb-d cameras. In *ACCV*, 2020. 9

Zongwei Wu, Guillaume Allibert, Fabrice Meriaudeau, Chao Ma, and Cédric Demonceaux. Hidanet: Rgb-d salient object detection via hierarchical depth awareness. *IEEE TIP*, 32:2160–2173, 2023. 6, 9, 23

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021. 5, 8, 19

Song Yan, Jinyu Yang, Jani Käpylä, Feng Zheng, Aleš Leonardis, and Joni-Kristian Kämäräinen. Depthtrack: Unveiling the power of rgbd tracking. In *ICCV*, 2021. 21

Yang Yang, Qi Qin, Yongjiang Luo, Yi Liu, Qiang Zhang, and Jungong Han. Bi-directional progressive guidance network for rgb-d salient object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(8):5346–5360, 2022. 6

Bowen Yin, Xuying Zhang, Qibin Hou, Bo-Yuan Sun, Deng-Ping Fan, and Luc Van Gool. Camoformer: Masked separable attention for camouflaged object detection. *arXiv preprint arXiv:2212.06570*, 2022. 8

Tao Yu, Zerong Zheng, Kaiwen Guo, Pengpeng Liu, Qionghai Dai, and Yebin Liu. Function4d: Real-time human volumetric capture from very sparse consumer rgbd sensors. In *CVPR*, 2021. 21

Yingjie Zhai, Deng-Ping Fan, Jufeng Yang, Ali Borji, Ling Shao, Junwei Han, and Liang Wang. Bifurcated backbone strategy for rgb-d salient object detection. *IEEE TIP*, 30:8727–8742, 2021. 6

Jiaming Zhang, Huayao Liu, Kailun Yang, Xinxin Hu, Ruiping Liu, and Rainer Stiefelhagen. Cmx: Cross-modal fusion for rgb-x semantic segmentation with transformers. *T-ITS*, 2023a. 5, 9, 15, 18, 20

Jiaming Zhang, Ruiping Liu, Hao Shi, Kailun Yang, Simon Reiß, Kunyu Peng, Haodong Fu, Kaiwei Wang, and Rainer Stiefelhagen. Delivering arbitrary-modal semantic segmentation. In *CVPR*, 2023b. 2, 5, 6, 18, 20, 22

Jing Zhang, Deng-Ping Fan, Yuchao Dai, Xin Yu, Yiran Zhong, Nick Barnes, and Ling Shao. RGB-D saliency detection via cascaded mutual information minimization. In *ICCV*, 2021a. 6

Miao Zhang, Shunyu Yao, Beiqi Hu, Yongri Piao, and Wei Ji. C^{2}dfnet: Criss-cross dynamic filter network for rgb-d salient object detection. *IEEE TMM*, 2022. 6

Xuying Zhang, Xiaoshuai Sun, Yunpeng Luo, Jiayi Ji, Yiyi Zhou, Yongjian Wu, Feiyue Huang, and Rongrong Ji. Rstnet: Captioning with adaptive attention on visual and non-visual words. In *CVPR*, 2021b. 9

Xuying Zhang, Bo-Wen Yin, Yuming Chen, Zheng Lin, Yunheng Li, Qibin Hou, and Ming-Ming Cheng. Temo: Towards text-driven 3d stylization for multi-object meshes. *arXiv preprint arXiv:2312.04248*, 2023c. 9

Xuying Zhang, Bowen Yin, Zheng Lin, Qibin Hou, Deng-Ping Fan, and Ming-Ming Cheng. Referring camouflaged object detection. *arXiv preprint arXiv:2306.07532*, 2023d. 8

Zhao Zhang, Zheng Lin, Jun Xu, Wen-Da Jin, Shao-Ping Lu, and Deng-Ping Fan. Bilateral attention network for rgb-d salient object detection. *IEEE TIP*, 30:1949–1961, 2021c. 6

Yifan Zhao, Jiawei Zhao, Jia Li, and Xiaowu Chen. Rgb-d salient object detection with ubiquitous target awareness. *IEEE TIP*, 30:7717–7731, 2021. 6

Heng Zhou, Chunna Tian, Zhenxi Zhang, Qizheng Huo, Yongqiang Xie, and Zhongbo Li. Multispectral fusion transformer network for rgb-thermal urban scene semantic segmentation. *IEEE GRSL*, 19:1–5, 2022a. 18

Jiayuan Zhou, Lijun Wang, Huchuan Lu, Kaining Huang, Xinchu Shi, and Bocong Liu. Mvsalnet: Multi-view augmentation for rgb-d salient object detection. In *ECCV*, 2022b. 6

Tao Zhou, Huazhu Fu, Geng Chen, Yi Zhou, Deng-Ping Fan, and Ling Shao. Specificity-preserving rgb-d saliency detection. In *ICCV*, 2021a. 6, 20, 23

Wujie Zhou, Jinfu Liu, Jingsheng Lei, Lu Yu, and Jenq-Neng Hwang. GMNet: Graded-feature multilabel-learning network for RGB-thermal urban scene semantic segmentation. *TIP*, 30:7790–7802, 2021b. 18

Wujie Zhou, Enquan Yang, Jingsheng Lei, Jian Wan, and Lu Yu. Pgdenet: Progressive guided fusion and depth enhancement network for rgb-d indoor scene parsing. *IEEE TMM*, 2022c. 5

