Supplementary Materials for Edge-RecViT: Efficient Vision Transformer via Semantic-Refined Dynamic Recursion

YiZhou Li Jinyi Xu Mingyu Yin Xianyi Zhao

1. Addition Experimental Results

1.1. Additional Visualizations for Edge-RecViT

This section presents the complete set of visualizations for Edge-RecViT in Fig. 1. These results provide a clear view of the model's behavior. It can be clearly observed that Edge-RecViT allocates deeper computation (shown in red) to edge regions and semantically complex areas that contain important classification cues.

1.2. Trend in ImageNet-1K.

In Fig. 2, the Top-1–vs–FLOPs and Top-1–vs–parameters plots place the Edge-RecViT points on or near the Pareto frontier relative to ViT/DeiT and token-adaptive/merging counterparts. Across different model scales, the performance curves of Edge-RecViT consistently shift upward and to the left, indicating higher accuracy under lower computational and parameter budgets. It demonstrates the robustness and scalability of Edge-RecViT, as the proposed strategy generalizes well across model sizes.

2. Analysis of DU Loss

In the EARR, we propose the DU loss to encourage balanced depth usage. In this section, we further explain why a DU-loss value approaching 1 indicates uniform token exits across all depths.

The DU loss is shown as

$$\mathcal{L}_{\mathrm{du}} = L \sum_{l}^{L} E_l A_l, \tag{1}$$

$$E_l = \frac{1}{N} \sum_{i=1}^{N} p_{i,l}, \tag{2}$$

$$A_l = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_{(d_i=l)}, \tag{3}$$

where

$$\mathbf{1}_{(d_i=l)} = \begin{cases} 1, & d_i = l, \\ 0, & d_i \neq l. \end{cases} \tag{4}$$

It measures the alignment between the expected depth distribution E_l (derived from soft depth probabilities $p_{i,l}$) and the actual depth allocation A_l (obtained from discrete depth assignments d_i). Both E_l and A_l are normalized proportions that sum to 1 over all L depth levels. Moreover, the factor L in Eq. 3 serves as a normalization term ensuring that the minimum value of $\mathcal{L}_{\mathrm{du}}$ is exactly 1.

When token usage is perfectly uniform, each depth receives exactly $1/L$ of the tokens, giving

$$E_l = A_l = \frac{1}{L}, \quad \forall l. \tag{5}$$

Under this condition, we can obtain

$$\mathcal{L}_{\mathrm{du}} = L \sum_{l=1}^{L} \frac{1}{L} \cdot \frac{1}{L} = L \cdot L \cdot \frac{1}{L^2} = 1. \tag{6}$$

Hence, a value of $\mathcal{L}_{\mathrm{du}}$ close to 1 indicates that the number of exiting tokens is evenly distributed across depths, with each depth receiving approximately N/L tokens. This balanced usage prevents degenerate allocation patterns (e.g., assigning most tokens to a single depth) and encourages EARR to produce stable and diverse depth assignments across the recursive transformer.

Disucssion. While the DU-loss effectively regularizes depth allocation and ensures stable token-level dynamic computation, it also introduces an interesting behavior in rare and highly extreme cases. For most natural images, semantic information is reasonably distributed across different regions of the scene, and encouraging balanced token exits leads to strong generalization and stable performance. However, for a small subset of atypical images in which only very few tokens contain meaningful semantics, such as the shadow cast on an empty ground or a small decoration on a blank wall. In these cases, the informative content exists only along the boundary of the shadow, or within the very few tokens that cover the small decorative object.

