PIXEL-LEVEL TASKS NEED STATE-AWARE COMPENSATION AND ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Parameter-efficient fine-tuning (PEFT) enables efficient adaptation of pre-trained models to downstream tasks by updating only a small subset of task-specific parameters, but still requires full model computation during inference. Although token reduction (TR) improves inference efficiency by assigning different computational paths to tokens, it introduces partial information loss as a consequence of reduced token computation. This challenge is even more pronounced in pixel-level tasks, which require dense and fine-grained spatial information for accurate prediction. Existing TR-PEFT methods address this challenge by increasing adaptation capacity or introducing explicit compensation pathways. However, these designs typically do not explicitly account for the distinct token states arising from TR-induced computational paths, resulting in a state-agnostic formulation. Our analysis reveals a design gap in state-agnostic approaches under TR, with observable effects on compensation effectiveness and adaptation differentiation across tokens. To address this gap, we propose a state-aware approach that leverages TR-induced token states to regulate compensation and adaptation accordingly. Our approach effectively improves compensation and adaptation, achieving better performance with reduced computational overhead across diverse pixel-level tasks. We further demonstrate competitive or superior performance compared with strong baseline methods.

1 INTRODUCTION

As pre-trained Vision Transformer models are widely applied to diverse downstream vision tasks (Dosovitskiy et al., 2021; Oquab et al., 2024; Touvron et al., 2021; Li et al., 2022; Strudel et al., 2021), the efficiency of both model adaptation and inference has become increasingly important. Parameter-efficient fine-tuning (PEFT) (Jia et al., 2022; Chen et al., 2022) reduces adaptation cost by optimizing only a small set of task-specific parameters, while token reduction (TR) (Rao et al., 2021; Liang et al., 2022; Bolya et al., 2023) improves inference efficiency by reducing token computation. Recent studies have therefore explored combining PEFT with TR to improve adaptation and inference efficiency (Zhao et al., 2024; Liu et al., 2025; Kim et al., 2025).

For pixel-level tasks such as image segmentation, combining PEFT with TR is more challenging because dense prediction depends heavily on fine-grained spatial information that may be lost during TR. To address this challenge, some TR-PEFT methods allow selected tokens to skip feed-forward network (FFN) computation within the frozen backbone without removing them from the sequence (Zhao et al., 2024). DAR (Lei et al., 2025b) further introduces a dedicated Token Compensator to recover TR-induced information loss. Current TR-PEFT methods for pixel-level tasks assign different computational paths to different tokens during TR, resulting in two token states: preserved and skipped. However, as illustrated in Figure 1, existing compensation and adaptation modules remain state-agnostic, without explicitly accounting for the distinct preserved and skipped token states. This mismatch motivates a closer examination of how state-agnostic compensation and adaptation behave under TR.

State-agnostic compensation is first examined from the perspective of TR-induced token states. Since preserved and skipped tokens follow different computational paths, their compensation demands are expected to differ. To quantify the response strength for different token states, we define Relative Response Strength (RRS), following prior work that characterizes residual responses by

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

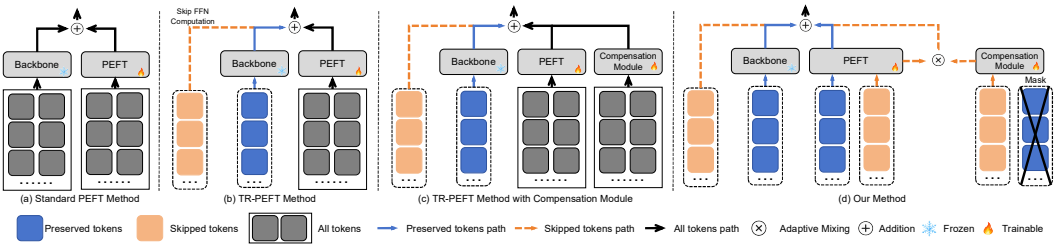

Figure 1: Comparison of adaptation and compensation paradigms in TR-PEFT. (a) Standard PEFT applies adaptation to all tokens without TR. (b) TR-PEFT allows selected tokens to skip FFN computation within the frozen backbone, while PEFT adaptation is still applied to all tokens. (c) TR-PEFT with an explicit Compensation Module introduces an additional compensation pathway for all tokens. (d) Our method uses TR-induced token states to differentiate token processing: preserved tokens follow the original PEFT pathway, while skipped tokens receive selective compensation and adaptive mixing to coordinate compensation with PEFT adaptation.

relative magnitude (Dotzel et al., 2024). RRS is computed as the ℓ_2 norm of a module's residual, normalized by the ℓ_2 norm of the corresponding input token feature. As shown in Figure 2(a), RRS reveals that the existing Token Compensator responds differently to preserved and skipped tokens, with generally stronger responses to skipped tokens.

In addition to characterizing differences in compensation strength, we examine the effectiveness of compensation for skipped tokens. As the Token Compensator is intended to recover information lost by skipping FFN computation, we evaluate compensation for skipped tokens using their corresponding full FFN residuals as references and compute the normalized mean squared error (NMSE) against the compensation residuals. Following prior work that normalizes MSE by the magnitude of the target representation (Deng & Zhang, 2021), NMSE measures the relative discrepancy between the compensation residual and its corresponding full FFN residual. As shown in Figure 2(b), the NMSE remains relatively stable in earlier layers but increases sharply in deeper layers at high skipping rates, suggesting that compensation becomes less effective at approximating the missing FFN residual. Taken together, these results indicate that the Token Compensator can implicitly produce different responses for preserved and skipped tokens, while compensation effectiveness degrades in deeper layers as skipping becomes more aggressive. These observations motivate a state-aware compensation design that explicitly accounts for whether a token is preserved or skipped.

The adaptation pathway offers another view of how the state-agnostic design behaves under TR. Since previous work has observed that TR tends to preserve boundary tokens for FFN computation (Lei et al., 2025b), we further examine whether adaptation also exhibits different response strengths for boundary and non-boundary tokens. Using the same RRS metric, we compare the adaptation strengths of the two token groups before and after introducing TR.

As shown in Figure 2(c)–(d), vanilla AdaptFormer exhibits clear differences in adaptation strength between boundary and non-boundary tokens. Notably, after introducing TR, the difference in adaptation strength between boundary and non-boundary tokens is markedly reduced under the current state-agnostic adaptation design. Together with the performance advantage of vanilla AdaptFormer, this contrast suggests that preserving the original difference in adaptation strength may be beneficial when introducing TR. However, the current state-agnostic adaptation design provides no explicit mechanism for maintaining this distinction after TR is introduced. Since TR itself creates preserved and skipped token states through different computational paths, these states provide an explicit signal for regulating adaptation under TR. This motivates a state-aware adaptation design.

Taken together, these findings suggest that state-agnostic design does not fully accommodate the distinct token states introduced by TR, motivating a state-aware design for compensation and adaptation. We therefore propose TRACE (Token-Reduction-Aware Compensation and Adaptation), a state-aware framework that leverages TR-induced token states to regulate compensation and adaptation. Specifically, preserved tokens retain the original PEFT pathway, while skipped tokens receive selective compensation that is adaptively coordinated with PEFT adaptation. Our main contributions are summarized as follows:

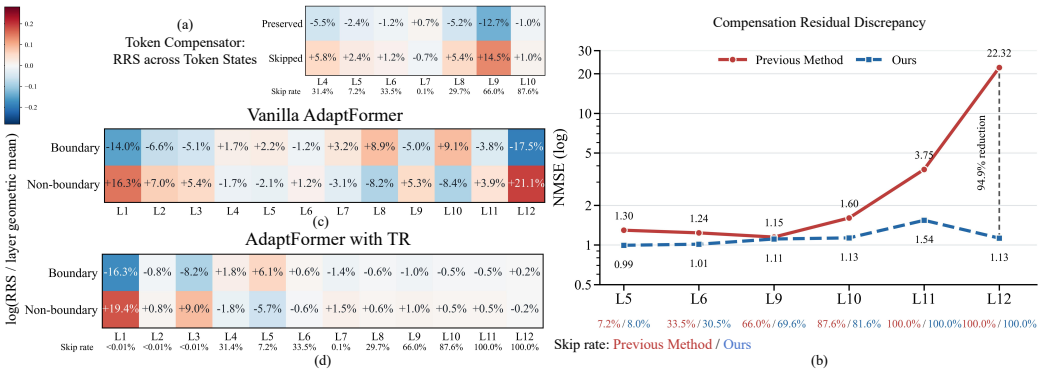

Figure 2: Analysis of state-agnostic compensation and adaptation under TR. (a) RRS of the Token Compensator for preserved and skipped tokens. (b) NMSE between the compensation residual and the corresponding full FFN residual for skipped tokens, comparing the previous method (Lei et al., 2025b) with ours; larger values indicate greater discrepancy. NMSE is shown on a logarithmic y-axis for better visualization. (c)–(d) RRS for boundary and non-boundary tokens in vanilla AdaptFormer and AdaptFormer with TR. In (a), (c), and (d), RRS is normalized by the layer-wise geometric mean; annotated values show relative deviations from this reference, while colors show the logarithm of the same ratio, with red and blue indicating above- and below-reference responses, respectively.

- **State-Aware Compensation and Adaptation.** We systematically analyze TR-PEFT and show that state-agnostic compensation does not consistently maintain effective compensation under aggressive TR, while state-agnostic adaptation does not preserve the original difference in adaptation strength between boundary and non-boundary tokens after TR is introduced.
- **Token-Reduction-Aware Compensation and Adaptation.** Based on these findings, we develop a Selective Compensation Module that focuses compensation on skipped tokens and an Adaptive Mixing Module that adaptively coordinates compensation with PEFT adaptation. A Shared Down-Projection is further introduced to provide a common projection basis for the compensation and adaptation pathways.
- **Performance and Computational Efficiency.** Our method demonstrates strong performance across several pixel-level tasks with substantially reduced computation, outperforming vanilla AdaptFormer on most evaluated datasets while using approximately 19% fewer FLOPs.

2 Related Work

Parameter-Efficient Fine-Tuning. Parameter-efficient fine-tuning (PEFT) adapts pre-trained models by updating only a small subset of parameters and has been widely used in vision tasks. Representative approaches include visual prompt tuning (Jia et al., 2022), adapter-based tuning (Chen et al., 2022), and low-rank adaptation (Hu et al., 2022). Building on these paradigms, subsequent studies have explored more flexible prompt designs (Ren et al., 2025; Huang et al., 2025; Shang et al., 2025) and lightweight task-specific adaptation mechanisms.

