ProSAM: Enhancing the Robustness of SAM-based Visual Reference Segmentation with Probabilistic Prompts

Supplementary Material

Appendix

Table of Contents

7. Theoretical Analysis of ProSAM and VRP-SAM

In this section, we present a formal analysis that reveals a deep connection between variational optimization, noise injection, and regularization. As already explained in Section 4.2, by using the reparameterization trick, variational optimization is accomplished by adding noise to the prompt embeddings during training. In the subsequent section, we demonstrate that adding noise to the prompt embeddings is mathematically equivalent to incorporating a regularization term that penalizes the Laplacian of the loss function. This equivalence not only provides a rigorous justification for our method but also elucidates how the induced flatness in the loss landscape enhances the robustness and generalization of the model.

7.1. Equivalence of Noise Injection and Laplacian Regularization

Proposition 1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a twice continuously differentiable function and let $\epsilon \in \mathbb{R}^n$ be an i.i.d. distributed random noise vector satisfying*

$$\mathbb{E}[\epsilon] = 0 \quad \textit{and} \quad \mathbb{E}[\epsilon\,\epsilon^T] = \sigma^2 I,$$

where I is the $n \times n$ identity matrix and $\sigma > 0$ is sufficiently small. Then, for any point $z \in \mathbb{R}^n$,

$$\mathbb{E}_\epsilon\Big[f(z+\epsilon)\Big] = f(z) + \frac{\sigma^2}{2}\Delta f(z) + O(\sigma^3),$$

where the Laplacian $\Delta f(z)$ is defined as

$$\Delta f(z) = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial z_i^2}(z).$$

The formal proof of Proposition 1 can be found in Section 7.4.

Corollary 1. *Assuming the $O(\sigma^3)$ term is negligible, minimizing $\mathbb{E}_\epsilon[f(z+\epsilon)]$ is equivalent to minimizing*

$$f(z) + \frac{\sigma^2}{2}\Delta f(z).$$

Mapping Corollary 1 to Variational Prompt Distribution Optimization in ProSAM. In ProSAM, we optimize a variational prompt distribution $q_\phi(z \mid I_r, M_r, I_t)$ using the reparameterization trick, where the sampled prompt embedding is expressed as

$$z = \mu_z + \epsilon,$$

with ϵ being a noise vector satisfying

$$\mathbb{E}[\epsilon] = 0 \quad \text{and} \quad \mathbb{E}[\epsilon\,\epsilon^T] = \sigma^2 I.$$

The segmentation loss is defined as

$$\mathcal{L}\big(f_S^M(z, F_S^I(I_t)), M_t\big),$$

which measures the deviation between the predicted mask $f_S^M(z, F_S^I(I_t))$ and the ground truth mask M_t. By applying Corollary 1 to this loss function, we obtain

$$\mathbb{E}_\epsilon\Big[\mathcal{L}\big(f_S^M(z, F_S^I(I_t)), M_t\big)\Big] \tag{11}$$

$$\approx \mathcal{L}\big(f_S^M(\mu_z, F_S^I(I_t)), M_t\big) + \tfrac{\sigma^2}{2}\Delta\mathcal{L}\big(f_S^M(\mu_z, F_S^I(I_t)), M_t\big). \tag{12}$$

This shows that minimizing the expected loss over the noisy prompt embeddings is equivalent to minimizing the standard segmentation loss plus an additional regularization term that penalizes the Laplacian (i.e., the curvature) of the loss with respect to the prompt embedding z at $z = \mu_z$.

7.2. The Robustness of ProSAM by Penalizing Laplacian

In this section, we explain why penalizing the Laplacian of the loss enhances the robustness of ProSAM. Following the conclusion in Section 7.1, injecting noise into the prompt embeddings during training is more than just a method for sampling from a variational distribution—it acts as an implicit regularizer that penalizes the Laplacian of the loss. Near local minima, small Laplacian indicates lower curvature, which is characterized by the Hessian matrix

$$\nabla^2\mathcal{L}(\mu_z)$$

of the loss function L with respect to the prompt embedding z, evaluated at the mean prompt μ_z. The overall curvature is then quantified by the trace of the Hessian, namely the Laplacian,

$$\Delta\mathcal{L}(\mu_z) = \mathrm{Tr}\left(\nabla^2\mathcal{L}(\mu_z)\right) = \sum_{i=1}^{n}\lambda_i,$$

where λ_i are the eigenvalues of $\nabla^2\mathcal{L}(\mu_z)$. Near a local minimum, where the segmentation loss is minimized, the loss function is typically convex or locally convex, ensuring that all eigenvalues satisfy $\lambda_i \geq 0$. Consequently, the Laplacian $\Delta\mathcal{L}(\mu_z)$ is nonnegative. Under this constraint, minimizing the Laplacian strictly leads to lower overall curvature. Following the conclusion from Section 7.1, when noise ϵ with variance σ^2 is added, the Laplacian is implicitly penalized, which effectively encourages the optimization process to favor flat minima over high curvature regions. For ProSAM, this is crucial because a flat loss landscape implies that the predicted mean prompt μ_z is robust to small perturbations, thereby enhancing the stability and generalization of segmentation performance, particularly on novel objects.

7.3. The Limitation of VRP-SAM Without Laplacian Regularization

In VRP-SAM, the prompt encoder is optimized solely by minimizing the segmentation loss:

$$\mathcal{L}(\mu_z) = \mathcal{L}\big(f_S^M(\mu_z, F_S^I(I_t)), M_t\big),$$

where μ_z is the learned prompt embedding, f_S^M denotes the SAM mask decoder, $F_S^I(I_t)$ is the image feature extraction, and M_t represents the ground truth mask. This objective ensures that the generated mask is close to the target mask but does not explicitly encourage the embedding μ_z to reside in the low-curvature area of the target prompt region R_{I_r,M_r,I_t}. As a result, the learned embedding may end up in an area where the loss function exhibits high curvature.

