

# Rethinking Token Reduction with Parameter-Efficient Fine-Tuning in ViT for Pixel-Level Tasks

Cheng Lei, Ao Li, Hu Yao, Ce Zhu, Le Zhang\*

University of Electronic Science and Technology of China

{202321011612, aoli, 202322011614}@std.uestc.edu.cn, {eczhu, lezhang}@uestc.edu.cn

## Abstract

*Parameter-efficient fine-tuning (PEFT) adapts pre-trained models to new tasks by updating only a small subset of parameters, achieving efficiency but still facing significant inference costs driven by input token length. This challenge is even more pronounced in pixel-level tasks, which require longer input sequences compared to image-level tasks. Although token reduction (TR) techniques can help reduce computational demands, they often lead to homogeneous attention patterns that compromise performance in pixel-level scenarios. This study underscores the importance of maintaining attention diversity for these tasks and proposes to enhance attention diversity while ensuring the completeness of token sequences. Our approach effectively reduces the number of tokens processed within transformer blocks, improving computational efficiency without sacrificing performance on several pixel-level tasks. We also demonstrate the superior generalization capability of our proposed method compared to challenging baseline models. The source code will be made available at <https://github.com/AVC2-UESTC/DAR-TR-PEFT>.*

## 1. Introduction

Recent advancements in visual pre-trained foundation models [2, 15, 16, 44, 67] have gained widespread popularity due to their exceptional generalization capabilities across a broad spectrum of tasks. Building on this success, parameter-efficient fine-tuning (PEFT) techniques [22, 23, 26] have further enhanced the practicality of these models by enabling efficient adaptation to various downstream tasks and data domains, requiring updates to only a minimal number of parameters. Despite the efficiency that PEFT methods bring to training, they still encounter significant computational challenges during inference. For image-level tasks like classification, these challenges can be alleviated using token reduction (TR) techniques [1, 3, 30, 32, 34, 49,

65]. In image classification, a class token is used, enabling the reduction of patch tokens without negatively impacting prediction accuracy, as fewer and more abstract tokens are sufficient to capture the overall content of the image.

At first glance, pixel-wise tasks such as segmentation [27, 38, 60] appear well-suited to benefit from token reduction (TR) due to their requirement for longer token sequences to capture fine-grained, high-resolution details. However, this very need introduces a significant trade-off between efficiency and accuracy: reducing the number of tokens can compromise the model’s ability to produce precise and detailed predictions. Although prior studies on dynamic transformers [32, 33, 49, 63] have shown that TR effectively reduces inference costs for image-level tasks, extending these techniques to pixel-level tasks remains challenging. This is partly because pixel-wise tasks are highly sensitive to spatial resolution and coherence. Specifically, reduced token sequences often lack a spatially coherent representation, which is problematic for pixel decoders that require tokens with a complete spatial structure to function effectively. Additionally, segmentation tasks demand maintaining semantic and boundary details across all tokens, and any loss of fine-grained information can degrade performance. Therefore, balancing token reduction while preserving both spatial structure and high-resolution details remains an unresolved challenge.

Moreover, the diversity of features captured by the attention mechanism is crucial for the success of pixel-level tasks. Namuk *et al.* [47] and Xie *et al.* [61] emphasize that a diverse set of attention heads allows the model to focus on various aspects of the input, facilitating the learning of rich and comprehensive representations. This diversity is essential for accurately capturing fine-grained details and complex patterns, which are critical for tasks such as segmentation and depth estimation. Without sufficient diversity, models may struggle to differentiate subtle features, ultimately leading to a decline in performance for pixel-level applications. We further examine existing TR methods from the perspective of diversity and discuss our findings below.

In Fig. 1 (a), we present the normalized mean atten-

---

\*Corresponding author.

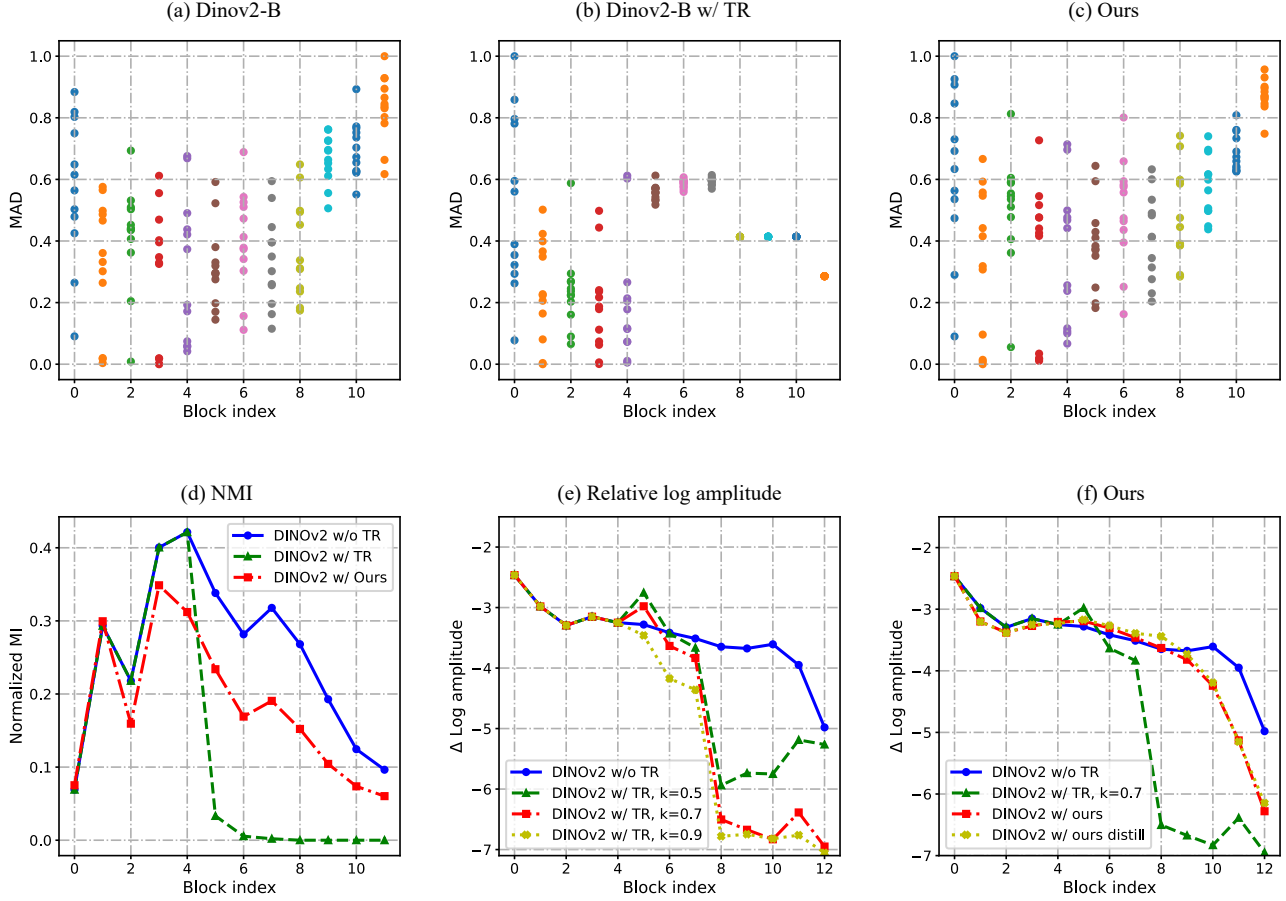


Figure 1. Comparison of the normalized mean attention distance (MAD) for (a) DINOv2 without token reduction (TR), (b) DINOv2 with TR, and (c) DINOv2 using our proposed method. Each point within a column corresponds to a specific attention head, with the vertical axis representing the normalized MAD. Higher values indicate a larger receptive field, while greater separation between points reflects increased diversity among attention heads. In (d), a lower value of normalized mutual information (MI) suggests a weaker dependence of the attention map on query tokens, implying that the attention distribution collapses into homogeneity, thereby reducing diversity. Notably, the normalized MI of DINOv2 with TR is significantly lower than that of DINOv2 without TR, especially after the 5<sup>th</sup> transformer block. Furthermore, the curves for different token activation rates of 0.5, 0.7, and 0.9 exhibit consistent patterns. In (e) and (f), a higher value of  $\Delta$  Log amplitude indicates that the representations leverage high-frequency components. DINOv2 with TR begins to show a reduction in high-frequency components after the 5<sup>th</sup> transformer block, whereas DINOv2 without TR only exhibits this reduction starting from the 10<sup>th</sup> block. Details of the procedure are provided in the supplementary materials.

tion distance (MAD) for DINOv2 [44]. The figure illustrates that nearly all transformer blocks exhibit diverse attention distributions, as reflected by the varying normalized MAD values across different attention heads in each column. However, when a fixed token reduction strategy [32, 34] is applied, this diversity significantly diminishes. As shown in Fig. 1 (b), the attention heads in the later transformer blocks tend to have nearly identical normalized MAD values, indicating a loss of variability.

To further quantify attention diversity, we use normalized mutual information (NMI) [53] in Fig. 1 (d), comparing the diversity of attention heads across transformer blocks. The green curve reveals that the NMI of DINOv2 with TR drops rapidly and remains much lower than that

of the model without TR. This sharp decline indicates that TR causes the attention mechanisms to collapse into homogeneous distributions. While such uniformity may not noticeably harm image classification performance, it can be detrimental to pixel-level tasks [47]. Such homogenization of attention patterns reduces the model’s expressive power, particularly undermining its performance in tasks that require a nuanced understanding of complex visual details.

Given that fixed TR leads to homogeneous attention patterns, we hypothesize that it may also cause a loss of high-frequency details. To explore this, we conduct a Fourier analysis following the methods outlined in [46, 47]. As shown in Fig. 1 (e), we present the relative log amplitude of Fourier-transformed representations, computed as the dif-

ference in amplitude between the highest and lowest frequencies. The analysis reveals that DINOv2 without TR retains more high-frequency components in the later transformer blocks compared to DINOv2 with TR. Notably, we observe that a higher token keep rate within the TR configuration is associated with a significant reduction in high-frequency components in the final transformer blocks. This counterintuitive result may be due to the increased uniformity among longer tokens at higher keep rates. Fixed TR inherently leads to the loss of high-frequency details, which are crucial for pixel-level tasks. High-frequency components capture important information about edges, textures, and subtle variations within an image. By reducing the number of tokens, the model risks discarding these features, resulting in diminished fidelity and performance.

We contend that the loss of high-frequency information can significantly degrade the performance of pixel-level tasks, which depend heavily on these components to capture essential fine-grained details such as edges, boundaries, and small-scale textures required for accurate predictions. When high-frequency details are diminished, the model’s ability to distinguish between different regions or objects is compromised, leading to blurred or imprecise outputs and ultimately reducing overall task performance.

Motivated by this phenomenon, in this paper, we seek to deepen the understanding of how TR affects attention mechanisms and visual representations, pinpointing the barriers and challenges that complicate the use of TR in PEFT for pixel-level tasks. We propose Diverse Attention Restorer (**DAR**) that preserves attention diversity, retains the high-frequency components of visual representations, and maintains the completeness of the token sequence, all while reducing the number of tokens processed within transformer blocks. By striking an optimal balance between performance and inference efficiency for pixel-level tasks, our method enhances the practical applicability of PEFT techniques in this crucial area. To summarize, we make the following contributions:

- We systematically analyze the limitations of current token reduction (TR) techniques and identify that they often lead to a loss of attention diversity and high-frequency information representation, which we believe are critical for pixel-level tasks.
- Based on these findings, we propose a decoupled design of the Token Compensator and PEFT Module. We further develop a learnable mask mechanism to address the perturbation issue caused by the FFN bias, enhancing the effectiveness of token reduction without compromising performance.
- Our method demonstrates outstanding performance across several pixel-level tasks with minimal FLOPs. Additionally, we showcase the superior generalization capability of our approach in zero-shot settings compared to

strong baseline models, highlighting its practical efficacy and efficiency.

## 2. Related Work

### 2.1. Pixel-Level Tasks

Pixel-level tasks involve classifying or detecting each pixel within an image, as seen in applications like segmentation [27]. In real-world scenarios, many critical applications rely on precise pixel-level analysis for accurate detection and interpretation. For example, in the medical domain, accurately segmenting pathological regions is essential. Tasks such as polyp segmentation [25] and skin lesion segmentation [6] are crucial for providing clinicians with reliable delineations to aid in effective diagnosis.

Additionally, salient object segmentation and camouflaged object segmentation [13, 14] play a vital role in computer vision by enhancing visual perception, enabling the accurate separation of relevant objects from complex backgrounds. In the field of remote sensing [17, 56], high-precision detection is required for the effective analysis and interpretation of satellite imagery, demonstrating the broad and impactful applications of pixel-level tasks.