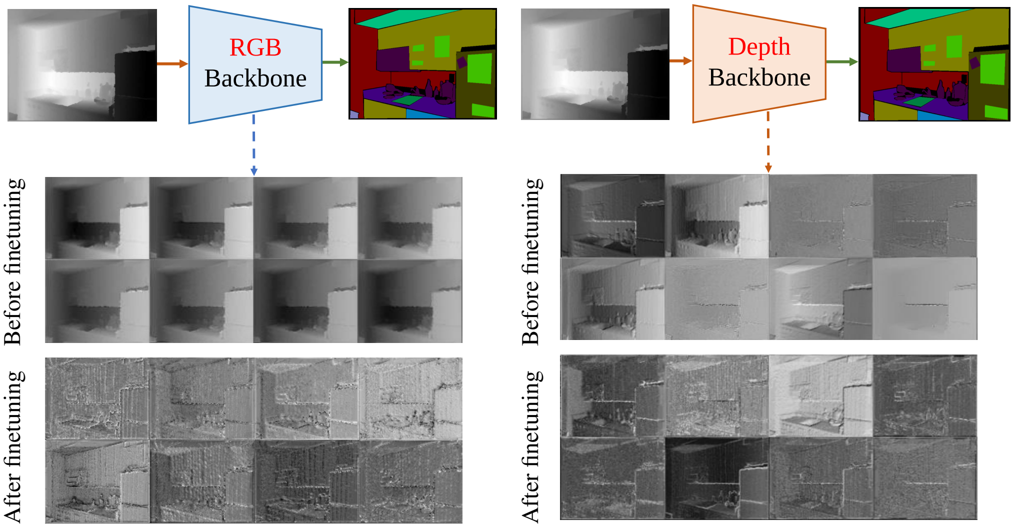
Wujie Zhou, Enquan Yang, Jingsheng Lei, and Lu Yu. Frnet: Feature reconstruction network for rgb-d indoor scene parsing. *JSTSP*, 16(4):677–687, 2022d. 5

Zhuangwei Zhuang, Rong Li, Kui Jia, Qicheng Wang, Yuanqing Li, and Mingkui Tan. Perception-aware multi-sensor fusion for 3d lidar semantic segmentation. In *ICCV*, 2021. 18

Appendix

In Sec. A, we first provide further analysis of our DFormer: 1) The efficiency in encoding the depth maps; 2) The effect of depth maps with different quality during pretraining. Then we present more detailed descriptions of DFormer in Sec. B and the experimental settings in Sec. C. Finally, we provide more visualization results of DFormer in Sec. D and future exploring directions in Sec. E.

A More analysis on DFormer



(a) RGB backbone (b) Depth backbone

Figure 7: Encoding depth maps with backbones pretrained on different types of training data. (a) Pretraining with RGB data. (b) Pretraining with depth maps. During finetuning, we only take the depth maps as input to see which backbone works better. We visualize some features from the two backbones as shown in the bottom part. Obviously, the backbone pretrained on depth maps can generate more expressive feature maps.

Table 12: Performance of the RGB pretrained and depth pretrained backbone for segmentation. The two backbones adopt the same pretraining settings and architecture but are pretrained on the ImageNet images and their depth maps, respectively.

Backbone	**#Params**	**FLOPs**	**mIoU(%)**
RGB	11.2M	13.7G	27.6
Depth	11.2M	13.7G	42.8

Why involving depth information in pretraining? The existing state-of-the-art RGB-D methods (Zhang et al., 2023a; Wang et al., 2022) tend to use models pretrained on RGB images to encode 3D geometry information for depth maps. We argue that the huge representation distribution shift caused by using RGB backbone to encode the depth maps may influence the extraction of the 3D geometry. To demonstrate this, we respectively use RGB and depth data to pretrain the RGB and depth backbone and then take only the depth maps as input for segmentation, as shown in Fig. 7. From Tab. 12, we can see under the same network architecture, the model pretrained on depth images performs much better than the one pretrained on RGB images, *i.e.,* yielding an improvement of 15% mIoU. To delve into the reasons, we visualize the feature maps as shown in the bottom part of Fig. 7. Before finetuning, the backbone pretrained on RGB data is not able to extract expressive features from the depth maps. After finetuning, the model using RGB backbones still struggles to extract diverse features from the depth maps. On the contrary, the features from the backbone pretrained on depth data are better. These experiments indicate that there exists significant difference between

RGB and depth maps and it is difficult to process the depth data with RGB pretrained weights. This also motivates us to involve depth data during ImageNet pretraining.