Token Reduction. Token reduction (TR) improves the inference efficiency of Vision Transformers by reducing token computation. Early methods mainly focused on token pruning or dynamic token selection (Rao et al., 2021; Liang et al., 2022), followed by broader exploration of token merging (Bolya et al., 2023; Norouzi et al., 2024) and learnable token retention (Chen et al., 2023). Recent approaches further explore task-aware token selection and spatial-preserving compression (Bergner et al., 2025; Mao et al., 2025).

Token Reduction with PEFT for Pixel-Level Tasks. For pixel-level tasks, selectively skipping FFN computation provides a practical token-reduction strategy, as it improves inference efficiency while preserving the complete token sequence. DyT (Zhao et al., 2024) follows this paradigm by allowing selected tokens to skip FFN computation and strengthening the adaptation pathway to mitigate the resulting representation loss. DAR (Lei et al., 2025b) further decouples compensation from adaptation by introducing a dedicated Token Compensator to recover information lost during token

reduction. These works represent two strategies for pixel-level TR-PEFT: compensating through strengthened adaptation or through an explicit compensation branch. Despite this progress, TR-PEFT for pixel-level tasks remains an underexplored direction with considerable room for further development.

3 METHODOLOGY

3.1 REVISITING THE TR-PEFT FRAMEWORK

Let $X_i = [E_i, P_i]$ denote the input token sequence to the i-th Transformer block with TR, where E_i contains the class and register tokens and P_i denotes the patch-token subset. The corresponding transformation can be formulated as

$$X_{i+1} = X_i + \mathcal{M}_{\mathrm{TR}}(X_i), \tag{1}$$

where X_{i+1} denotes the output token sequence of the i-th Transformer block, and $\mathcal{M}_{\mathrm{TR}}(\cdot)$ denotes the backbone transformation under TR. Within $\mathcal{M}_{\mathrm{TR}}(\cdot)$, the Token Selector produces an intermediate token-selection signal $\mathbf{S}_i$, which determines whether each patch token undergoes or skips FFN computation. The special tokens in E_i are excluded from token selection and remain preserved. Existing TR-PEFT methods introduce a PEFT branch in parallel with the backbone transformation under TR for downstream adaptation (Lei et al., 2023; Zhao et al., 2024):

$$X_{i+1} = X_i + \mathcal{M}_{\mathrm{TR}}(X_i) + \mathcal{F}(X_i), \tag{2}$$

where $\mathcal{F}(\cdot)$ denotes the PEFT branch. Under this formulation, the PEFT branch serves not only for task adaptation but also as an implicit compensator for representation degradation introduced by TR (Lei et al., 2025b).

DAR (Lei et al., 2025b) further decouples task adaptation from compensation by assigning them to independent branches. Specifically, the PEFT branch focuses on task adaptation, while an independent compensation branch is introduced to preserve spatial structure and restore high-frequency information. Accordingly, the decoupled TR-PEFT framework can be formulated as

$$X_{i+1} = X_i + \mathcal{M}_{\mathrm{TR}}(X_i) + \mathcal{F}(X_i) + \mathcal{G}(P_i), \tag{3}$$

where $\mathcal{G}(\cdot)$ denotes the independent compensation branch. However, the token-selection signal $\mathbf{S}_i$ is not explicitly propagated to the PEFT or compensation branches, leaving both compensation and PEFT adaptation largely state-agnostic and without explicit coordination between them.

To address this limitation, we establish a state-aware coordination mechanism for compensation and PEFT adaptation. Specifically, the token-selection signal $\mathbf{S}_i$ is propagated beyond computational routing to guide selective compensation and adaptation–compensation coordination. Accordingly, the decoupled formulation is extended as

$$X_{i+1} = X_i + \mathcal{M}_{\mathrm{TR}}(X_i) + \mathcal{A}\left(\mathcal{F}(X_i), \mathcal{G}(P_i, \mathbf{S}_i), \mathbf{S}_i\right). \tag{4}$$

Here, $\mathcal{G}(P_i, \mathbf{S}_i)$ denotes the state-aware compensation branch, which uses the token-selection signal $\mathbf{S}_i$ to selectively compensate skipped tokens. $\mathcal{A}(\cdot)$ denotes the adaptive coordination mechanism that retains the original PEFT response for preserved tokens while adaptively balancing PEFT adaptation and compensation for skipped tokens. For clarity, we define the corresponding token-wise responses as

$$R_i^{\mathrm{peft}} = \mathcal{F}(X_i), \qquad R_i^{\mathrm{scm}} = \mathcal{G}(P_i, \mathbf{S}_i), \qquad R_i^{\mathrm{mix}} = \mathcal{A}\left(R_i^{\mathrm{peft}}, R_i^{\mathrm{scm}}, \mathbf{S}_i\right), \tag{5}$$

where R_i^{peft} and R_i^{scm} denote the PEFT adaptation response and compensation response, respectively. R_i^{mix} denotes the coordinated token response, where the PEFT and compensation responses are adaptively combined.

While preserving the decoupled roles of token selection, compensation, and PEFT adaptation, our formulation introduces an explicit information flow from the token-selection signal to selective compensation and adaptation–compensation coordination.

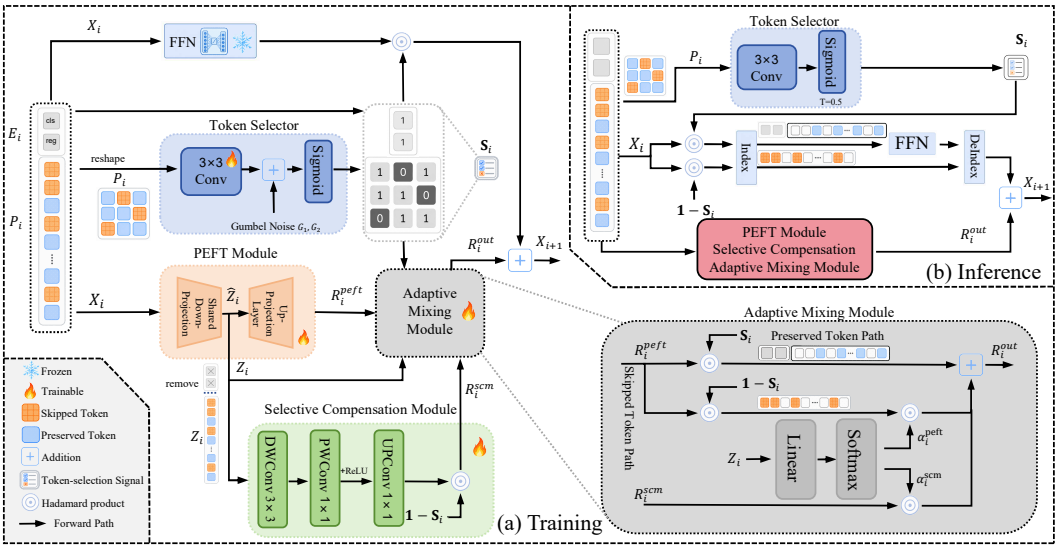

Figure 3: Overall architecture of the proposed method. The token-selection signal generated by the Token Selector guides selective compensation and adaptation-compensation coordination. A Shared Down-Projection provides a shared low-dimensional representation for PEFT adaptation, compensation, and mixing-weight estimation, while the Adaptive Mixing Module adaptively combines PEFT adaptation and compensation for skipped tokens and retains the original PEFT responses of preserved tokens. During inference, only preserved tokens are processed by the FFN to reduce computational cost.

3.2 Our Method

Building on the decoupled TR-PEFT framework, we develop a state-aware architecture that explicitly coordinates token selection, compensation, and PEFT adaptation. As illustrated in Figure 3, the framework consists of a Token Selector, a Selective Compensation Module (SCM), a Shared Down-Projection (SDP), and an Adaptive Mixing Module (AMM). The Token Selector produces a token-selection signal, which is further propagated to guide selective compensation and adaptation–compensation coordination. The detailed designs of these components are described below.

Token Selector. Given the patch-token sequence P_i, the Token Selector first reshapes the patch tokens into their spatial arrangement and predicts token-wise selection scores using a lightweight 3×3 convolution (Lei et al., 2025b). To facilitate optimization during training, Gumbel noise (Herrmann et al., 2020) is introduced before the sigmoid operation, and the resulting probabilities are binarized using a fixed threshold to obtain the token-selection signal $\mathbf{S}_i$:

$$Q_i = \sigma\left(\frac{C_s(P_i) + G_1 - G_2}{\tau}\right), \tag{6}$$

$$(\mathbf{S}_i)_{m,n} = \begin{cases} 0, & (Q_i)_{m,n} < T, \\ 1, & (Q_i)_{m,n} \geq T. \end{cases} \tag{7}$$

Here, $C_s(\cdot)$ denotes the selection predictor, $\sigma(\cdot)$ is the sigmoid function, $G_1, G_2 \sim \mathrm{Gumbel}(0, 1)$, τ is the temperature parameter, and $T = 0.5$ denotes the fixed selection threshold. A value of $(\mathbf{S}_i)_{m,n} = 1$ indicates a preserved patch token that undergoes FFN computation, whereas $(\mathbf{S}_i)_{m,n} = 0$ indicates a skipped patch token.

The class and register tokens are always preserved and are excluded from token selection. During training, gradients are estimated with straight-through estimation (Appendix B.2), and the binary selection signal is applied after the frozen FFN to avoid perturbations caused by its bias terms (Lei et al., 2025b). During inference, the Gumbel noise is removed, and $\mathbf{S}_i$ is used to index the preserved patch tokens before FFN computation and de-index the processed tokens to their original positions afterward, so that only preserved tokens are processed by the FFN.

Shared Down-Projection. To provide a shared low-dimensional basis for PEFT adaptation, compensation, and adaptation–compensation coordination, Shared Down-Projection (SDP) reuses the PEFT down-projection to transform X_i, reducing parameter overhead.

$$\hat{Z}_i = W_{\text{down}} X_i, \qquad Z_i = \text{Patch}(\hat{Z}_i). \tag{8}$$

Here, $\hat{Z}_i$ denotes the complete down-projected token sequence, while Z_i denotes its patch-token subset. The complete projected representation $\hat{Z}_i$ is used for PEFT adaptation, while its patch-token subset Z_i is provided to the compensation branch and the Adaptive Mixing Module.

Selective Compensation Module. To explicitly couple compensation with the token-selection signal, we introduce a Selective Compensation Module (SCM) that applies compensation only to skipped tokens. Using the shared patch-token representation Z_i, SCM first generates an intermediate compensation response through a lightweight convolutional transformation consisting of a 3×3 depth-wise convolution, a 1×1 point-wise convolution, a ReLU activation, and a 1×1 up-projection layer.

$$\widetilde{R}_i^{\text{scm}} = \text{UpConv}\left(\text{ReLU}\left(\text{PWConv}\left(\text{DWConv}(Z_i)\right)\right)\right). \tag{9}$$

The generated compensation response is then modulated by the token-selection signal $\mathbf{S}_i$, so that only skipped tokens receive compensation.

$$R_i^{\text{scm}} = (\mathbf{1} - \mathbf{S}_i) \odot \widetilde{R}_i^{\text{scm}}. \tag{10}$$

Therefore, SCM suppresses compensation for preserved tokens and retains it only for skipped tokens.