### 2.2. Token Reduction

Token reduction has become a widely adopted strategy in Vision Transformers (ViTs) to enhance inference efficiency. Several methods have incorporated token reduction into the full training process of ViTs, such as EViT [32], DynamicViT [49], and DiffRate [3]. In addition, recent studies have leveraged token reduction to enable parameter-efficient fine-tuning of ViT models. Prominent examples include CoDA [30], Dynamic-Tuning [65], and Sparse-Tuning [34]. Moreover, ToMe [1] introduces a training-free approach designed to improve the inference efficiency of ViTs.

### 2.3. Parameter-Efficient Fine-Tuning (PEFT)

PEFT techniques aim to minimize computational overhead by fine-tuning a small subset of model parameters while keeping most of the parameters unchanged. One notable method is LoRA [23], initially popularized for language tasks and has recently gained attention in vision applications. VPT [26] is a simple yet effective approach for fine-tuning visual models by adding learnable prompt tokens to the input sequence. Another approach, the adapter method, introduces small, learnable modules between transformer layers [18]. A serial adapter strategy has been used effectively for language tasks, as demonstrated by the NLP Adapter [22, 48], while AdaptFormer [4] employs a parallel strategy optimized for vision tasks. More recently, EVP [35, 59] has emerged, combining the parameter efficiency of VPT with the robustness of adapters and incorporating

high-frequency priors to further enhance performance.

### 3. Methodology

#### 3.1. Facilitating TR-PEFT for Pixel-level Tasks

Let  $\mathbf{X}^i$  be the input tokens to the  $i^{\text{th}}$  transformer block with TR. This process can be formulated as follows:

$$\mathbf{X}^{i+1} = \mathbf{X}^i + \mathcal{M}(\text{TR}(\mathbf{X}^i)), \quad (1)$$

where  $\mathcal{M}$  represents the entire transformer block or any of its internal modules, such as the self-attention mechanism or feed-forward networks (FFN). While the token reduction process effectively reduces the number of tokens, the residual connection [10, 19] is designed to mitigate the loss of information and structural integrity. However, the inherent limitations of the residual connection can also lead to inadequate feature aggregation, which is critical for pixel-level tasks. To overcome this limitation, there is a strong need for an additional token compensator that can preserve spatial structure and further enhance the representation of the reduced token sequence.

**Adapter as a token compensator.** In CoDA [30] and Dynamic Tuning [65], the authors introduce an adapter [4, 22] in parallel with the token reduction module. Thus, Eq. (1) can be rewritten as:

$$\mathbf{X}^{i+1} = \mathbf{X}^i + \mathcal{M}(\text{TR}(\mathbf{X}^i)) + \text{Adapter}(\mathbf{X}^i), \quad (2)$$

We posit that this adapter serves a dual function: it not only facilitates the learning of new information from the incoming data but also plays a crucial role in preserving the spatial structure inherent in the original, unreduced tokens. Thus, the adapter can be viewed as a composition of these two essential functions.

$$\text{Adapter} = \mathcal{F} \circ \mathcal{G}, \quad (3)$$

where  $\mathcal{F}$  represents the function that aims to learn new knowledge, potentially realized through a PEFT module, and  $\mathcal{G}$  denotes the token compensator, which is designed to retain the spatial integrity of the unreduced tokens while aggregating information that may be advantageous for pixel-level tasks.

**Decoupled design of token compensator and PEFT module.** Usually, visual tokens are represented as the concatenation of class tokens, register tokens [8], and patch tokens.

$$\mathbf{X}^i = \text{concat}[\mathbf{E}^i, \mathbf{P}^i], \quad (4)$$

where  $\mathbf{E}^i \in \mathbb{R}^{l \times d}$  denotes class and register tokens with length  $l$  and embedding dimension  $d$ , and  $\mathbf{P}^i \in \mathbb{R}^{p \times d}$  is patch tokens derived from the images with length  $p$ .  $\mathcal{F}$  receives the entire token sequence  $\mathbf{X}^i$  to learn new knowledge, while  $\mathcal{G}$  processes only the patch tokens  $\mathbf{P}^i$  to preserve spatial structure. We believe that separating the design

of the PEFT module and the token compensator provides greater flexibility and efficiency for model adaptation. This can be formulated as:

$$\mathbf{X}^{i+1} = \mathbf{X}^i + \mathcal{M}(\text{TR}(\mathbf{X}^i)) + \mathcal{F}(\mathbf{X}^i) + \mathcal{G}(\mathbf{P}^i), \quad (5)$$

The rationale behind the decoupled design of the PEFT module and the token compensator lies in the need to address distinct requirements for model adaptation efficiently. By decoupling these components, the PEFT module can be optimized specifically for learning new knowledge from incoming data, focusing on parameter efficiency and minimizing computational overhead. Meanwhile, the token compensator can be independently designed to preserve the spatial structure and integrity of the original tokens, ensuring that crucial details necessary for tasks like segmentation or depth estimation are maintained. Additionally, this architectural separation facilitates the application of the token compensator across a diverse range of PEFT methodologies, thereby promoting generalizability beyond the confines of adapter tuning [4, 22].

#### 3.2. Our Method

Building on the preceding discussion, we propose the implementation of a token compensator that functions independently from adapters or other PEFT modules. This design is specifically intended to preserve the spatial structure of the tokens. Furthermore, we introduce the idea of using a learnable mask as the token reduction strategy. This adaptive mechanism enables dynamic token selection, allowing the model to adjust which tokens are activated during processing based on the input’s characteristics. Together, the independent token compensator and the learnable mask are designed to restore attention diversity and enhance the high-frequency components within visual representations, ultimately improving performance on pixel-level tasks.

**Strategy for reducing tokens.** A simple strategy for reducing tokens involves using a learnable mask  $\mathbf{M}^i$  to index the complete token sequence  $\mathbf{X}^i$ . However, in a given batch, each image token sequence  $\mathbf{X}^i$  can generate a different mask  $\mathbf{M}^i$ , resulting in varying numbers of active tokens. This variability leads to different lengths  $p$  for  $\mathbf{X}^i$  within a single batch. While this approach is acceptable during inference, it hinders the use of parallel computation during training. To address this issue, the authors in [49] directly sparsify the token sequence  $\mathbf{X}^i$  by performing a Hadamard product between  $\mathbf{X}^i$  and a mask  $\bar{\mathbf{M}}^i \in \mathbb{R}^{(l+p) \times d}$ . The mask  $\bar{\mathbf{M}}^i$  is obtained by repeating  $\mathbf{M}^i$  along the last dimension, leading to the formulation:

$$\bar{\mathbf{X}}^i = \text{FFN}(\mathbf{X}^i \odot \bar{\mathbf{M}}^i). \quad (6)$$

where  $\odot$  denotes the Hadamard product. While this approach demonstrates effectiveness when the model is



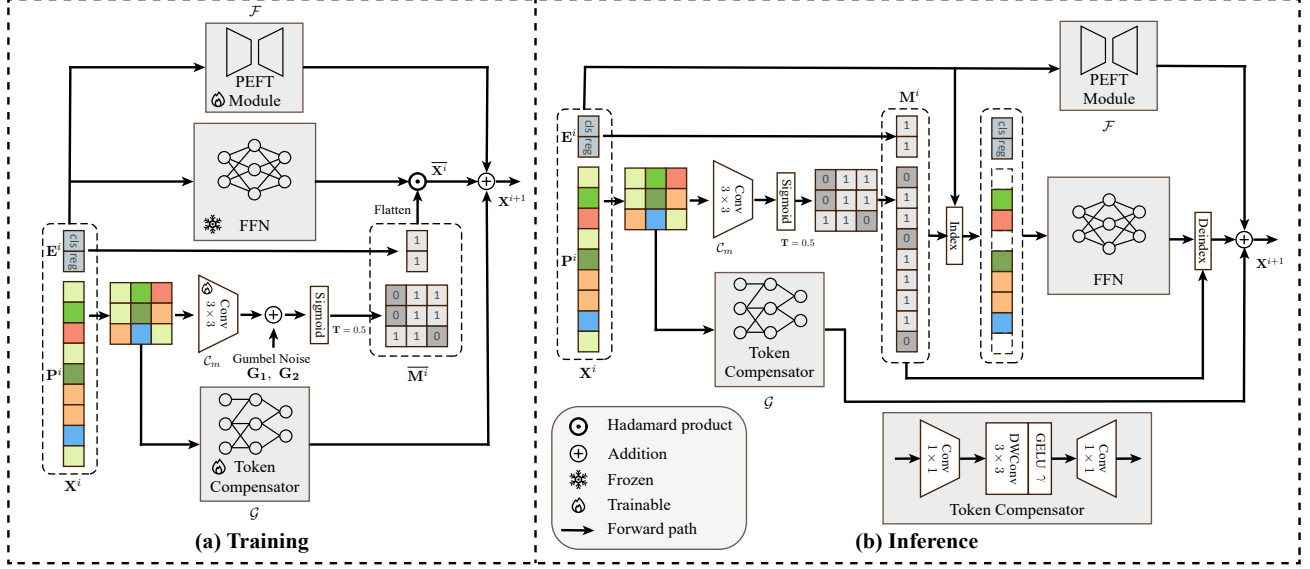


Figure 2. Overall architecture of the proposed method. During training, the PEFT module, token compensator, and token reduction components are tuned, while FFN and other modules within the transformer block remain fixed. A masking technique is applied after the FFN to prevent perturbation of the bias weights. During inference, the mask is utilized to reduce the length of the patch tokens  $\mathbf{P}^i$ , thereby lowering the computational resource requirements.

trained from scratch, our empirical observations indicate that it encounters significant convergence difficulties within the context of PEFT, as illustrated in Fig. 3. We hypothesize this is due to *the perturbation of the FFN bias* from the pre-trained weights. We take the first linear layer of an MLP FFN for instance.

$$\mathbf{Z} = (\mathbf{X}^i \odot \overline{\mathbf{M}}^i) \mathbf{W} + \mathbf{b}. \quad (7)$$

where  $\mathbf{W} \in \mathbb{R}^{d \times rd}$  and  $\mathbf{b} \in \mathbb{R}^{rd}$  are the weight and bias, respectively, and  $r$  is FFN ratio. Here,  $\mathbf{Z}$  denotes the output of the linear layer. The bias term  $\mathbf{b}$  is not affected by the mask  $\overline{\mathbf{M}}^i$ . To address this issue, we apply the mask  $\overline{\mathbf{M}}^i$  again after the FFN to block the perturbation, resulting in the equation

$$\overline{\mathbf{X}}^i = \overline{\mathbf{M}}^i \odot \text{FFN}(\mathbf{X}^i \odot \overline{\mathbf{M}}^i). \quad (8)$$

This adjustment ensures successful model convergence. However, applying  $\overline{\mathbf{M}}^i$  before the FFN becomes ineffective. Since the FFN performs channel-wise transformations, masking prior to the FFN has negligible impact on spatial feature interactions. The benefits of masking are most pronounced when applied after the FFN, as depicted in Fig. 2 (a). This strategy maintains the spatial structure of the input while effectively reducing the biases introduced by the FFN. Thus, we have designed our method to leverage this strategic placement of masking as follows:

$$\overline{\mathbf{X}}^i = \overline{\mathbf{M}}^i \odot \text{FFN}(\mathbf{X}^i). \quad (9)$$

**Mask generation.** We directly generate the mask from the patch token sequence  $\mathbf{P}^i$  by a  $3 \times 3$  convolution layer

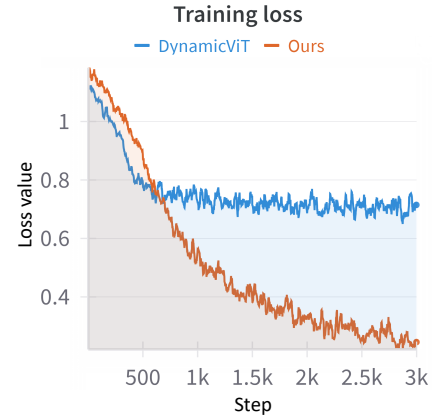


Figure 3. Training Curves of Different TR methods. DynamicViT, based on Eq. (6), faces convergence difficulties. In contrast, our approach, which utilizes Eq. (8), converges successfully.