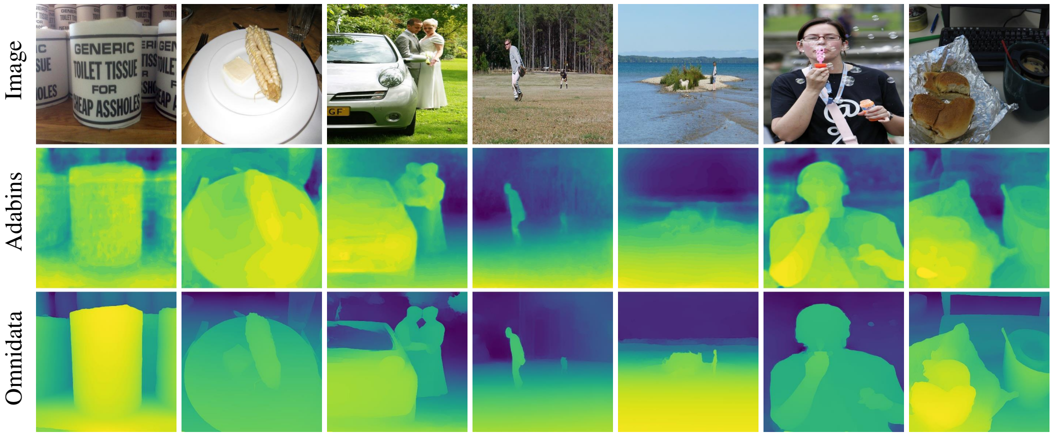


Figure 8: Comparisons of estimated depth maps that generated by Adabins (Bhat et al., 2021) and more advanced Omnidata (Eftekhar et al., 2021). We visualize the depth maps in color for better comparisons.

Impact on the quality of depth maps. In the main paper, we use a depth estimation method Adabins (Bhat et al., 2021) to predict depth maps for the ImageNet-1K (Russakovsky et al., 2015) dataset. Both the ImageNet images and the generated depth maps are used to pretrain our DFormer. Here, we explore the effect of depth maps in different quality on the performance of DFormer. To this end, we also choose a more advanced depth estimation method, *i.e.,* Omnidata (Eftekhar et al., 2021), to generate the depth maps. In Fig. 8, we visualize the generated depth maps that are generated by these two methods. We use the Omnidata-predicted depth maps to replace the Adabins-predicted ones during RGB-D pretraining. Then, we use this to pretrain the DFormer-Base and it achieves nearly the same performance, *i.e.,* 55.6 mIoU on NYUDepth v2. This indicates our DFormer is robust to the quality of depth maps.

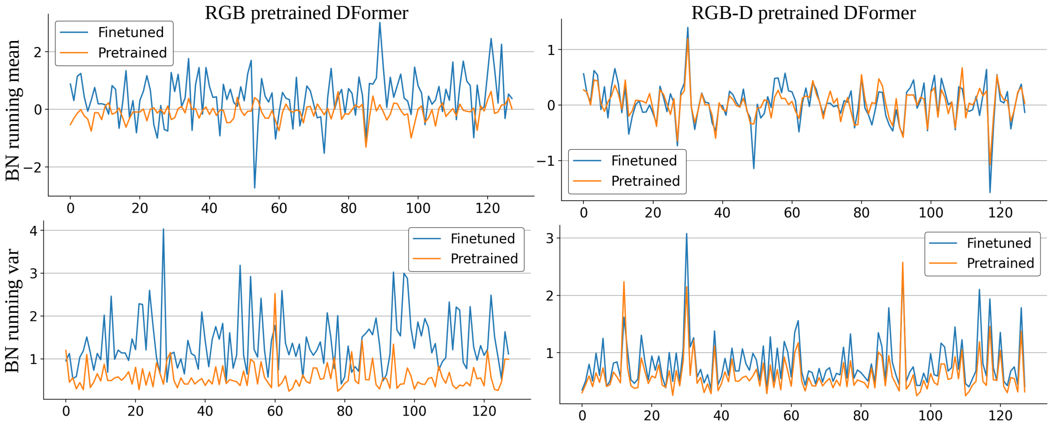


Figure 9: The statistics of distribution shift on the finetuning of DFormer-S that uses different pretraining manners. The BN layer at the first block of Stage 2 is chosen for this visualization.

Observations towards the distribution shift. Interacting the RGB and depth features within the RGB-pretrained encoders would bring in drastic changing of the feature distribution, which makes the previous statistics of batch normalization incompatible with the input features. Following (Chen et al., 2021b), we visualize the statistics of the BN layers during the finetuning process to reflect the distribution shift. Specifically, we visualize the BN layer for a random layer from DFormer in Fig. 9 to observe the statistics of the fused features. For the RGB-D pretrained DFormer, the running mean and variance of the BN layer only have slight changes after finetuning, illustrating the learned RGB-D representations are transferable for the RGB-D segmentation tasks. In contrast, for the RGB pretrained one, the statistics of the BN layer are changed sharply after finetuning, which indicates

