Your ViT is Secretly an Image Segmentation Model

Your ViT is Secretly an Image Segmentation Model

Tommie Kerssies[1] Niccolò Cavagnero[2,*] Alexander Hermans[3] Narges Norouzi[1]
Giuseppe Averta[2] Bastian Leibe[3] Gijs Dubbelman[1] Daan de Geus[1,3]

[1]Eindhoven University of Technology [2]Polytechnic of Turin [3]RWTH Aachen University

Abstract

Vision Transformers (ViTs) have shown remarkable performance and scalability across various computer vision tasks. To apply single-scale ViTs to image segmentation, existing methods adopt a convolutional adapter to generate multi-scale features, a pixel decoder to fuse these features, and a Transformer decoder that uses the fused features to make predictions. In this paper, we show that the inductive biases introduced by these task-specific components can instead be learned by the ViT itself, given sufficiently large models and extensive pre-training. Based on these findings, we introduce the Encoder-only Mask Transformer (EoMT), which repurposes the plain ViT architecture to conduct image segmentation. With large-scale models and pre-training, EoMT obtains a segmentation accuracy similar to state-of-the-art models that use task-specific components. At the same time, EoMT is significantly faster than these methods due to its architectural simplicity, e.g.*, up to* 4× *faster with ViT-L. Across a range of model sizes, EoMT demonstrates an optimal balance between segmentation accuracy and prediction speed, suggesting that compute resources are better spent on scaling the ViT itself rather than adding architectural complexity. Code: https://www.tue-mps.org/eomt/.*

1. Introduction

The Vision Transformer (ViT) [23] has proven to be a strong, scalable, and generally applicable architecture for computer vision [1, 21, 51, 67]. Recently, research has shown that ViTs are very suitable for large-scale pre-training [5, 28, 51, 68], resulting in generalizable models that achieve high performance on many downstream tasks. A particularly well-researched task is image segmentation, *e.g.*, semantic, instance, and panoptic segmentation [39]. To achieve state-of-the-art segmentation performance with ViTs, they are typically combined with several computationally intensive and task-specific components such as ViT-Adapter and Mask2Former (M2F) [13, 15]. In these methods, an *adapter* [13, 63] is first applied in parallel to the ViT, using convolutional layers and interacting with the ViT to extract multi-scale features. Second, these multi-scale features are fed to a Mask Transformer module [14, 15, 35, 66], consisting of a *pixel decoder* and a *Transformer decoder*. The pixel decoder fuses and enhances information across multiple feature scales. The Transformer decoder then introduces learnable object queries that attend to the multi-scale features via cross-attention. These queries are finally used to generate segmentation mask and class predictions.

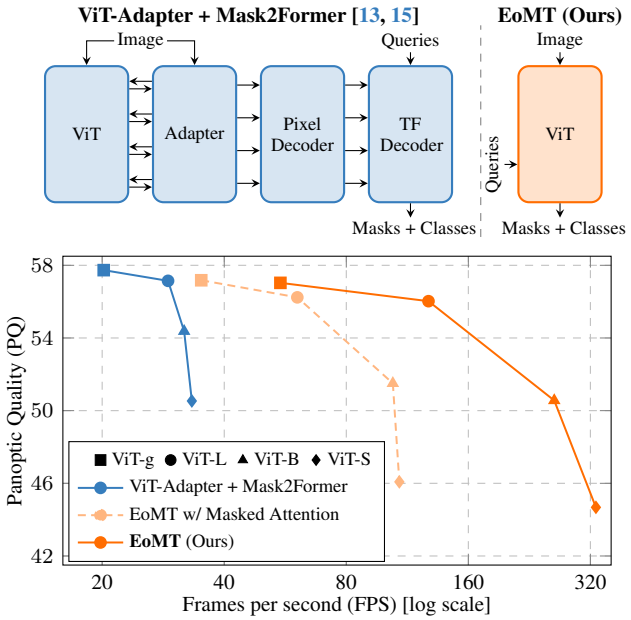


Figure 1. **ViT-Adapter + Mask2Former *vs*. EoMT (Ours).** EoMT demonstrates an optimal balance between Panoptic Quality (PQ) and FPS across different sizes of DINOv2 [51] pre-trained ViTs [23]. Evaluation on COCO *val2017* [43], see Tab. 3.

