

## A Different Generative Approach to Ref-COD — Less Effective than GRCOD

### A. Notes on Other Generative Models

We first review the Latent Diffusion Model used in our paper. It consists of an auto-encoder (VAE) and a UNet. The auto-encoder facilitates a two-way transformation between the RGB image  $\mathbf{I}_c \in \mathbb{R}^{H \times W \times 3}$  and the latent space  $\mathbf{z} \in \mathbb{R}^{h \times w \times c}$ . Both the forward and backward processes of diffusion are carried out in the latent space, and we denote the noisy latent code at time  $t$  as  $\mathbf{z}^{(t)} = \sqrt{\alpha_t} \mathbf{z} + \sqrt{1 - \alpha_t} \epsilon$ , where  $\bar{\alpha}_t = \prod_{s=1}^t (1 - \beta_s)$  is the noise schedule.  $\beta_s$  is the variance sampled from a variance schedule  $\beta_t \in (0, 1)^T$ . The UNet can be considered as a series of equally weighted denoiser  $\epsilon_\theta(\mathbf{z}^{(t)}, t)$ . The training objective  $\mathcal{L}$  can be simplified as:

$$\mathcal{L} = \mathbb{E}_{\mathbf{z}, \epsilon \sim \mathcal{N}(0,1), t \in \mathcal{U}(T)} \left[ \left\| \epsilon - \epsilon_\theta(\mathbf{z}^{(t)}, t) \right\|_2^2 \right] \quad (1)$$

Furthermore, to simplify comprehension and narration, we can reparametrize the output of UNet  $\epsilon_\theta$  as the form of v-prediction  $v_\theta$ . The training objective can be further elaborated as:

$$\mathcal{L} = \mathbb{E}_{\mathbf{z}, \epsilon \sim \mathcal{N}(0,1), t \in \mathcal{U}(T)} \left[ \left\| \mathbf{z} - v_\theta(\mathbf{z}^{(t)}, t) \right\|_2^2 \right] \quad (2)$$

This implies that the goal of every training round is to denoise  $\mathbf{z}^{(t)}$  to  $\mathbf{z}$  for any time step  $t$ .

Secondly, we present our task definition, using one-reference camouflaged object segmentation as an illustration. Given a data triplet  $(\mathbf{I}_r, \mathbf{G}_r, \mathbf{I}_c)$ , here  $\mathbf{I}_r$  and  $\mathbf{I}_c$  denote the support reference image and query camouflaged image respectively, both sharing an overlapping category  $c$ .  $\mathbf{G}_r$  is the mask of category  $c$  in the support reference image. Our task is to predict the mask corresponding to camouflaged category  $c$  in  $\mathbf{I}_c$ . In the strict one-reference camouflaged object segmentation setting, the category sets of the training set and the test set are disjoint.

Our objective is to fully utilize the priors in the Latent Diffusion Model and equip it with Few-reference Camouflaged Object Semantic Segmentation capabilities. This leads us to reuse the original VAE to convert  $\mathbf{I}_r$ ,  $\mathbf{I}_c$  and  $\mathbf{G}_c$  into latent variables  $\mathbf{z}_r$ ,  $\mathbf{z}_c$  and  $\mathbf{z}_p$ . Thus, our task is further simplified to explore how to improve the structure of UNet to  $v_\theta^*$  so that it can accept  $\mathbf{z}_r$ ,  $\mathbf{z}_c$  and  $\mathbf{G}_r$  as inputs, and use  $\mathbf{z}_p$  as supervision.

This supervised approach in the latent space has been certified effective in tasks such as depth estimation and semantic segmentation. Concretely, our training objective  $\mathcal{L}$  is transformed into:

$$\mathcal{L} = \mathbb{E}_{(\mathbf{z}_r, \mathbf{z}_c, \mathbf{G}_r, \mathbf{z}_p) \sim \mathcal{D}} \left[ \left\| \mathbf{z}_p - v_\theta^*(\mathbf{z}_r, \mathbf{z}_c, \mathbf{G}_r) \right\|_2^2 \right] \quad (3)$$

where  $\mathcal{D}$  represents the constructed training dataset. In addition, we omitted the input of time  $t$ . Our early experiments revealed that performing multiple steps of noise addition and denoising during training did not bring performance improvement.

### B. Discussion

**Limitations** Our method, as the first diffusion-based Ref-COD model, proposes a simple and intuitive design, which maximizes the retention of the generative framework of LDM. There is still a lot of room for improvement in performance (especially in the n-reference Ref-COD setting), including more sophisticated model design and more optimized training strategies. We hope that our method can serve as a diffusion-based Ref-COD baseline to inspire more researchers to invest in this field.