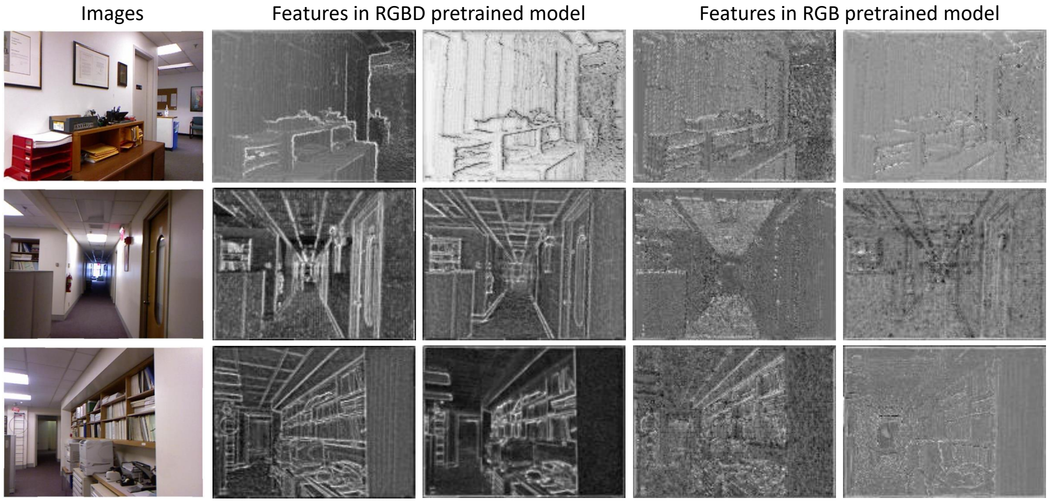

Images
Features in RGBD pretrained model
Features in RGB pretrained model

Figure 10: Visualization of the features within the finetuned models that load RGB-pretrained and RGBD-pretrained weights. The features are randomly picked from the first stage output in the DFormer-S.

the encoding is mismatched. The situation forces RGB pretrained weights to adapt the input features fused by the two modalities. We also visualize some features of DFormer that uses the RGB and RGB-D pretraining, as shown in Fig. 10.

B Details of DFormer

Structure. In the main paper, we only present the structure map of the interaction modules due to the limited space. The detailed structure of our RGB-D block is presented in Fig. 11. The GAA, LEA and base modules comprise the block, and each of them is essential and contributes to the performance improvement. GAA and LEA aims to conduct interactions between different modalities globally and locally, while the base module is only responsible for encoding RGB features. As the first stage focuses on encoding low-level features, we do not use GAA in the first stage for the sake of efficiency. Moreover, the MLP layers for the RGB and depth features are independent.

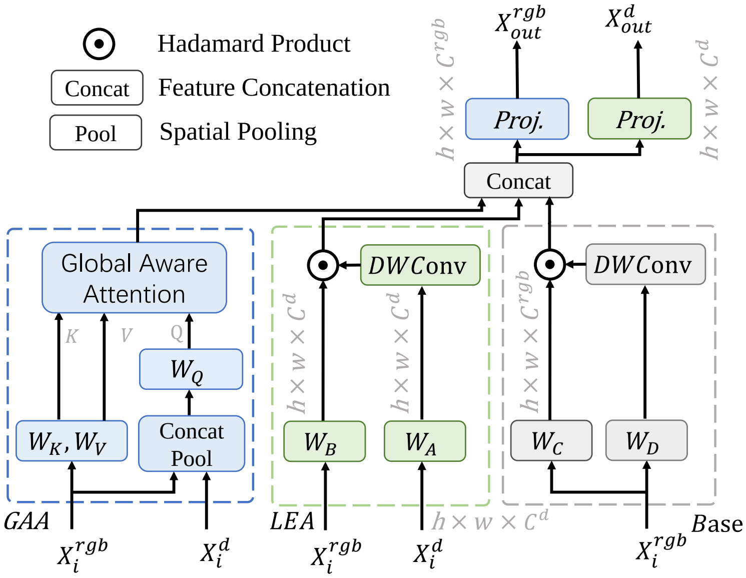
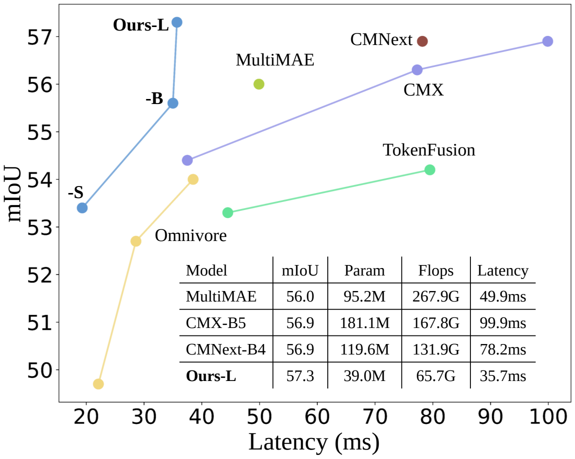


Figure 11: Detailed structure of our RGB-D block in DFormer.

Inference time analysis. Real-time inference of an RGB-D model plays a key role in a wide spectrum of downstream applications, as stated by (Chen et al., 2020a). To this end, we conduct experiments to explore the real-time potential of our DFormer and other methods. To ensure fairness, all comparisons are performed on the same device, *i.e.,* a single 3090 RTX GPU, and the same image resolution, *i.e.,* 480 × 640. As illustrated in Fig. 12, our DFormer-L achieves 57.2% mIoU with 35.7 ms latency, while the latency of the current state-of-the-art CMNext is 78.2 ms. Remarkably, our DFormer-S can process an image within 20ms and achieve about 50 frames per second (FPS) with competitive performance on NYU Depthv2, *i.e.,* 53.6% mIoU.



Model	mIoU	Param	Flops	Latency
MultiMAE	56.0	95.2M	267.9G	49.9ms
CMX-B5	56.9	181.1M	167.8G	99.9ms
CMNext-B4	56.9	119.6M	131.9G	78.2ms
Ours-L	57.3	39.0M	65.7G	35.7ms

Figure 12: Performance vs. latency when processing 480×640 images.