$\mathcal{C}_m : \mathbb{R}^{d \times h \times w} \rightarrow \mathbb{R}^{1 \times h \times w}$ , where  $h = w = \sqrt{p}$  is the height and width of the reshaped patch tokens. Then, we apply a sigmoid function followed by a threshold operation to generate masks  $\mathbf{M}_P^i$  for patch tokens. To ensure differentiability, we add Gumbel noise [21] prior to the sigmoid function. This process can be formulated as follows:

$$\mathbf{S}^i = \sigma \left( \frac{\mathcal{C}_m(\mathbf{P}^i) + \mathbf{G}_1 - \mathbf{G}_2}{\tau} \right), \quad (10)$$

$$(\mathbf{M}_P^i)_{m,n} = \begin{cases} 0, & (\mathbf{S}^i)_{m,n} < T \\ 1, & (\mathbf{S}^i)_{m,n} \geq T \end{cases} \quad (11)$$

where  $\mathbf{G}_1, \mathbf{G}_2 \sim \text{Gumbel}(0, 1)$ ,  $\tau$  is the temperature pa-

parameter set to 5.0 by default, and  $\sigma$  is the sigmoid function. The threshold value  $T$  is set to 0.5.  $(\mathbf{M}_P^i)_{m,n}$  and  $(\mathbf{S}^i)_{m,n}$  denote the element of  $\mathbf{M}_P^i$  and  $\mathbf{S}^i$  respectively. Then, We choose not to apply masking to class and register tokens, thus setting  $\mathbf{M}_E^i = \mathbf{1} \in \mathbb{R}^{l \times 1}$ . Finally, we concatenate the flattened  $\mathbf{M}_P^i \in \mathbb{R}^{p \times 1}$  with  $\mathbf{M}_E^i$  to obtain the final mask  $\mathbf{M}^i$ .

$$\mathbf{M}^i = \text{concat}[\mathbf{M}_E^i, \mathbf{M}_P^i]. \quad (12)$$

**Masking in training and inference.** As illustrated in Fig. 2 (a), we repeat the values in  $\mathbf{M}^i$  along the embedding dimension to obtain  $\bar{\mathbf{M}}^i \in \mathbb{R}^{(l+p) \times d}$ , during training. During inference, we remove the Gumbel noise and apply  $\mathbf{M}^i$  directly to  $\mathbf{X}^i$  to determine which tokens to mask at FFN, as shown in Fig. 2.

**Token compensator.** The goal of the token compensator is to preserve the spatial structure of the tokens and restore high-frequency information. To achieve this, we design a simple and lightweight token compensator by incorporating local biases and enhancing high-frequency components using convolution layers [46]. Fig. 2 illustrates the architecture of our token compensator. The formulation of the token compensator (TC) is given as:

$$\text{TC}(z) = \text{Conv}(\gamma(\text{DWConv}(\text{Conv}(z)))), \quad (13)$$

where  $\gamma$  is the GELU function, DWConv refers to the  $\times 3$  depth-wise convolution, and Conv denotes the  $1 \times 1$  convolution layer. To maintain the computational efficiency of the token compensator, we maintain a low-rank design strategy for two  $1 \times 1$  convolution layers. This approach minimizes the number of additional parameters.

**Loss function.** In our approach, we incorporate a composite loss function that consists of the segmentation loss  $\mathcal{L}_{seg}$ , the distillation loss  $\mathcal{L}_{cos}$ , and a regularization term  $\mathcal{L}_{rate}$ , which regulates the token activation rate  $k$ . The segmentation is employed to regularize the generation of segmentation maps, combining binary cross-entropy (BCE) and DICE. We directly adopt the Ada loss from [65] to control the token activation rate. For the distillation process, we employ cosine similarity to align the learned representations between the fine-tuned and pre-trained models. The total loss function is thus formulated as:

$$\mathcal{L} = \mathcal{L}_{seg} + \lambda_1 \mathcal{L}_{rate} + \lambda_2 \mathcal{L}_{cos} \quad (14)$$

where  $\lambda_1$  and  $\lambda_2$  are hyperparameters that balance the contributions of the  $\mathcal{L}_{rate}$  and  $\mathcal{L}_{cos}$  and  $\mathcal{L}_{seg} = \lambda_{bce} \mathcal{L}_{bce} + \lambda_{dice} \mathcal{L}_{dice}$ . The complete formulae for each loss function is provided in *Supp.*

## 4. Experiments

### 4.1. Experimental Setting

**Datasets.** We conduct experiments across three different scenes, focusing on real-world tasks. The datasets include

simple scenes, complex scenes, and medical scenes. For simple scenes, we utilize the DUTS [57] dataset for salient object segmentation (SOS) and the CUHK [50] dataset for defocus blur detection. For complex scenes, we employ the COD10K [14] and CAMO [28] datasets for camouflaged object segmentation (COS) [13, 14]. In the medical domain, we use the Kvasir [25] and ISIC 2017 [6] datasets for polyp segmentation and skin lesion segmentation, respectively. For additional results on more datasets, please refer to the *Supp.*

**Implementation details.** Our method was implemented using the PyTorch and the experiments were conducted on a RTX4060Ti GPU. In our implementation, the backbone was a pre-trained *DINOv2-B* [44] model, while the remaining modules were randomly initialized. We utilized the AdamW [40] optimizer for all experiments, with an initial learning rate set at  $1.5 \times 10^{-4}$  and a cosine decay scheduler. During the training phase, we followed the resolution used in DINOv2, resizing input images to  $518 \times 518$  pixels and applying random horizontal flips. For additional implementation details, please refer to *Supp.*

**Evaluation metrics.** To thoroughly evaluate the model’s performance, we employed three widely-used metrics for binary segmentation: weighted F-measure ( $F_\beta^w$ ) [42], S-measure ( $S_\alpha$ ) [5], and mean E-measure ( $E_\phi$ ) [11, 12].

**Loss weights.** We employed a straightforward approach for assigning loss weights. Specifically, we set  $\lambda_{bce} = 1.0$  and  $\lambda_{dice} = 0.5$  for mask loss  $\mathcal{L}_{mask}$ . Additionally, We set  $\lambda_1 = 2.0$  and  $\lambda_2 = 0.1$ .

### 4.2. Quantitative Results

**Comparison with PEFT methods.** In Tab. 1, all PEFT methods outperform the linear probing and decoder-only methods, highlighting the significance of PEFT in enhancing foundation models for challenging scene understanding in difficult data domains. Specifically, in the medical domain, our methods demonstrate superior performance compared to DyT, as well as methods that do not utilize token reduction, particularly referring to weighted f-measure  $F_\beta^w$ . In simple scenes, our methods consistently outperform DyT and achieve comparable performance to PEFT methods without token reduction, such as AdaptFormer and VPT, while exceeding the performance of other PEFT methods without token reduction, including EVP and LoRA. In the complex domain, our methods continue to show an advantage over DyT, reinforcing the effectiveness of our approach across varying levels of scene complexity.

**Comparison with pixel-level TR methods.** We also include additional comparison with TR methods based on full fine-tuning for pixel-level tasks in Sec. 12.2 of *Supp.*

**Analysis of inference efficiency.** In Tab. 1, we present a comparative analysis of the number of parameters and FLOPs for PEFT methods. Notably, our method demon-

Table 1. Quantitative and efficiency comparison of three different scenes using parameter-efficient fine-tuning (PEFT), both without and with token reduction (TR). In terms of the backbone, “Total Params.” denotes the total number of parameters. “Trainable Params. (M) / Ratio (%)” indicates the number of trainable parameters and their proportion relative to the total number. “FLOPs” refers to the floating-point operations per second. Bold values indicate the best performance for configurations both without TR and with TR, respectively. Ours<sup>†</sup> denotes our method with distillation.

Methods		Total Params. (M)	Trainable Params. (M) / Ratio (%)	FLOPs (G)	FPS	Simple						Complex						Medical					
						DUTS [57]			CUHK [50]			COD10K [14]			CAMO [28]			Kvasir [25]			ISIC 2017 [6]		
						$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$	$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$	$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$	$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$	$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$	$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$
w/o TR	Linear	85.51	0 / 0.00%	117.18	24.20	.658	.841	.832	.675	.659	.670	.657	.860	.866	.707	.853	.852	.482	.782	.747	.571	.779	.755
	Decoder Only	85.51	0 / 0.00%	117.18	24.20	.756	.879	.889	.723	.683	.743	.769	.888	.918	.796	.886	.904	.723	.870	.872	.772	.839	.839
	VPT [26]	86.10	0.59 / 0.68%	122.52	27.98	.896	.931	.938	.800	.752	.809	.816	.903	.936	.850	.908	.933	.891	.939	.951	.860	.872	.883
	AdaptFormer [4]	86.70	1.19 / 1.37%	118.70	27.71	.906	.937	.946	.767	.712	.791	.826	.906	.941	.866	.911	.939	.935	.957	.967	.874	.855	.874
	LoRA [23]	86.86	1.33 / 1.53%	118.90	26.45	.897	.932	.942	.764	.712	.803	.814	.897	.935	.852	.905	.934	.936	.956	.966	.880	.862	.882
	EVP [35]	86.25	0.74 / 0.86%	117.44	27.82	.887	.929	.935	.771	.736	.791	.803	.900	.933	.832	.901	.921	.857	.934	.943	.854	.866	.875
w/ TR	DyT [65]	86.71	1.20 / 1.38%	95.51	29.02	.859	.921	.930	.735	.691	.759	.796	.894	.934	.839	.901	.932	.897	.944	.963	.843	.868	.884
	Ours	87.86	2.57 / 2.92%	90.83	28.84	.895	.933	.944	.793	.749	.812	.810	.899	.935	.842	.902	.932	.938	.960	.975	.871	.862	.881
	Ours†	87.86	2.57 / 2.92%	91.54	28.11	.901	.935	.947	.755	.698	.776	.808	.899	.936	.848	.905	.933	.940	.959	.974	.875	.863	.882

strates the smallest FLOPs among the evaluated approaches, indicating a significant advantage in inference efficiency. Our method also achieves a competitive  $F_{\beta}^w$  of 0.901 on the DUTS dataset, closely aligning with the performance of AdaptFormer, which records a  $F_{\beta}^w$  of 0.906. It is important to highlight that AdaptFormer does not implement TR, which suggests that our approach effectively balances complexity and performance.

### 4.3. Recovering High-Frequency Components

In Fig. 1 (f), we show the relative log amplitude of Fourier-transformed representations for both TR and our method. The results highlight that our model effectively restores high-frequency components, particularly in the middle blocks, between the 6<sup>th</sup> and 10<sup>th</sup> layers. While our approach leverages a convolution-based token compensator to enhance high-frequency recovery, we also observe that further distillation of the fine-tuned model with the pre-trained model significantly boosts this process. This enhancement is likely due to the pre-trained model’s prior exposure to high-frequency components during MIM pre-training. Moreover, the learnable masks integrated into the fine-tuned model may also serve as a latent regularizer, guiding the model to focus on high-frequency components and local patterns. This mechanism improves the model’s ability to capture intricate details, which are essential for high-precision pixel-level tasks.

### 4.4. Ablation Study

In order to assess the effectiveness of each component of our methods, we conducted ablation studies on DUTS and Kvasir datasets.

**Different components.** In Tab. 2, we present an ablation study evaluating four combinations of components within our proposed methods. we assess the impact of applying TR both with and without the inclusion of the token compensator (TC), as well as the effects of utilizing distillation in

Table 2. Ablation study on various aspects. The rows highlighted in gray represent our method (default setting). **Bold** values denote the optimal performance.

Methods		DUTS			Kvasir		
		$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$	$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$
Components	No TR	.906	.937	.946	.935	.957	.967
	DyT-TD	.859	.921	.930	.897	.944	.963
	Our TR	.886	.930	.942	.899	.942	.959
	Our TR + Distill.	.887	.931	.943	.928	.954	.973
	Our TR + TC	.895	.933	.944	.938	<b>.960</b>	<b>.975</b>
	Our TR + TC + Distill.	<b>.901</b>	<b>.935</b>	<b>.947</b>	<b>.940</b>	.959	.974
Rank of token compensator	24 (0.52 M)	.878	.927	.939	.910	.946	.960
	48 (1.38 M)	.895	.933	.944	<b>.938</b>	<b>.960</b>	<b>.975</b>
	96 (2.90 M)	<b>.896</b>	<b>.934</b>	<b>.945</b>	.906	.943	.958
Token activation rate	DyT-0.3 (66.39 GFLOPs)	.861	.909	.922	.856	.918	.936
	DyT-0.5 (95.51 GFLOPs)	.859	.921	.930	.897	.944	.963
	DyT-0.7 (107.38 GFLOPs)	.871	.928	.934	.908	.938	.954
	Ours-0.3 (67.91 GFLOPs)	.863	.917	.926	.911	.935	.950
	Ours-0.5 (90.83 GFLOPs)	<b>.895</b>	<b>.933</b>	<b>.944</b>	<b>.938</b>	<b>.960</b>	<b>.975</b>
	Ours-0.7 (106.14 GFLOPs)	.883	.932	.940	.920	.951	.963

these configurations. Our results indicate that the configuration incorporating both the token compensator and distillation yields the highest performance on the DUTS dataset. In the Kvasir dataset, the method without distillation demonstrates a slight performance advantage on S-measure and E-measure. This observation implies that while distillation is beneficial for certain tasks, the inherent characteristics of the Kvasir dataset may favor configurations that prioritize direct token manipulations.