On the other hand, we believe that our method is not limited to Ref-COD. Our framework has the potential to unify few-reference segmentation and open vocabulary segmentation by leveraging prompts from different modalities.

#### B.1. More details on generation process

In the above section we have discussed the generation process. In addition to the final choice of one-step image-to-mask sampling, we also tried multi-step noise-to-mask sampling and multi-step image-to-mask sampling. Here we detail the training objectives of these three generation processes.

**one-step image-to-mask sampling** We directly input the image and let the UNet output the mask. This process can be described as:

$$\mathcal{L}_1 = \mathbb{E}_{(\mathbf{z}_r, \mathbf{z}_c, \mathbf{z}_p) \sim \mathcal{D}} \left[ \left\| \mathbf{z}_p - v_\theta^*(\mathbf{z}_r, \mathbf{z}_c, \mathbf{z}_p) \right\|_2^2 \right] \quad (4)$$

**Multi-step noise-to-mask generation** We add noise to camouflaged mask  $\mathbf{z}_p$ ,  $\mathbf{z}_p^{(t)} = \sqrt{\alpha_t} \mathbf{z}_p + \sqrt{1 - \alpha_t} \epsilon$ , and during inference we use  $\mathbf{z}_p^{(0)}$  as the mask prediction. The supervised form is as follows:

$$\mathcal{L}_2 = \mathbb{E}_{(\mathbf{z}_r, \mathbf{z}_c, \mathbf{z}_p, \mathbf{z}_p^{(t)}) \sim \mathcal{D}, \epsilon \sim \mathcal{N}(0,1), t \in \mathcal{U}(T)} \left[ \left\| \mathbf{z}_p - v_\theta^*(\mathbf{z}_p^{(t)}, \mathbf{z}_r, \mathbf{z}_c, \mathbf{z}_p, t) \right\|_2^2 \right] \quad (5)$$

**Multi-step image-to-mask generation** We add image(as noise) to the camouflaged mask  $\mathbf{z}_p$ ,  $\mathbf{z}_p^{(t)} = \sqrt{\alpha_t} \mathbf{z}_p + \sqrt{1 - \alpha_t} \mathbf{z}_c$ . The supervised form is as follows:

$$\mathcal{L}_3 = \mathbb{E}_{(\mathbf{z}_r, \mathbf{z}_c, \mathbf{z}_p, \mathbf{z}_p^{(t)}) \sim \mathcal{D}, t \in \mathcal{U}(T)} \left[ \left\| \mathbf{z}_p - v_\theta^*(\mathbf{z}_p^{(t)}, \mathbf{z}_r, \mathbf{z}_c, \mathbf{z}_p, t) \right\|_2^2 \right] \quad (6)$$

#### B.2. Cross-attention tokenized interaction

In the paper, we only discussed how to inject information from the reference mask based on the reference fusion

self-attention mechanism method. Here we discuss how to inject information from the reference mask based on the Tokenized Interaction Cross-Attention method. There are also the following four ways.

1. **Concatenation** We can convert the reference mask  $\mathbf{G}_r$  into an RGB image, encode  $I_r$  and  $\mathbf{G}_r$  into token sequences using CLIP image encoder respectively, concatenate them on the sequence, and finally use them as the input of cross-attention.
2. **Multiplication** We can directly multiply  $\mathbf{G}_r$  on the image  $I_r$  to form the image  $\mathbf{I}_r^* = I_r \cdot \mathbf{G}_r$ , and finally encode  $\mathbf{I}_r^*$  into a token sequence using CLIP image encoder as the input of cross-attention.
3. **Addition** We can also directly add  $\mathbf{G}_r$  to the image  $I_r$  to form the image  $\mathbf{I}_r^* = 0.5I_r + 0.5\mathbf{G}_r$ . Similarly, we encode  $\mathbf{I}_r^*$  into a token sequence using CLIP image encoder as the input of cross-attention.
4. **Attention Mask** We can use  $\mathbf{G}_r$  as an attention mask to control self-attention.

To reduce the randomness caused by initial noise, enhance the influence of reference prompts, and ensure consistency between outputs and inputs, we employ classifier-free guidance (CFG) [?]. The query camouflaged latent  $z_c$  and condition  $\tau$  are randomly set to null embedding with probability  $p = 0.05$  in the training stage.