Table 13: Ablation towards the base modules within the building block of DFormer-S.

DWConv Setting	Attention Operation	Param	Flops	NYUDepthV2
DWConv 7 × 7	Hadamard Product	18.7M	25.6G	51.9
DWConv 5 × 5	Hadamard Product	18.7M	23.9G	51.6
DWConv 9 × 9	Hadamard Product	18.7M	27.1G	51.9
DWConv 7 × 7	Addition	18.7M	25.0G	51.3
DWConv 7 × 7	Concatanation	19.3M	26.9G	51.7

More detailed ablations on the block. In the main paper, we have provided the ablation experiments about the components of our RGB-D block, the pooling size of our GAA, as well as the fusion manners in GAA and LEA. Here we provide more results for the modules that encode RGB features, as shown in Tab. 13. All the models are pretrained for 100 epochs using DFormer-S. Note that the results in the table only ablates the structure within the module (gray part in Fig. 11) that only processes the RGB features.

Table 14: Results on MFNet (RGB-T) (Ha et al., 2017).

Method	Backbone	mIoU
FuseSeg (Sun et al.)	DenseNet-161	54.5
LASNet (Li et al.)	ResNet-152	54.9
FEANet (Deng et al.)	ResNet-152	55.3
MFTNet (Zhou et al.)	ResNet-152	57.3
GMNet (Zhou et al.)	ResNet-50	57.3
DooDLeNet (Frigo et al.)	ResNet-101	57.3
CMX (Zhang et al.)	MiT-B2	58.2
CMX (Zhang et al.)	MiT-B4	59.7
CMNeXt (Zhang et al.)	MiT-B4	59.9
(RGB) DFormer	Ours-L	59.5
(RGB-D) DFormer	Ours-L	**60.3**

Table 15: Results on KITTI-360 (RGB-L) (Liao et al.).

Method	Backbone	mIoU
HRFuser (Broedermann et al.)	HRFormer-T	48.7
PMF (Zhuang et al.)	SalsaNext	54.5
TokenFusion (Wang et al.)	MiT-B2	54.6
TransFuser (Prakash et al.)	RegNetY	56.6
CMX (Zhang et al.)	MiT-B2	64.3
CMNeXt (Zhang et al.)	MiT-B2	65.3
(RGB) DFormer	Ours-L	65.2
(RGB-D) DFormer	Ours-L	**66.1**

B.1 Comparison on the MFNet and KITTI-360

For a more comprehensive comparison, we compare our DFormer and more other state-of-the-art methods on MFNet (Ha et al., 2017) and KITTI-360 (Liao et al., 2021) in Tab. 15 and Tab. 14, as a supplement to the main paper.

Table 16: Detailed configurations of the proposed DFormer. 'C' = ('C_{rgb}','C_d'), which respectively represent the channel number of the RGB part and the depth part in different stages. 'N_i' is the number of building blocks in i-th stage. 'Expansion' is the expand ratio for the number of channels in MLPs. 'Decoder dimension' denotes the channel dimension in the decoder.

Stage	Output size	Expansion	DFormer-T	DFormer-S	DFormer-B	DFormer-L
Stem	$\frac{H}{4} \times \frac{W}{4}$	-	$C = (16, 8)$	$(32, 16)$	$(32, 16)$	$(48, 24)$
1	$\frac{H}{4} \times \frac{W}{4}$	8	$C = (32, 16), N_1 = 3$	$(64, 32), 2$	$(64, 32), 3$	$(96, 48), 3$
2	$\frac{H}{8} \times \frac{W}{8}$	8	$C = (64, 32), N_2 = 3$	$(128, 64), 2$	$(128, 64), 3$	$(192, 96), 3$
3	$\frac{H}{16} \times \frac{W}{16}$	4	$C = (128, 64), N_3 = 5$	$(256, 128), 4$	$(256, 128), 12$	$(288, 144), 12$
4	$\frac{H}{32} \times \frac{W}{32}$	4	$C = (256, 128), N_4 = 2$	$(512, 256), 2$	$(512, 256), 2$	$(576, 288), 3$
Decoder dimension			512	512	512	512
Parameters (M)			6.0	18.7	29.5	39.0

C Experimental Details

C.1 ImageNet Pretraining

We provide DFormers' ImageNet-1K pretraining settings in Tab. 17. All DFormer variants use the same settings, except the stochastic depth rate. Besides, the data augmentation strategies related to color, *e.g.,* auto contrast, are only used for RGB images, while other common strategies are simultaneously performed on RGB images and depth maps, *e.g.,* random rotation.