Adaptive Mixing Module. To enable state-aware PEFT adaptation, we introduce an Adaptive Mixing Module that estimates token-wise mixing weights from the shared low-dimensional representation. Specifically, the projected feature Z_i is first mapped to two token-wise mixing logits through a linear layer. A softmax operation is then applied to normalize the logits and obtain the mixing weights for the PEFT and compensation branches:

$$[\alpha_i^{\text{peft}}, \alpha_i^{\text{scm}}] = \text{Softmax}\left(W_{\text{mix}} Z_i\right). \tag{11}$$

For skipped tokens, the PEFT and compensation responses are adaptively combined, while preserved tokens directly retain their original PEFT responses.

$$R_i^{\text{mix}} = \mathbf{S}_i \odot R_{i,\text{patch}}^{\text{peft}} + (\mathbf{1} - \mathbf{S}_i) \odot \alpha_i^{\text{peft}} \odot R_{i,\text{patch}}^{\text{peft}} + \alpha_i^{\text{scm}} \odot R_i^{\text{scm}}. \tag{12}$$

The PEFT responses corresponding to the special tokens E_i bypass adaptive mixing and are concatenated back with the mixed patch-token responses afterward.

$$R_i^{\text{out}} = \left[R_{i,E}^{\text{peft}} \,\|_{\text{seq}}\, R_i^{\text{mix}}\right]. \tag{13}$$

Here, W_{mix} denotes the learnable linear projection used to produce two mixing logits from the projected token representation Z_i. α_i^{peft} and α_i^{scm} denote the token-wise mixing weights for the PEFT and compensation branches, respectively. The PEFT responses for patch tokens and the special tokens E_i are represented by $R_{i,\text{patch}}^{\text{peft}}$ and $R_{i,E}^{\text{peft}}$, while $\|_{\text{seq}}$ indicates concatenation along the token dimension. The resulting R_i^{out} therefore realizes state-aware PEFT adaptation by retaining the original PEFT responses for preserved tokens while adaptively balancing PEFT adaptation and compensation for skipped tokens before being added to the Transformer residual pathway.

Spatial Feature Distillation. In addition to the state-aware coordination above, we introduce an auxiliary spatial feature distillation objective to preserve local representations under TR. Specifically, the L12 patch-token features are processed by a lightweight convolutional distillation head and aligned with the corresponding features from the frozen DINOv2 teacher (Oquab et al., 2024) using cosine similarity. The resulting cosine loss is denoted as $\mathcal{L}_{\cos}$. The distillation head is used only during training and removed at inference, introducing no additional inference-time computation. Further architectural and implementation details are provided in the Appendix B.

Loss Function. The overall training objective consists of the segmentation loss, token activation-rate regularization, and feature distillation loss. The segmentation loss combines binary cross-entropy and Dice losses. We adopt the Ada loss from Dynamic-Tuning (Zhao et al., 2024) as $\mathcal{L}_{\text{rate}}$ to regulate

Table 1: Comparison of parameter-efficient fine-tuning methods with and without token reduction (TR) across multiple pixel-level segmentation benchmarks. "Total Params." and "Trainable Params." exclude the decoder, and "Ratio" is computed as "Trainable Params." / "Total Params." with the decoder excluded from both. FLOPs are measured under a unified protocol. The best segmentation results within each TR setting are highlighted in **bold**. ^{LD} denotes the distillation strategy in DAR, and ^{SD} denotes our spatial feature distillation.

TR Setting	Method	Total Params. (M)	Trainable Params. (M) / Ratio (%)	FLOPs (G)	Simple						Complex						Medical					
					DUTS-TE			DUT-OMRON			COD10K			CAMO			ISIC 2017			PolypDB		
					$F_\beta^w\uparrow$	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F_\beta^w\uparrow$	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F_\beta^w\uparrow$	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F_\beta^w\uparrow$	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F_\beta^w\uparrow$	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F_\beta^w\uparrow$	$S_\alpha\uparrow$	$E_\phi\uparrow$
w/o TR	Linear	85.51	0 / 0.00%	151.71	.658	.841	.832	.509	.755	.734	.657	.860	.866	.707	.853	.852	.571	.779	.755	.649	.849	.830
w/o TR	Decoder Only	85.51	0 / 0.00%	155.27	.756	.879	.889	.623	.796	.807	.769	.888	.918	.796	.886	.904	.772	.839	.839	.813	.898	.909
w/o TR	VPT (Jia et al., 2022)	86.10	0.59 / 0.68%	158.35	.896	.931	.938	.784	.873	.877	.816	.903	.936	.850	.908	.933	.860	**.872**	**.883**	.903	.940	.956
w/o TR	AdaptFormer (Chen et al., 2022)	86.70	1.19 / 1.37%	155.07	**.906**	**.937**	**.946**	**.795**	.874	.884	**.826**	**.906**	**.941**	**.866**	**.911**	**.939**	.874	.855	.874	**.927**	**.949**	**.966**
w/o TR	LoRA (Hu et al., 2022)	86.86	1.33 / 1.53%	155.27	.897	.932	.942	.793	**.877**	**.886**	.814	.897	.935	.852	.905	.934	**.880**	.862	.882	.919	.946	.962
w/o TR	EVP (Liu et al., 2023)	86.25	0.74 / 0.86%	156.84	.887	.929	.935	.776	.873	.879	.803	.900	.933	.832	.901	.921	.854	.866	.875	.906	.942	.958
w/ TR	DyT (Zhao et al., 2024)	86.71	1.20 / 1.38%	125.04	.859	.921	.930	.733	.843	.848	.796	.894	.934	.839	.901	.932	.843	.868	.884	.864	.923	.938
w/ TR	DAR (Lei et al., 2025b)	87.86	2.57 / 2.92%	130.01	.895	.933	.944	.779	.869	.883	.810	.899	.935	.842	.902	.932	.871	.862	.881	.913	.945	.963
w/ TR	DAR^{LD} (Lei et al., 2025b)	87.86	2.57 / 2.92%	130.01	.901	.935	.947	.784	.872	.887	.808	.899	.936	.848	**.905**	.933	.875	.863	.882	.920	.947	.964
w/ TR	Ours	87.46	2.17 / 2.48%	125.82	**.912**	**.938**	.949	.798	.875	.892	**.844**	**.908**	**.940**	.853	.902	.931	**.885**	.876	.888	.914	.942	.959
w/ TR	Ours^{SD}	87.46	2.17 / 2.48%	125.82	**.912**	**.938**	**.950**	**.801**	**.879**	**.896**	.837	.907	.937	**.857**	**.905**	**.934**	.864	**.880**	**.890**	**.922**	**.950**	**.967**

the token activation rate toward a target ratio, while $\mathcal{L}_{\cos}$ aligns the student and teacher patch-token representations. The total objective is formulated as

$$\mathcal{L} = \mathcal{L}_{\mathrm{seg}} + \lambda_1 \mathcal{L}_{\mathrm{rate}} + \lambda_2 \mathcal{L}_{\cos}, \tag{14}$$

where the segmentation loss is defined as

$$\mathcal{L}_{\mathrm{seg}} = \lambda_{\mathrm{bce}} \mathcal{L}_{\mathrm{bce}} + \lambda_{\mathrm{dice}} \mathcal{L}_{\mathrm{dice}}. \tag{15}$$

Here, λ_1 and λ_2 balance the token activation-rate regularization and feature-distillation terms, while λ_{bce} and λ_{dice} weight the BCE and Dice losses, respectively. The complete formulations of all loss terms are provided in Appendix B.

4 Experiments

4.1 Experimental Settings

Datasets. We evaluate our method on three types of pixel-level segmentation tasks. For salient object segmentation, we train on DUTS and evaluate on DUTS-TE (Wang et al., 2017) and DUT-OMRON (Yang et al., 2013). For camouflaged object segmentation, we jointly train on the training sets of CAMO (Le et al., 2019) and COD10K (Fan et al., 2022), and evaluate separately on their corresponding test sets. Medical image segmentation is evaluated on ISIC 2017 (Codella et al., 2018) and PolypDB (Jha et al., 2025) using their official splits.

Implementation Details. We adopt DINOv2-B/14 (Oquab et al., 2024) as the frozen visual backbone with an input resolution of 518×518. The newly introduced modules are optimized using AdamW with an initial learning rate of 1.5×10^{-4}, a cosine learning-rate schedule, and a batch size of 10. All experiments are conducted on a single NVIDIA GeForce RTX 3090 GPU, with detailed task-specific configurations provided in Appendix C.

Token Reduction and Training Objective. The Token Selector is applied to all Transformer blocks with a target token activation rate of 0.5, using a Gumbel-Sigmoid temperature of $\tau = 5$ during training and a selection threshold of $T = 0.5$ during inference. We use $\lambda_{\mathrm{bce}} = 1.0$, $\lambda_{\mathrm{dice}} = 0.5$, and $\lambda_1 = 2.0$. Ours^{SD} additionally uses $\lambda_2 = 0.1$ for spatial feature distillation.

Evaluation Metrics. We evaluate segmentation performance using three widely adopted metrics: weighted F-measure (F_β^w) (Margolin et al., 2014), S-measure (S_α) (Fan et al., 2017), and mean E-measure (E_ϕ) (Fan et al., 2018). To assess computational efficiency, we further report the total number of parameters, the number and ratio of trainable parameters, and FLOPs. FLOPs are measured at an input resolution of 518×518. Baseline results are reproduced or taken from prior work, with details provided in Appendix C.

Table 2: Ablation study of the proposed components. We evaluate the individual and combined effects of SCM, SDP, and AMM under different mixing strategies on DUTS-TE and ISIC 2017.

Setting	SCM	SDP	AMM	Mixing	DUTS-TE			ISIC 2017		
					$F_\beta^w \uparrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$
No TR	–	–	–	–	.906	.937	.946	.874	.855	.874
DAR	×	×	×	–	.895	.933	.944	.871	.862	.881
Variant	✓	×	×	–	.901	.936	.947	.880	.873	.887
Variant	✓	✓	×	–	.899	.937	.948	.874	.870	.887
Variant	✓	×	✓	Adaptive	.900	.934	.944	.877	.874	.884
Variant	✓	✓	✓	Fixed 0.5	.906	.937	.948	.865	.872	.878
Ours	✓	✓	✓	Adaptive	**.912**	**.938**	**.949**	**.885**	**.876**	**.888**

Table 3: Performance comparison of DAR and our method under different token activation rates on DUTS-TE and ISIC 2017. The best results are highlighted in **bold**.