As explained in Section 7.2, the curvature at the embedding μ_z is characterized by the Hessian $\nabla^2\mathcal{L}(\mu_z)$ of the loss function, and its trace, the Laplacian $\Delta\mathcal{L}(\mu_z)$ can be large if μ_z is near a boundary or a sharp region of the loss landscape. Without a regularization term that penalizes this curvature—such as the additional term $\frac{1}{2}\sigma^2\,\Delta\mathcal{L}(\mu_z)$ obtained via noise injection—the optimizer is not explicitly guided to find flatter regions. Consequently, small perturbations in the embedding can lead to significant increases in loss, making the model more sensitive to noise and less robust. This sensitivity is particularly problematic when segmenting novel objects, where the embedding must generalize well to unseen variations. Hence, the absence of Laplacian regularization in VRP-SAM can result in unstable prompt embeddings and degraded segmentation performance.

7.4. Mathematical Proofs

Proof. **Step 1. Taylor Expansion**
Since f is twice continuously differentiable, we can write the second-order Taylor expansion of $f(z+\epsilon)$ about the point z:

$$f(z+\epsilon) = f(z) + \nabla f(z)^T\epsilon + \frac{1}{2}\epsilon^T H_f(z)\epsilon + R(\epsilon) \tag{13}$$

, where

- $\nabla f(z)$ is the gradient of f at z,
- $H_f(z)$ is the Hessian matrix of f at z,
- $R(\epsilon)$ is a remainder term of order $O(\|\epsilon\|^3)$.

Step 2. Taking the Expectation
Taking the expectation with respect to ϵ, we obtain:

$$\mathbb{E}_\epsilon\Big[f(z+\epsilon)\Big] = \mathbb{E}_\epsilon\left[f(z) + \nabla f(z)^T\epsilon + \frac{1}{2}\epsilon^T H_f(z)\epsilon + R(\epsilon)\right].$$

Since $f(z)$ is constant and $\mathbb{E}_\epsilon[\epsilon] = 0$, we have:

$$\mathbb{E}_\epsilon\Big[f(z+\epsilon)\Big] = f(z) + \frac{1}{2}\mathbb{E}_\epsilon\Big[\epsilon^T H_f(z)\epsilon\Big] + O(\sigma^3). \tag{14}$$

Step 3. Evaluating the Quadratic Term
Express the quadratic form as:

$$\epsilon^T H_f(z)\epsilon = \sum_{i=1}^{n}\sum_{j=1}^{n} H_f(z)_{ij}\,\epsilon_i\,\epsilon_j.$$

Taking the expectation, we have:

$$\mathbb{E}_\epsilon\Big[\epsilon^T H_f(z)\epsilon\Big] = \sum_{i=1}^{n}\sum_{j=1}^{n} H_f(z)_{ij}\,\mathbb{E}_\epsilon[\epsilon_i\,\epsilon_j].$$

Given that $\mathbb{E}_\epsilon[\epsilon_i\,\epsilon_j] = \sigma^2$ if $i = j$ and 0 otherwise, it follows that:

$$\mathbb{E}_\epsilon\Big[\epsilon^T H_f(z)\epsilon\Big] = \sigma^2 \sum_{i=1}^{n} H_f(z)_{ii} = \sigma^2\ \mathrm{Tr}(H_f(z)).$$

Recall that the Laplacian of f is defined as:

$$\Delta f(z) = \mathrm{Tr}(H_f(z)) = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial z_i^2}(z).$$

Step 4. Final Expression

Substituting the evaluated quadratic term into our expectation, we obtain:

$$\mathbb{E}_\epsilon\Big[f(z+\epsilon)\Big] = f(z) + \frac{1}{2}\sigma^2\,\Delta f(z) + O(\sigma^3).$$

This completes the proof. □

7.5. Advantage of Student-t over Gaussian

Following Equation 14, we observed that the second-order Taylor terms of $\mathbb{E}[L(z+\epsilon)]$ are identical for any zero-mean noise with covariance $\sigma^2 I$. Third-order terms also vanish because the noise distributions are symmetric and have zero odd moments. The distinction appears first in the fourth-order term, which depends on the fourth central moment

$$m_4 = \mathbb{E}[\epsilon_i^4].$$

For a Gaussian $\mathcal{N}(0, \sigma^2)$, one has

$$m_4^{\mathcal{N}} = 3\,\sigma^4,$$

whereas for a Student–t with $\nu > 4$ degrees of freedom, scaled to variance σ^2,

$$m_4^t = \frac{3\,\nu}{\nu - 4}\,\sigma^4 > 3\,\sigma^4.$$

Recalling that the fourth-order correction in the expected loss is

$$\frac{1}{24}\sum_{i,j,k,\ell} \frac{\partial^4 L}{\partial z_i \partial z_j \partial z_k \partial z_\ell}(z)\,\mathbb{E}[\epsilon_i\epsilon_j\epsilon_k\epsilon_\ell],$$

and that only index-pairings $(i = j = k = \ell)$ and $(i = j \neq k = \ell)$ survive, the larger m_4 of the Student–t directly amplifies the contribution

$$\frac{m_4}{24}\sum_i Q_{iiii} + \frac{3\,\sigma^4}{24}\sum_{i\neq j} Q_{iijj},$$

where $Q_{ijkl} = \frac{\partial^4 L}{\partial z_i \partial z_j \partial z_k \partial z_\ell}$. Consequently, Student–$t$ noise imposes a strictly stronger fourth-order "push" against high curvature than Gaussian noise, driving the mean prompt deeper into flatter regions of the loss landscape and yielding greater empirical robustness.

8. Model Architecture

In this section, the model architecture of our variational prompt encoder is described in detail. For a fair and straightforward comparison, our variational prompt encoder closely follows the model architecture of VRP-SAM [32] (see Section 3.2). As shown in Figure 5, our variational prompt encoder is composed of two major components: feature augmentation and prompt distribution prediction.