We also adopt CFG in the inference stage. Specifically, our model outputs the  $\tilde{z}_t(z_c, \tau)$  on the basis of three conditional outputs  $\tilde{z}_t(z_c, \tau)$ ,  $\tilde{z}_t(\emptyset, \emptyset)$ ,  $\tilde{z}_t(z_c, \emptyset)$  (Eq. 7).

$$\begin{aligned} \tilde{z}_t(z_c, \tau) = & \tilde{z}_t(\emptyset, \emptyset) \\ & + \gamma_c \cdot (\tilde{z}_t(z_c, \emptyset) - \tilde{z}_t(\emptyset, \emptyset)) \\ & + \gamma_\tau \cdot (\tilde{z}_t(z_c, \tau) - \tilde{z}_t(z_c, \emptyset)), \end{aligned} \quad (7)$$

where  $\gamma_q$  and  $\gamma_\tau$  control the guidance of query camouflaged image and reference prompt, respectively.

**Meta-architecture.** We investigate the three meta-architectures by applying different training strategies. In general, the model with all parameters trainable performs best. In addition, we study the effect of parameter-efficient fine-tuning on the models using LoRA. Performance degradation is observed for both architectures.

When we further reduce the rank from 4 to 1, the performance of the one-step image-to-mask sampling slightly degrades. This phenomenon suggests that because SD was originally designed for generative tasks, its limited expressive capacity hinders transfer to segmentation tasks, and Multi-step generation is more sensitive to this characteristic.

### B.3. 1-reference to N-reference

So far, we have primarily explored the training and inference processes specifically designed for 1-reference scenarios. A natural question arises: can this framework be extended to n-reference settings? To address this, we first present the simplest and most straightforward method for adaptation, which requires only minor modifications during the inference phase to accommodate n-reference tasks.

In the paper, we introduced how to inject the information of the inference image into the features of the query camouflaged image using the reference fusion self-attention mechanism. In inference, our support set  $R$  may contain more than one image,  $S = \{I_{r1}, I_{r2}, \dots, I_{rn}\}$ . We encode each image into the features  $\mathbf{I}_{ri}$ . Correspondingly, after mapping, we can obtain a series of  $\mathbf{Q}_{ri}$ ,  $\mathbf{K}_{ri}$ ,  $\mathbf{V}_{ri}$  and  $\mathbf{Q}_{ci}$ ,  $\mathbf{K}_{ci}$ ,  $\mathbf{V}_{ci}$ . We can concatenate  $\mathbf{K}_{ci}$  and  $\mathbf{K}_{ri}$  to form  $\mathbf{K}_{cr} = [\mathbf{K}_{ci}, \mathbf{K}_{r1}, \mathbf{K}_{r2}, \dots, \mathbf{K}_{rn}]$ , and similarly we can obtain  $\mathbf{V}_{cr} = [\mathbf{V}_{ci}, \mathbf{V}_{r1}, \mathbf{V}_{r2}, \dots, \mathbf{V}_{rn}]$ . Finally, our reference fusion self-attention layer can be represented as:

$$z_c^* = \text{FusionA}(z_c, z_r) = \text{Att}(\mathbf{Q}_c, \mathbf{K}_{cr}, \mathbf{V}_{cr}) \quad (8)$$

While the aforementioned solutions enable N-reference inference, their performance does not match that of state-of-the-art (SOTA) models. This discrepancy primarily arises because the model receives only a single support image during the training phase, which leads to inconsistencies when transitioning to the inference phase with 5-reference or 10-reference configurations.

To address this issue, we explore improvements from both the inference and training perspectives. From the perspective of inference, transitioning from 1-reference to N-reference involves concatenating the keys and values of additional support samples, which significantly increases the number of keys and values processed during inference. To address this, we implement random sampling of the keys and values from the reference samples during inference, ensuring that their quantity matches that of the training phase. Another more straightforward idea is to introduce multiple reference samples during the training phase. In this way, the model can learn how to utilize multiple reference images during training. we randomly select 1 to N reference samples as input using reference fusion self-attention during a single training iteration. Our experiments demonstrate that improvements during the training phase are more effective than those during the inference phase.