In this paper, we explore whether these additional components, as visualized in Fig. 1 (top left), are truly necessary to obtain state-of-the-art performance. We hypothesize that with increasingly extensive pre-training and for larger ViTs, the positive effect of these additional components decreases, making them nearly irrelevant. There are two key reasons for this: (*i*) Large-scale pre-training, particularly

[*] Work done while visiting RWTH.

when combined with objectives like masked image modeling (*e.g.*, DINOv2 [51]), teaches the ViT to extract dense, fine-grained semantic information essential for segmentation [36]. Therefore, we expect that additional components are no longer required to aid the ViT in extracting this information. (ii) Larger ViTs have more learnable parameters. We expect that this increased capacity allows large ViTs to accurately conduct image segmentation without additional components. If these hypotheses hold, the simplicity and efficiency of segmentation models could be significantly enhanced by removing these task-specific components, while only minimally affecting their segmentation accuracy.

To confirm these hypotheses, we experimentally assess the effect of gradually removing the aforementioned components, in combination with several types of pre-training and different model sizes. This process ultimately leads us to a conceptually simple model with an architecture that only minimally differs from the plain ViT. We call this model the *Encoder-only Mask Transformer* (EoMT). Crucially, EoMT repurposes the ViT blocks to not only extract image features, but also enable interaction between learnable object queries and image features, to finally predict a mask and class label for each of these queries. This highly simplified design is visualized in Fig. 1 (top right).

A key innovation of EoMT is that it does not require masked attention during inference, unlike existing Mask2Former-like architectures. Models use masked attention to constrain each query to cross-attend only within the intermediate segmentation mask predicted for that query. While this improves segmentation accuracy, it also harms efficiency as it requires additional operations. To overcome this, we present a novel *mask annealing* strategy, where masked attention is initially fully enabled during training, but then gradually phased out as training progresses, allowing for efficient, masked-attention-free inference.

Overall, EoMT offers several advantages: (i) By not requiring additional components and by enabling inference without masked attention, EoMT greatly reduces computational requirements and latency. (ii) Due to its architectural simplicity, EoMT is significantly easier to implement than existing approaches that use additional components. (iii) By relying purely on the Transformer architecture [57] of the ViT, EoMT can fully and directly leverage ongoing and future developments related to Transformers, without being bottlenecked by additional non-optimized modules. This applies not only to general improvements like FlashAttention [18, 19] and specialized hardware [16, 24], but also to vision-specific advances like token merging [7, 48, 50] and vision foundation models [28, 51].

Through our experimental analysis, we find that removing task-specific components has only a minimal impact on segmentation accuracy when using a large pre-trained model like DINOv2 [51], confirming our hypotheses. By removing these components, EoMT achieves performance competitive with the state of the art, while being much faster. As shown in Fig. 1, EoMT obtains a considerably better balance between prediction speed and segmentation quality. To illustrate this: ViT-Adapter + M2F with ViT-B achieves a Panoptic Quality (PQ) of 54.4 at 32 frames per second (FPS). In contrast, despite using a larger model, EoMT runs significantly faster with ViT-L at 128 FPS while also achieving a higher PQ of 56.0.

In summary, we make the following contributions: **(1)** We assess the necessity of task-specific components of state-of-the-art image segmentation models and find that they become less relevant when scaling up model size and pre-training. **(2)** We present the *Encoder-only Mask Transformer* (EoMT), a simple and efficient model that repurposes the ViT blocks for segmentation and obtains state-of-the-art performance without requiring inefficient task-specific components. **(3)** We propose a *mask annealing* training strategy that enables significantly faster inference by removing the need for masked attention.

2. Related Work

Image segmentation. Image segmentation is a fundamental task in computer vision, for which the goal is to divide an image into pixel-level segments based on semantics, by providing a segmentation mask and class label for each segment. For semantic segmentation, the objective is to output a single segment for each class in the image. Instance segmentation, on the other hand, requires a segment for each individual object instance, but disregards uncountable entities like 'road' or 'sky'. Finally, panoptic segmentation [39] combines these tasks and requires (i) segments per individual instance for countable classes called *things* (*e.g.*, 'person' or 'car'), and (ii) a single segment per class for uncountable classes called *stuff* (*e.g.*, 'road' or 'sky').

Traditionally, segmentation methods had specialized architectures that were tailored for only one of these tasks [11, 33, 38, 55, 69]. Recently, however, the emergence of the *Mask Transformer* framework [8, 14, 59] has enabled the adoption of a unified architecture and training pipeline for all three segmentation tasks [3, 10, 14, 15, 35, 40, 41, 66]. The versatility of these models is enabled by learnable object queries, which adaptively learn to represent a single segment, whether it is a *stuff* class or a *thing* instance. In this work, we investigate state-of-the-art Mask Transformer models and propose a minimalistic, efficient ViT-based model that is competitive with more complex architectures while being significantly more efficient.