8.1. Feature Augmentation

In this component, the visual features of the reference image I_r and the target image I_t are augmented and enhanced with reference annotations M_r. First, both the reference image I_r and the target image I_t are encoded into F_{I_r} and F_{I_t} using a frozen pre-trained image encoder f_I followed by a learnable pointwise convolutional layer. To obtain a reference annotation embedding F_{M_r}, the reference image embedding F_{I_r} within the annotated region M_r is fed into an average pooling layer. Next, F_{M_r} is concatenated with both the reference image embedding F_{I_r} and the reference annotation M_r in a pointwise manner, and then transformed by another pointwise convolutional layer to produce the final enhanced reference feature F_r^v. To obtain the enhanced target feature F_t^v, a pseudo-mask of target image M_t^{pseudo} is generated by evaluating the pixel-wise similarity map through the comparison of high-level features of reference and target image. Then, the similarity map is normalized into [0,1] and serves as the pseudo mask M_t^{pseudo} for the target image. Similarly, the enhanced target feature F_t^v is obtained by transforming a concatenation of M_t^{pseudo}, F_{M_r} and F_{I_t} with a learnable pointwise convolution layer.

8.2. Prompt Distribution Prediction

Given the enhanced reference feature F_r^v and target feature F_t^v, a variational prompt distribution $q_\phi(\boldsymbol{z}|I_r, M_r, I_t)$ is predicted via attention mechanisms. First, a set of learnable queries $Q \in \mathbb{R}^{m\times c}$ is initialized and interacted with the reference feature F_r^v through a cross-attention layer and a self-attention layer, to generate query vectors $Q_r' \in \mathbb{R}^{m\times c}$ containing information about the object to be segmented. These query vectors Q_r' then interact with the target feature F_t^v via another cross-attention layer and a subsequent self-attention layer to produce the prompt features $Q_t' \in \mathbb{R}^{m\times c}$. Finally, two linear transformation heads are employed to predict the mean $\hat{\boldsymbol{\mu}}_z$ and standard deviation $\hat{\boldsymbol{\sigma}}_{\boldsymbol{z}}$ of the variational prompt distribution $q_\phi(\boldsymbol{z}|I_r, M_r, I_t)$, respectively. For the quantitative comparison against VRP-SAM with two linear layers appended at the end of prompt encoder, please refer to Section 10.2.

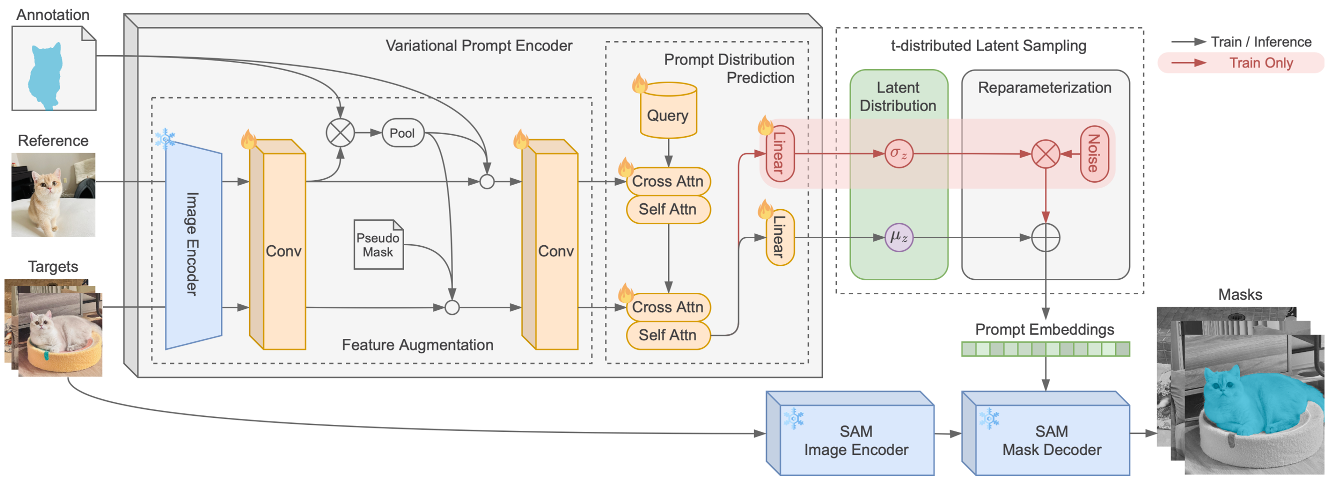

Figure 5. The detailed model architecture of ProSAM. The only trainable module in ProSAM is the variational prompt encoder, which is composed of two components: feature augmentation and prompt distribution prediction. Specifically, the feature augmentation aims to extract the enhanced reference and target feature to guide the learning of prompt distribution. The prompt distribution prediction module is responsible for predicting the variational prompt distribution to guide the SAM in mask generation for the target images.

9. Additional Verification Study

In addition to the studies presented in Section 5.3, we also designed a verification study that does not require training a deep learning model, allowing for a direct comparison between the underlying principles of the variational and non-variational prompt encoders.

Specifically, given the pre-trained SAM mask decoder and SAM image encoder, we learn the prompt embedding or variational prompt distribution via gradient descent for a given object (i.e., image-mask pair). To learn the prompt embedding via gradient descent, the gradient $\frac{\partial \mathcal{L}}{\partial \boldsymbol{z}}$ will be computed and used to directly update $\boldsymbol{z}$, which is treated as a parameterized vector rather than being predicted by the prompt encoder. Similarly, to learn the multivariate prompt distribution, the gradient $\frac{\partial \mathcal{L}}{\partial \hat{\boldsymbol{\mu}}_{\boldsymbol{z}}}$ and $\frac{\partial \mathcal{L}}{\partial \hat{\boldsymbol{\sigma}}_{\boldsymbol{z}}}$ will be utilized to update the parameterized vector $\hat{\boldsymbol{\mu}}_{\boldsymbol{z}}$ and $\hat{\boldsymbol{\sigma}}_{\boldsymbol{z}}$. Essentially, the learned prompt embedding reflects the fundamental principles of a non-variational prompt encoder (e.g., VRP-SAM), while the learned prompt distribution captures the core principles of a variational prompt encoder (e.g., ProSAM).

For the experiment of learning prompt embedding, we ran 200 experiments to generate 200 prompt embeddings. The initial prompt embeddings are randomly drawn from the normal distribution with a mean of 0 and a standard deviation of 25, and the loss function is formulated as the same loss function as VRP-SAM (see Equation 4). For the experiment of learning variational prompt distribution, we learn a single multivariate prompt distribution following the same formulation and reparameterization trick in Equation 5 and Equation 7 via gradient descent, given the loss function presented in Equation 10. For both experiments, the gradient descent optimization process is stopped when the loss value does not improve by more than 0.0003 over 100 consecutive epochs. For other experimental settings not mentioned above, we follow the same practice as in Section 5.1.