When valid semantics are concentrated in only a handful of tokens, large areas of uniform background can still

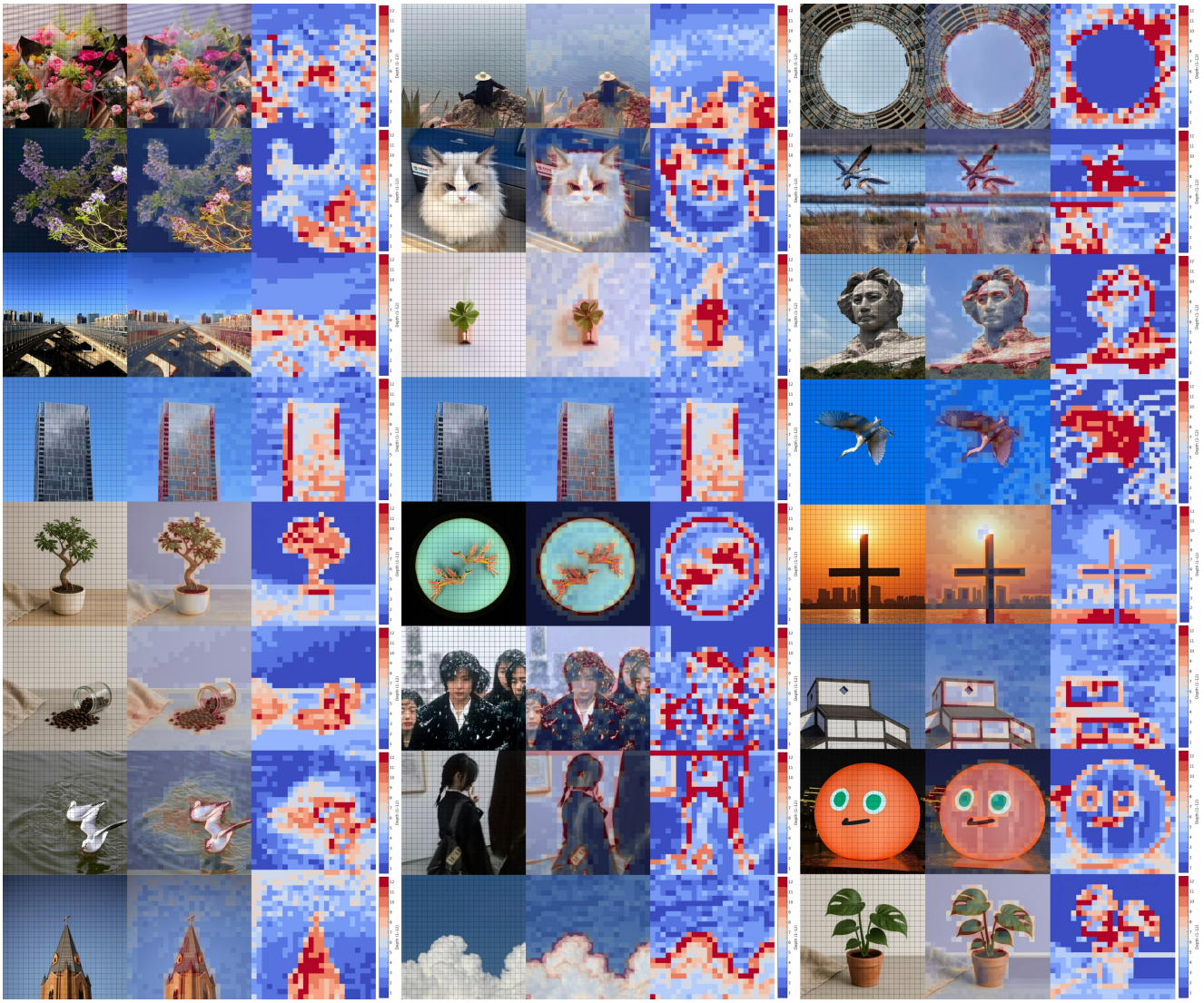


Figure 1. Additional visualizations for Edge-RecViT.

be assigned shallow but non-zero depths (one to three layers), despite not requiring additional processing. This behavior reveals an inherent trade-off introduced by the DU-loss. It prevents the ranker from collapsing and stabilizes depth allocation for typical images, yet it may lead to little over-computation in rare scenarios where semantic content is extremely sparse. This is a direction that we can further investigate in future work.