**Rank of token compensator.** Furthermore, Tab. 2 compares the performance of three different ranks of the token compensator. The rank of 48 achieves the best results on the Kvasir dataset, while it performs slightly lower than the rank of 96 on the DUTS dataset. Considering the trade-off between performance and the number of parameters, we have chosen a rank of 48 to achieve an optimal balance.

**Token activation rate.** Tab. 2 presents the impact of varying token activation rates of 0.3, 0.5, and 0.7 on model performance. Our methods achieve optimal results on both the



Figure 4. Visualization of the effects of our token reduction on DINOv2-B. The masked regions indicate areas of lower relevance, allowing the model to focus on informative parts of the images. Notably, some objects retain only their boundaries such as the red mushroom.

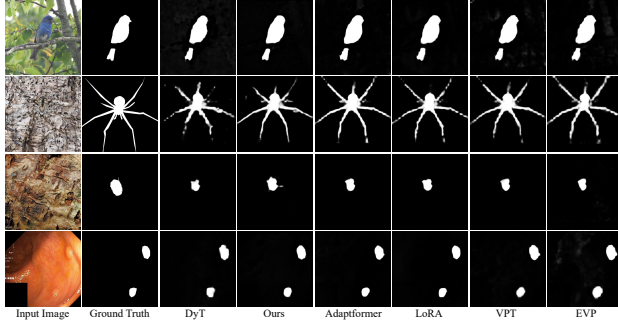


Figure 5. Segmentation outputs from various methods. Zoom in for a clearer view.

DUTS and Kvasir datasets when the token activation rate is set to 0.5. Furthermore, at this setting, our approach demonstrates a reduced number of FLOPs compared to DyT.

#### 4.5. Analysis of Generalization Ability

Previous research [34, 65] have primarily focused on task-specific effectiveness, often ignoring the capacity of models to generalize to novel tasks. To address this gap, we employ a zero-shot COS setting [29] to further evaluate the generalization capabilities of our proposed methods. Specifically, we utilize our well-trained model on the DUTS dataset for SOS, to directly address the problem of COS. As demonstrated in Tab. 3, our methods outperform previous approaches, such as CaMF [29] and GenSAM [24], particularly in terms of S-measure and E-measure. These results suggest that the restored diversity of attention and the high-frequency components in representations contribute significantly to the model’s effectiveness in this zero-shot COS which is biased towards high-frequency information [29].

#### 4.6. Visualization

**Masked tokens.** In Fig. 4, we illustrate the effects of our token reduction method. In most images, the informative areas are retained while less relevant regions are masked, demonstrating that our approach effectively reduces computational requirements by prioritizing the processing of useful areas. Notably, we observe that certain objects, such as those in the bottom left and middle images, have only their

Table 3. Comparison with other supervised (S), weakly-supervised (WS), and zero-shot (ZS) COS methods. **Bold** values denote the best performance.

Methods	Sup.	CAMO			COD10K		
		$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$	$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$
SINet [13]	S	.606	.751	.829	.551	.771	.806
ZoomNeXt [45]	S	.857	.889	.945	.827	.898	.956
CRNet [20]	WS	.641	.735	.815	.576	.733	.805
GenSAM [24]	ZS	.659	.719	.775	.681	.775	.838
CaMF [29]	ZS	.729	.788	.814	<b>.717</b>	.808	.832
Ours	ZS	<b>.730</b>	<b>.812</b>	<b>.838</b>	.695	<b>.812</b>	<b>.844</b>

boundaries unmasked. This suggests that our method emphasizes the local patterns and high-frequency components of the images, resulting in a relatively higher masking rate to further conserve computational resources.

**Segmentation maps.** As shown in Fig. 5, our method produces segmentation maps with smoother boundaries compared to other methods. Additional results are provided in the *Supp.*

## 5. Conclusion

In this paper, we investigate the impact of TR on attention mechanisms and visual representations, identifying barriers that impede its application within PEFT for pixel-level tasks. We introduce a novel TR-PEFT method that effectively preserves attention diversity, high-frequency components, and the completeness of the token sequence while reducing the length of tokens. This innovative approach aims to achieve an optimal balance between performance and inference efficiency, thereby enhancing the practical applicability of PEFT methods in pixel-level tasks.

## Acknowledgements

This work was supported by the Key Program for International Cooperation of Ministry of Science and Technology of China (No.2024YFE0100700) and the National Natural Science Foundation of China (NSFC) under Grant 62020106011.



## References

- [1] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations*, 2023. 1, 3
- [2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 1
- [3] Mengzhao Chen, Wenqi Shao, Peng Xu, Mingbao Lin, Kaipeng Zhang, Fei Chao, Rongrong Ji, Yu Qiao, and Ping Luo. Diffrate: Differentiable compression rate for efficient vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17164–17174, 2023. 1, 3
- [4] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022. 3, 4, 7, 2, 5, 6
- [5] Ming-Ming Cheng and Deng-Ping Fan. Structure-measure: A new way to evaluate foreground maps. *IJCV*, 129(9): 2622–2638, 2021. 6
- [6] Noel CF Codella, David Gutman, M Emre Celebi, Brian Helba, Michael A Marchetti, Stephen W Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, et al. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pages 168–172. IEEE, 2018. 3, 6, 7, 2, 4
- [7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 4
- [8] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations*, 2024. 4
- [9] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1
- [10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 4
- [11] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In *ICCV*, 2017. 6
- [12] 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*. AAAI Press, 2018. 6
- [13] Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In *CVPR*, 2020. 3, 6, 8
- [14] Deng-Ping Fan, Ge-Peng Ji, Ming-Ming Cheng, and Ling Shao. Concealed object detection. *IEEE TPAMI*, 44(10): 6024–6042, 2022. 3, 6, 7, 2, 5
- [15] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. *arXiv preprint arXiv:2211.07636*, 2022. 1
- [16] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *Image and Vision Computing*, page 105171, 2024. 1, 3
- [17] Shengxi Gui, Shuang Song, Rongjun Qin, and Yang Tang. Remote sensing object detection in the deep learning era—a review. *Remote Sensing*, 16(2):327, 2024. 3
- [18] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*, 2022. 3
- [19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4
- [20] Ruozhen He, Qihua Dong, Jiaying Lin, and Rynson W.H. Lau. Weakly-supervised camouflaged object detection with scribble annotations. *AAAI*, 37(1):781–789, 2023. 8
- [21] Charles Herrmann, Richard Strong Bowen, and Ramin Zabih. Channel selection using gumbel softmax. In *European conference on computer vision*, pages 241–257. Springer, 2020. 5
- [22] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, 2019. 1, 3, 4
- [23] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 1, 3, 7, 2, 4, 5, 6
- [24] Jian Hu, Jiayi Lin, Shaogang Gong, and Weitong Cai. Relax image-specific prompt requirement in sam: A single generic prompt for segmenting camouflaged objects. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12511–12518, 2024. 8
- [25] Debesh Jha, Pia H Smedsrud, Michael A Riegler, Pål Halvorsen, Thomas De Lange, Dag Johansen, and Håvard D Johansen. Kvasir-seg: A segmented polyp dataset. In *MultiMedia modeling: 26th international conference, MMM 2020, Daejeon, South Korea, January 5–8, 2020, proceedings, part II 26*, pages 451–462. Springer, 2020. 3, 6, 7, 2, 4



- [26] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision (ECCV)*, 2022. 1, 3, 7, 2, 4, 5, 6
- [27] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 1, 3
- [28] Trung-Nghia Le, Tam V. Nguyen, Zhongliang Nie, Minh-Triet Tran, and Akihiro Sugimoto. Anabran network for camouflaged object segmentation. *CVIU*, 184:45–56, 2019. 6, 7, 2, 5
- [29] Cheng Lei, Jie Fan, Xinran Li, Tianzhu Xiang, Ao Li, Ce Zhu, and Le Zhang. Towards real zero-shot camouflaged object segmentation without camouflaged annotations. *arXiv preprint arXiv:2410.16953*, 2024. 8
- [30] Tao Lei, Junwen Bai, Siddhartha Brahma, Joshua Ainslie, Kenton Lee, Yanqi Zhou, Nan Du, Vincent Zhao, Yuxin Wu, Bo Li, et al. Conditional adapters: Parameter-efficient transfer learning with fast inference. *Advances in Neural Information Processing Systems*, 36:8152–8172, 2023. 1, 3, 4
- [31] Guanbin Li and Yizhou Yu. Visual saliency based on multi-scale deep features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5455–5463, 2015. 2, 5
- [32] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations*, 2022. 1, 2, 3
- [33] Mingbao Lin, Mengzhao Chen, Yuxin Zhang, Chunhua Shen, Rongrong Ji, and Liujuan Cao. Super vision transformer. *International Journal of Computer Vision*, 131(12): 3136–3151, 2023. 1
- [34] Ting Liu, Xuyang Liu, Liangtao Shi, Zunnan Xu, Siteng Huang, Yi Xin, and Qunjun Yin. Sparse-Tuning: Adapting vision transformers with efficient fine-tuning and inference. *arXiv preprint arXiv:2405.14700*, 2024. 1, 2, 3, 8
- [35] Weihuang Liu, Xi Shen, Chi-Man Pun, and Xiaodong Cun. Explicit visual prompting for low-level structure segmentations. In *CVPR*, pages 19434–19445, 2023. 3, 7, 2, 5, 6
- [36] Yifei Liu, Mathias Gehrig, Nico Messikommer, Marco Cannici, and Davide Scaramuzza. Revisiting token pruning for object detection and instance segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2658–2668, 2024. 3, 4
- [37] Yang Liu, Qiang Zhou, Jing Wang, Zhibin Wang, Fan Wang, Jun Wang, and Wei Zhang. Dynamic token-pass transformers for semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1827–1836, 2024. 3, 4
- [38] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1
- [39] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 3
- [40] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6, 3
- [41] Yunqiu Lyu, Jing Zhang, Yuchao Dai, Aixuan Li, Bowen Liu, Nick Barnes, and Deng-Ping Fan. Simultaneously localize, segment and rank the camouflaged objects. In *CVPR*, 2021. 2, 5
- [42] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps? In *CVPR*, pages 248–255, 2014. 6
- [43] Vida Movahedi and James H Elder. Design and perceptual validation of performance measures for salient object segmentation. In *2010 IEEE computer society conference on computer vision and pattern recognition-workshops*, pages 49–56. IEEE, 2010. 2, 5
- [44] 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, 2023. 1, 2, 6, 3
- [45] Youwei Pang, Xiaoqi Zhao, Tian-Zhu Xiang, Lihe Zhang, and Huchuan Lu. Zoomnext: A unified collaborative pyramid network for camouflaged object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 8
- [46] Namuk Park and Songkuk Kim. How do vision transformers work? In *International Conference on Learning Representations*, 2022. 2, 6, 1
- [47] Namuk Park, Wonjae Kim, Byeongho Heo, Taekyung Kim, and Sangdoo Yun. What do self-supervised vision transformers learn? In *International Conference on Learning Representations*, 2023. 1, 2
- [48] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. AdapterFusion: Non-destructive task composition for transfer learning. In *EACL*, pages 487–503, 2021. 3
- [49] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021. 1, 3, 4
- [50] Jianping Shi, Li Xu, and Jiaya Jia. Discriminative blur detection features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2965–2972, 2014. 6, 7, 2
- [51] Juan Silva, Aymeric Histace, Olivier Romain, Xavier Dray, and Bertrand Granado. Toward embedded detection of polyps in wce images for early diagnosis of colorectal cancer. *International journal of computer assisted radiology and surgery*, 9:283–293, 2014. 2
- [52] Przemysław Skurowski, Hassan Abdulameer, J Błaszczyk, Tomasz Depta, Adam Kornacki, and P Kozieł. Animal