The visualization results on a sample image are presented in Figure 6. First, from Figure 6(b), we can see that both VRP-SAM and ProSAM are able to generate faithful prompts with IoU close to 0.96 and BCE close to zero. Notably, the prompt embeddings sampled from our variational prompt distribution consistently perform better than at least 75% of 200 prompt embeddings learned by VRP-SAM, with higher IoU and lower BCE value. From the scatter plot of projected prompt embeddings via t-SNE [34] in Figure 6(c), we can observe that the prompts sampled from our variational prompt distribution are clearly clustered in the center, while solely learning a single observation of prompt embeddings lie in the boundary of our variational prompt distribution. This observation assures that the proposed variational prompt encoder can indeed produce more robust prompts that are closer to the center of the target prompt region $\mathcal{R}_{I_r,M_r,I_t}$, compared with the non-variational prompt encoder employed by VRP-SAM.

10. Additional Quantitative Evaluations

To thoroughly assess the effectiveness of ProSAM, more quantitative evaluations have been conducted and presented here due to the page limit. First, we analyze our confusion matrix compared with VRP-SAM confusion matrix in Section 10.1 for a detailed comparison. Secondly, to conduct a fair comparison against VRP-SAM with the same number of parameters as ours, we present an ablation study on VRP-SAM with two linear layers appended in Section 10.2.

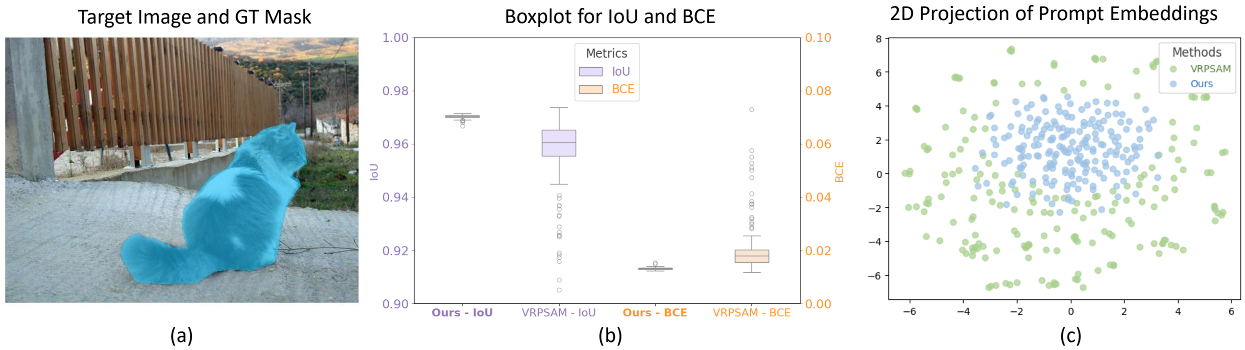

Figure 6. The visualization of the learned prompt embeddings by VRP-SAM and our method through gradient descent. For a sample image from COCO-20^i presented in Figure (a), we analyze the generated prompt embeddings and their associated mask predictions in Figure (b) and (c). Specifically, in Figure (b), the IoU (left y-axis) and BCE (right y-axis) are computed between the predicted masks and the ground-truth mask. In Figure (c), the 2D projection of prompt embeddings via t-SNE is visualized.

Additionally, an ablation study on different inference strategies has been conducted in Section 5.4. Lastly, we present more quantitative results on different choice of image encoder in Section 10.3. Again, similar to the results presented in Section 5, the experimental results presented here for both VRP-SAM and our method are conducted under identical experimental settings to ensure a fair comparison.

10.1. Confusion Matrix Comparison with VRP-SAM

As demonstrated in Section 5.2, we have outperformed the state-of-the-art method VRP-SAM on both COCO-20^i and PASCAL-5^i (see Table 1), and surpassed VRP-SAM under the significant domain shift from COCO-20^i to PASCAL-5^i (see Table 3). The question is whether ProSAM will potentially suffer from a greater false negative rate (FNR) while predicting a mean prompt that is more aligned with the center of target prompt region $\mathcal{R}_{I_r,M_r,I_t}$ (defined in Section 4.2). The detailed evaluations of ProSAM and VRP-SAM on True Positive Rate (TPR), True Negative Rate (TNR), False Positive Rate (FPR), and False Negative Rate (FNR) have been presented in Table 8. As you can see, for every fold in PASCAL-5^i, ProSAM can obtain a higher TPR and TNR in terms of pixel-level accuracy, while reducing the FPR and FNR systematically.

Table 8. A detailed comparison of ProSAM predictions with VRP-SAM predictions. At here, the True Positive Rate (TPR), True Negative Rate (TNR), False Positive Rate (FPR), and False Negative Rate (FNR) have been computed for both ProSAM and VRP-SAM predictions on PASCAL-5^i.

Methods	Metrics	PASCAL-5^i			
		F-0	F-1	F-2	F-3
ProSAM	TPR(%)	**78.47**	**68.97**	**66.97**	**66.37**
VRPSAM		78.14	68.21	65.88	64.39
ProSAM	TNR(%)	**11.55**	**19.91**	**17.82**	**16.74**
VRPSAM		11.54	19.89	17.22	16.67
ProSAM	FPR(%)	**9.21**	**9.5**	**12.75**	**14.13**
VRPSAM		9.53	10.25	13.84	16.12
ProSAM	FNR(%)	**0.78**	**1.62**	**2.46**	**2.75**
VRPSAM		**0.78**	1.64	3.05	2.82

10.2. VRP-SAM with Same Number of Parameters as ProSAM

As described in Section 5.1 and Section 8, the major difference in our model architecture compared with VRP-SAM is that we append two linear layers at the end of the variational prompt encoder to predict the mean and variance of the prompt distributions. Therefore, ProSAM has more learnable parameters resulting from these two linear layers. To conduct a fair comparison under the same number of learnable parameters, we trained a VRP-SAM with two linear layers appended at the end of their prompt encoder while keeping other experimental settings the same. From Table 9, we can see that appending two linear layers at the end of VRP-SAM prompt encoder fails to boost the VRP-SAM performance. In other words, with the same number of learnable parameters, ProSAM still surpasses VRP-SAM by a large margin.