3. FLOPs Analysis of Edge-RecViT

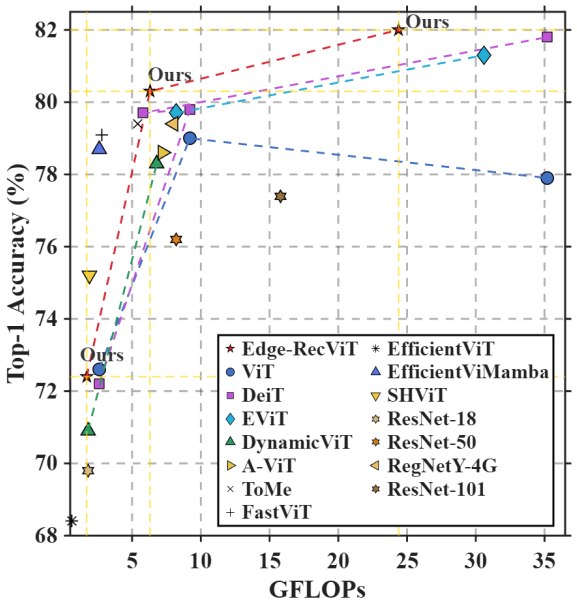
This section provides a theoretical analysis of the FLOPs optimization achieved by Edge-RecViT and presents the corresponding measurements to validate the experimental findings. The theoretical estimates closely match the empirical results reported in the main text (Table 1), demonstrating the reliability of our computational model.

3.1. Theoretical Analysis

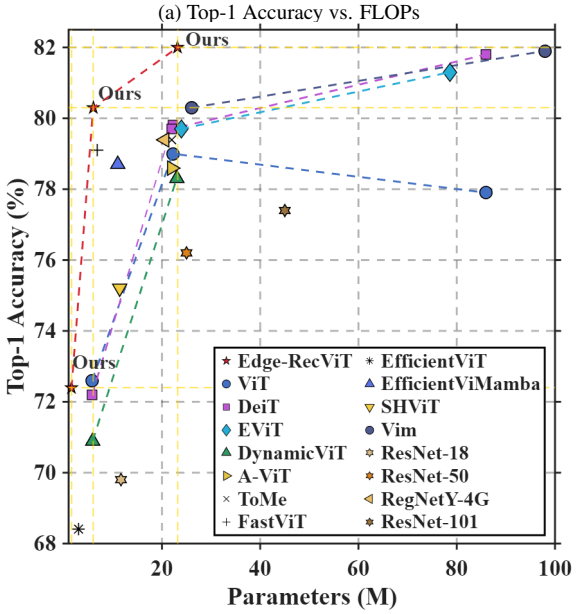
We begin by introducing the FLOPs formulation for a standard transformer block. We then compute and compare the FLOPs of the full-depth baseline (where Edge-RecViT operates without EARR) and the dynamic-depth variant enabled by EARR.

Standard Transformer Block. Standard complexity analysis estimates the FLOPs of transformer layers by analytically computing the required Multiply–Accumulate Operations (MACs) [1–4].

Each transformer layer can be viewed as consisting of two main computational components: the multi-head self-attention module (MHSA) and the two-layer feed-forward



(a) Top-1 Accuracy vs. FLOPs



(b) Top-1 Accuracy vs. Parameters

Figure 2. Comparison of Edge-RecViT and baseline models on ImageNet-1K.

network (FFN) [5]. The bias terms, softmax operations, and normalization layers can be ignored because they contribute negligibly to the total MACs. Thus the overall MACs of the l-th layer can therefore be expressed as

$$\mathrm{MAC}_{\text{layer}}(l) = \mathrm{MAC}_{\text{attn}}(l) + \mathrm{MAC}_{\text{ffn}}(l). \tag{7}$$

Let the hidden dimension of each layer be C, and let N_l be the number of active tokens at layer l. The MACs of the MHSA module is

$$\mathrm{MAC}_{\text{attn}}(l) \approx 4N_lC^2 + 2N_l^2C, \tag{8}$$

where the term $4N_lC^2$ comes from the four linear projections, corresponding to the query, key, value, and output transformations. The term $2N_l^2C$ comes from computing attention scores and applying them to the value vectors.