Table 17: **DFormer ImageNet-1K pretraining settings**. All the pretraining experiments are conducted on 8 NVIDIA 3090 GPUs.

pretraining config	DFormer-T	DFormer-S	DFormer-B	DFormer-L
input size	224^2	224^2	224^2	224^2
weight init	trunc. normal (0.2)	trunc. normal (0.2)	trunc. normal (0.2)	trunc. normal (0.2)
optimizer	AdamW	AdamW	AdamW	AdamW
base learning rate	1e-3	1e-3	1e-3	1e-3
weight decay	0.05	0.05	0.05	0.05
optimizer momentum	β_1, β_2=0.9, 0.999	β_1, β_2=0.9, 0.999	β_1, β_2=0.9, 0.999	β_1, β_2=0.9, 0.999
batch size	1024	1024	1024	1024
training epochs	300	300	300	300
learning rate schedule	cosine decay	cosine decay	cosine decay	cosine decay
warmup epochs	5	5	5	5
warmup schedule	linear	linear	linear	linear
layer-wise lr decay	None	None	None	None
randaugment	(9, 0.5)	(9, 0.5)	(9, 0.5)	(9, 0.5)
mixup	0.8	0.8	0.8	0.8
cutmix	1.0	1.0	1.0	1.0
random erasing	0.25	0.25	0.25	0.25
label smoothing	0.1	0.1	0.1	0.1
stochastic depth	0.1	0.1	0.15	0.2
head init scale	None	None	None	None
gradient clip	None	None	None	None
exp. mov. avg. (EMA)	None	None	None	None

C.2 RGB-D Semantic Segmentation Finetuning

The finetuning settings for NYUDepth v2 and SUNRGBD datasets are listed in Tab. 18. The batch sizes, input sizes, base learning rate, epochs and stochastic depth are different for the two datasets.

Datasets. Following the common experiment settings of RGB-D semantic segmentation methods (Xie et al., 2021; Guo et al., 2022a), we finetune and evaluate DFormer on two widely used datasets, *i.e.,* NYUDepthv2 (Silberman et al., 2012) and SUN-RGBD (Song et al., 2015). To be spe-

Table 18: **DFormer finetuning settings on NYUDepthv2/SUNRGBD**. Multiple stochastic depth rates, input sizes and batch sizes are for NYUDepthv2 and SUNRGBD datasets respectively. All the finetuning experiments for RGB-D semantic segmenations are conducted on 2 NVIDIA 3090 GPUs.

pretraining config	DFormer-T	DFormer-S	DFormer-B	DFormer-L
input size	480 × 640 / 480^2	480 × 640 / 480^2	480 × 640 / 480^2	480 × 640 / 480^2
optimizer	AdamW	AdamW	AdamW	AdamW
base learning rate	6e-5/8e-5	6e-5/8e-5	6e-5/8e-5	6e-5/8e-5
weight decay	0.01	0.01	0.01	0.01
batch size	8/16	8/16	8/16	8/16
epochs	500/300	500/300	500/300	500/300
optimizer momentum	β_1, β_2=0.9, 0.999	β_1, β_2=0.9, 0.999	β_1, β_2=0.9, 0.999	β_1, β_2=0.9, 0.999
training epochs	300	300	300	300
learning rate schedule	linear decay	linear decay	linear decay	linear decay
warmup epochs	10	10	10	10
warmup schedule	linear	linear	linear	linear
layer-wise lr decay	None	None	None	None
aux head	None	None	None	None
stochastic depth	0.1/0.1	0.1/0.1	0.1/0.1	0.15/0.2

cific, NYUDepthv2 (Silberman et al., 2012) contains 1,449 RGB-D samples covering 40 categories, where the resolution of all RGB images and depth maps is unified as 480 × 640. Particularly, 795 image-depth pairs are used to train the RGB-D model, and the remaining 654 are utilized for testing. SUN-RGBD (Song et al., 2015) includes 10,335 RGB-D images with 530 × 730 resolution, where the objects are in 37 categories. All samples of this dataset are divided into 5,285 and 5,050 splits for training and testing, respectively.

Implementation Details. Following SegNeXt Guo et al. (2022a), we employ Hamburger (Geng et al., 2021), a lightweight head, as the decoder to build our RGB-D semantic segmentation network. During finetuning, we only adopt two common data augmentation strategies, *i.e.,* random horizontal flipping and random scaling (from 0.5 to 1.75). The training images are cropped and resized to 480 × 640 and 480 × 480 respectively for NYU Depthv2 and SUN-RGBD benchmarks. Cross-entropy loss is utilized as the optimization objective. We use AdamW (Kingma & Ba, 2015) as our optimizer with an initial learning rate of 6e-5 and the poly decay schedule. Weight decay is set to 1e-2. During testing, we employ mean Intersection over Union (mIoU), which is averaged across semantic categories, as the primary evaluation metric to measure the segmentation performance. Following recent works (Zhang et al., 2023a; Wang et al., 2022; Zhang et al., 2023b), we adopt multi-scale (MS) flip inference strategies with scales $\{0.5, 0.75, 1, 1.25, 1.5\}$.

C.3 RGB-D Salient Object Detection

Datasets. We finetune and test DFormer on five popular RGB-D salient object detection datasets. The finetuning dataset consists of 2,195 samples, where 1,485 are from NJU2K-train (Ju et al., 2014) and the other 700 samples are from NLPR-train (Peng et al., 2014). The model is evaluated on eight datasets, *i.e.,* DES (Cheng et al., 2014) (135 samples), NLPR-test (Peng et al., 2014) (300), NJU2K-test (Ju et al., 2014) (500), STERE (Niu et al., 2012) (1,000), SIP (Fan et al., 2020) (929). The comparison of our method and other state-of-the-art methods is shown in the Tab. 2.