Table 9. A quantitative comparison against VRP-SAM with the exactly same number of learnable parameters as ours. To be specific, 2 linear layers have been appended at the end of VRP-SAM prompt encoder to ensure identical model architecture as ours (see the second row below).

Methods	Metrics	PASCAL-5^i			
		F-0	F-1	F-2	F-3
ProSAM	mIOU(%)	**75.26**	**77.57**	**70.29**	**65.22**
VRPSAM+2Linear		74.04	76.55	69.71	63.98
VRPSAM		74.01	76.77	69.46	64.34

10.3. Choices of Image Encoder

In addition to ResNet-50 [7] and DINOv2 [26], we also experimented on adopting VGG-16 [31] as the image encoder. From Table 10, we can see that ProSAM with VGG-16 surpasses VRP-SAM with VGG-16 for all different folds. Also, for both VRP-SAM and ProSAM, the performance with VGG-16 generally performs worse than the performance with ResNet-50. This indicates that ResNet-50 can extract more accurate semantic-aware visual features and thereby enable ProSAM to learn better prompts.

Table 10. The quantitative evaluations of ProSAM with different image encoders such as ResNet-50 and VGG-16.

Methods	Image Encoder	PASCAL-5^i				
		F-0	F-1	F-2	F-3	Mean
VRP-SAM	ResNet-50	74.01	76.77	69.46	64.34	71.14
	VGG-16	69.72	74.74	67.12	61.84	68.35
ProSAM	ResNet-50	**75.26**	**77.57**	**70.09**	**65.22**	**72.04**
	VGG-16	70.53	75.30	68.25	62.99	69.27

11. Qualitative Evaluations

To qualitatively evaluate the effectiveness of ProSAM, we first present a qualitative comparison with VRP-SAM on COCO-20^i in Figure 7, then showcase the generalizability of ProSAM on diverse image styles in Figure 8 and lastly demonstrate our capability of handling challenging cases in Figure 9.

11.1. Qualitative Comparison with VRP-SAM

After a thorough qualitative analysis of masks generated by ProSAM and VRP-SAM across multiple datasets, we observed a general trend: our generated masks are less prone to artifacts, such as small holes or disconnected regions, which often appear in the masks produced by VRP-SAM.

For example, in Figure 7, VRP-SAM predictions on ”car” and ”banana” exhibit many small holes and pixelated artifacts in the masked region, whereas our predictions are consistently more robust with fewer pixelated artifacts. One key reason is that the mean prompts of our learned prompt distribution are more robust and precise than prompts predicted by VRP-SAM because our mean prompts are encouraged to be more closely aligned with the center of the target prompt region during the training. Thus, the masks generated by our mean prompts have higher quality. It is also interesting to see that VRP-SAM masks tend to have more false positives, which is consistent with our findings in Section 10.1. Taking ”car” and ”clock” in Figure 7 as examples: VRP-SAM wrongly perceives the road as ”car”; the entire spire is incorrectly predicted as ”clock” by VRP-SAM. However, by taking advantage of the robustness of our predicted mean prompts, our mask predictions on ”car” and ”clock” are accurate and precise with much fewer false positives. For ”fork”, VRP-SAM not only predicts more false positives but also wrongly treats other silverware (e.g., spoon and knife) as a ”fork”, while we generate a more accurate mask for ”fork” by leveraging a more optimal prompt encoder.

11.2. Generalizability on Diverse Image Styles

To evaluate the generalizability of ProSAM on images with novel and unseen styles, we conducted experiments on images featuring complex scenes and diverse styles. Specifically, both reference images and target images were collected from the internet, and the reference annotations were curated by prompting SAM with bounding boxes. As demonstrated in Figure 8, even though the model is trained on general-style images only (COCO-20^i), ProSAM can consistently generate high-quality masks with precise and clean boundaries, regardless of whether the target images are artistic paintings or photorealistic scenes. The ability to maintain such performance, even across vastly different image styles, is particularly impressive, as it requires no retraining or fine-tuning of the model. This strongly highlights the zero-shot segmentation capability of ProSAM in open-world scenarios.

11.3. Capability of Handling Challenging Cases

In image segmentation, certain challenging scenarios often cause segmentation methods to fall short. One such challenge arises when target objects have irregular shapes and non-uniform boundaries, which can lead to artifacts along object edges. Another common difficulty occurs when an image contains multiple target objects, as some objects may be overlooked, either receiving no masks or being assigned low-quality masks.

To better showcase our capability in understanding visual references and handling these challenges, we present qualitative results for these two scenarios in Figure 9. The visualization results demonstrate that ProSAM effectively generates high-quality masks even in the presence of complex shapes and multiple target objects. A key reason behind this strong performance is that our variational prompt encoder jointly learns multiple prompt distributions to guide SAM, enabling it to capture both non-uniform object boundaries and multiple objects within a scene. For example, in Figure 9, even though the motorcycle has an irregular boundary, our predictions accurately capture its complexity, producing a high-quality mask. Additionally, for the target images containing 15 goats, ProSAM successfully detects and segments all of them, demonstrating its robustness in handling multiple target objects.