Vision Transformers. The Transformer architecture [57] was originally developed for Natural Language Processing (NLP), where its ability to model long-range dependencies revolutionized the field. To harness the power of this

architecture for computer vision, the Vision Transformer (ViT) [23] was introduced. ViTs divide images into fixed-size patches, project them into an embedding space to form tokens, and then process these tokens using multiple Transformer blocks. By design, there are key differences between Convolutional Neural Networks (CNNs) and ViTs. (i) ViTs process images at a fixed resolution due to the fixed patch size. In contrast, CNNs (*e.g.*, ResNet [32]) typically contain various downscaling steps, allowing them to output feature maps of multiple resolutions [44]. (ii) ViTs process images globally leveraging self-attention, while CNNs process images locally by applying convolutional filters.

For tasks like image segmentation, multi-scale features and local processing are claimed to be beneficial for performance [13]. To introduce these properties into Transformers, some works propose alternative architectures [30, 31, 45, 64] that incorporate local attention and token downsampling. However, as discussed later, these models deviate from the plain ViT architecture, preventing them from leveraging advancements in large-scale pre-training. An alternative approach extends ViTs with a CNN-based *adapter* to produce multi-scale features [13, 63]. In this work, however, we find that the necessity of these adapters and other task-specific components greatly diminishes when using extensively pre-trained large ViTs for image segmentation. This allows a much simpler and more efficient model while maintaining performance competitive with state-of-the-art approaches. Similar to our work, UViT [12] explores the use of single-scale features from the plain ViT for instance recognition tasks, but it still relies on complex task-specific decoders. Meanwhile, YOLOS [27] adopts an encoder-only ViT for instance recognition but is restricted exclusively to object detection. Moreover, neither method demonstrates that scaling up model size and pre-training enables a simple ViT-based model to be competitive with complex state-of-the-art architectures, which we do with EoMT.

Large-scale visual pre-training. For tasks like image segmentation, it is common practice to initialize a model's backbone with pre-trained weights to improve downstream performance over random initialization. Initially, such pre-training relied on supervised image classification on ImageNet [22]. More recently, pre-training has been scaled up to massive datasets using weakly- [52] or self-supervised [9] learning. Currently, vision foundation models (VFMs) that use masked image modeling, like DINOv2 [51] and EVA-02 [28], provide the best pre-training for image segmentation [36]. Notably, all these VFMs use the ViT architecture for its scalability. This means that models with non-ViT backbones, such as Swin [45] and ConvNeXt [46], which are commonly used for image segmentation, cannot leverage VFM pre-training due to their incompatible architectures. In contrast, EoMT *can* benefit from VFM initialization, as it is fully ViT-based.

3. Towards Encoder-only Mask Transformer

3.1. Preliminaries

Vision Transformers. Vision Transformers first divide an image $\boldsymbol{I} \in \mathbb{R}^{3\times H\times W}$ into N non-overlapping patches of shape $(p \times p)$, where H and W are the image height and width, and p is the pre-determined patch size. Subsequently, these patches are linearly projected into patch tokens $\boldsymbol{X}^0 \in \mathbb{R}^{D\times N}$ and processed by L Transformer blocks [57]. Each Transformer block applies multi-head self-attention (`MHSA`) and a two-layer multi-layer perceptron (`MLP`) with a non-linear activation function. Concretely, for each block i,

$$\begin{aligned} \boldsymbol{Z}^i &= \boldsymbol{X}^i + \texttt{MHSA}(\texttt{Norm}(\boldsymbol{X}^i)); \\ \boldsymbol{X}^{i+1} &= \boldsymbol{Z}^i + \texttt{MLP}(\texttt{Norm}(\boldsymbol{Z}^i)), \end{aligned} \tag{1}$$

where `Norm` is Layer Normalization [4]. The result of this process is a set of final patch tokens $\boldsymbol{X}^L$. Reordering these tokens yields spatial features $\boldsymbol{F}^{\text{vit}} \in \mathbb{R}^{D\times \frac{H}{p}\times \frac{W}{p}}$, where the patch size p determines the resolution.

To achieve state-of-the-art image segmentation with ViTs, recent works have proposed several components that further process the resulting patch tokens and interact with the ViT at different levels.