For cross-attention tokenized interaction, information of the reference mask can also be injected in the same four ways. There are just some slight differences in the implementation details. We carry out a comparison of two interaction methods paired with four injection methods; these eight combinations are then verified experimentally. Overall, we observe that reference fusion self-

attention (RFSA) mechanism outperforms Tokenized Interaction Cross-Attention (TCA). We attribute this mainly to the preservation and flexible utilization of information from the reference image by Ref-COD. Conversely, TCA, which only compresses support image to tokens via the CLIP image encoder, leads to some information loss. Notably, within the RFSA, the Concatenation method surpassed the other three. It offered a more free-form handling of RGB images and MASK information via subsequent learnable convolutional layers, compared to other hard injection methods. In the case of TCA, the Attention Mask method seems more apt as other operations are actually constrained by the CLIP image encoder. The CLIP image encoder itself is not good at dealing with mask information. Of course, we believe that there is still room for further exploration here, referring to FGVP.

#### B.4. Post processing

The original prediction of the model is an RGB three-channel image. We first average over the channel dimension to obtain a single-channel  $\hat{\mathbf{G}}_c \in [0, 1]^{H \times W}$ . Then we tried two thresholding methods, absolute threshold  $\tau_a$  and relative threshold  $\tau_r$ . The absolute threshold is a fixed value, and the final binary mask  $\mathbf{G}_c$  can be represented as:

$$\mathbf{G}_c = \begin{cases} 1, & \text{if } \hat{\mathbf{G}}_c > \tau_a \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

Using relative threshold, we have:

$$\mathbf{G}_c = \begin{cases} 1, & \text{if } \hat{\mathbf{G}}_c > \tau_r \max(\hat{\mathbf{G}}_c) \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

Table 1. Comparison of different thresholding methods

$\tau_r$	0.2	0.25	0.3	0.35	0.4
mIoU	77.56	77.60	77.48	77.4	77.11
$\tau_a$	0.1	0.15	0.2	0.25	0.3
mIoU	76.65	77.24	76.98	76.58	76

#### B.5. More ablation studies

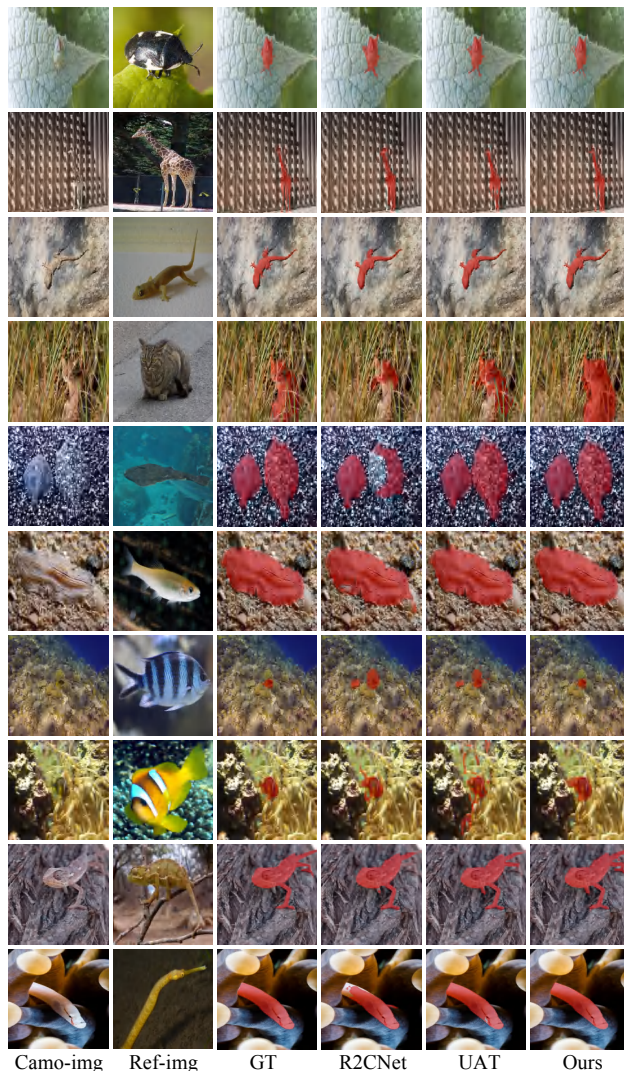
**Multiplication** We found in the experiment that Multiplication can be directly applied to RGB images, and another choice is to apply it to the latent space.