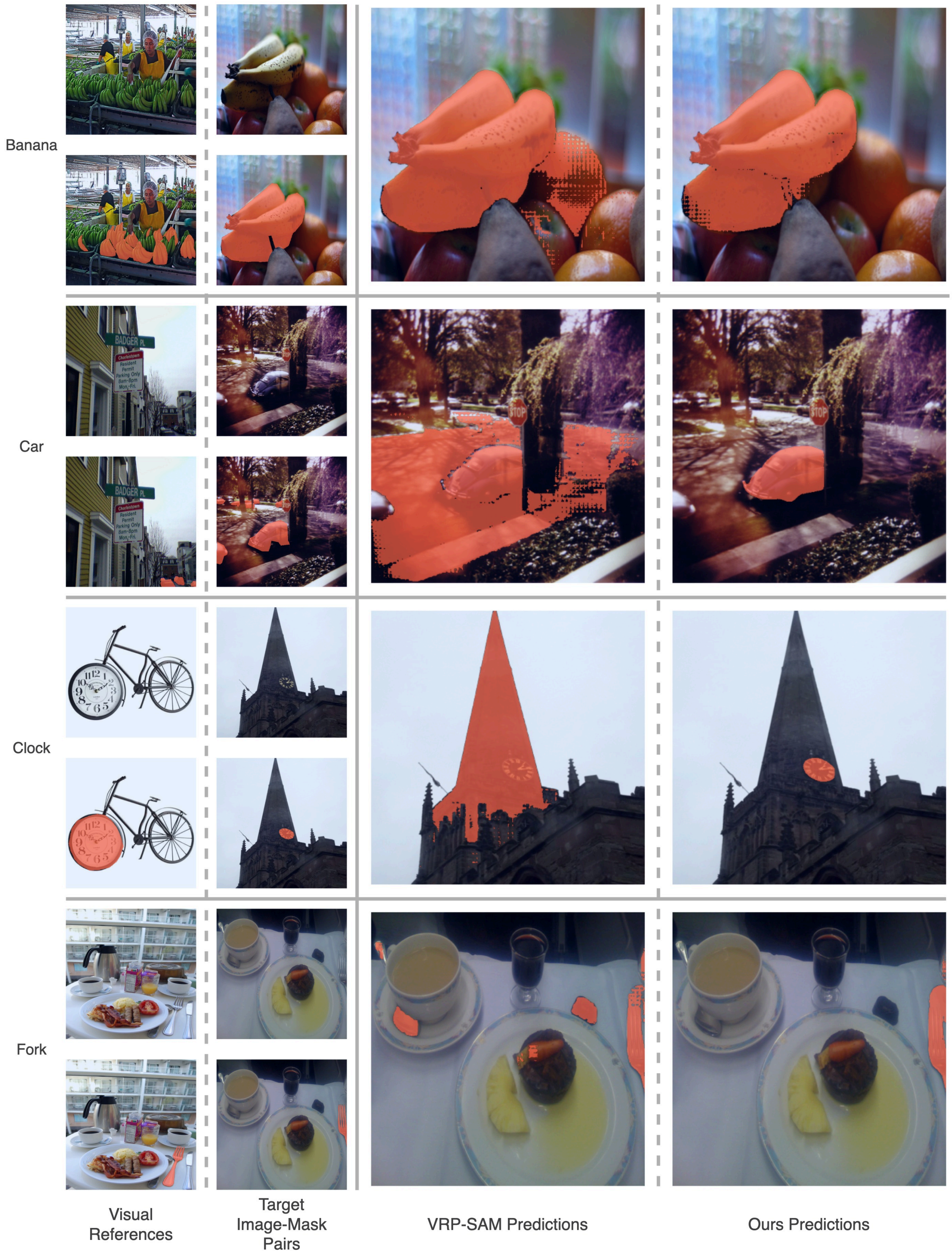

Figure 7. Qualitative comparison between VRP-SAM and ProSAM on COCO-20^i.

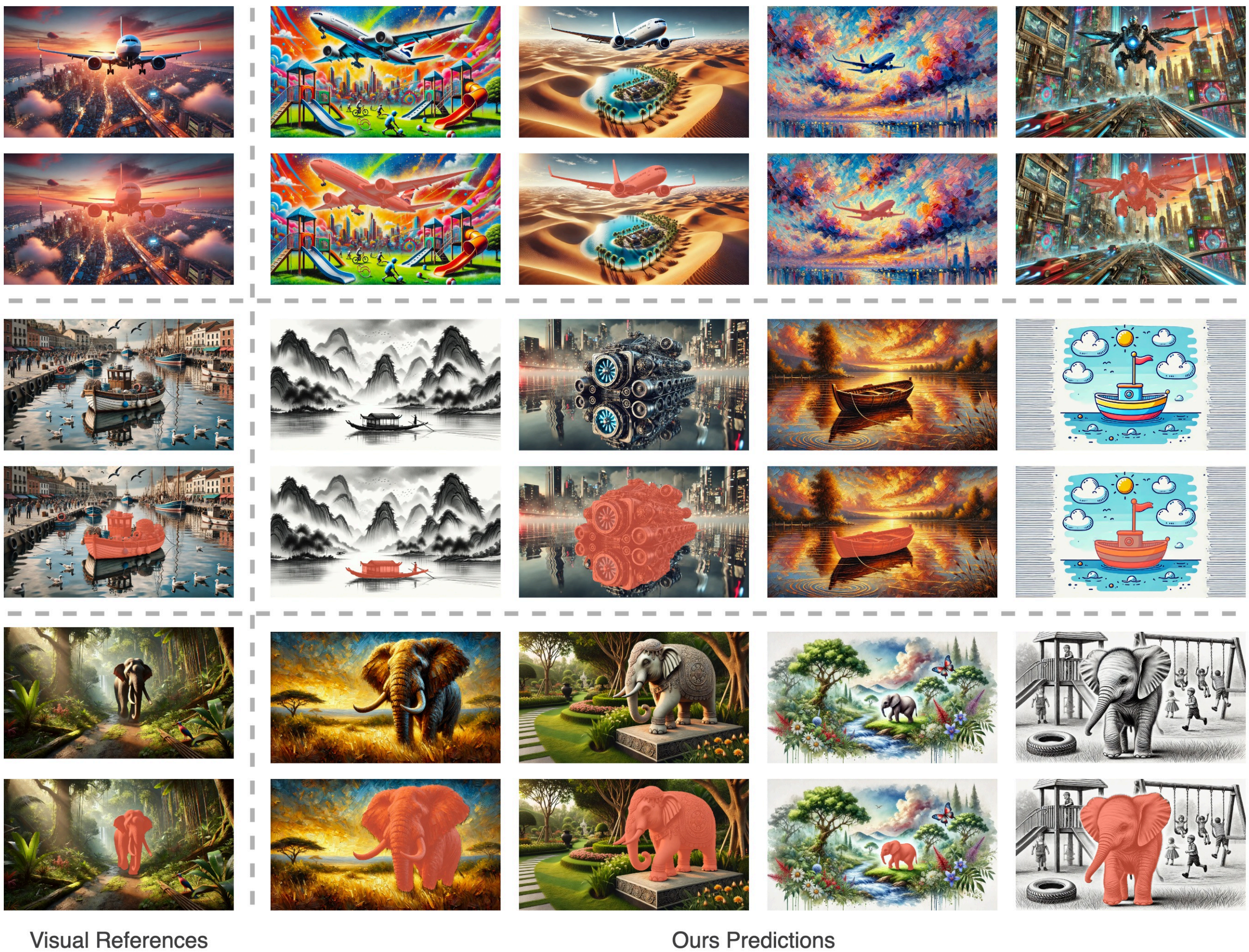

Figure 8. Qualitative results of ProSAM (trained on COCO-20^i) across diverse image styles. Both the reference images and target images were collected from the internet.

