Appendix

A Group Attention Block

The Group Attention Block (GAB) is a residual learning process employing the group guidance operation, suggested by Fan et al. [7]. It focuses on more important information about objects through attention with guidance from the prior segmentation map and gradually improves the map through a sequence of four group attention (GA) operations. Each GA_s operation, as shown in Fig. I, consists of 3 steps: 1) splitting the input feature $\mathbf{g}'^{(n)}_i$ into multiple ($s \in \{1, 8, 16, 32\}$) groups along the channel, 2) concatenating the guidance map $\mathbf{p}^{(n)}_i$ among the split features $\mathbf{g}'^{(n)}_{i,j}$, where $j = 1, \ldots, s$, and three finally producing an improved guidance map and feature map with convolution operations and residual connection.

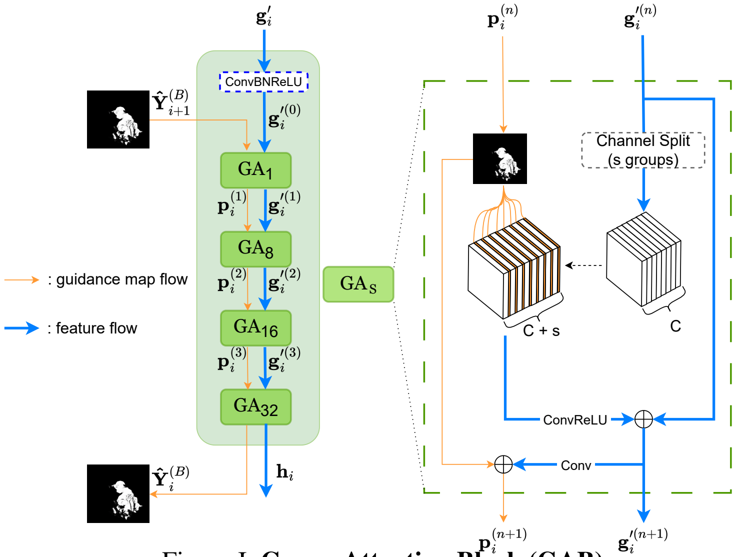

Figure I: **Group Attention Block (GAB).**

Formally, the n^{th} GA operation of the GAB at level i is given by

$$1)\ \mathbf{g}'^{(n)}_i \rightarrow \left\{\mathbf{g}'^{(n)}_{i,1}, \mathbf{g}'^{(n)}_{i,2}, \ldots, \mathbf{g}'^{(n)}_{i,s}\right\} \tag{5}$$

$$2)\ \mathbf{g}'\mathbf{p}^{(n)}_i = \texttt{Cat}\left[\mathbf{g}'^{(n)}_{i,1}, \mathbf{p}^{(n)}_i, \ldots, \mathbf{g}'^{(n)}_{i,s}, \mathbf{p}^{(n)}_i\right] \tag{6}$$

$$3)\ \mathbf{g}'^{(n+1)}_i = \mathbf{g}'^{(n)}_i \oplus \texttt{ReLU} \circ \texttt{Conv3}(\mathbf{g}'\mathbf{p}^{(n)}_i), \tag{7}$$

$$\mathbf{p}^{(n+1)}_i = \mathbf{p}^{(n)}_i \oplus \texttt{Conv3}(\mathbf{g}'^{(n+1)}_i) \tag{8}$$

where $i = 1, \ldots, L$, $n \in \{0, 1, 2, 3\}$, $\mathbf{g}'^{(0)}_i = \mathbf{g}'_i$, $\mathbf{g}'^{(4)}_i = \mathbf{h}_i$, $\mathbf{p}^{(0)}_i = \hat{\mathbf{Y}}^{(B)}_{i+1}$, and $\mathbf{p}^{(4)}_i = \hat{\mathbf{Y}}^{(B)}_i$. `Cat` indicates concatenation along the channel dimension. The convolution operations reduce the channel number from $C + s$ to C for the feature maps and from $C + s$ to 1 for guidance maps, which in turn becomes the guidance for the next GA operation. For the first GAB stage, a coarse map from the Enrich Decoder ($\hat{\mathbf{Y}}^{(E)}$) is used as the guidance segmentation map. Each

GAB iteratively refines the output segmentation map, which is used as guidance for the first GA operation at the next GAB stage.

B Parameter Size Comparison

Comparing the parameter size of architectures is essential to justify that the performance increase is not merely due to larger model capacity. To ascertain that the performance of ENTO is not merely due to an increase in parameter size, we conduct a comparative analysis with parameter sizes of other models, presented in Tab. I. As ENTO is versatile and can adapt to different backbone encoders, we choose to only compare the decoder parameter sizes, by leaving out the backbone encoder size for all models. The results reveal that our decoder parameter, at 4.17M, is smaller than all other models except SINet-V2. This indicates that the enhancement in performance of our method is not a consequence of increased parameters but rather the result of an efficient architectural design.