- camouflage analysis: Chameleon database. *Unpublished manuscript*, 2(6):7, 2018. [2](#), [5](#)
- [53] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617, 2002. [2](#)
- [54] Nima Tajbakhsh, Suryakanth R Gurudu, and Jianming Liang. Automated polyp detection in colonoscopy videos using shape and context information. *IEEE transactions on medical imaging*, 35(2):630–644, 2015. [2](#), [6](#)
- [55] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. [1](#)
- [56] Di Wang, Jing Zhang, Bo Du, Gui-Song Xia, and Dacheng Tao. An empirical study of remote sensing pretraining. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–20, 2023. [3](#)
- [57] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *CVPR*, 2017. [6](#), [7](#), [2](#), [5](#)
- [58] Zhehao Wang, Xian Lin, Nannan Wu, Li Yu, Kwang-Ting Cheng, and Zengqiang Yan. Dtmformer: Dynamic token merging for boosting transformer-based medical image segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5814–5822, 2024. [3](#), [4](#)
- [59] W.Liu, X.Shen, C.-M.Pun, and X.Cun. Explicit visual prompting for universal foreground segmentations. *arXiv preprint arXiv:2305.18476*, 2023. [3](#)
- [60] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *European Conference on Computer Vision*. Springer, 2018. [1](#)
- [61] Zhenda Xie, Zigang Geng, Jingcheng Hu, Zheng Zhang, Han Hu, and Yue Cao. Revealing the dark secrets of masked image modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14475–14485, 2023. [1](#)
- [62] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1155–1162, 2013. [2](#), [5](#), [6](#)
- [63] Hongxu Yin, Arash Vahdat, Jose Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-ViT: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. [1](#)
- [64] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019. [4](#)
- [65] Wangbo Zhao, Jiasheng Tang, Yizeng Han, Yibing Song, Kai Wang, Gao Huang, Fan Wang, and Yang You. Dynamic tuning towards parameter and inference efficiency for vit adaptation. *arXiv preprint arXiv:2403.11808*, 2024. [1](#), [3](#), [4](#), [6](#), [7](#), [8](#), [2](#), [5](#)
- [66] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. [2](#), [4](#)
- [67] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *International Conference on Learning Representations (ICLR)*, 2022. [1](#), [3](#)

# Rethinking Token Reduction with Parameter-Efficient Fine-Tuning in ViT for Pixel-Level Tasks

## Supplementary Material

### 6. Attention Distance Analysis

In this paper, we measure the average distance spanned by attention weights at different layers. This attention distance is analogous to receptive field size in CNNs [9]. Below, we outline the process for calculating the normalized mean attention distance. For simplicity, we omit the block index  $i$ . First, we compute the distance matrix based on  $\mathbf{P}$ . This involves calculating the distances of each token relative to all other tokens, resulting in a distance matrix  $\mathbf{D} \in \mathbb{R}^{p \times p}$ , where  $p$  denotes the length of the patch token. Following the principles of self-attention [9, 55], we first compute the query embedding  $\mathbf{Q}$  and key embedding  $\mathbf{K}$  of the patch tokens  $\mathbf{P}$ . Next, we obtain the attention weights, denoted as  $\mathbf{A}$  using the formula  $\mathbf{A} = \mathbf{Q}\mathbf{K}^\top$ . We represent the attention weights for each head as  $\mathbf{A}_j$ . Subsequently, we derive the weighted distance matrix  $\mathbf{W}_j$  by calculating  $\mathbf{W}_j = \mathbf{D} \odot \mathbf{A}_j$ . Finally, for the mean attention distance of  $j^{th}$  head, we can use the following formula:

$$d_j = \frac{1}{p} \sum_{m=1}^p \sum_{n=1}^p (\mathbf{W}_j)_{m,n} \quad (15)$$

To compute the normalized mean attention distance, we apply min-max normalization to  $\mathbf{d} = [d_j]_{j=1}^N$ , where  $N$  denotes the number of heads in the attention.

### 7. Mutual Information Analysis

Normalized Mutual Information is used to measure the attention collapse [47]. Let  $p_Q(q)$  and  $p_K(k)$  be the spatial distribution of query embeddings  $\mathbf{Q}$  and key embeddings  $\mathbf{K}$  and assume that these query tokens are uniformly distributed since a single query token is given for each spatial coordinate. That is  $p_Q(q) = \frac{1}{N}$ . Our goal is to measure the mutual information of the  $p_Q(q)$  and  $p_K(k)$ .

$$I(q; k) = \sum p_{QK}(q, k) \log \frac{p_{QK}(q, k)}{p_Q(q)p_K(k)}, \quad (16)$$

where  $p_{QK}(q, k) = \pi(k|q)p_Q(q)$  represents the joint distribution of  $p_Q(q)$  and  $p_K(k)$  and  $\pi(k|q)$  denotes the conditional distribution, which is the attention weights after softmax normalization. Since  $p_Q(q)$  is constant,  $p_K(k) = p_{QK}(q, k) = \pi(k|q)p_Q(q)$ . Then, we get normalized mutual information  $I_{norm}$  by

$$I_{norm} = \frac{I(q; k)}{\sqrt{H(q)H(k)}}, \quad (17)$$

where  $H(q)$  and  $H(k)$  are the entropy of  $p_Q(q)$  and  $p_K(k)$ , respectively.

### 8. Fourier Analysis

Following [46, 47], let  $\mathbf{P}$  be patch token sequence. We begin by applying the fast Fourier transform (FFT) to  $\mathbf{P}$ , followed by a conversion to obtain the log amplitude:

$$\delta = \log |\text{FFT}(\mathbf{P})| \quad (18)$$

Subsequently, we extract the half-diagonal components, denoted as  $\delta'$ . The relative log amplitudes are then computed as follows:

$$\Delta = \delta' - \delta'_{max}, \quad (19)$$

where  $\delta'_{max}$  represents the maximum amplitude, identified as the first element of  $\delta'$ .

### 9. Decoder

The application of PEFT to complex downstream tasks necessitates the incorporation of a decoder that introduces non-linearity, in contrast to a simple linear layer in [44]. In this work, we aim to demonstrate the capacity of PEFT methods to effectively transfer knowledge and the inductive bias introduced by our approach. To mitigate the effects of inductive bias associated with intricately designed decoders, such as those based on convolutional architectures like UperNet [60], we implement a MLP-based decoder. The structure of our decoder can be represented as follows:

$$\text{Decoder}(\cdot) = \text{MLP}(\text{Up}(\text{MLP}(\cdot))), \quad (20)$$

where Up denotes an upsampling operation using bilinear interpolation.

### 10. Loss Functions

The total loss function in our approach is a composite of three components: the segmentation loss  $\mathcal{L}_{seg}$ , the distillation loss  $\mathcal{L}_{cos}$ , and a regularization term  $\mathcal{L}_{rate}$ , which controls the mask rate  $k$ . The overall loss is defined as:

$$\mathcal{L} = \mathcal{L}_{seg} + \lambda_1 \mathcal{L}_{rate} + \lambda_2 \mathcal{L}_{cos} \quad (21)$$

where  $\lambda_1$  and  $\lambda_2$  are hyperparameters that balance the contributions of the  $\mathcal{L}_{rate}$  and  $\mathcal{L}_{cos}$ .

The segmentation loss  $\mathcal{L}_{seg}$  combines binary cross-entropy (BCE) and DICE to regularize the generation of segmentation masks.

$$\mathcal{L}_{seg} = \lambda_{bce} \mathcal{L}_{bce} + \lambda_{dice} \mathcal{L}_{dice}, \quad (22)$$

where  $\lambda_{bce}$ , and  $\lambda_{dice}$  are hyperparameters that balance the contributions of the BCE and DICE loss. The BCE loss  $\mathcal{L}_{bce}$  and DICE loss  $\mathcal{L}_{dice}$  are defined as:

$$\mathcal{L}_{bce} = -\frac{1}{H \times W} \sum_{m=1}^H \sum_{n=1}^W \left( \left( \mathbf{Y} \log \hat{\mathbf{Y}} \right)_{m,n} + \left( (\mathbf{1} - \mathbf{Y}) \log (\mathbf{1} - \hat{\mathbf{Y}}) \right)_{m,n} \right), \quad (23)$$

$$\mathcal{L}_{dice} = 1 - \frac{\epsilon + 2 \sum_{m=1}^H \sum_{n=1}^W (\mathbf{Y} \odot \hat{\mathbf{Y}})_{m,n}}{\epsilon + \sum_{m=1}^H \sum_{n=1}^W (\mathbf{Y} + \hat{\mathbf{Y}})_{m,n}}, \quad (24)$$

where  $\mathbf{Y}$  and  $\hat{\mathbf{Y}} \in \mathbb{R}^{H \times W}$  represent the ground truth and predicted segmentation maps, respectively, with values in the range  $[0, 1]$ . The term  $\epsilon$  is a small constant added to prevent division by zero.

We directly adopt the Ada loss from [65] to control the mask rate.

$$\mathcal{L}_{rate} = \left( \frac{1}{N \times h \times w} \sum_{i=1}^N \sum_{m=1}^h \sum_{n=1}^w (\mathbf{M}_P^i)_{m,n} - k \right)^2 \quad (25)$$

where  $\mathbf{M}_P^i \in \mathbb{R}^{H' \times W'}$  denotes the mask at  $i^{\text{th}}$  transformer block for patch tokens,  $N$  represent the total number of transformer blocks, and  $k$  is the target mask rate.

For distillation, we use cosine similarity to align the representations of the fine-tuned and pre-trained models. The logits from both models are processed through an MLP-based distillation head, yielding  $\dot{\mathbf{X}}^N$  and  $\ddot{\mathbf{X}}^N$ , where  $\dot{\mathbf{X}}^N, \ddot{\mathbf{X}}^N \in \mathbb{R}^{(l+p) \times d}$ . The distillation loss is defined as:

$$\mathcal{L}_{cos} = 1 - \sum_{i=1}^{l+p} \frac{\dot{\mathbf{X}}_i^N \cdot \ddot{\mathbf{X}}_i^N}{\|\dot{\mathbf{X}}_i^N\| \cdot \|\ddot{\mathbf{X}}_i^N\|} \quad (26)$$

## 11. Implementation Details

**Baselines.** We compare our method with several approaches, including linear probing, fine-tuning of the decoder only, and VPT [26], which inserts learnable tokens into the hidden states of each transformer block. Additionally, we consider AdaptFormer [4], which adds trainable low-rank MLP layers in parallel to the FFN layer within a transformer block, and LoRA [23], which incorporates trainable low-rank linear layers alongside the frozen linear weights. Finally, we include EVP [35], which integrates high-frequency priors with parallel adapters in the transformer blocks.

Table 4 presents the implementation details for training binary segmentation tasks.

**Salient object segmentation.** We utilize four widely recognized datasets for salient object segmentation: DUTS [57],

ECSSD [62], SOD [43], and HKU-IS [31]. The DUTS dataset comprises 10,553 training images and 5,019 testing images. The ECSSD dataset includes 1,000 testing images, while the SOD dataset consists of 300 testing images. Additionally, the HKU-IS dataset contains 4,447 testing images. All methods are trained on the training set of DUTS [57] and evaluated on the testing sets of DUTS, ECSSD, SOD, and HKU-IS.

**Defocus blur detection.** Following the methodology presented in [50], we conduct training on the CUHK dataset [50], which consists of a total of 704 samples exhibiting partial defocus. The network is trained using 604 images from the CUHK dataset, with testing performed on the remaining images.

**Camouflaged object segmentation.** To assess our methods, we select four commonly utilized datasets for camouflaged object segmentation. The COD10K dataset [14] comprises 3,040 training samples and 2,026 testing samples. The CAMO dataset [28] provides 1,000 images for training and 250 for testing. The NC4K dataset [41] contains 4,121 testing samples, while the CHAMELEON dataset [52] includes 76 testing images. In alignment with [14], we train our methods on the training sets of COD10K and CAMO, and we evaluate their performance on the testing sets of COD10K, CAMO, NC4K, and CHAMELEON.

**Polyp segmentation.** For polyp segmentation, we employ three datasets: Kvasir [25], ETIS [51], and CVC-ColonDB [54]. The Kvasir dataset includes 1,100 training images and 196 testing images. The ETIS dataset contains 196 images, while CVC-ColonDB comprises 612 images. Our training is conducted on the training set of Kvasir, with evaluation performed using the testing sets of ETIS, CVC-ColonDB, and Kvasir.

**Skin lesion segmentation.** For skin lesion segmentation, we focus on the ISIC 2017 dataset [6], which provides 2,000 training images and 600 testing images.

**Semantic segmentation.** We utilize two foundational datasets for semantic segmentation: ADE20K [66] and Cityscapes [7]. Following [44], we conduct training and testing on ADE20K and Cityscapes, respectively. As presented in Table 5, we train for a total of 8 epochs on ADE20K and 48 epochs on Cityscapes, thereby ensuring that the number of iterations remains approximately consistent across both datasets.