Method	Token Activation Rate	DUTS-TE			ISIC 2017		
		$F_\beta^w \uparrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$
DAR	0.3	.886	.923	.935	.862	.847	.850
	0.5	.895	.933	.944	.871	.862	.881
	0.7	.902	.937	.947	.885	.863	.878
Ours	0.3	.892	.920	.931	.868	.865	.878
	0.5	**.912**	.938	**.949**	.885	**.876**	**.888**
	0.7	.911	**.939**	**.949**	**.887**	**.876**	.884

4.2 Quantitative Results

Comparison against Standard PEFT Baselines. As shown in Tab. 1, our method achieves competitive or better segmentation performance while substantially reducing computational cost compared with standard PEFT baselines. On DUTS-TE, Ours outperforms AdaptFormer across all three metrics while reducing FLOPs from 155.07 G to 125.82 G, a reduction of 29.25 G. Competitive or improved performance is also observed on DUT-OMRON and COD10K, further supporting a favorable performance–efficiency trade-off.

Comparison against Token-Reduced PEFT Baselines. Compared with existing token-reduced PEFT approaches, Ours achieves stronger overall segmentation performance at a comparable computational cost. Its FLOPs remain close to the lowest-cost token-reduced baseline. Overall, Ours achieves a favorable performance–efficiency trade-off under token reduction.

Performance across Pixel-Level Tasks. Beyond salient object segmentation, our method also performs well on camouflaged object and medical image segmentation benchmarks. Ours achieves strong results on COD10K and remains competitive on CAMO. On medical image segmentation benchmarks, Ours achieves competitive performance, with improvements on several metrics. The distilled variant further improves performance in several settings, suggesting that spatial feature distillation can provide complementary gains. These results support the applicability of the proposed design across diverse pixel-level segmentation tasks.

4.3 Approximating Missing FFN Residuals

As discussed in the Introduction, the missing FFN residual provides a natural reference for evaluating compensation mismatch. Figure 2(b) compares this discrepancy between the previous method and ours, showing that TRACE yields a substantially smaller mismatch, particularly in deeper Transformer blocks where token skipping becomes more aggressive. At L12, the NMSE decreases from 22.32 to 1.13, corresponding to a 94.9% reduction in compensation mismatch. The discrepancy also remains relatively low in deeper layers with high skipping rates. These results suggest that the state-aware compensation in TRACE provides a closer and more stable approximation of the missing FFN residual, particularly under high token-skipping rates in deeper layers.

4.4 Ablation Study

We conduct ablation studies on DUTS-TE and ISIC 2017 to examine the contribution of each proposed component and the performance of our method under different token activation rates.

Effect of Core Components. As shown in Tab. 2, replacing the original compensation design with SCM consistently improves performance on both DUTS-TE and ISIC 2017, demonstrating the effectiveness of state-aware compensation for skipped tokens. Adding either SDP or adaptive mixing individually leads to only limited or inconsistent improvements, suggesting that their benefits become more evident when the proposed components are jointly integrated. To evaluate the effect of adaptive mixing, we further fix both AMM weights, α_i^{peft} and α_i^{scm}, to 0.5. When SCM, SDP, and AMM are jointly employed, adaptive mixing achieves better performance than fixed mixing on

both datasets, leading to the best overall results. These results highlight the complementarity of the proposed components.

Effect of Token Activation Rate. Tab. 3 compares the previous method with our method under different target token activation rates. Our method achieves its best overall performance at an activation rate of 0.5, while reducing the rate to 0.3 degrades performance and increasing it to 0.7 provides little additional benefit. These results suggest that an activation rate of 0.5 is sufficient for our method to approach performance saturation.

4.5 Zero-Shot Generalization

To evaluate cross-task generalization, we directly apply the model trained on DUTS for salient object segmentation to CAMO and COD10K without further fine-tuning. As shown in Tab. 4, we compare our method with supervised (S), weakly-supervised (WS), and zero-shot (ZS) approaches. Our method achieves competitive cross-task generalization across both datasets, with spatial feature distillation further improving zero-shot performance. These results indicate that the learned representations can transfer from salient object segmentation to camouflaged object segmentation without task-specific fine-tuning. Further details and results are provided in the Appendix C.2.2.

Table 4: Zero-shot comparison on CAMO and COD10K. **Bold** values indicate the best ZS results.

Method	Sup.	CAMO			COD10K		
		$F_\beta^w \uparrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$
SINet (Fan et al., 2020)	S	.606	.751	.829	.551	.771	.806
ZoomNeXt (Pang et al., 2024)	S	.857	.889	.945	.827	.898	.956
CRNet (He et al., 2023)	WS	.641	.735	.815	.576	.733	.805
GenSAM (Hu et al., 2024)	ZS	.659	.719	.775	.681	.775	.838
CaMF (Lei et al., 2025a)	ZS	.729	.788	.814	.717	.808	.832
Ours	ZS	.728	.800	.820	.749	.825	.855
Ours^{SD}	ZS	**.746**	**.815**	**.848**	**.750**	**.834**	**.866**

4.6 Visualization

Segmentation Visualization. As shown in Fig. 4(a), our method produces more complete object boundaries and preserves finer structural details.

Token-Selection Visualization. Fig. 4(b) shows that boundary tokens tend to be preserved in deeper layers. Additional visualizations are provided in Appendix D.

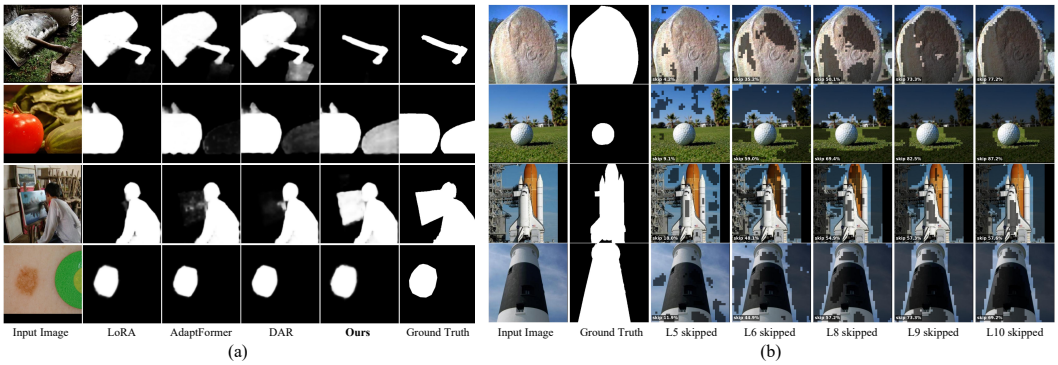

Figure 4: Qualitative visualization of (a) segmentation results and (b) layer-wise token selection.

5 Conclusion

In this paper, we revisit TR-PEFT from the perspective of TR-induced token states and show that state-agnostic compensation and adaptation do not fully account for these states, affecting compensation effectiveness and adaptation differentiation. To address these issues, we introduce TRACE, a state-aware TR-PEFT framework that explicitly regulates compensation and adaptation according to token states, improving compensation effectiveness while adaptively modulating adaptation. Extensive experiments demonstrate that TRACE achieves a favorable performance–efficiency trade-off and strong zero-shot generalization across pixel-level tasks.

AI USE STATEMENT

In this work, we used generative AI tools to assist with code generation and editing during implementation. We did not use generative AI tools to develop the core research ideas, methodological design, experimental design, or interpretation of the experimental results. Additionally, we used generative AI tools to assist with drafting parts of the manuscript and polishing the writing for clarity and readability. We also used generative AI tools to assist with literature search and reference checking. All AI-assisted code was manually reviewed, verified, and tested by the authors, and all AI-assisted text was carefully reviewed and revised. We take responsibility for the final content of this work, including text, claims, code, and other artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed architecture, training objectives, and optimization procedure in the main paper. The experimental section specifies the datasets, evaluation metrics, backbone configuration, token-reduction settings, and major hyperparameters used in our experiments. Additional implementation details, training configurations, and supplementary analyses are provided in the Appendix. To further support reproducibility, anonymous source code and configuration files will be provided to the reviewers and Area Chairs during the discussion period.

REFERENCES

Benjamin Bergner, Christoph Lippert, and Aravindh Mahendran. Token cropr: Faster vits for quite a few tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9740–9750, June 2025.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=JroZRaRw7Eu.

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 (ICCV)*, pp. 17164–17174, October 2023.

Shoufa Chen, Chongjian GE, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 16664–16678. Curran Associates, Inc., 2022. doi: 10.52202/068431-1212. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/69e2f49ab0837b71b0e0cb7c555990f8-Paper-Conference.pdf.

Noel C. F. Codella, David Gutman, M. Emre Celebi, Brian Helba, Michael A. Marchetti, Stephen W. Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, and Allan Halpern. 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)*, pp. 168–172, 2018. doi: 10.1109/ISBI.2018.8363547.

Xiang Deng and Zhongfei Zhang. Comprehensive knowledge distillation with causal intervention. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 22158–22170. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/b9f35816f460ab999cbc168c4da26ff3-Paper.pdf.

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. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Jordan Dotzel, Carly Jiang, Mohamed Abdelfattah, and Zhiru Zhang. Opportunities for post-training dynamic layer sparsity in large vision and language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 8280–8284, June 2024.

Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 4558–4567, 2017. doi: 10.1109/ICCV.2017.487.

Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. Enhanced-alignment measure for binary foreground map evaluation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, pp. 698–704. AAAI Press, 2018. ISBN 9780999241127.

Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Deng-Ping Fan, Ge-Peng Ji, Ming-Ming Cheng, and Ling Shao. Concealed object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6024–6042, 2022. doi: 10.1109/TPAMI.2021.3085766.

Ruozhen He, Qihua Dong, Jiaying Lin, and Rynson W.H. Lau. Weakly-supervised camouflaged object detection with scribble annotations. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i1.25156. URL https://doi.org/10.1609/aaai.v37i1.25156.

Charles Herrmann, Richard Strong Bowen, and Ramin Zabih. Channel selection using gumbel softmax. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*, pp. 241–257, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58583-9.

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. URL https://openreview.net/forum?id=nZeVKeeFYf9.

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 Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pp. 12511–12518. AAAI Press, 2024. doi: 10.1609/AAAI.V38I11.29144. URL https://doi.org/10.1609/aaai.v38i11.29144.

Lingyun Huang, Jianxu Mao, Junfei Yi, Ziming Tao, and Yaonan Wang. Cvpt: Cross visual prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 848–858, October 2025.