Model	**Decoder Params**	$S_\alpha \uparrow$	$F_\beta^w \uparrow$	$E_\phi \uparrow$	$M \downarrow$
SINet	23.35M	0.808	0.723	0.871	0.058
MGL-R	42.04M	0.833	0.740	0.867	0.052
PFNet	20.90M	0.829	0.745	0.888	0.053
SINet-V2	1.69M	0.847	0.770	0.903	0.048
SegMaR	42.44M	0.841	0.781	0.896	0.046
ZoomNet	6.78M	0.853	0.784	0.912	0.043
FSPNet	188.15M	0.879	0.816	0.915	0.048
FEDER-R2N	18.52M	0.862	-	0.913	0.042
ENTO(Ours)	4.17M	**0.904**	**0.864**	**0.942**	**0.029**

Table I: **Comparative analysis of decoder parameter sizes across different models.** Performance metrics are evaluated on the NC4K dataset.

C Implementation Details

All input images are resized to the desired input size and augmented by random flipping and rotation. ENTO can adopt various backbones for the encoder. We report the main results in Tab. 2 using PVTv2-B4 [32] pretrained on ImageNet-1K and with 768 × 768 input size. For fair comparison with previous state-of-the-art models, and to demonstrate the versatility of our architecture with any encoder, we also report ablation results using a comparable backbone and input resolution in Tab. 3. For training our model, we set the learning rate to 0.01 by default, except for the backbone (0.001). We linearly warm up the learning rate for the first half of training and decrease it to 0 for the rest. We use SGD optimizer with 0.9 momentum and 0.0005 weight decay. We train our model up to 100 epochs and set the batch size to 16. We conduct experiments on a single NVIDIA A100 GPU, taking about 10 hours to train a model. The Implementation details for other backbones are provided in Appendix D.

D Impartial Comparison Setting

Encoder Backbone	Resolution	Best Baseline	Feature Level (L)	Channels	CABs per Level	SABs per Level	Batch Size		Learning Rate	Training Epohcs
							Ours	Base		
PVTv2-B2 [32]	352 × 352	HitNet [12]	4	64	6	6	16	-	1e-2	100
	704 × 704	HitNet [12]	4	64	6	6	16	16	1e-2	100
ViT [3]	384 × 384	FSPNet [13]	4	64	6	6	16	2	1e-2	100
Res2Net50 [5]	352 × 352	SINet-V2 [7]	3	64	5	5	32	36	1e-2	100
	384 × 384	FEDER-R2N [10]	4	96	4	4	32	36	1e-2	120
ResNet50 [11]	576 × 576	ZoomNet [25]	3	64	5	5	8	8	1e-2	100

Table II: **Experiment settings for impartial comparison.** In all cases, we try to match the level of features and other parameters used in the best baseline setting.

In Tab. 5, the main text demonstrates the experimental results of our model under various combinations of encoder backbones and resolutions compared to the best-performing baseline models. Since different backbones extract features in different numbers of layers and channels, we adapt some model architectures or training settings according to them, mostly following the settings of baseline models. For each backbone, features from the last L layers are used. Tab. 3 shows the optimal settings of our model on each backbone condition according to our experiments. The channels column indicates the number of feature channels that are matched in the encoder, and the following decoder architecture is also adapted to the input channel. For backbone training, the learning rate is applied at 1/10 of the specified rate in the table. For a fair comparison with ZoomNet [25], we use the highest resolution (×1.5) images among the multi-scale setting on inputs of the ZoomNet.

E Impact of Input Resolution

We experiment with various resolutions and report the performance of our proposed model in Tab. III. We select various resolutions ranging from 352 × 352 to 896 × 896, considering the resolution choices in literature and average image size of the datasets we utilized: COD10K (740 × 963) and NC4K (529 × 709).

As shown in Tab. III, higher resolutions of input images lead to improved performance until the resolution reaches the average size of each dataset. We observed no performance improvement or even slight degradation when the resolution exceeded the average size. Tak-

	COD10K (Avg. Size: 740 × 963)				NC4K (Avg. Size: 529 × 709)			
Resolution	$S_\alpha \uparrow$	$F_\beta^w \uparrow$	$E_\phi \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$F_\beta^w \uparrow$	$E_\phi \uparrow$	$M \downarrow$
352 × 352	0.871	0.779	0.928	0.023	0.891	0.842	0.936	0.031
384 × 384	0.877	0.791	0.931	0.022	0.897	0.850	0.939	**0.029**
416 × 416	0.881	0.799	0.937	0.021	0.898	0.852	0.940	**0.029**
480 × 480	0.889	0.815	0.941	0.020	0.900	0.858	0.941	**0.029**
704 × 704	0.901	0.841	0.949	**0.018**	0.902	0.861	0.940	**0.029**
768 × 768	0.904	0.845	0.948	**0.018**	**0.904**	**0.864**	**0.942**	**0.029**
896 × 896	**0.905**	**0.847**	**0.950**	**0.018**	0.902	0.861	0.940	**0.029**

Table III: **Ablation study on different input resolution.** Bolded result shows our choice of input resolution.

ing into account these aspects and computational complexity, we select 768 × 768 as the representative resolution for our full model performance reported in Tab. 2.