## 12. More Experiment Results

### 12.1. Ablation Study

**TR placement.** Attention mechanisms incur notable computational overhead for long sequences. Our ablation study in Tab. 6 evaluates the effects of TR placement, demonstrating that pre-attention TR application significantly degrades pixel-level task performance. For instance,  $F_\beta^w$  scores on

Table 4. Experimental settings for binary segmentation datasets. We train all methods using the same hyperparameters.

Configuration	DUTS	CUHK	COD10K+CAMO	Kvasir	ISIC 2017
Optimizer			AdamW [40]		
Base learning rate			1.5e-4		
Weight decay			1e-4		
Batch size			10		
Learning rate schedule			Cosine decay [39]		
Warmup epochs	4	10	10	10	5
Training epochs	16	40	50	40	20

Table 5. Experimental settings for semantic segmentation datasets. We train all methods using the same hyperparameters.

Configuration	ADE20K	Cityscapes
Optimizer	AdamW [40]	
Base learning rate	1.5e-4	
Weight decay	1e-4	
Batch size	4	
Learning rate schedule	Cosine decay [39]	
Warmup epochs	1	6
Training epochs	8	48

Table 6. Ablation study on TR placement and different backbones. The rows highlighted in gray represent our method (default setting). **Bold** values denote the optimal performance under TR.

Methods		DUTS			Kvasir		
		$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$	$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$
TR Placement	No TR	.906	.937	.946	.935	.957	.967
	Attention	.878	.926	.937	.865	.921	.933
	FFN	<b>.895</b>	<b>.933</b>	<b>.944</b>	<b>.938</b>	<b>.960</b>	<b>.975</b>
Backbones	iBOT-Base [67]	.826	.893	.905	.925	.957	.968
	iBOT-Large [67]	.847	.904	.917	.941	.964	.973
	EVA02-Base [16]	.883	.923	.934	.953	.967	.979
	EVA02-Large [16]	.901	.934	.943	<b>.965</b>	<b>.974</b>	<b>.984</b>
	DINOv2-Base [44]	.895	.933	.944	.938	.960	.975
	DINOv2-Large [44]	<b>.908</b>	<b>.940</b>	<b>.949</b>	.958	.969	.980

DUTS decline to 0.878 (vs. 0.895 when TR is applied to FFN), with analogous performance reductions observed on the Kvasir dataset. This aligns with [65]’s findings in image classification. Future work should explore effective TR application in attention for pixel-level tasks without compromising performance.

**Different backbones.** To assess the generalization and scalability of our method, we conduct experiments on backbones with diverse attention properties (e.g. DINOv2 [44], EVA02 [16], iBOT [67]). As shown in Tab. 6, our method achieves consistent performance across these backbones. Furthermore, when scaling to larger variants, performance improves with model size, empirically demonstrating scalability.

## 12.2. Segmentation Results

**Comparison with recent TR methods.** To evaluate our method against existing TR approaches for pixel-level tasks, we conduct comparisons in Tab. 7. We evaluate recent TR methods specialized for distinct pixel-level tasks, including DTMFormer [58] in medical image segmentation, DoViT [37] in semantic segmentation, and SViT [36] in instance or semantic segmentation. Experiments are performed on established benchmarks, including Kvasir and ISIC 2017 for medical image segmentation, and ADE20K and Cityscapes for semantic segmentation, ensuring consistency with prior experimental setups.

**Semantic Segmentation.** In Table 8, our methods demonstrate superior performance relative to DyT on both ADE20K and Cityscapes. Notably, our approach outperforms AdaptFormer, a method that does not incorporate TR. This advantage may stem from our method’s ability to maintain the diversity of attention while the high-frequency components aggregated by the token compensator positively influence multi-class segmentation. This effect is particularly significant on ADE20K, which includes 151 classes, in contrast to Cityscapes, which has only 34 classes. Additionally, our methods achieve performance comparable to that of PEFT methods without TR on the Cityscapes dataset.

**Salient object segmentation.** Table 9 presents the results of our evaluation on salient object segmentation (SOS). Our method demonstrates superior performance compared to DyT across all SOS datasets, while achieving performance levels comparable to methods that do not utilize TR. This suggests that our approach effectively enhances fundamental segmentation capabilities.

**Camouflaged object segmentation.** In Table 10, we demonstrate the performance of our method on four camouflaged object segmentation (COS) datasets. Our approach achieves superior performance relative to DyT across all datasets, showcasing its effectiveness in handling more complex scenes. Furthermore, it attains performance levels comparable to those of methods that do not incorporate TR. This finding suggests that our method significantly enhances segmentation capabilities in challenging scenarios



Table 7. Comparison with full fine-tuning (full FT) TR methods.

Methods		Total Params. (M)	Trainable Params. (M) / Ratio (%)	Semantic Segmentation		Medical					
				ADE20K [66]	Cityscapes [7]	Kvasir [25]		ISIC 2017 [6]			
				mIOU (%)	mIOU (%)	$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$	$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$
Full FT w/ TR	DTMFormer [58]	33.6	33.6 / 100.0%	-	-	.870	.926	.941	.776	.830	.821
	DoViT [37]	88.5	88.5 / 100.0%	46.5	<b>76.4</b>	-	-	-	-	-	-
	SViT [36]	48.0	48.0 / 100.0%	44.8	76.2	-	-	-	-	-	-
PEFT w/ TR	DyT [65]	86.7	1.2 / 1.4%	49.5	50.6	.897	.944	.963	.843	<b>.868</b>	<b>.884</b>
	Ours	87.9	2.6 / 2.9%	<b>52.6</b>	67.3	<b>.938</b>	<b>.960</b>	<b>.975</b>	.871	.862	.881

Table 8. Quantitative and efficiency comparison of semantic segmentation using parameter-efficient fine-tuning (PEFT), both without and with token reduction (TR).

Methods		FLOPs (G)	mIOU		
			ADE20K [66]	Cityscapes [7]	Avg.
w/o TR	Linear	117.08	45.9	60.9	54.1
	VPT [26]	122.52	49.2	61.8	55.5
	LoRA [23]	118.90	50.7	61.6	56.2
	AdaptFormer [4]	118.70	<b>51.5</b>	<b>68.8</b>	<b>60.2</b>
w/ TR	DyT [65]	97.87	49.5	50.6	50.0
	Ours	107.55	<b>52.6</b>	<b>67.3</b>	<b>60.0</b>

associated with camouflaged objects.

**Polyp segmentation.** In Table 11, we present the performance of our method on several medical segmentation datasets. Our approach demonstrates superior performance compared to DyT across all datasets, highlighting its effectiveness in addressing the complexities inherent in medical images. Notably, our method outperforms all PEFT methods without TR including AdaptFormer on Kvasir and CVC-ColonDB. This finding indicates the robustness and adaptability of our methods in medical segmentation tasks.

### 12.3. VTAB-1K Results

To evaluate the performance of our method on classification tasks and its adaptation capabilities when training data is limited, we employ the VTAB-1K [64] benchmark. Following the approach outlined in [65], we utilize a Vision Transformer (ViT) pretrained on ImageNet-21k, maintaining the same training scheme. In contrast to our segmentation configuration, we adopt a significantly reduced rank setting for this classification task, specifically configuring the adapter rank at 6 and the token compensator rank at 2.

Results are summarized in Tab. 12. For TR-PEFT methods on natural images, our approach achieves one first-rank and one second-rank result, outperforming DyT, which ranks second in both instances. In specialized domains, AdaptFormer demonstrates superior performance across most datasets, while our method and DyT achieve comparable performance. Conversely, in structured domains, our method exhibits limitations due to limited dataset scale and significant domain gaps between structured images and pre-training data, necessitating larger training datasets than DyT

to achieve comparable performance.

## 13. More Visualization

### 13.1. Masked Tokens

We provide additional visualizations to illustrate the effects of our token reduction methods across various scenes, including simple, complex, and medical contexts. Fig. 6 displays the effects of our token reduction in a simple scene, exemplified by the DUTS dataset. Fig. 7 demonstrates the effects in a complex scene. Fig. 8 and Fig. 9 showcase the impact of our token reduction methods in polyp segmentation and skin lesion segmentation, respectively.

Our method progressively masks tokens along object boundaries as network layers increase. In Fig. 6, we observed that our approach tends to mask inner object regions while preserving boundary integrity. In other words, our method retains only the boundary information of objects, thereby achieving a relatively higher mask ratio within a single image. In Fig. 7, Fig. 8, and Fig. 9, the masking strategy effectively identifies concealed objects that require fine-grained processing, demonstrating the effectiveness of our approach. This strategy thereby enhances the model’s confidence in target identification across various segmentation tasks.

### 13.2. Segmentation Maps

To further demonstrate the effectiveness of our method, we present additional visualizations of segmentation maps across diverse scenarios, including simple, complex, and medical contexts. Specifically, Fig. 10 illustrates segmentation results in a simple scene using the DUTS dataset as a representative example. Fig. 11 highlights segmentation performance in complex scenes, while Fig. 12 showcases results in medical applications, specifically polyp segmentation and skin lesion segmentation.

As depicted in Fig. 10, our method achieves superior performance in simple scenes. The segmented regions generated by our approach exhibit smoother boundaries and finer details, such as continuous thin lines, compared to competing methods. Additionally, the segmentation maps produced by our method contain minimal noise, which can be

Table 9. Quantitative comparison of salient object segmentation.

Methods		DUTS [57]			ECSSD [62]			SOD [43]			HKU-IS [31]		
		$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$	$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$	$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$	$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$
w/o TR	Linear	.658	.841	.832	.777	.892	.875	.715	.810	.785	.741	.881	.879
	Decoder Only	.756	.879	.889	.849	.916	.921	.787	.834	.837	.817	.906	.924
	VPT [26]	.896	.931	.938	.934	.951	.958	.830	.826	.824	.914	.940	.954
	AdaptFormer [4]	<b>.906</b>	<b>.937</b>	<b>.946</b>	<b>.943</b>	<b>.955</b>	<b>.962</b>	<b>.865</b>	<b>.850</b>	<b>.859</b>	<b>.926</b>	<b>.946</b>	<b>.960</b>
	LoRA [23]	.897	.932	.942	.927	.949	.954	.854	.851	.855	.911	.940	.954
	EVP [35]	.887	.929	.935	.930	.949	.955	.846	.836	.832	.912	.940	.953
w/ TR	DyT [65]	.859	.921	.930	.902	.938	.944	.837	.849	.855	.886	.931	.945
	Ours	.895	.933	.944	.924	<b>.948</b>	<b>.958</b>	<b>.860</b>	<b>.851</b>	<b>.863</b>	.908	<b>.943</b>	<b>.960</b>
	Ours†	<b>.901</b>	<b>.935</b>	<b>.947</b>	<b>.932</b>	<b>.948</b>	.957	.856	.850	.857	<b>.917</b>	.942	.958

Table 10. Quantitative comparison of camouflaged object segmentation.

Methods		COD10K [14]			CAMO [28]			NC4K [41]			CHAMELEON [52]		
		$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$	$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$	$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$	$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$
w/o TR	Linear	.657	.860	.866	.707	.853	.852	.750	.889	.891	.746	.892	.888
	Decoder Only	.769	.888	.918	.796	.886	.904	.831	.907	.925	.825	.908	.924
	VPT [26]	.816	.903	.936	.850	.908	.933	.869	.918	.941	.860	<b>.926</b>	.951
	AdaptFormer [4]	<b>.826</b>	<b>.906</b>	<b>.941</b>	<b>.866</b>	<b>.911</b>	<b>.939</b>	<b>.879</b>	<b>.922</b>	<b>.946</b>	<b>.873</b>	<b>.926</b>	<b>.952</b>
	LoRA [23]	.814	.897	.935	.852	.905	.934	.867	.913	.939	.855	.920	.944
	EVP [35]	.803	.900	.933	.832	.901	.921	.859	.916	.937	.847	.917	.938
w/ TR	DyT [65]	.796	.894	.934	.839	.901	.932	.851	.910	.937	.834	.909	.943
	Ours	<b>.810</b>	<b>.899</b>	.935	.842	.902	.932	.853	.914	.939	.834	.915	.936
	Ours†	.808	<b>.899</b>	<b>.936</b>	<b>.848</b>	<b>.905</b>	<b>.933</b>	<b>.859</b>	<b>.915</b>	<b>.939</b>	<b>.860</b>	<b>.924</b>	<b>.952</b>

attributed to the integration of a learnable mask mechanism that effectively suppresses irrelevant regions.