For the FFN with expansion ratio $r = 4$, the input of size $N_l \times C$ is projected to $N_l \times rC$ and then back to $N_l \times C$. The total number of MACs can be computed as:

$$\mathrm{MAC}_{\text{ffn}}(l) \approx 8N_lC^2. \tag{9}$$

Combining the attention and FFN terms 8 and 9 into Eq. 7, the total MACs for a single transformer layer can be expressed as:

$$\mathrm{MAC}_{\text{layer}}(l) \approx 12N_lC^2 + 2N_l^2C. \tag{10}$$

Full-depth Baseline (12 layers). For a full-depth Edge-RecViT (with EARR disabled), the major computational cost comes from three components: the patch embedding, the ranker, and the 12-layer transformer block.

For an 224×224 input resolution, 16×16 patch embedding generates 16×16 token grid, resulting in $N_1 = 14^2 + 1 = 197$, including the [CLS] token, and the hidden width is $C = 768$ with $L = 12$.

The patch embedding layer is implemented as a 16×16 convolution with three input channels and C output channels. Its FLOPs are

$$\mathrm{FLOPs}_{\text{patch}} = 2 \cdot 14^2 \cdot 3 \cdot 16^2 \cdot C \approx 0.231 \text{ GFLOPs}. \tag{11}$$

The Edge-Aware Ranker (EARR) is a two-layer MLP with a hidden width of $2C$ and L output logits, applied once to all N_0 tokens. Its FLOPs are

$$\mathrm{FLOPs}_{\text{ranker}} = 2N_0\big(C \cdot 2C + 2C \cdot L\big) \approx 0.472 \text{ GFLOPs}. \tag{12}$$

If all tokens traverse every layer in Edge-RecViT (i.e., with EARR disabled), then $N_l = N_1$ holds for every layer l. Substituting this into Eq. 10, the total FLOPs of the transformer blocks become:

$$\begin{aligned}\mathrm{FLOPs}_{\text{fixed}} &= 2L\big(12N_0C^2 + 2N_0^2C\big) \\ &= 34.90 \text{ GFLOPs}.\end{aligned} \tag{13}$$

Finally, we obtain the total FLOPs:

$$\begin{aligned}\mathrm{FLOPs}_{\text{Full}} &= \mathrm{FLOPs}_{\text{patch}} + \mathrm{FLOPs}_{\text{ranker}} + \mathrm{FLOPs}_{\text{fixed}} \\ &\approx 35.60 \text{ GFLOPs}.\end{aligned} \tag{14}$$

We note that this overall computational cost is essentially aligned with a DeiT-Base style backbone under the same configuration [4, 5], further confirming the reasonableness of our computation.

Edge-RecViT (with dynamic length). In practical inference, Edge-RecViT uses EARR to assign tokens to different recursive depths based on their semantic complexity. As a result, the transformer's computation is substantially smaller than that of the full-depth version. Next, we derive the theoretical FLOPs reduction rate through a formal analysis.

The computational costs of the patch embedding and the ranker remain identical to the full-depth setting (see Eq. 11 and Eq. 12), and are therefore omitted here for brevity.

For the recursive transformer module, let

$$\alpha_l = \frac{N_l}{N_1} \tag{15}$$

denote the fraction of tokens that remain active at layer l, where $\alpha_1 = 1$ and α_l This depth-wise decay is enforced by the DU-loss during training (see Sec. 3.3). Using Eq. 10 and substituting $N_l = \alpha_l N_1$, we obtain the FLOPs of the dynamic-depth transformer:

$$\begin{aligned}\text{FLOPs}_{dynamic} &= 2\sum_{l=1}^{L}\left(12N_lC^2 + 2N_l^2C\right) \\ &= 2\sum_{l=1}^{L}\left(12\alpha_l N_0C^2 + 2\alpha_l^2 N_0^2C\right) \\ &= 24N_0C^2\sum_{l=1}^{L}\alpha_l + 4N_0^2C\sum_{l=1}^{L}\alpha_l^2. \end{aligned} \tag{16}$$

