From Adaptation to Generalization: Adaptive Visual Prompting for Medical Image Segmentation

A. Implementation Details

A.1. Experimental Environment

All experiments were conducted using a single NVIDIA GeForce RTX 3090 GPU (24GB VRAM). Models were implemented in PyTorch. Publicly available datasets were used for evaluation.

A.2. Data Preprocessing

A.2.1. Image Processing

All images were resized to $352\times352\times3$ for consistency across models, except for SwinUNet, which was resized to $256\times256\times3$ due to architectural constraints. For polyp segmentation, images were normalized using ImageNet statistics, with mean $[0.485, 0.456, 0.406]$ and standard deviation $[0.229, 0.224, 0.225]$. For OC/OD segmentation, min-max normalization was applied.

A.2.2. Domain Augmentation

To simulate domain shifts, we employed Random Amplitude Spectrum Synthesis (RASS)[2], which perturbs the FFT amplitude spectrum of input images. Given an original amplitude $A(u, v)$, the augmented amplitude is defined as:

$$A^{\text{aug}}(u, v) = \delta(u, v) \cdot A(u, v), \quad \delta(u, v) \sim \mathcal{N}(1, \sigma^2(u, v)), \tag{1}$$

where the perturbation strength $\sigma(u, v)$ is:

$$\sigma(u, v) = (2\alpha\sqrt{(u^2 + v^2)/(H^2 + W^2)})^{\gamma}. \tag{2}$$

We use $(\alpha, \beta, \gamma) \in (3, 0.25, 6), (10, 0.5, 1)$ to define multiple synthetic domains, as suggested in[2].

A.2.3. APEX Training

Backbone models were trained using their official implementations. After acceptance, we will release the pretrained weights for all models. For training the APEX, we use a prompt memory with $J = 150$ slots, each storing a 24-dimensional learnable prompt vector. The domain feature encoder consists of four MLP layers with dimensions $[72, 72, 36, 24]$, and the prompt decoder consists of layers $[36, 72, 72, 108]$. The prompt is defined with spatial dimensions $6\times6\times3$ and is embedded into the low-frequency region of the FFT amplitude.

Hyperparameters	Values
Memory Slot Size (J)	150
Memory Feature Size (K)	24
Domain Feature Encoder Layers	[72, 72, 36, 24]
Prompt Decoder Layers	[36, 72, 72, 108]
Prompt Dimensions	$6 \times 6 \times 3$ (=108)
Memory Temperature	1
LFC Temperature	0.7
Learning Rate	1e-3
Batch Size	16 (4 for DUCK-Net)
Weight Decay	1e-2

Table 1. Hyperparameters for optimizing APEX.

We set the temperature parameter for soft addressing in memory to 1.0 to control the sharpness of slot selection. A lower temperature leads to more selective slot usage, which can help prevent uniform optimization across all slots. The LFC loss weight λ_{LFC} is set to 0.7.

We use a batch size of 16 for all models except DUCK-Net, which requires a smaller batch size of 4 due to higher GPU memory usage. Optimization is performed using AdamW [1] with a learning rate of 1e-3 and a weight decay of 1e-2. A summary of key hyperparameters is provided in Table 1.

B. Computational Cost

B.1. Parameter Efficiency

The proposed APEX module introduces only **0.036M** additional parameters, encompassing the domain feature encoder, prompt memory, and prompt decoder. In comparison, the backbone models require substantially larger capacities: PraNet (32.5M), UNet (34.5M), ResUNet (22.5M), TransUNet (105.4M), SwinUNet (27.1M), and DUCK-Net (152.0M) parameters.

This highlights the remarkable parameter efficiency of APEX, introducing only a marginal overhead, approximately **1,000 times fewer parameters** than typical back-

Task	Method	Mean Difference ± Standard Error	95%CI	p-value
OC/OD	VPT	12.09±0.84	[10.42, 13.76]	p<0.0001
	FVP	12.10±0.84	[10.43, 13.77]	p<0.0001
	A2XP	8.23±0.63	[6.99, 9.48]	p<0.0001
	VPAD	9.14±0.71	[7.74, 10.53]	p<0.0001
Polyp	VPT	2.35±0.82	[0.74, 3.97]	p<0.005
	FVP	2.07±0.83	[0.43, 3.71]	p<0.01
	A2XP	2.35±0.82	[0.74, 3.97]	p<0.005
	VPAD	2.41±0.77	[0.95, 3.98]	p<0.005

Table 2. Statistical significance analysis of performance improvements by paired t-test.

bone models. Despite its lightweight design, APEX significantly enhances model generalization, demonstrating that domain-adaptive prompting can be achieved with minimal computational burden.

B.2. Inference Time

APEX introduces minimal overhead during inference, making it highly suitable for deployment in real-world, resource-constrained medical scenarios. Since the adaptive prompting mechanism requires only a few additional forward passes through lightweight MLP layers, the impact on latency is negligible. For instance, using U-Net as the backbone, the baseline inference time is 25ms on GPU and 744ms on CPU. With APEX, the total inference time increases marginally to 30ms on GPU and 764ms on CPU—corresponding to only a 5ms and 20ms overhead, respectively.

C. Statistical Analysis

To validate the effectiveness of our method, we perform a paired t-test to assess whether the performance improvements are statistically significant. This test quantifies the consistency and reliability of the observed gains across different settings. We compare the segmentation performance of our method against four representative visual prompting baselines: VPT, FVP, A2XP, and VPAD.