Debesh Jha, Nikhil Kumar Tomar, Vanshali Sharma, Quoc-Huy Trinh, Koushik Biswas, Hongyi Pan, Ritika K. Jha, Gorkem Durak, Alexander Hann, Jonas Varkey, Hang Viet Dao, Long Van Dao, Binh Phuc Nguyen, Nikolaos Papachrysos, Brandon Rieders, Peter Thelin Schmidt, Enrik Geissler, Tyler Berzin, Pål Halvorsen, Michael A. Riegler, Thomas de Lange, and Ulas Bagci. PolypDB: A curated multi-center dataset for development of AI algorithms in colonoscopy, 2025. URL https://arxiv.org/abs/2409.00045.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, pp. 709–727, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19827-4.

Kwonyoung Kim, Jungin Park, Jin Kim, Hyeongjun Kwon, and Kwanghoon Sohn. Faster parameter-efficient tuning with token redundancy reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30189–30198, June 2025.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4015–4026, October 2023.

Trung-Nghia Le, Tam V. Nguyen, Zhongliang Nie, Minh-Triet Tran, and Akihiro Sugimoto. Anabranch network for camouflaged object segmentation. *Computer Vision and Image Understanding*, 184:45–56, 2019. ISSN 1077-3142. doi: https://doi.org/10.1016/j.cviu.2019.04.006. URL https://www.sciencedirect.com/science/article/pii/S1077314219300608.

Cheng Lei, Jie Fan, Xinran Li, Tian-Zhu Xiang, Ao Li, Ce Zhu, and Le Zhang. Towards real zero-shot camouflaged object segmentation without camouflaged annotations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(12):11990–12004, 2025a. doi: 10.1109/TPAMI.2025.3600461.

Cheng Lei, Ao Li, Hu Yao, Ce Zhu, and Le Zhang. Rethinking token reduction with parameter-efficient fine-tuning in vit for pixel-level tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14954–14964, June 2025b.

Tao Lei, Junwen Bai, Siddhartha Brahma, Joshua Ainslie, Kenton Lee, Yanqi Zhou, Nan Du, Vincent Zhao, Yuexin Wu, Bo Li, Yu Zhang, and Ming-Wei Chang. Conditional adapters: Parameter-efficient transfer learning with fast inference. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 8152–8172. Curran Associates, Inc., 2023. doi: 10.52202/075280-0357. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/19d7204af519eae9993f7f72377a0ec0-Paper-Conference.pdf.

Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, pp. 280–296, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-20077-9.

Youwei Liang, Chongjian GE, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. EVit: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=BjyvwnXXVn_.

Nian Liu, Ni Zhang, Kaiyuan Wan, Ling Shao, and Junwei Han. Visual saliency transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4722–4732, October 2021.

Ting Liu, Xuyang Liu, Liangtao Shi, Zunnan Xu, Yue Hu, Siteng Huang, Yi Xin, Bineng Zhong, and Donglin Wang. Sparse-tuning: Adapting vision transformers with efficient fine-tuning and inference, 2025. URL https://arxiv.org/abs/2405.14700.

Weihuang Liu, Xi Shen, Chi-Man Pun, and Xiaodong Cun. Explicit visual prompting for low-level structure segmentations. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19434–19445, 2023. doi: 10.1109/CVPR52729.2023.01862.

Yunqiu Lv, Jing Zhang, Yuchao Dai, Aixuan Li, Bowen Liu, Nick Barnes, and Deng-Ping Fan. Simultaneously localize, segment and rank the camouflaged objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11591–11601, June 2021.

Junzhu Mao, Yang Shen, Jinyang Guo, Yazhou Yao, Xiansheng Hua, and Hengtao Shen. Prune and merge: Efficient token compression for vision transformer with spatial information preserved. *IEEE Transactions on Multimedia*, 27:4670–4683, 2025. doi: 10.1109/TMM.2025.3535405.

Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2014. doi: 10.1109/CVPR.2014.39.

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*, pp. 49–56, 2010. doi: 10.1109/CVPRW.2010.5543739.

Narges Norouzi, Svetlana Orlova, Daan de Geus, and Gijs Dubbelman. Algm: Adaptive local-then-global token merging for efficient semantic segmentation with plain vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15773–15782, June 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features Without Supervision. *Transactions on Machine Learning Research*, 2024. URL https://mlanthology.org/tmlr/2024/oquab2024tmlr-dinov2/.

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*, 46(12):9205–9220, 2024. doi: 10.1109/TPAMI.2024.3417329.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 13937–13949. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/747d3443e319a22747fbb873e8b2f9f2-Paper.pdf.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollar, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 28085–28128, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/45c1f6a8cbf2da59ebf2c802b4f742cd-Paper-Conference.pdf.

Li Ren, Chen Chen, Liqiang Wang, and Kien Hua. Da-vpt: Semantic-guided visual prompt tuning for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4353–4363, June 2025.

Chikai Shang, Mengke Li, Yiqun Zhang, Zhen Chen, Jinlin Wu, Fangqing Gu, Yang Lu, and Yiu-Ming Cheung. Pro-vpt: Distribution-adaptive visual prompt tuning via prompt relocation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1558–1568, October 2025.

Przemysław Skurowski, Hassan Abdulameer, Jakub Błaszczyk, Tomasz Depta, Adam Kornacki, and Przemysław Kozieł. Animal camouflage analysis: CHAMELEON database, 2018. URL https://www.polsl.pl/rau6/chameleon-database-animal-camouflage-analysis/. Unpublished manuscript.

Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7262–7272, October 2021.

Lv Tang, Peng-Tao Jiang, Zhi-Hao Shen, Hao Zhang, Jin-Wei Chen, and Bo Li. Chain of visual perception: Harnessing multimodal large language models for zero-shot camouflaged object detection. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM

'24, pp. 8805–8814, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400706868. doi: 10.1145/3664647.3680730. URL https://doi.org/10.1145/3664647.3680730.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10347–10357. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/touvron21a.html.

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 *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3796–3805, 2017. doi: 10.1109/CVPR.2017.404.

Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3166–3173, 2013. doi: 10.1109/CVPR.2013.407.

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. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 114765–114796. Curran Associates, Inc., 2024. doi: 10.52202/079017-3643. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/d0241a0fb1fc9be477bdfde5e0da276a-Paper-Conference.pdf.

A Analysis Details

A.1 Relative Response Strength

To quantify the response strength of a module relative to the magnitude of its input token feature, we define the Relative Response Strength (RRS), inspired by prior analyses based on relative residual magnitude (Dotzel et al., 2024). For the j-th patch token in the i-th Transformer block, RRS is defined as

$$\mathrm{RRS}_{i,j} = \frac{\|R_{i,j}\|_2}{\|H_{i,j}\|_2 + \epsilon}, \tag{16}$$

where $H_{i,j}$ denotes the input feature of the analyzed module, $R_{i,j}$ denotes the corresponding module response, and ϵ is a small constant for numerical stability. For PEFT adaptation, $R_{i,j}$ corresponds to the patch-token response in R_i^{peft}; for compensation, it corresponds to the response produced by the compensation module. All statistics are computed over patch tokens only.

Preserved and skipped tokens. The token-selection signal $\mathbf{S}_i$ partitions patch tokens into two TR-induced token states. Tokens with a selection value of 1 are categorized as preserved tokens and undergo FFN computation, whereas tokens with a value of 0 are categorized as skipped tokens and bypass FFN computation. Let $\mathcal{T}_{s,i}$ denote the set of all patch tokens in state s at block i over the entire evaluation set, where $s \in \{\mathrm{preserved}, \mathrm{skipped}\}$, and $|\mathcal{T}_{s,i}|$ denotes the number of tokens in this set. The mean RRS is computed as

$$\bar{r}_{s,i} = \frac{\sum_{j \in \mathcal{T}_{s,i}} \mathrm{RRS}_{i,j}}{|\mathcal{T}_{s,i}|}. \tag{17}$$

Boundary and non-boundary tokens. For the spatial analysis, a patch is categorized as a boundary token if it overlaps both the ground-truth region and the region outside the ground truth. All remaining patches are categorized as non-boundary tokens.

The analysis is conducted on DUTS-TE using image-level aggregation. For image n and Transformer block i, let $\mathcal{B}_{n,i}$ and $\mathcal{N}_{n,i}$ denote the boundary and non-boundary token sets, respectively. Their image-level response strengths are computed as

$$r_{\mathrm{bnd}}^{(n,i)} = \frac{1}{|\mathcal{B}_{n,i}|} \sum_{j \in \mathcal{B}_{n,i}} \mathrm{RRS}_{i,j}, \tag{18}$$

and

$$r_{\mathrm{non}}^{(n,i)} = \frac{1}{|\mathcal{N}_{n,i}|} \sum_{j \in \mathcal{N}_{n,i}} \mathrm{RRS}_{i,j}. \tag{19}$$

The image-level statistics are then averaged over the dataset:

$$\bar{r}_{\mathrm{bnd},i} = \frac{1}{N} \sum_{n=1}^{N} r_{\mathrm{bnd}}^{(n,i)}, \qquad \bar{r}_{\mathrm{non},i} = \frac{1}{N} \sum_{n=1}^{N} r_{\mathrm{non}}^{(n,i)}. \tag{20}$$

Layer-wise normalization. For visualization, we normalize the mean RRS of the two groups under comparison by their geometric mean. For each Transformer block i, let $\bar{r}_{a,i}$ and $\bar{r}_{b,i}$ denote the mean RRS values of the two groups, corresponding to either preserved versus skipped tokens or boundary versus non-boundary tokens. The geometric-mean reference is defined as

$$g_i = \sqrt{\bar{r}_{a,i} \bar{r}_{b,i}}. \tag{21}$$

The log-normalized value used for heatmap coloring is defined as

$$z_{s,i} = \log \frac{\bar{r}_{s,i}}{g_i}, \tag{22}$$

and the percentage displayed in each cell is

$$\Delta_{s,i} = \left(\frac{\bar{r}_{s,i}}{g_i} - 1 \right) \times 100\%. \tag{23}$$

For both $z_{s,i}$ and $\Delta_{s,i}$, positive and negative values indicate response strengths above and below the within-block geometric-mean reference, respectively.

A.2 Normalized Mean Squared Error

To quantify how closely the compensation residual approximates the corresponding full FFN residual, we use normalized mean squared error (NMSE), following prior feature-matching analyses that normalize reconstruction error by the energy of the target representation (Deng & Zhang, 2021).

For the j-th skipped patch token in the i-th Transformer block, let $R_{i,j}^{\mathrm{ffn}}$ denote the full FFN residual obtained from the original FFN path, and let $R_{i,j}^{\mathrm{comp}}$ denote the compensation residual used by the model. The full FFN residual is computed only as a reference for analysis and is not used during normal inference.