In complex scenes (Fig. 11), our method maintains its advantage, particularly in preserving the continuity of thin structures—a challenge where alternative approaches often produce fragmented results. For medical applications (Fig. 12), our method demonstrates higher confidence levels in delineating abnormal regions. Notably, in cases involving ambiguous abnormal areas (e.g., the second row of Fig. 12), our approach maintains clearer distinctions compared to other methods, which exhibit uncertainty in such regions.

## 14. Limitations and Future Work

While the proposed method demonstrates improved efficiency in FFN computations through TR, two primary limitations remain. First, it does not directly address the computational overhead of attention mechanisms in long-sequence tasks. Our experiments in Tab. 6 show that applying token reduction before attention layers degrades performance in pixel-level tasks, as evidenced by  $F_{\beta}^w$  scores of 0.878

compared to 0.895 when token reduction is applied to FFN on DUTS. This aligns with prior work [65], which identified similar challenges in image classification, indicating that token reduction disrupts spatial dependencies essential for attention-based feature refinement. Second, the method demonstrates limited adaptability in structured image domains such as medical or satellite imagery, where significant domain gaps exist between pretraining data and target tasks. Achieving performance comparable to DyT in these domains requires substantially larger training datasets, underscoring a dependency on data scale that may limit practicality in resource-constrained settings.

Future research should focus on optimizing the integration of token reduction with attention mechanisms to minimize computational cost in pixel-level tasks. This could involve developing TR techniques that preserve attention dynamics without significantly compromising performance. Additionally, exploring the method’s adaptability to distant image domains is essential. Further evaluation should extend to multimodal understanding and generation, to assess the broader applicability and robustness of TR strategies. Addressing these challenges could advance the de-

Table 11. Quantitative comparison of polyp segmentation.

Methods		Kvasir [25]			ETIS [62]			CVC-ColonDB [54]		
		$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$	$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$	$F_{\beta}^w \uparrow$	$S_{\alpha} \uparrow$	$E_{\phi} \uparrow$
w/o TR	Linear	.482	.782	.747	.327	.726	.680	.535	.816	.787
	Decoder Only	.723	.870	.872	.531	.774	.725	.737	.863	.858
	VPT [26]	.891	.939	.951	.587	.814	.778	.817	.900	.906
	AdaptFormer [4]	.935	<b>.957</b>	<b>.967</b>	<b>.719</b>	<b>.873</b>	<b>.877</b>	<b>.864</b>	<b>.912</b>	<b>.930</b>
	LoRA [23]	<b>.936</b>	.956	.966	.641	.839	.839	.850	.909	.929
	EVP [35]	.857	.934	.943	.618	.846	.831	.806	.898	.907
w/ TR	DyT [65]	.897	.944	.963	.665	.859	.861	.827	.900	.919
	Ours	.938	<b>.960</b>	<b>.975</b>	.681	.864	<b>.871</b>	.862	<b>.917</b>	<b>.936</b>
	Ours <sup>†</sup>	<b>.940</b>	.959	.974	<b>.691</b>	<b>.868</b>	.862	<b>.865</b>	.912	.929

Table 12. Quantitative comparison of polyp segmentation.

	Natural							Specialized				Structured								Avg.
	CIFAR-100	Caltech101	DTD	Flowers102	Pets	SVHN	SUN397	Camelyon	EuroSAT	Resisc45	Retinopathy	Clevr-Count	Clevr-Dist	DMLab	KITTI-Dist	dSpr-Loc	dSpr-Ori	sNORB-Azim	sNORB-Elev	
<i>Parameter-efficient fine-tuning</i>																				
AdaptFormer	70.8	91.2	<b>70.5</b>	<b>98.8</b>	<b>90.9</b>	<b>86.6</b>	<b>54.3</b>	83.0	<b>95.8</b>	<b>84.4</b>	<b>76.3</b>	81.9	64.3	49.3	<b>80.3</b>	76.3	45.7	31.7	41.1	72.3
LoRA	67.1	91.4	69.4	<b>98.8</b>	90.4	85.3	54.0	84.9	95.3	<b>84.4</b>	73.6	<b>82.9</b>	<b>69.2</b>	49.8	78.5	75.7	47.1	31.0	<b>44.0</b>	72.3
VPT	<b>78.8</b>	90.8	65.8	98.0	88.3	78.1	49.6	81.8	96.1	83.4	68.4	68.5	60.0	46.5	72.8	73.6	47.9	32.9	37.8	69.4
<i>Parameter-efficient fine-tuning with token reduction</i>																				
DyT	70.4	94.2	68.6	98.0	90.3	86.5	51.5	<b>87.1</b>	95.3	84.2	72.2	79.2	60.8	51.0	79.9	79.7	<b>55.1</b>	<b>34.0</b>	40.9	<b>72.5</b>
Ours	68.3	<b>94.4</b>	68.8	98.6	89.9	84.7	52.2	<b>87.1</b>	95.7	83.8	72.6	81.1	60.7	<b>51.1</b>	79.9	<b>83.0</b>	50.9	30.0	43.3	72.4

velopment of efficient and accurate vision architectures for diverse real-world applications.

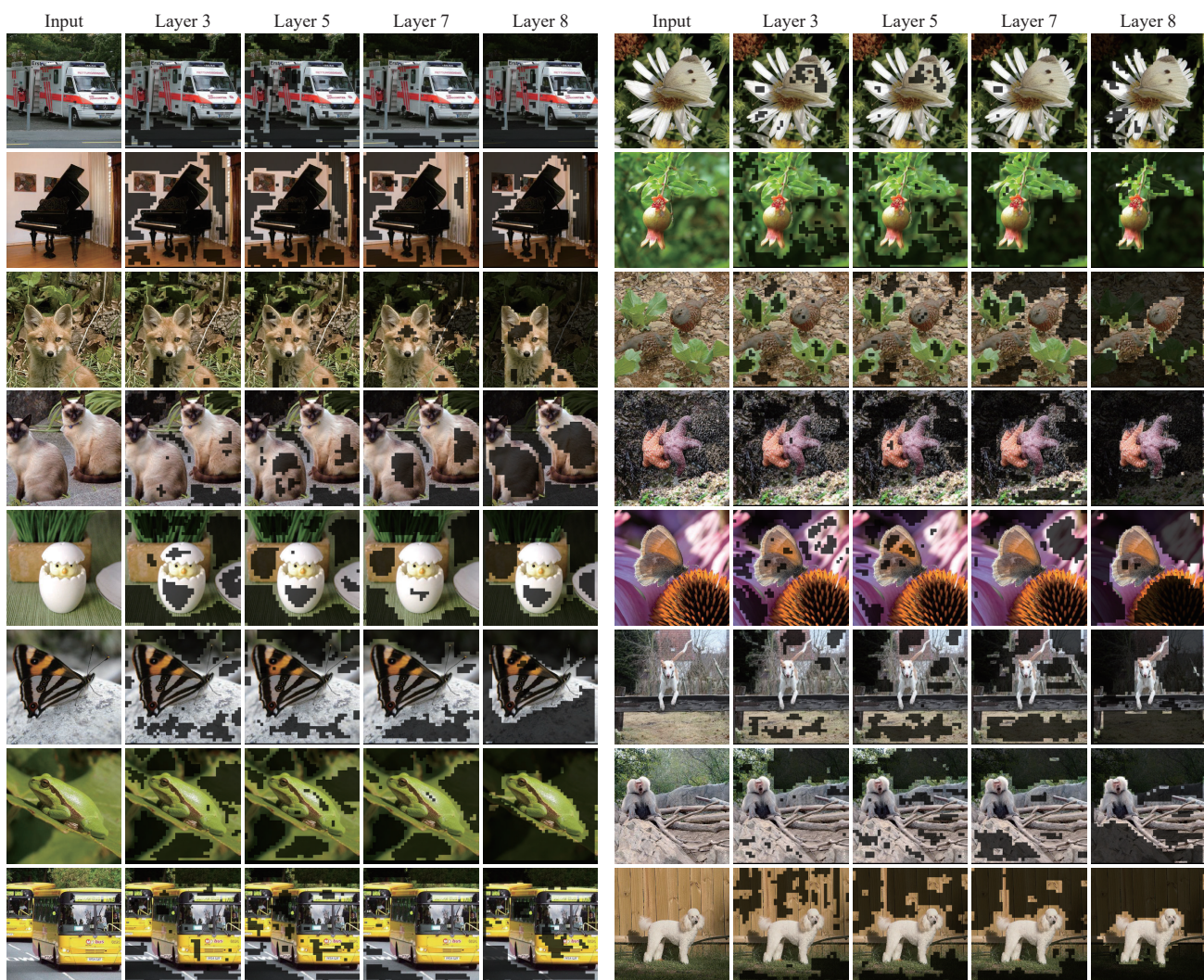


Figure 6. Visualization of the effects of our token reduction on DUTS dataset.



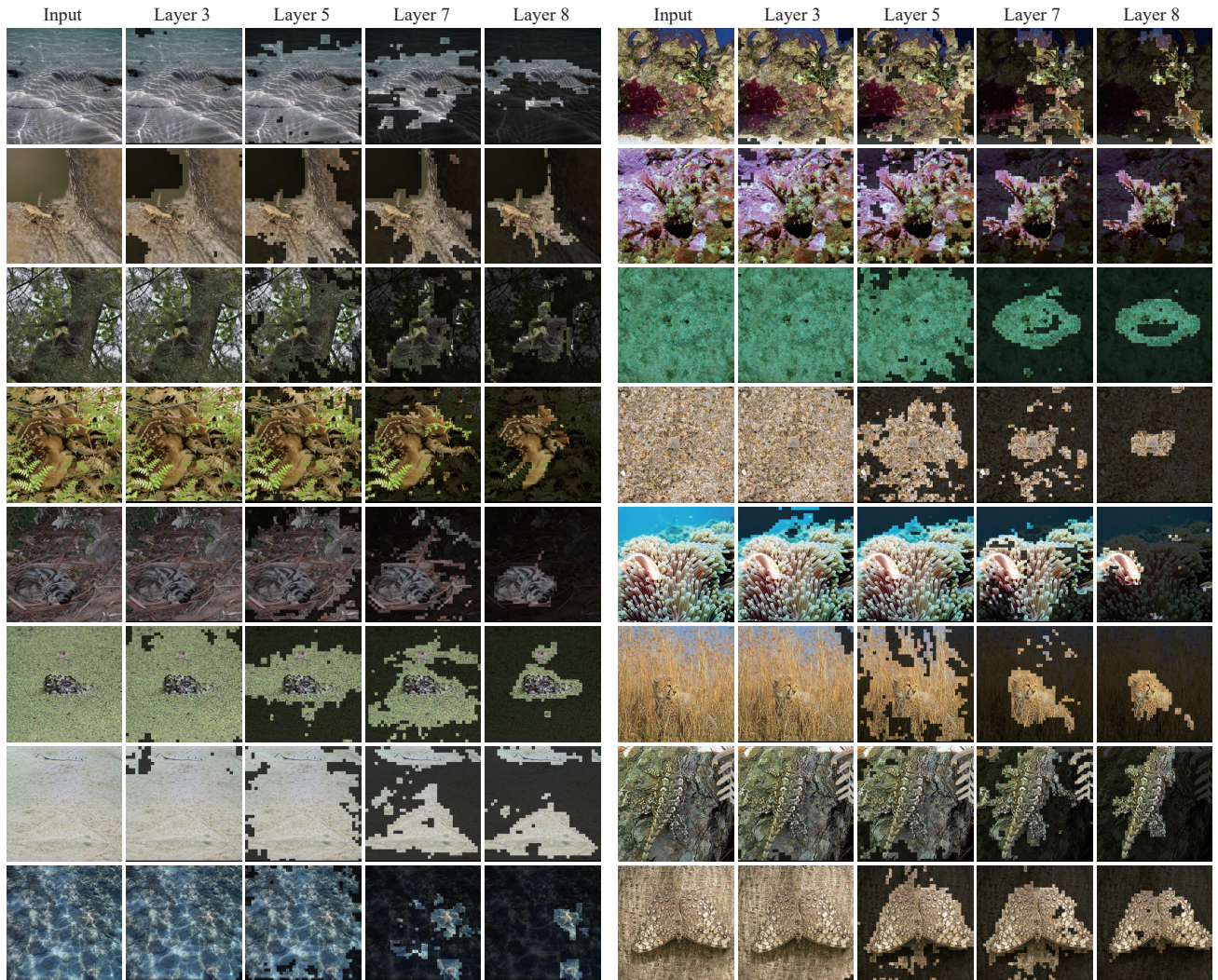


Figure 7. Visualization of the effects of our token reduction on COD10K and CAMO dataset.