Adapters. To introduce convolutional biases and enable multi-scale feature extraction, an *adapter* is applied in parallel to the ViT to inject and extract features [13, 63]. Concretely, the ViT-Adapter [13], which we study in this work, first applies a CNN to the input image to extract multi-scale features at resolutions $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$, and $\frac{1}{32}$. Subsequently, the ViT-Adapter repeatedly injects these CNN features into the ViT through multi-scale deformable attention [71], applies several ViT blocks, and extracts refined features from the ViT into the multi-scale CNN features. The output of the adapter is a set of multi-scale ViT- and CNN-based features, $\{\boldsymbol{F}_4, \boldsymbol{F}_8, \boldsymbol{F}_{16}, \boldsymbol{F}_{32}\}$, with $\boldsymbol{F}_i \in \mathbb{R}^{D\times \frac{H}{i}\times \frac{W}{i}}$.

Mask Transformers. To make segmentation predictions with these features, state-of-the-art models follow the *Mask Transformer* framework [8, 14, 59]. In this work, we study the state-of-the-art method Mask2Former (M2F) [15]. As a first step, to further enhance the features extracted by the ViT-Adapter, M2F applies a *pixel decoder*. This pixel decoder takes the features $\{\boldsymbol{F}_4, \boldsymbol{F}_8, \boldsymbol{F}_{16}, \boldsymbol{F}_{32}\}$ and applies a series of multi-scale deformable attention layers [71], outputting processed features $\{\widehat{\boldsymbol{F}}_4, \widehat{\boldsymbol{F}}_8, \widehat{\boldsymbol{F}}_{16}, \widehat{\boldsymbol{F}}_{32}\}$. In this process, multi-scale features from different backbone layers are processed into a consistent yet scale-specific representation.

The final component of M2F, the *Transformer decoder*, generates and outputs the actual segmentation predictions. As inputs, it takes not only the multi-scale features from the pixel decoder, but also a set of K learned queries $\mathcal{Q}^0 = \{\boldsymbol{q}_i^0 \in \mathbb{R}^D\}_{i=1}^K$. In the Transformer decoder, each of these queries learns to represent one individual segment per image (*i.e.*, a *stuff* class or a *thing* instance). To do so,

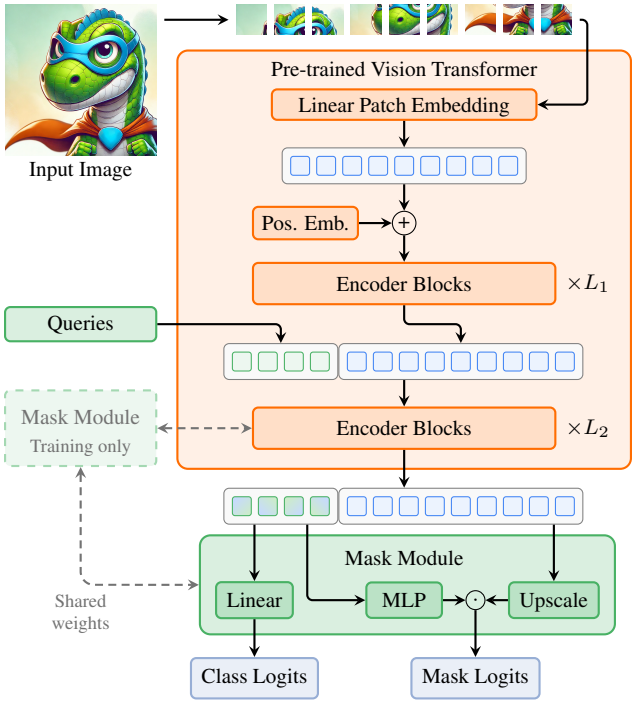


Figure 2. **EoMT architecture.** Learnable queries are concatenated to the patch tokens after the first L_1 ViT encoder blocks. These concatenated tokens are then jointly processed by the last L_2 blocks and used to predict class and mask logits.

these queries are subjected to J blocks with cross-attention to the multi-scale features and multi-head self-attention between queries, yielding processed queries $\mathcal{Q}^J$. The Transformer decoder then generates the segmentation predictions by predicting a class label and segmentation mask for each query $\boldsymbol{q}_i^J$. Class logits $\boldsymbol{c}_i^J \in \mathbb{R}^C$ are predicted by applying a linear layer to $\boldsymbol{q}_i^J$. Mask logits $\boldsymbol{m}_i^J \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4}}$ are obtained by first applying an `MLP` to yield mask embedding $\widehat{\boldsymbol{q}}_i^J$ and subsequently taking the dot product between the features $\widehat{\boldsymbol{F}}_4$ and $\widehat{\boldsymbol{q}}_i^J$. By assigning each ground-truth segment to a unique query and supervising both class and mask predictions, the model learns the segmentation task.