Only skipped patch tokens are included in the calculation. Let $\mathcal{S}_i$ denote the set of all skipped patch tokens at block i over the entire evaluation set. The NMSE is computed as

$$\mathrm{NMSE}_i = \frac{\sum_{j \in \mathcal{S}_i} \left\| R_{i,j}^{\mathrm{comp}} - R_{i,j}^{\mathrm{ffn}} \right\|_2^2}{\sum_{j \in \mathcal{S}_i} \left\| R_{i,j}^{\mathrm{ffn}} \right\|_2^2 + \epsilon}. \tag{24}$$

A lower NMSE indicates that the compensation residual more closely approximates the omitted full FFN residual. The numerator and denominator are accumulated over all skipped patch tokens in DUTS-TE before computing the final ratio. The block-wise NMSE values in Figure 2(b) are displayed on a logarithmic vertical scale for visualization. The logarithmic scale does not alter the underlying NMSE values.

B Additional Method Details

B.1 Decoder

Following prior work on TR-PEFT for pixel-level tasks (Lei et al., 2025b), we employ a lightweight MLP-based decoder to minimize additional task-specific inductive bias. Given the final backbone representation, the decoder is formulated as

$$\mathrm{Decoder}(\cdot) = \mathrm{MLP}\left(\mathrm{Up}\left(\mathrm{MLP}(\cdot)\right)\right), \tag{25}$$

where $\mathrm{Up}(\cdot)$ denotes bilinear upsampling. The same decoder architecture is used across all compared methods.

B.2 Training Objective And details

Training Objective. The overall training objective consists of the segmentation loss, the token activation-rate regularization, and the spatial feature distillation loss:

$$\mathcal{L} = \mathcal{L}_{\mathrm{seg}} + \lambda_1 \mathcal{L}_{\mathrm{rate}} + \lambda_2 \mathcal{L}_{\cos}, \tag{26}$$

where λ_1 and λ_2 control the contributions of the token activation-rate regularization and spatial feature distillation losses, respectively.

The segmentation loss combines binary cross-entropy (BCE) and Dice loss:

$$\mathcal{L}_{\mathrm{seg}} = \lambda_{\mathrm{bce}} \mathcal{L}_{\mathrm{bce}} + \lambda_{\mathrm{dice}} \mathcal{L}_{\mathrm{dice}}. \tag{27}$$

Given the ground-truth mask Y and the predicted probability map $\hat{Y}$, the BCE loss is defined as

$$\mathcal{L}_{\mathrm{bce}} = -\frac{1}{HW} \sum_{m=1}^{H} \sum_{n=1}^{W} \left[Y_{m,n} \log \hat{Y}_{m,n} + (1 - Y_{m,n}) \log(1 - \hat{Y}_{m,n}) \right], \tag{28}$$

and the Dice loss is

$$\mathcal{L}_{\mathrm{dice}} = 1 - \frac{\epsilon + 2 \sum_{m=1}^{H} \sum_{n=1}^{W} Y_{m,n} \hat{Y}_{m,n}}{\epsilon + \sum_{m=1}^{H} \sum_{n=1}^{W} \left(Y_{m,n} + \hat{Y}_{m,n} \right)}, \tag{29}$$

where ϵ is a small constant for numerical stability.

Following DyT (Zhao et al., 2024), we use the activation-rate regularization $\mathcal{L}_{\mathrm{rate}}$ to constrain the average token-selection rate toward the target rate k:

$$\mathcal{L}_{\mathrm{rate}} = \left(\frac{1}{Nhw} \sum_{i=1}^{N} \sum_{m=1}^{h} \sum_{n=1}^{w} (\mathbf{S}_i)_{m,n} - k \right)^2, \tag{30}$$

where $\mathbf{S}_i$ denotes the patch-token selection signal at the i-th Transformer block, N denotes the number of Transformer blocks, and $h \times w$ denotes the spatial resolution of the patch-token grid.

Token Selector training details. During training, we use a straight-through estimator: the forward pass uses a binary mask, while gradients to the Token Selector flow through the Gumbel-Sigmoid probabilities.

B.3 Spatial Feature Distillation Details

We employ spatial feature distillation heads to align the final-layer patch-token representations of the student model with those of a frozen pre-trained DINOv2 (Oquab et al., 2024) teacher. Let

$$F_s, F_t \in \mathbb{R}^{B \times N_p \times C} \tag{31}$$

denote the normalized patch-token features from the final Transformer block of the student and teacher, respectively, where B is the batch size, N_p is the number of patch tokens, and C is the feature dimension.

The spatial feature distillation head first restores the token sequence to its two-dimensional patch grid:

$$F \in \mathbb{R}^{B\times N_p\times C} \to \widetilde{F} \in \mathbb{R}^{B\times C\times h\times w}, \qquad hw = N_p. \tag{32}$$

The reshaped feature is then processed by a lightweight spatial projection head:

$$\widetilde{H} = \mathrm{Conv}^{(2)}_{1\times 1}\left(\mathrm{DWConv}_{3\times 3}\left(\mathrm{ReLU}\left(\mathrm{Conv}^{(1)}_{1\times 1}(\widetilde{F})\right)\right)\right), \tag{33}$$

followed by flattening back to the token sequence:

$$H = \mathrm{Flatten}(\widetilde{H}) \in \mathbb{R}^{B\times N_p\times C}. \tag{34}$$

The first 1×1 convolution performs channel projection, the 3×3 depthwise convolution aggregates local spatial information from neighboring patches within each channel, and the final 1×1 convolution performs channel mixing. Padding of one is used in the depthwise convolution to preserve the spatial resolution. In our implementation, $C = 768$.

The student and teacher representations are processed through independent spatial feature distillation heads, yielding H_s and H_t, respectively. The spatial feature distillation loss is defined using token-wise cosine distance:

$$\mathcal{L}_{\cos} = \frac{1}{BN_p}\sum_{b=1}^{B}\sum_{j=1}^{N_p}\left(1 - \frac{H_{s,b,j}\cdot H_{t,b,j}}{\|H_{s,b,j}\|_2\,\|H_{t,b,j}\|_2}\right). \tag{35}$$

All patch tokens in the final Transformer block contribute equally to the spatial feature distillation loss, without additional weighting according to TR-induced token states or spatial regions.

The complete training objective is

$$\mathcal{L} = 1.0\,\mathcal{L}_{\mathrm{bce}} + 0.5\,\mathcal{L}_{\mathrm{dice}} + 2.0\,\mathcal{L}_{\mathrm{rate}} + 0.1\,\mathcal{L}_{\cos}, \tag{36}$$

with the target token-selection rate set to $k = 0.5$.

The spatial feature distillation branch is used only during training. During inference, the teacher backbone and both spatial feature distillation heads are removed, introducing no additional inference cost.

C Additional Implementation Details and Experimental Results

C.1 Implementation Details

Baselines. We compare our method with several representative lightweight adaptation, PEFT, and TR-PEFT approaches. Specifically, we include Linear and Decoder Only as lightweight adaptation baselines, and VPT (Jia et al., 2022), which introduces learnable prompt tokens into the input sequence of Transformer layers. We further consider AdaptFormer (Chen et al., 2022), which inserts trainable bottleneck MLP adapters in parallel with the FFN, LoRA (Hu et al., 2022), which introduces trainable low-rank updates to frozen linear layers, and EVP (Liu et al., 2023), which incorporates high-frequency priors through parallel adaptation modules. For TR-PEFT baselines, we include DyT (Zhao et al., 2024), which dynamically skips FFN computation for a subset of tokens, and DAR (Lei et al., 2025b), which further introduces an explicit Token Compensator to mitigate information loss caused by token reduction. Unless otherwise specified, all PEFT and TR-PEFT baselines use the same backbone and decoder architecture for a fair comparison. Except for the results on PolypDB and DUT-OMRON, the baseline results are taken from previously reported results (Lei et al., 2025b;a).

Salient object segmentation. For salient object segmentation, all methods are trained on the DUTS training set and evaluated on DUTS-TE (Wang et al., 2017), ECSSD (Yan et al., 2013), SOD (Movahedi & Elder, 2010), HKU-IS (Li & Yu, 2015), and DUT-OMRON (Yang et al., 2013). DUTS contains 10,553 training images and 5,019 testing images. ECSSD, SOD, and HKU-IS contain 1,000, 300, and 4,447 testing images, respectively. DUT-OMRON contains 5,168 images with pixel-level saliency annotations and includes scenes with multiple salient objects and relatively complex backgrounds. We use DUT-OMRON as an additional cross-dataset benchmark to evaluate the generalization of the trained models.

Table 5: Experimental settings for binary segmentation tasks.

Configuration	DUTS	COD10K+CAMO	PolypDB	ISIC 2017
Optimizer	AdamW	AdamW	AdamW	AdamW
Base learning rate	1.5×10^{-4}	1.5×10^{-4}	1.5×10^{-4}	1.5×10^{-4}
Weight decay	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}
Batch size	10	10	10	10
LR schedule	Cosine decay	Cosine decay	Cosine decay	Cosine decay
Warmup epochs	4	10	5	5
Training epochs	16	50	20	20

Camouflaged object segmentation. For camouflaged object segmentation, all methods are trained on the combined training sets of COD10K (Fan et al., 2022) and CAMO (Le et al., 2019), and evaluated on COD10K, CAMO, NC4K (Lv et al., 2021), and CHAMELEON (Skurowski et al., 2018). COD10K contains 3,040 training images and 2,026 testing images, while CAMO contains 1,000 training images and 250 testing images. NC4K contains 4,121 images and is used as an additional large-scale test set, while CHAMELEON contains 76 annotated images. Following the common evaluation protocol, the 4,040 training images from COD10K and CAMO are jointly used for model training.

Polyp segmentation. For polyp segmentation, we conduct experiments on PolypDB (Jha et al., 2025), which contains 3,934 polyp images with pixel-level annotations across five imaging modalities: Blue Light Imaging, Flexible Imaging Color Enhancement, Linked Color Imaging, Narrow Band Imaging, and White Light Imaging. Following the official dataset split, we use 3,150 images for training, 392 for validation, and 392 for testing.

Skin lesion segmentation. For skin lesion segmentation, we conduct experiments on the ISIC 2017 dataset (Codella et al., 2018). The dataset contains 2,000 training images, 150 validation images, and 600 testing images with expert pixel-level lesion annotations. We follow the official dataset split.

Table 5 summarizes the training settings for the binary segmentation tasks.

C.2 More Experiment Results

C.2.1 Segmentation Results

Salient object segmentation. Table 6 reports the results on four salient object segmentation datasets. Compared with existing TR-PEFT methods, our method achieves strong performance on DUTS-TE and ECSSD, while remaining competitive on SOD and HKU-IS. The spatial feature distillation variant further improves several metrics, but the gains are not consistent across all datasets. Overall, these results show that the proposed state-aware compensation and adaptation design is effective for TR-PEFT, while also indicating room for further improvement in achieving more consistent performance across different salient object segmentation datasets.