Implementation Details. We set the output channel of Hamburger (Geng et al., 2021) head to 1, which is further added on the top of our RGB-D backbone to build the RGB-D salient object detection network. For model finetuning, we adopt the same data augmentation strategies and model optimizer as in SPNet (Zhou et al., 2021a). For performance evaluation, we adopt four golden metrics of this task, *i.e.,* Structure-measure (S) (Fan et al., 2017), mean absolute error (M) (Perazzi et al., 2012), max F-measure (F) (Margolin et al., 2014), and max E-measure (E) (Fan et al., 2018).

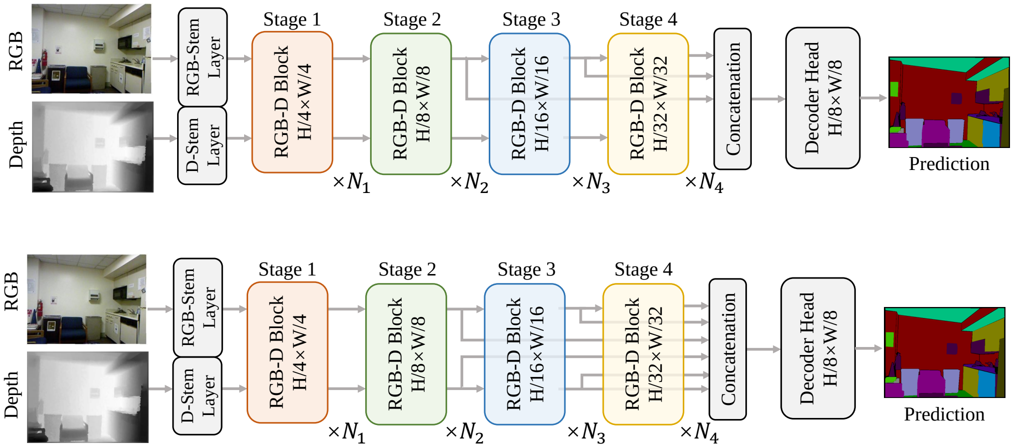


Figure 13: Detailed illustration for the input features of the decoder head. Top: Only the features in the RGB branch are sent to the decoder. Bottom: The features of both RGB and depth branches are sent to the decoder.

C.4 More details about the decoder input features

Benefiting from the powerful RGB-D pretraining, the features of the RGB branch can better fuse the information of the two modalities compared to the ones with RGB pretraining. Thus, our decoder only takes the RGB features X_i^{rgb} instead of both X_i^{rgb} and X_i^d as input. The detailed structures of the two forms are shown in Fig. 13.

D More visualization results

In this section, we provide more visualization results in Fig. 14. Our DFormer produces higher segmentation accuracy than the current state-of-the-art CMNext (MiT-B4). Moreover, the visual comparisons on RGB-D salient object detection are shown in Fig. 15.

E Future Work

We argue that there are a number of foreseeable directions for future research: 1) Applications on a wider range of RGB-D downstream tasks. Considering that our pretrained DFormer backbones own better RGB-D representation capabilities, it is promising to apply them to more RGB-D tasks, *e.g.,* RGB-D face anti-spoofing (George & Marcel, 2021), 3D visual grounding (Liu et al., 2021a), RGB-D tracking (Yan et al., 2021), and human volumetric capture (Yu et al., 2021). 2) Extension to other modalities. Based on the well-designed framework, it is interesting to produce multi-modal representations by substituting depth with other sensor data in other modalities, such as thermal and lidar data. The incorporation of additional modalities can facilitate the development of specialized and powerful encoders to meet the needs of various tasks. 3) More lightweight models. Current RGB-D methods are usually too heavy to deploy to mobile devices. Although DFormer is efficient, there is still a gap and development space in practical applications. We hope this paper could inspire researchers to develop better RGB-D encoding method, pushing the boundaries of what's feasible in real-world scenarios.