For an 224×224 input resolution, the empirical depth histogram yields:

$$\sum_{l=1}^{L}\alpha_l \approx 8.21, \qquad \sum_{l=1}^{L}{\alpha_l}^2 \approx 6.64, \tag{17}$$

and substituting these values gives:

$$\text{FLOPs}_{\text{dynamic}} \approx 23.69 \text{ GFLOPs}. \tag{18}$$

Including the patch embedding and ranker FLOPs, the total computation of Edge-RecViT is:

$$\begin{aligned}\text{FLOPs}_{\text{Edge-RecViT}} &= 0.231 + 0.472 + 23.69 \\ &= 24.39 \text{ GFLOPs}. \end{aligned} \tag{19}$$

3.2. Experimental Verification

To validate the theoretical FLOPs accounting presented above, we implement a custom FLOPs profiler that instruments each sub-module—patch embedding, Edge-Aware Ranker, self-attention, and feed-forward networks—for a single 224×224 image. Under the same ViT-Base configuration (sequence length 197, hidden size 768, 12 layers, intermediate size 3072, 12 heads), the profiler reports a total of 24.39 GFLOPs decomposed as in Table 1.

Table 1. Empirical FLOPs Accounting for Edge-RecViT (224×224 Input)

Component	GFLOPs	Share
Patch Embedding	0.231	0.95%
Ranker (EARR)	0.472	1.94%
Attention	8.433	34.6%
MLP	15.256	62.6%
Total	24.39	100%

These results confirm that: (1) the overhead of the EARR remains in the low single-digit percentage range, and (2) the self-attention versus feed-forward split closely matches the canonical 1/3 vs. 2/3 ratio observed in ViT.

Overall, these empirical measurements align closely with our theoretical estimate of 24.39 GFLOPs and verify that dynamic token routing removes inactive tokens from all subsequent computations. Owing to the quadratic dependence of self-attention on the number of active tokens, this dynamic computation mechanism yields slightly more than linear savings in practice.

4. Training and Initialization Details

In this section, we detail our training pipeline from three complementary aspects: parameter freezing strategy, parameter mapping from pretrained DeiT model, and empirical hyperparameter setting for the STD loss and DU loss.

4.1. Parameter Freezing Strategy

For ImageNet-1K, we perform end-to-end fine-tuning of Edge-RecViT for 300 epochs starting from DeiT (non-distilled) model, with all transformer blocks kept trainable and no layers frozen.

For CIFAR-10/100, we adopt a two-stage training scheme. In the first stage, we initialize the model following the parameter-mapping strategy described in Sec. 4.2 and temporarily freeze head layer, shared layer, and tail layer, while updating only the patch embedding, the ranker, and the classification head for 200 epochs. In the second stage, we unfreeze all transformer layers and continue training for an additional 300 epochs under the same optimization settings. Unless otherwise stated, all main tables report the performance at convergence of this second stage. Throughout training on the CIFAR datasets, we fix the batch size to 32, which in our preliminary experiments provided a favorable empirical trade-off between gradient noise, optimization stability, and generalization. All remaining training details strictly follow the configuration specified in the Experiment Setup section.

4.2. Parameter Mapping Strategy

We perform parameter mapping for the transformer block of Edge-RecViT from the pretrained DeiT model. Since Edge-RecViT is structured with a head layer, a shared middle layer, and a tail layer, a natural strategy is to map the shallow, intermediate, and deep layers of the DeiT backbone to these three modules, respectively.

Our experiments show that initializing Edge-RecViT with parameters drawn from deeper layers consistently outperforms mappings that rely on very shallow or mid-shallow layers. As the mapping indices shift toward deeper layers, the Top-1 accuracy increases steadily, and this trend remains consistent across all scales of Edge-RecViT, including the Tiny, Small, and Base variants.

4.3 Hyperparameter Setting Strategy

We summarize the empirical hyperparameter configuration for the ranker regularization losses, which we found to be robust across all Edge-RecViT variants. STD Loss is crucial for stabilizing the ranker and is therefore always enabled. In practice, we set the hyperparameter λ_{std} of STD Loss to 1×10^{-3}, which provides a good balance between stability and accuracy. DU Loss is likewise indispensable for preventing the ranker's depth allocation from collapsing into degenerate patterns. When the DU Loss weight is too small, the depth allocation typically requires many more epochs to converge, whereas a larger weight allows the ranker to reach a stable depth allocation within roughly 100 training epochs. Throughout our experiments, $\lambda_{\mathrm{du}} = 1\times10^{-1}$ consistently yielded fast and stable convergence and is therefore adopted as our default setting.