Table 2. Comparison of different Multiplication methods

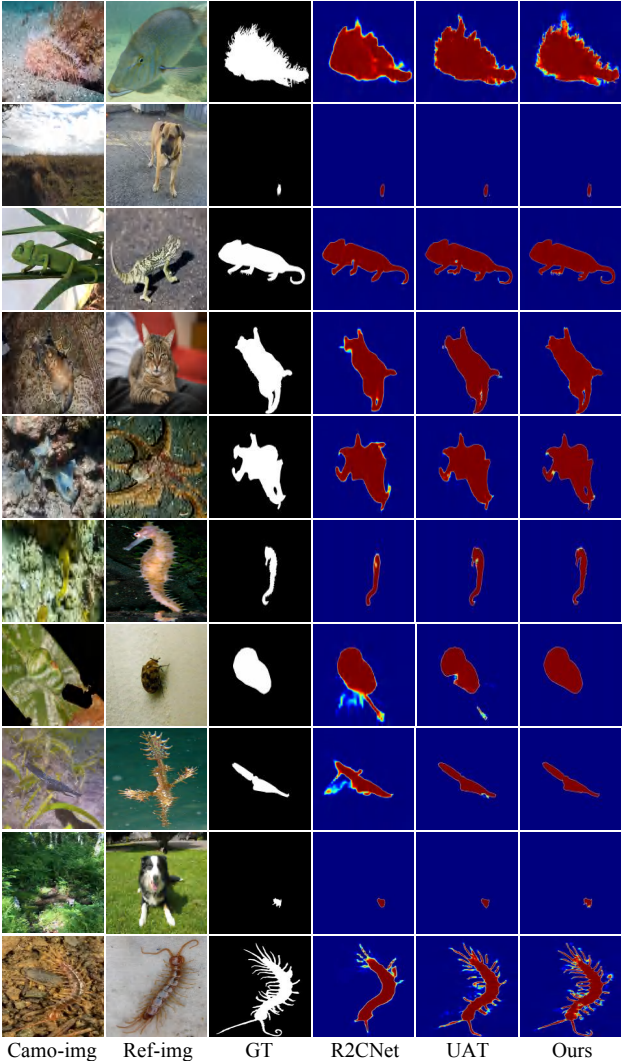
Multiplication	mIoU
latent	62.11
RGB	63.18

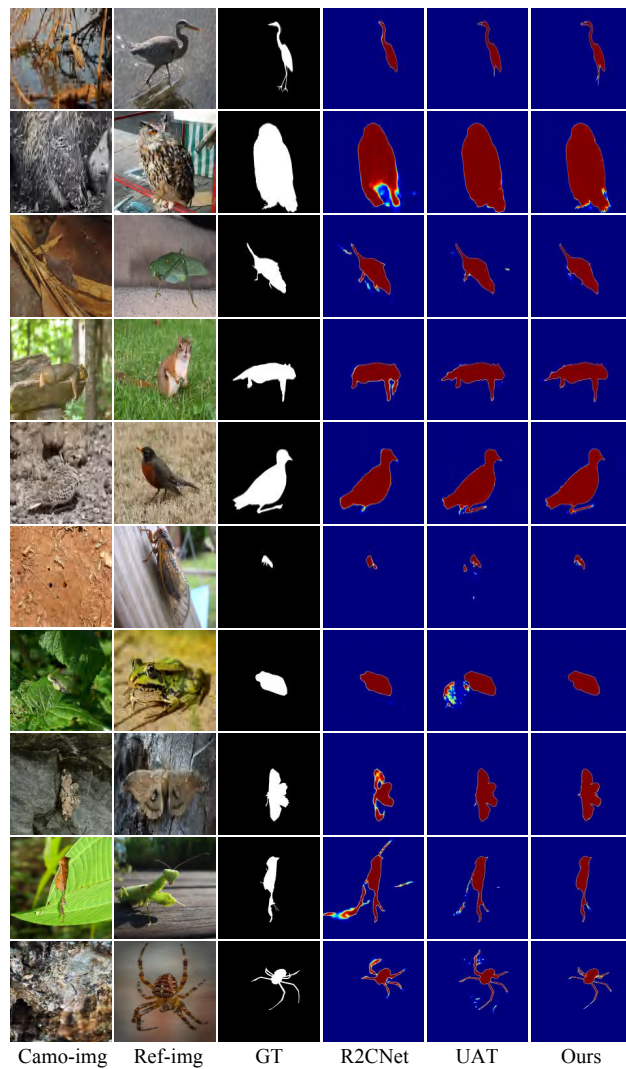
As shown in Tab. 2, the Multiplication method directly applied to RGB images achieved better results. However, the overall disparity is not significant.

#### B.6. Other visualization



Camo-img Ref-img GT R2CNet UAT Ours





Camo-img Ref-img GT R2CNet UAT Ours