Camouflaged object segmentation. Table 7 reports the results on COD10K, CAMO, NC4K, and CHAMELEON. Our method achieves strong performance compared with existing TR-PEFT baselines across these challenging datasets. On COD10K, our method achieves the best results across all three metrics among TR-PEFT methods, while also obtaining strong performance on CAMO, NC4K, and CHAMELEON. Compared with DAR, the proposed method yields clear improvements in F_β^w on all four datasets, with particularly noticeable gains on COD10K and CHAMELEON. Ours^{SD} further improves the performance on CAMO across all three evaluation metrics. Overall, these results show that the proposed state-aware compensation and adaptation design remains effective across different camouflaged object segmentation datasets.

Table 6: Comparison on salient object segmentation datasets. The best results within each TR setting are highlighted in **bold**.

Method	DUTS-TE			ECSSD			SOD			HKU-IS		
	$F_\beta^w \uparrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$
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 (Jia et al., 2022)	.896	.931	.938	.934	.951	.958	.830	.826	.824	.914	.940	.954
AdaptFormer (Chen et al., 2022)	**.906**	**.937**	**.946**	**.943**	**.955**	**.962**	**.865**	.850	**.859**	**.926**	**.946**	**.960**
LoRA (Hu et al., 2022)	.897	.932	.942	.927	.949	.954	.854	**.851**	.855	.911	.940	.954
EVP (Liu et al., 2023)	.887	.929	.935	.930	.949	.955	.846	.836	.832	.912	.940	.953
w/ TR												
DyT (Zhao et al., 2024)	.859	.921	.930	.902	.938	.944	.837	.849	.855	.886	.931	.945
DAR (Lei et al., 2025b)	.895	.933	.944	.924	**.948**	**.958**	.860	**.851**	**.863**	.908	**.943**	**.960**
DAR^{LD} (Lei et al., 2025b)	.901	.935	.947	.932	**.948**	.957	.856	.850	.857	.917	.942	.958
Ours	**.912**	**.938**	.949	.935	**.948**	.955	.850	.836	.847	**.919**	.940	.955
Ours^{SD}	**.912**	**.938**	**.950**	**.937**	**.948**	.956	**.863**	.841	.855	.917	.938	.954

Table 7: Comparison on camouflaged object segmentation datasets.

Method	COD10K			CAMO			NC4K			CHAMELEON		
	$F_\beta^w \uparrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$
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 (Jia et al., 2022)	.816	.903	.936	.850	.908	.933	.869	.918	.941	.860	.926	.951
AdaptFormer (Chen et al., 2022)	**.826**	**.906**	**.941**	**.866**	**.911**	**.939**	**.879**	**.922**	**.946**	**.873**	**.926**	**.952**
LoRA (Hu et al., 2022)	.814	.897	.935	.852	.905	.934	.867	.913	.939	.855	.920	.944
EVP (Liu et al., 2023)	.803	.900	.933	.832	.901	.921	.859	.916	.937	.847	.917	.938
w/ TR												
DyT (Zhao et al., 2024)	.796	.894	.934	.839	.901	.932	.851	.910	.937	.834	.909	.943
DAR (Lei et al., 2025b)	.810	.899	.935	.842	.902	.932	.853	.914	.939	.834	.915	.936
DAR^{LD} (Lei et al., 2025b)	.808	.899	.936	.848	**.905**	.933	.859	.915	.939	.860	.924	.952
Ours	**.844**	**.908**	**.940**	.853	.902	.931	**.879**	**.919**	**.941**	**.884**	**.930**	**.954**
Ours^{SD}	.837	.907	.937	**.857**	**.905**	**.934**	.876	**.919**	**.941**	.881	.929	.949

C.2.2 Zero-Shot Generalization

To further evaluate cross-task generalization, we directly evaluate the DUTS-trained models on four camouflaged object segmentation benchmarks, including CAMO, NC4K, COD10K, and CHAMELEON, without any additional fine-tuning or camouflaged-object annotations. This setting evaluates whether the representations learned from salient object segmentation can generalize to the different visual characteristics of camouflaged objects.

As shown in Table 8, we compare our method with a diverse set of zero-shot baselines. SAM (Kirillov et al., 2023) and SAM2 (Ravi et al., 2025) are large-scale promptable segmentation foundation models with strong zero-shot capabilities, while GenSAM (Hu et al., 2024) adapts SAM to camouflaged object segmentation through cross-modal prompt generation and test-time adaptation. MM-CPF (Tang et al., 2024) further exploits multimodal large language models to enhance camouflaged-object perception in a zero-shot setting. We also include several models trained on DUTS and directly transferred to camouflaged object segmentation, including VST (Liu et al., 2021), SINet-V2 (Fan et al., 2022), CRNet (He et al., 2023), and EVP (Liu et al., 2023), as well as CaMF (Lei et al., 2025a), which is specifically designed for zero-shot transfer from salient to camouflaged object segmentation.

As shown in Table 8, our method demonstrates promising zero-shot transferability across the four camouflaged object segmentation benchmarks. Without additional camouflaged-object annotations,

Table 8: Zero-shot comparison on four camouflaged object segmentation benchmarks.

Method	Pub.	Setting	Data	CAMO			NC4K			COD10K			CHAMELEON		
				$F_\beta^w \uparrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$
SAM (Kirillov et al., 2023)	ICCV23	ZS	SA-1B	.597	.643	.639	.696	.767	.776	.673	.730	.737	.595	.635	.647
SAM2 (Ravi et al., 2025)	ICLR25	ZS	SA-V	.184	.444	.401	.251	.512	.482	.271	.549	.521	–	–	–
GenSAM (Hu et al., 2024)	AAAI24	ZS	SA-1B	.659	.719	.775	.665	.754	.781	.681	.775	.838	.680	.764	.807
MMCPF (Tang et al., 2024)	MM24	ZS	SA-1B	.683	.751	.772	.686	.769	.796	.596	.769	.826	–	–	–
VST (Liu et al., 2021)	ICCV21	ZS	DUTS	.563	.651	.683	.685	.752	.790	.544	.676	.732	.548	.659	.686
SINet-V2 (Fan et al., 2022)	TPAMI22	ZS	DUTS	.437	.583	.575	.551	.667	.668	.404	.604	.598	.291	.515	.461
CRNet (He et al., 2023)	AAAI23	ZS	DUTS	.445	.593	.575	.599	.694	.697	.443	.611	.590	.377	.562	.509
EVP-SegFormer (Liu et al., 2023)	CVPR23	ZS	DUTS	.637	.701	.733	.756	.788	.822	.652	.716	.764	.642	.713	.713
EVP-EVA02-L (Liu et al., 2023)	–	ZS	DUTS	.671	.750	.775	.783	.837	.862	.674	.781	.816	.663	.760	.768
CaMF (Lei et al., 2025a)	TPAMI25	ZS	DUTS	.729	.788	.814	.818	.861	.882	.717	.808	.832	.720	.805	.817
Ours	–	ZS	DUTS	.728	.800	.820	.812	.856	.876	.749	.825	.855	.719	.795	.803
Ours^{SD}	–	ZS	DUTS	**.746**	**.815**	**.848**	**.820**	**.868**	**.891**	**.750**	**.834**	**.866**	**.753**	**.838**	**.856**

the proposed TR-PEFT framework already achieves competitive performance, indicating that the learned representations retain certain cross-task generalization ability. However, the transfer performance varies across different target datasets. By introducing spatial feature distillation, our method further improves zero-shot performance across all four benchmarks, suggesting that preserving spatial representations can enhance the transferability of efficient adaptation. These results also suggest that the design of distillation strategies for preserving transferable representations under token reduction remains an important direction for further improving zero-shot generalization across pixel-level tasks.

D More Visualization

D.1 Token Reduction Visualization

We provide additional visualizations to further illustrate the behavior of token reduction across different pixel-level tasks, including salient object segmentation, camouflaged object segmentation, polyp segmentation, and skin lesion segmentation.

For salient object scenes (Fig. 5), token reduction mainly suppresses computation over homogeneous background regions and less informative areas, while preserving more tokens around object boundaries and structurally important regions. As the network goes deeper, the skipping ratio increases, yet the preserved tokens remain concentrated on regions relevant to object localization and contour delineation. This indicates that the token-selection mechanism can effectively reduce computation while preserving object structures in relatively simple scenes.

For camouflaged object scenes (Fig. 6), the model still tends to skip more computation for redundant background regions with similar appearances while preserving more tokens around object-related areas and boundaries. However, due to the weak foreground-background contrast in camouflaged scenes, some boundary information may be gradually reduced as the network becomes deeper and the skipping ratio increases. This observation suggests that aggressive token reduction may make it more difficult to preserve fine-grained spatial details in camouflaged scenes.

For medical image segmentation, including polyp and skin lesion segmentation, the token-selection behavior varies across different scenarios. For polyp segmentation (Fig. 7), the preserved tokens are mainly concentrated on polyp regions and object boundaries, while computation over redundant background regions is reduced, allowing the model to preserve the overall object structure. For skin lesion segmentation (Fig. 8), the selection becomes less consistent when lesion boundaries are unclear or appearance differences are subtle, where aggressive token reduction may fail to preserve fine-grained lesion details. These observations indicate that preserving task-relevant spatial information remains challenging for medical images with ambiguous structures.

D.2 Segmentation Map Visualization

We further provide qualitative comparisons of segmentation maps across different pixel-level tasks, including salient object segmentation, polyp segmentation, and skin lesion segmentation.

For salient object segmentation (Fig. 9), our method produces more complete object regions with clearer boundaries and fewer noisy predictions. For medical image segmentation (Figs. 10 and 11), our method produces more precise target delineation with smoother boundaries and fewer false predictions. Overall, these qualitative results show that the proposed method can preserve spatial details and improve segmentation quality while reducing token computation.

E Limitations and Future Work

Although TRACE improves the efficiency of pixel-level adaptation, there remains a trade-off between token reduction and fine-grained spatial information preservation. In challenging scenarios with ambiguous boundaries or weak foreground-background contrast, aggressive token reduction may still fail to preserve some task-relevant spatial details. Exploring more effective state-aware adaptation and compensation under aggressive token skipping remains an important direction for future research.

Future work will explore more adaptive token reduction strategies that jointly consider computational efficiency and spatial information preservation. Another promising direction is to investigate how token reduction affects transferable representations and how to better preserve such representations for stronger generalization across pixel-level tasks.