F Analysis on Imperfect Ground Truths in CAMO Dataset

As evidenced by the performance results in Tab. 2, most of the COD models perform the worst in the CAMO [16] dataset. Similarly, ENTO's performance in the CAMO dataset is far lower than for other test datasets. We shed some light on why this may be so.

As evidenced in Fig. II, for some of the images in the CAMO dataset, the ground truth masks are not detailed and lump parts of the object together with the background. In some cases, crucial parts of an object are not included in the mask as well. In all such cases, ENTO successfully recovers the "mistakes" of the ground truths, leading to a more detailed and fine-grained segmentation map. Additionally, compared to baseline model, ZoomNet [25], ENTO more successfully recovers the missing parts of the objects in the ground truth and generates a more fine-grained segmentation map, matching the actual object. Such success in the model would have been penalized in the evaluation process, as they do not match the ground truth masks.

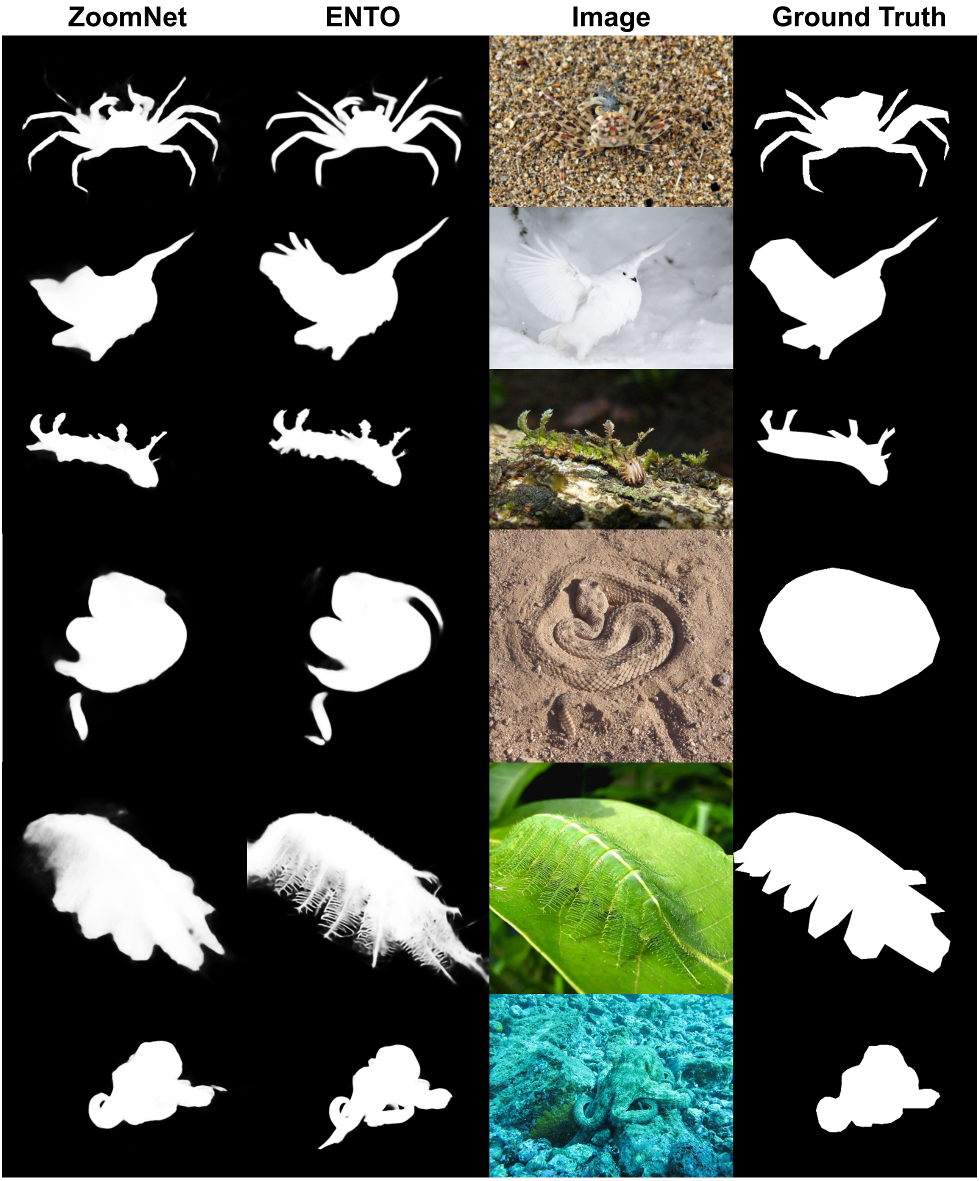

Figure II: **Examples of our model outperforming the ground truths in the CAMO dataset.** The ground truths map neglects details of the object, lump parts of objects together, and miss crucial parts of the objects. Our model is able to capture all these details, while baseline model, ZoomNet [25], misses such details.