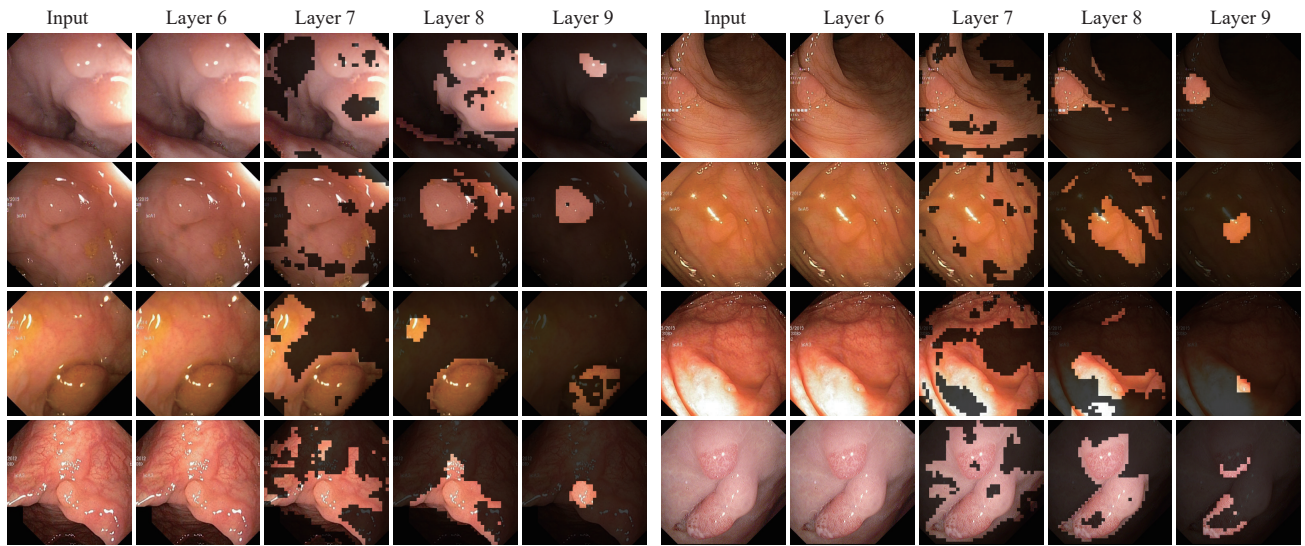


Figure 8. Visualization of the effects of our token reduction on Kvasir dataset.

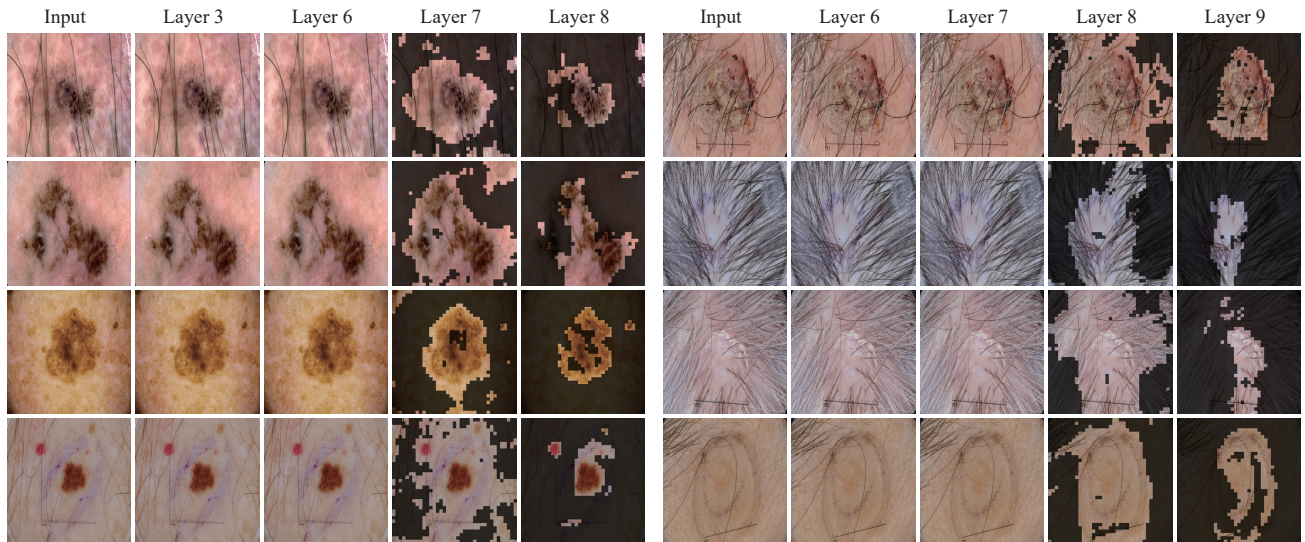


Figure 9. Visualization of the effects of our token reduction on ISIC 2017 dataset.



Figure 10. Visualization of the effects of our token reduction on ISIC 2017 dataset.

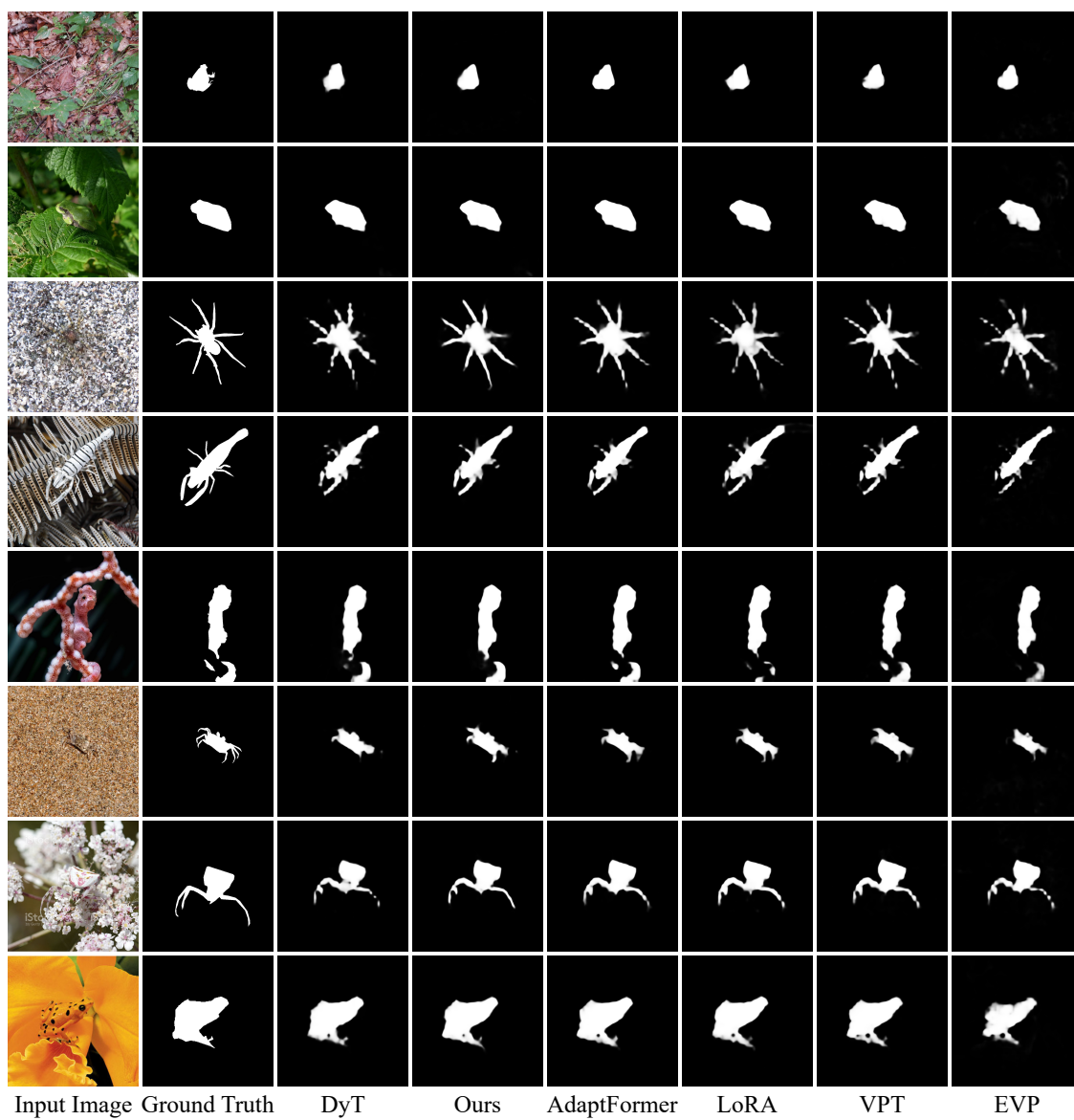


Figure 11. Visualization of the effects of our token reduction on ISIC 2017 dataset.

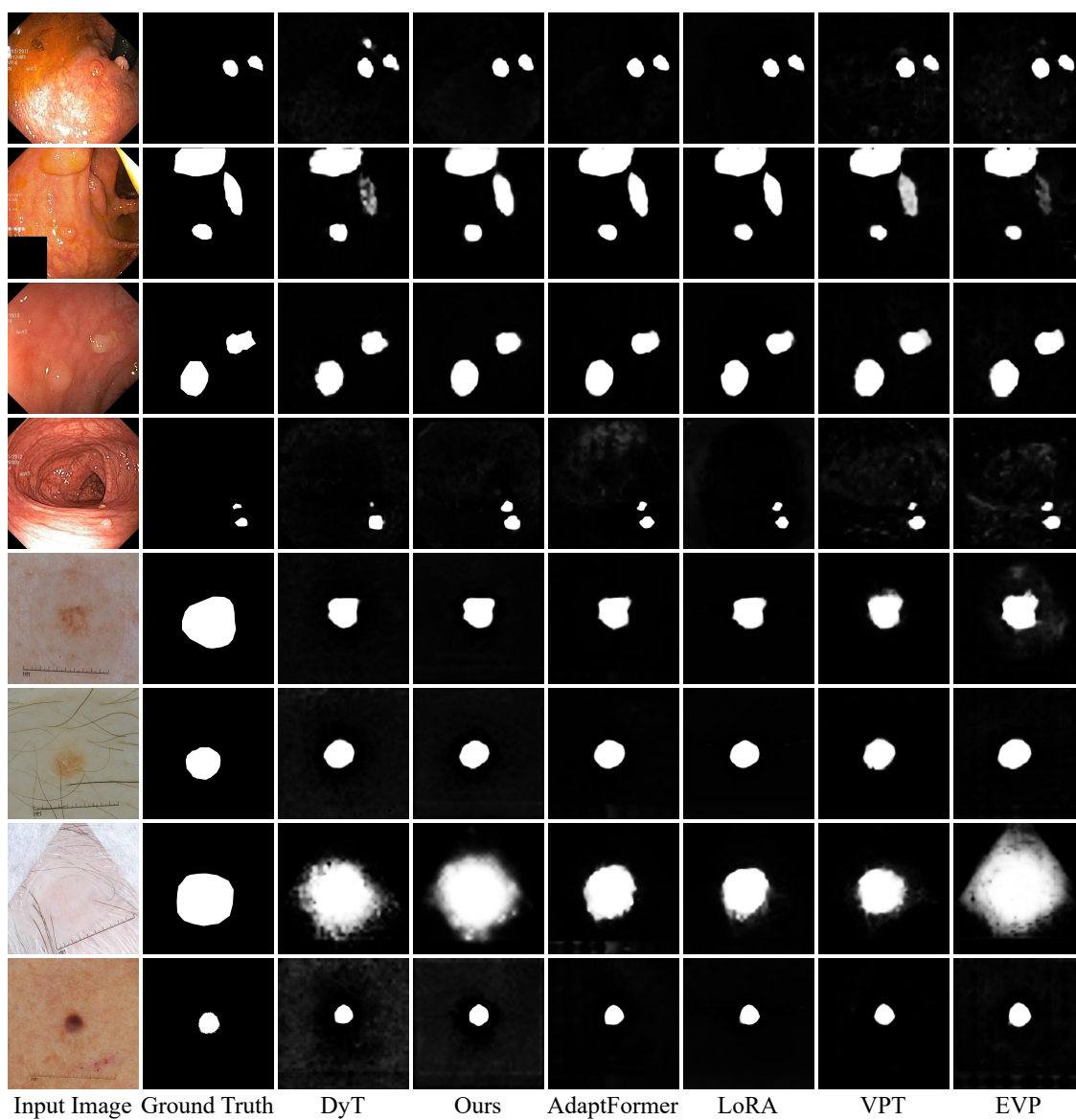


Figure 12. Visualization of the effects of our token reduction on ISIC 2017 dataset.