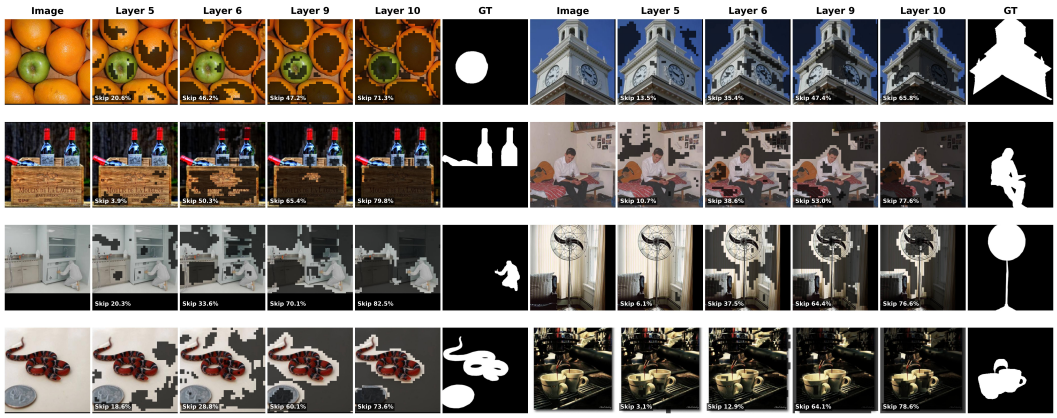

Figure 5: Visualization of the effects of our token reduction on the DUTS dataset.

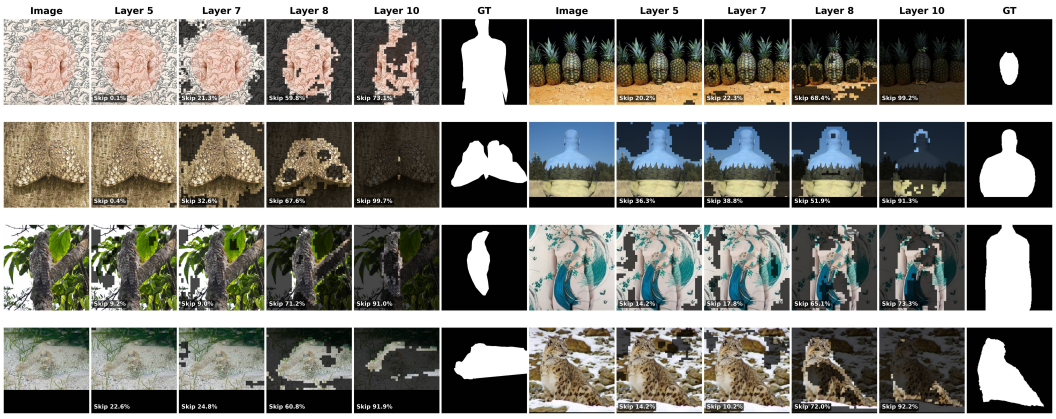

Figure 6: Visualization of the effects of our token reduction on the CAMO dataset.

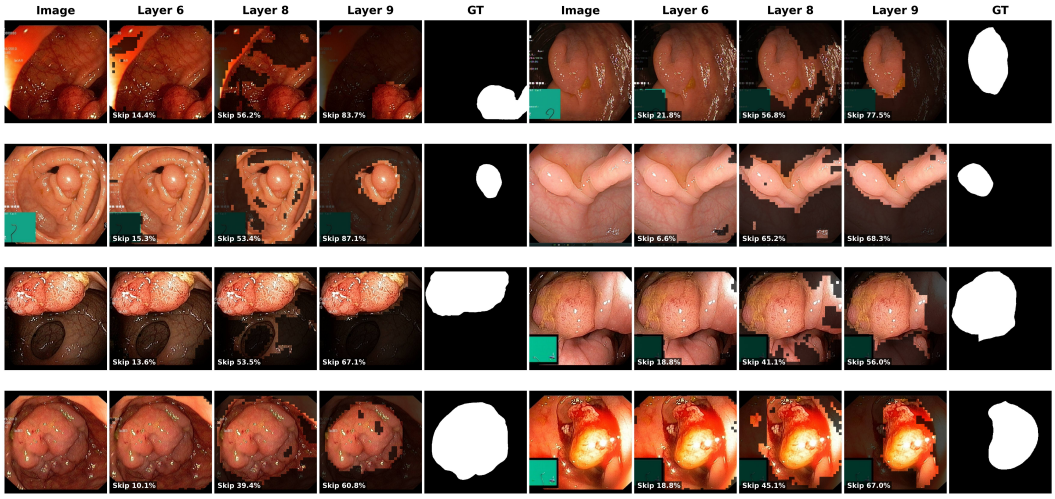

Figure 7: Visualization of the effects of our token reduction on the PolypDB dataset.

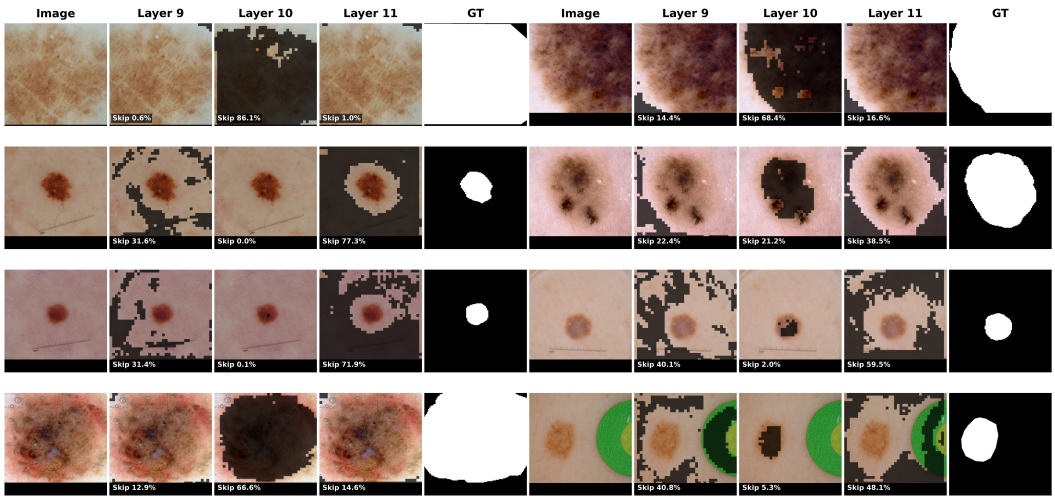

Figure 8: Visualization of the effects of our token reduction on the ISIC 2017 dataset.

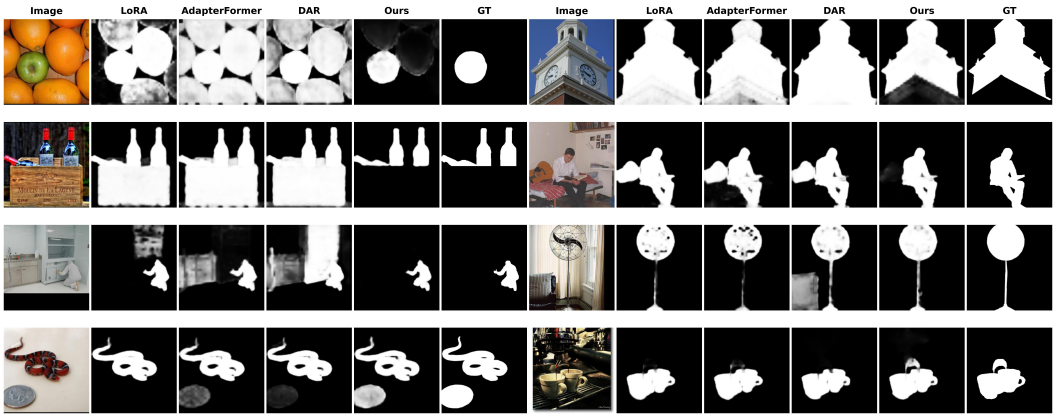

Figure 9: Qualitative segmentation results on the DUTS dataset.

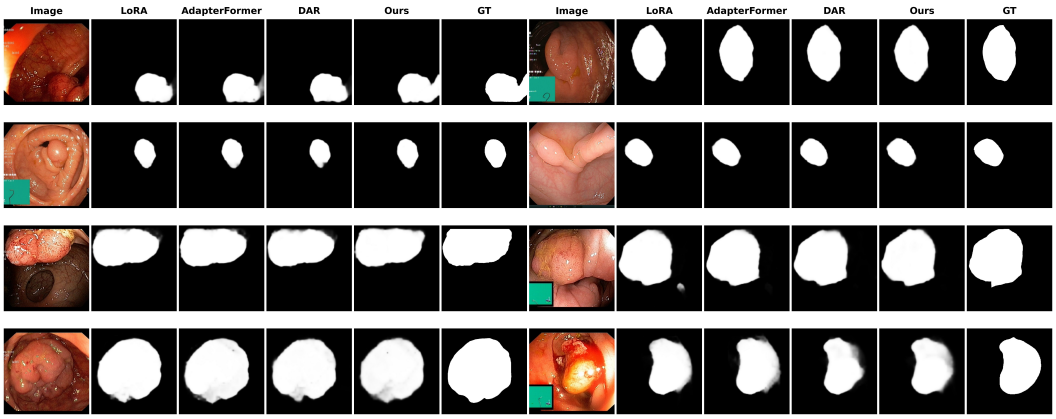

Figure 10: Qualitative segmentation results on the PolypDB dataset.

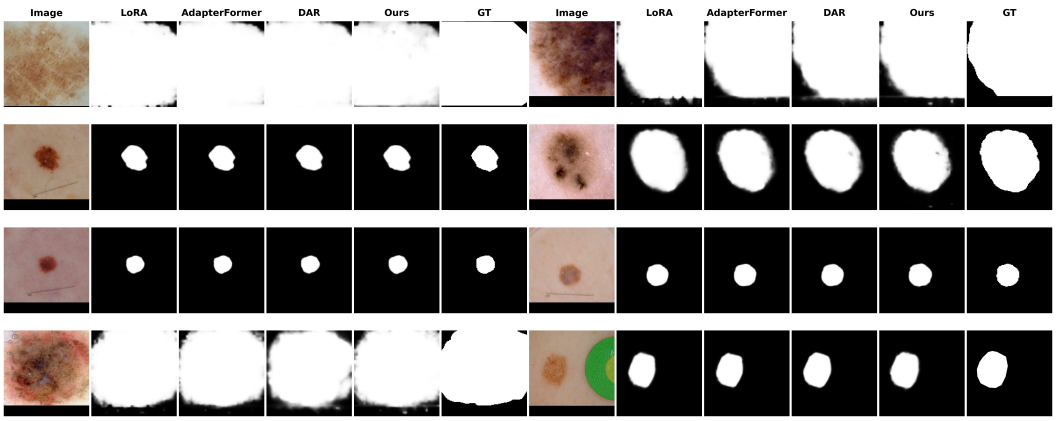

Figure 11: Qualitative segmentation results on the ISIC 2017 dataset.