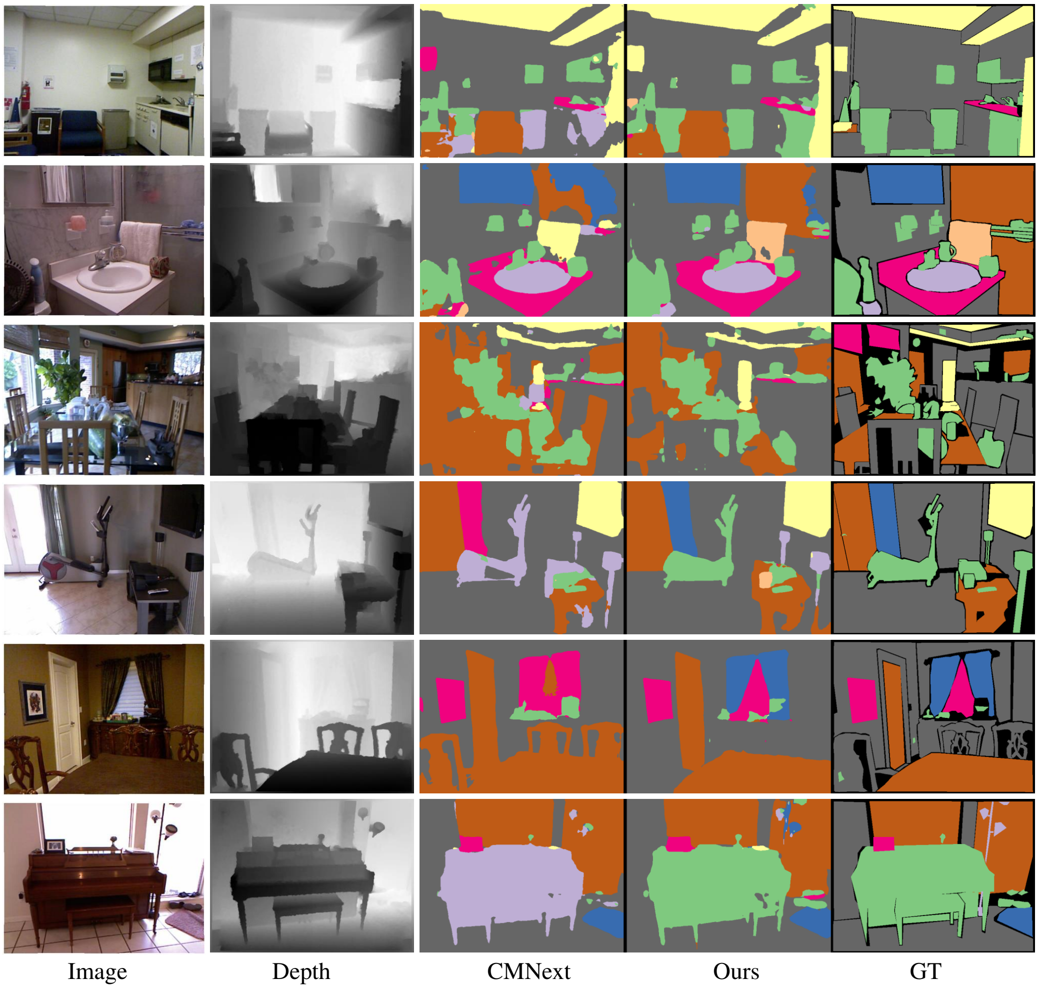


Figure 14: Qualitative comparison of DFormer-L (Ours) and CMNext (MiT-B4) (Zhang et al., 2023b) on the NYU Depthv2 (Silberman et al., 2012) dataset.

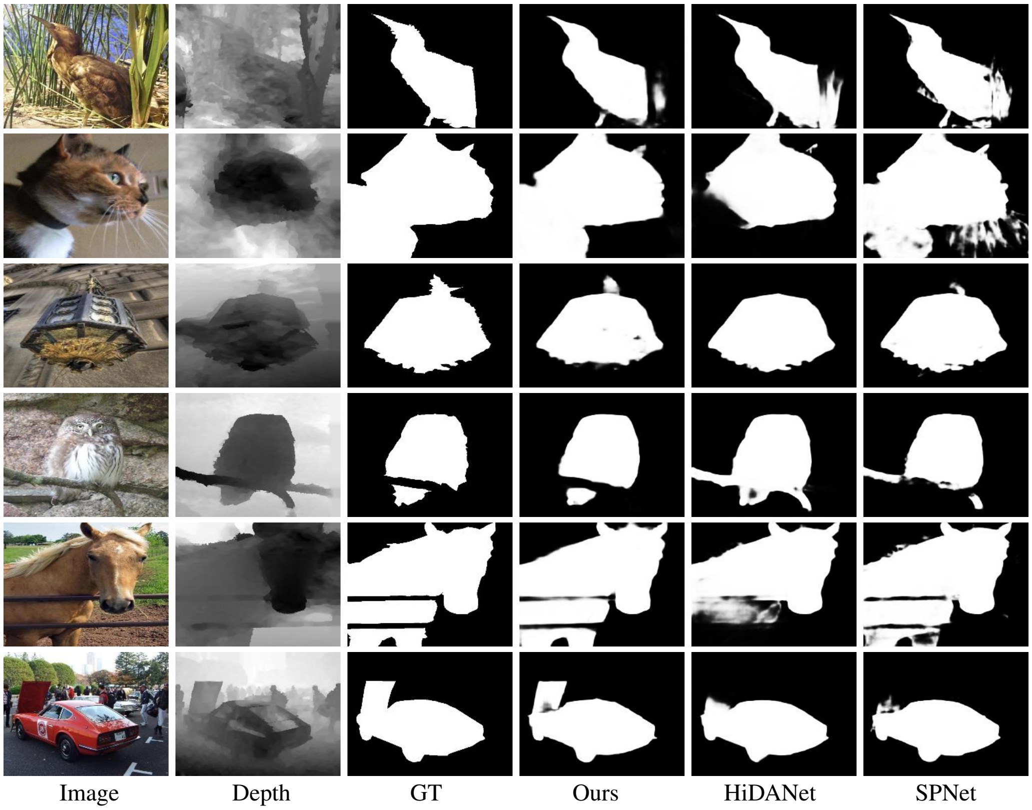


Figure 15: Qualitative comparison of DFormer-L (Ours), HiDANet (Wu et al., 2023), and SPNet Zhou et al. (2021a) on the NJU2K dataset.