5. Performance on Image Segmentation Task

We further evaluated Edge-RecViT on another downstream task, image segmentation on the COCO dataset, to verify the generalization ability of the proposed algorithm. The results are shown in Table 2. Edge-RecViT achieves higher segmentation accuracy than existing models of similar size. At the same time, the model has a smaller parameter size. To demonstrate that the compact model size leads to practical runtime benefits, we also measured the inference throughput of the model in Table 3. The results show that Edge-RecViT achieves higher throughput. This confirms that the smaller model size brings clear improvements in computational efficiency while maintaining strong segmentation performance.

Table 2. COCO instance segmentation comparison.

Method	Backbone	Params (M)	FLOPs (G)	AP
Detectron2 [facebook][6]	Res-X101-FPN	106	250	39.5
	R101-FPN	63	180	43.7
Mask2Former [CVPR '22][7]	Swin-Tiny	47.42	200	45.0
	Swin-Small	68.74	270	46.3
	Swin-Base	106.90	440	46.7
Mask DINO [CVPR '23][8]	ResNet-50	52	286	46.3
RMT [CVPR '24][9]	RMT-S	46	262	44.9
	RMT-B	73	373	46.1
Spatial-Mamba [ICLR '25][10]	Spatial-Mamba-T	46	261	45.0
	Spatial-Mamba-S	67	340	46.1
Ours(Tiny)	**Edge-RecViT**	**23.3**	**145**	**46.8**
Ours (Small)	**Edge-RecViT**	**26.0**	**194**	**47.9**
Ours (Base)	**Edge-RecViT**	**30.9**	**317**	**48.3**

Table 3. Throughput comparison on COCO 2017 val (5k images) on 8×A100-40GB GPUs with input resolution 1024.

Model	Params (M)	Single-GPU (img/s)	8-GPU (img/s)
Edge-RecViT (Tiny)	23.3	11.5	87.6
Edge-RecViT (Small)	26.0	9.9	74.9
Edge-RecViT (Base)	30.9	7.8	59.5
Detectron2(ResX101)	106	9.7	62.1

References

[1] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao, "A survey on visual transformer," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 45, no. 1, pp. 87–110, 2023. 2

[2] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM computing surveys*, vol. 54, no. 10s, pp. 1–41, 2022.

[3] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, "Training data-efficient image transformers and distillation through attention," in *Proceedings of the International Conference on Machine Learning (ICML)*, vol. 139, 2021, pp. 10 347–10 357.

[4] A. Steiner, A. Kolesnikov, X. Zhai, R. Wightman, J. Uszkoreit, and L. Beyer, "How to train your vit? data, augmentation, and regularization in vision transformers," *Transactions on Machine Learning Research*, vol. 2022, 2022. [Online]. Available: https://openreview.net/forum?id=4nPswr1KcP 2, 3

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations (ICLR)*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy 3

[6] Facebook AI Research, "Detectron2: A pytorch-based modular object detection library," 2019. [Online]. Available: https://github.com/facebookresearch/detectron2 5

[7] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 1280–1289. 5

[8] F. Li, H. Zhang, H. Xu, S. Liu, L. Zhang, L. M. Ni, and H.-Y. Shum, "Mask dino: Towards a unified transformer-based framework for object detection and segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 3041–3050. 5

[9] Q. Fan, H. Huang, M. Chen, H. Liu, and R. He, "Rmt: Retentive networks meet vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 5641–5651. 5

[10] C. Xiao, M. Li, Z. Zhang, D. Meng, and L. Zhang, "Spatial-mamba: Effective visual state space models via structure-aware state fusion," in *International Conference on Learning Representations (ICLR)*, 2025. [Online]. Available: https://openreview.net/forum?id=iDe1mtxqK5 5