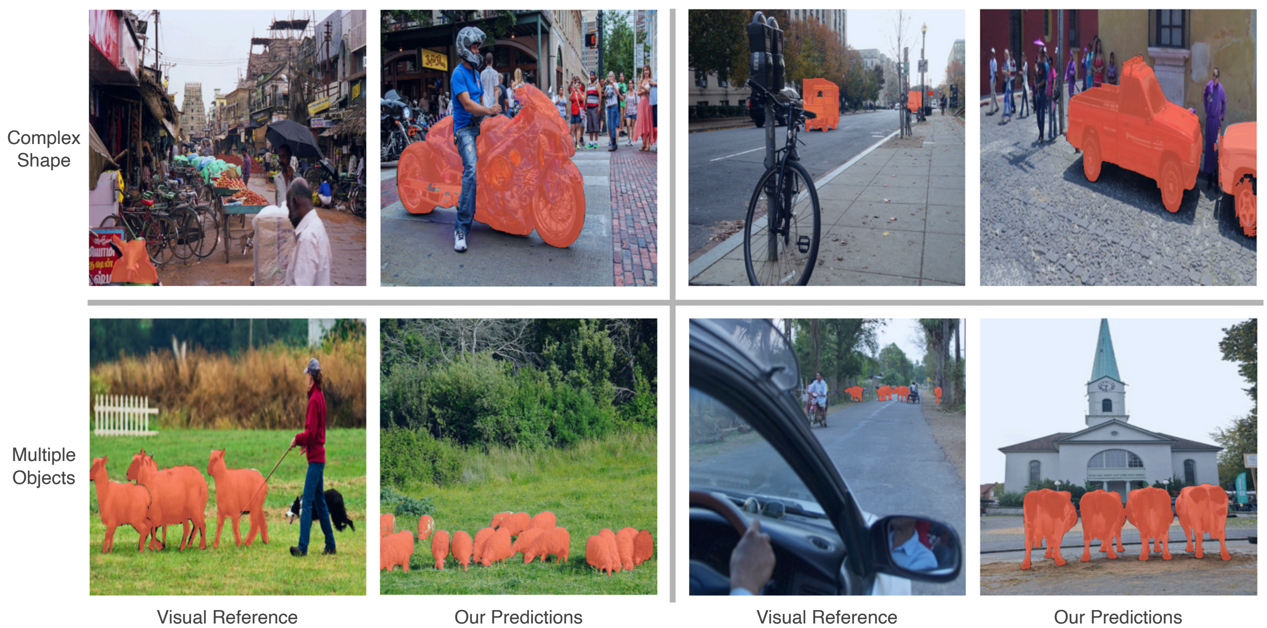

Figure 9. Qualitative results of ProSAM on two famous challenging cases including segmenting objects with irregular shape and segmenting multiple target objects.

References

[1] Najmeh Abiri and Mattias Ohlsson. Variational auto-encoders with student's t-prior. *arXiv preprint arXiv:2004.02581*, 2020. 5

[2] Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *Journal of machine learning research*, 18(153):1–43, 2018. 4

[3] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017. 2

[4] Malik Boudiaf, Hoel Kervadec, Ziko Imtiaz Masud, Pablo Piantanida, Ismail Ben Ayed, and Jose Dolz. Few-shot segmentation without meta-learning: A good transductive inference is all you need? In *Proc. CVPR*, pages 13979–13988, 2021. 7

[5] Peng Ding. On the conditional distribution of the multivariate t distribution. *The American Statistician*, 70(3):293–295, 2016. 4

[6] Qi Fan, Wenjie Pei, Yu-Wing Tai, and Chi-Keung Tang. Self-support few-shot semantic segmentation. In *Proc. ECCV*, pages 701–719, 2022. 6

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, pages 770–778, 2016. 3, 6

[8] Wenbin He, Suphanut Jamonnak, Liang Gou, and Liu Ren. CLIP-S4: Language-guided self-supervised semantic segmentation. In *Proc. CVPR*, pages 11207–11216, 2023. 1

[9] Sunghwan Hong, Seokju Cho, Jisu Nam, Stephen Lin, and Seungryong Kim. Cost aggregation with 4d convolutional swin transformer for few-shot segmentation. In *Proc. ECCV*, pages 108–126, 2022. 6

[10] Qing Jiang, Feng Li, Zhaoyang Zeng, Tianhe Ren, Shilong Liu, and Lei Zhang. T-Rex2: Towards generic object detection via text-visual prompt synergy. In *Proc. ECCV*, pages 38–57, 2024. 1, 2

[11] Joakim Johnander, Johan Edstedt, Michael Felsberg, Fahad Shahbaz Khan, and Martin Danelljan. Dense Gaussian processes for few-shot segmentation. In *Proc. ECCV*, pages 217–234. Springer, 2022. 7

[12] Juno Kim, Jaehyuk Kwon, Mincheol Cho, Hyunjong Lee, and Joong-Ho Won. t^3-variational autoencoder: Learning heavy-tailed data with student's t and power divergence. In *Proc. ICLR*, 2024. 5

[13] Diederik P Kingma. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2, 8

[14] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proc. ICCV*, pages 4015–4026, 2023. 1