The statistical results are summarized in Table 2. As indicated, our method achieves significantly better performance, with p-values consistently below 0.01. These results confirm that the improvements introduced by our framework are not only consistent but also statistically significant, reinforcing the robustness and generalizability of our approach across diverse domains.

D. More Visualization

Figure 1 and 2 shows the qualitative comparison on OC/OD and polyp segmentation tasks.

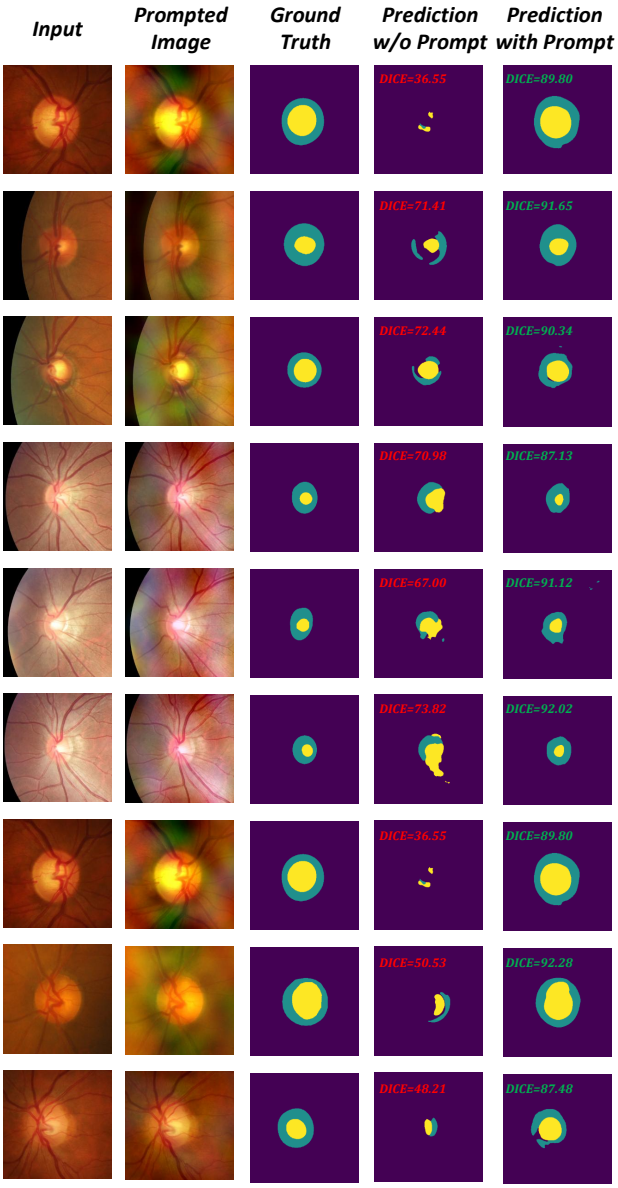

Figure 1. Qualitative comparison between before and after prompting on OC/OD segmentation task.

References

[1] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 1

[2] Qiang Qiao, Wenyu Wang, Meixia Qu, Kun Su, Bin Jiang, and Qiang Guo. Medical image segmentation via single-source domain generalization with random amplitude spectrum synthesis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 435–445. Springer, 2024. 1

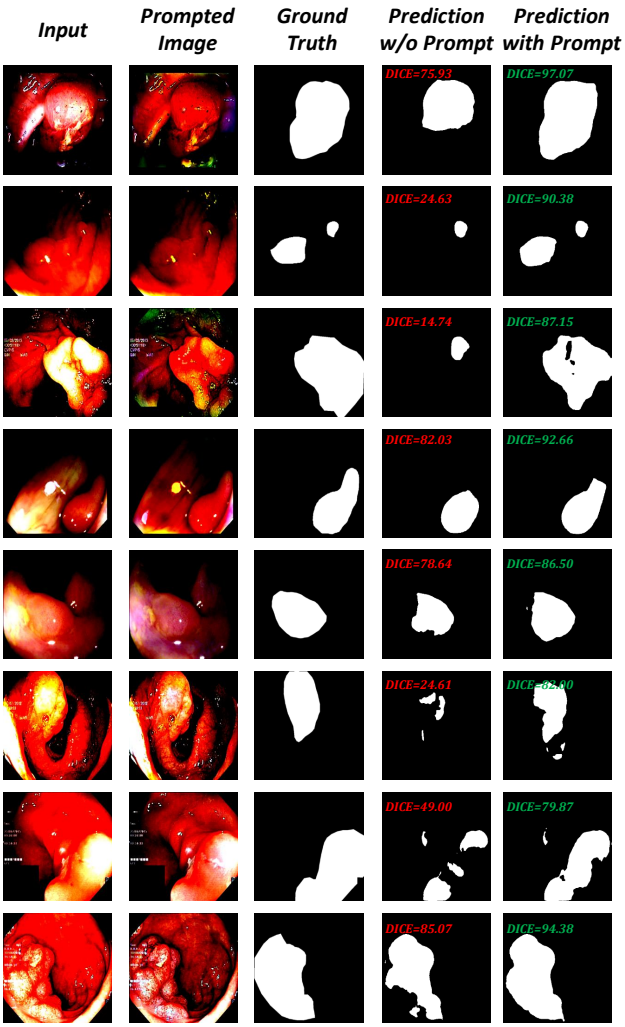

Figure 2. Qualitative comparison between before and after prompting on polyp segmentation task. *In the case of polyp images, we visualized normalized images since we prompt into the normalized images with ImageNet statistics.