A key feature of the M2F Transformer decoder is the use of masked cross-attention. In each of the J blocks, prior to cross-attention between queries and image features, an intermediate segmentation mask and class label are predicted and supervised per query using the above procedure. The predicted masks are then used to mask the attention, allowing a query to only attend to the image region that corresponds to its predicted segmentation mask. This masked attention results in improved segmentation accuracy [15].

3.2. Removing Task-Specific Components

To study the importance of the aforementioned task-specific components, we gradually remove them while assessing the effect on segmentation accuracy. We continue until all task-specific components are removed, ending up with our simple *Encoder-only Mask Transformer* (EoMT). The different configurations are described in this section and visualized in Fig. A, with the results reported in Sec. 4.2.

Removing the adapter. Removing the adapter eliminates the inductive biases and multi-resolution features provided by a CNN. However, we hypothesize that with sufficient pre-training, large ViTs can learn these features without requiring additional components. In the absence of an adapter, we construct a feature pyramid by upscaling the ViT output features $\boldsymbol{F}^{\text{vit}} = \boldsymbol{F}_{16}$ (patch size 16×16) to compute $\boldsymbol{F}_4$ and $\boldsymbol{F}_8$, and downscaling them to compute $\boldsymbol{F}_{32}$. We follow the approach of ViTDet [42] by using transposed convolutions for upscaling and normal convolutions for downscaling. For each scale of the feature pyramid, we independently up- or downscale $\boldsymbol{F}^{\text{vit}}$ with a sequence of operations. We repeat a (transposed) 2×2 convolution with stride 2×2, GELU activation, depthwise 3×3 convolution, and final `Norm`, until the required scales are reached. This approach mimics the multi-scale feature extraction of the ViT-Adapter in a much simpler manner.

Removing the pixel decoder. Without the adapter, the resulting features no longer originate from different stages of a hierarchical backbone and thus should not need further consolidation. As such, the heavy pixel decoder should be obsolete, and we remove it by directly feeding the simplified feature pyramid to the Transformer decoder. Specifically, instead of $\{\widehat{\boldsymbol{F}}_4, \widehat{\boldsymbol{F}}_8, \widehat{\boldsymbol{F}}_{16}, \widehat{\boldsymbol{F}}_{32}\}$, we input $\{\boldsymbol{F}_4, \boldsymbol{F}_8, \boldsymbol{F}_{16}, \boldsymbol{F}_{32}\}$ to the Transformer decoder.

Removing multi-scale feature processing. To further simplify, we question the necessity of using features at multiple scales, since all features are derived from a shared single-scale feature map $\boldsymbol{F}^{\text{vit}}$. Therefore, we do not generate multi-scale features $\boldsymbol{F}_8, \boldsymbol{F}_{32}$. In the Transformer decoder, instead, queries cross-attend exclusively to the ViT output $\boldsymbol{F}^{\text{vit}} = \boldsymbol{F}_{16}$. We only upscale $\boldsymbol{F}^{\text{vit}}$ to $\boldsymbol{F}_4$ to compute the mask logits via dot product with the mask embeddings $\widehat{\boldsymbol{q}}_i^J$, to ensure high-resolution output masks.

3.3. Encoder-only Mask Transformer

By removing the previous task-specific components, the model is reduced to a ViT with a single-scale Transformer decoder. The next step is to completely remove the decoder. This requires minor modifications to the plain ViT architecture, such that it can perform image segmentation without a dedicated decoder. We call the resulting method the *Encoder-only Mask Transformer* (EoMT).

Querying the ViT for masks. Differently from standard Mask Transformers for image segmentation [14, 15, 34, 35, 40, 66], which adopt heavy and complex ad-hoc decoders, EoMT only uses the architecture of the plain ViT with a few extra learned queries and a small mask prediction module.