[15] Chunbo Lang, Gong Cheng, Binfei Tu, and Junwei Han. Learning what not to segment: A new perspective on few-shot segmentation. In *Proc. CVPR*, pages 8057–8067, 2022. 6

[16] Chunbo Lang, Gong Cheng, Binfei Tu, and Junwei Han. Learning what not to segment: A new perspective on few-shot segmentation. In *Proc. CVPR*, pages 8057–8067, 2022. 5

[17] Xudong Lin, Yueqi Duan, Qiyuan Dong, Jiwen Lu, and Jie Zhou. Deep variational metric learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 689–704, 2018. 2

[18] Jie Liu, Yanqi Bao, Guo-Sen Xie, Huan Xiong, Jan-Jakob Sonke, and Efstratios Gavves. Dynamic prototype convolution network for few-shot semantic segmentation. In *Proc. CVPR*, pages 11553–11562, 2022. 6

[19] Yuanwei Liu, Nian Liu, Qinglong Cao, Xiwen Yao, Junwei Han, and Ling Shao. Learning non-target knowledge for few-shot semantic segmentation. In *Proc. CVPR*, pages 11573–11582, 2022. 6

[20] Yang Liu, Muzhi Zhu, Hengtao Li, Hao Chen, Xinlong Wang, and Chunhua Shen. Matcher: Segment anything with one shot using all-purpose feature matching. *arXiv preprint arXiv:2305.13310*, 2023. 1, 2, 5, 6

[21] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[22] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 6

[23] Juhong Min, Dahyun Kang, and Minsu Cho. Hypercorrelation squeeze for few-shot segmentation. In *Proc. ICCV*, pages 6941–6952, 2021. 6, 7

[24] Seonghyeon Moon, Samuel S Sohn, Honglu Zhou, Sejong Yoon, Vladimir Pavlovic, Muhammad Haris Khan, and Mubbasir Kapadia. HM: Hybrid masking for few-shot segmentation. In *Proc. ECCV*, pages 506–523, 2022. 7

[25] Khoi Nguyen and Sinisa Todorovic. Feature weighting and boosting for few-shot segmentation. In *Proc. ICCV*, pages 622–631, 2019. 5

[26] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 8, 6

[27] Bohao Peng, Zhuotao Tian, Xiaoyang Wu, Chengyao Wang, Shu Liu, Jingyong Su, and Jiaya Jia. Hierarchical dense correlation distillation for few-shot segmentation. In *Proc. CVPR*, pages 23641–23651, 2023. 6

[28] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. SAM 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 1

[29] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. *arXiv preprint arXiv:1709.03410*, 2017. 5

[30] Xinyu Shi, Dong Wei, Yu Zhang, Donghuan Lu, Munan Ning, Jiashun Chen, Kai Ma, and Yefeng Zheng. Dense

cross-query-and-support attention weighted mask aggregation for few-shot segmentation. In *Proc. ECCV*, pages 151–168, 2022. 6, 7

[31] Karen Simonyan. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6

[32] Yanpeng Sun, Jiahui Chen, Shan Zhang, Xinyu Zhang, Qiang Chen, Gang Zhang, Errui Ding, Jingdong Wang, and Zechao Li. VRP-SAM: Sam with visual reference prompt. In *Proc. CVPR*, pages 23565–23574, 2024. 1, 2, 3, 5, 6, 7

[33] Zhuotao Tian, Hengshuang Zhao, Michelle Shu, Zhicheng Yang, Ruiyu Li, and Jiaya Jia. Prior guided feature enrichment network for few-shot segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(2):1050–1065, 2020. 6, 7

[34] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9 (11), 2008. 4

[35] Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. In *Proc. CVPR*, pages 6830–6839, 2023. 5, 6

[36] Xinlong Wang, Xiaosong Zhang, Yue Cao, Wen Wang, Chunhua Shen, and Tiejun Huang. SegGPT: Towards segmenting everything in context. In *Proc. ICCV*, pages 1130–1140, 2023. 5, 6

[37] Xiaoqi Wang, Wenbin He, Xiwei Xuan, Clint Sebastian, Jorge Piazentin Ono, Xin Li, Sima Behpour, Thang Doan, Liang Gou, Han-Wei Shen, and Liu Ren. USE: Universal segment embeddings for open-vocabulary image segmentation. In *Proc. CVPR*, pages 4187–4196, 2024. 1

[38] Haoyu Xie, Changqi Wang, Mingkai Zheng, Minjing Dong, Shan You, Chong Fu, and Chang Xu. Boosting semi-supervised semantic segmentation with probabilistic representations. In *Proceedings of the AAAI conference on artificial intelligence*, pages 2938–2946, 2023. 2

[39] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side adapter network for open-vocabulary semantic segmentation. In *Proc. CVPR*, pages 2945–2954, 2023. 1

[40] Boyu Yang, Chang Liu, Bohao Li, Jianbin Jiao, and Qixiang Ye. Prototype mixture models for few-shot semantic segmentation. In *Proc. ECCV*, pages 763–778, 2020. 7

[41] Tianyuan Yu, Da Li, Yongxin Yang, Timothy M Hospedales, and Tao Xiang. Robust person re-identification by modelling feature uncertainty. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 552–561, 2019. 2

[42] Gengwei Zhang, Guoliang Kang, Yi Yang, and Yunchao Wei. Few-shot segmentation via cycle-consistent transformer. *Proc. NeurIPS*, 34:21984–21996, 2021. 6

[43] Gengwei Zhang, Guoliang Kang, Yi Yang, and Yunchao Wei. Few-shot segmentation via cycle-consistent transformer. *Proc. NeurIPS*, 34:21984–21996, 2021. 7

[44] Gengwei Zhang, Guoliang Kang, Yi Yang, and Yunchao Wei. Few-shot segmentation via cycle-consistent transformer. *Proc. NeurIPS*, 34:21984–21996, 2021. 5

[45] Renrui Zhang, Zhengkai Jiang, Ziyu Guo, Shilin Yan, Junting Pan, Xianzheng Ma, Hao Dong, Peng Gao, and Hongsheng Li. Personalize segment anything model with one shot. *arXiv preprint arXiv:2305.03048*, 2023. 1, 2, 5, 6

[46] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *Proc. NeurIPS*, 36, 2024. 6