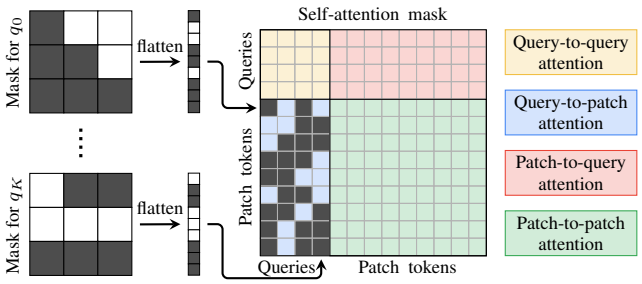


Figure 3. **Masked self-attention during training.** In the final L_2 blocks of EoMT, patch tokens and queries are jointly processed by self-attention. During training, the intermediate mask predictions are used to mask the query-to-patch portion of the attention operation, mimicking the masked cross-attention of M2F [15].

An overview of EoMT is provided in Fig. 2.

The first L_1 Transformer encoder blocks of the ViT are unchanged and only process the input image. After these encoder blocks, we introduce a set of K learnable queries that are concatenated to the patch tokens. These are then jointly processed by the remaining L_2 ViT encoder blocks, following Eq. (1). The final blocks thus have to process the patch tokens as before, but also replace the Transformer decoder that processes the queries.

A standard Mask Transformer introduces (i) interaction between individual queries through self-attention, enabling queries to coordinate the objects they should attend to, and (ii) transfer of information from visual tokens to object queries through query-to-patch cross-attention. Normally, these operations are performed sequentially. In contrast, by using the `MHSA` operation of the ViT, EoMT performs them jointly in a single layer (see Fig. 3).

In addition to the ViT, we introduce a small *mask module* to predict masks and corresponding classes for each query. Here, we follow the same design as M2F [15], passing the query tokens through a linear layer to predict class logits and using a three-layer `MLP` followed by a dot product with the upscaled image features $\boldsymbol{F}_4$ to obtain mask logits.

To enable masked attention between queries and image features during training, as in the M2F Transformer decoder, we additionally apply the previously introduced mask module before each of the last L_2 ViT blocks, to predict intermediate segmentation masks for each query. In turn, these masks can be used to mask the query-to-patch attention in the plain self-attention block, constraining the attention of each query to the segmentation mask that is predicted for it. This is visualized in Fig. 3.

Mask annealing. While using masked attention during training improves performance, predicting intermediate masks and applying them to the self-attention operation for each block during inference is computationally expensive and inefficient. To address this, we propose a *mask annealing* scheme that gradually phases out masked attention over the course of training. Specifically, in each block

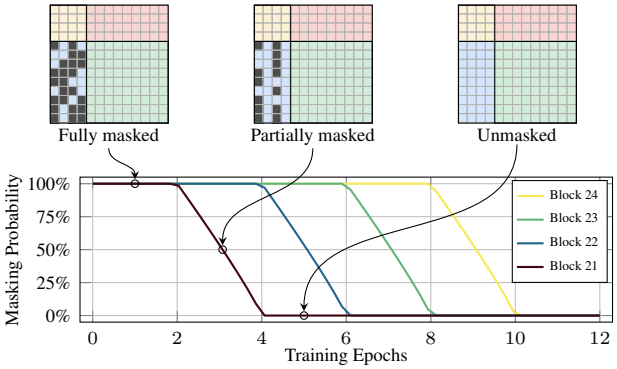


Figure 4. **Mask annealing during training.** Self-attention is initially masked in the final L_2 ($=$ 4 for ViT-L) EoMT blocks. The masking probability is gradually annealed, starting from early blocks, until it is no longer needed at the end of training.

with masked attention, we apply the mask for each query with a probability P_{mask}, which starts at 1.0 and decays block by block to 0.0 throughout the training, as shown in Fig. 4. This strategy allows the model to initially benefit from masked attention to aid convergence, while gradually learning to operate without it, thus maintaining high performance. By eliminating the need for masked attention during inference, the final model leverages the highly optimized plain ViT architecture without additional intermediate modules or modified attention operations, ensuring optimal efficiency.

4. Experiments

4.1. Experimental Setup

Datasets. We use widely adopted benchmarks for image segmentation: COCO [43] and ADE20K [70] for panoptic, Cityscapes [17] and ADE20K for semantic, and COCO for instance segmentation.

Models and training. Unless specified otherwise, we use DINOv2-L [51] with a 640×640 input size and a 16×16 patch size. For EoMT with ViT-L, we use $L_2 = 4$, which provides optimal performance (see Appendix B and Tab. D). We train our models in mixed precision with the AdamW [47] optimizer (learning rate 10^{-4}), layer-wise learning rate decay (LLRD) [5] (factor 0.8), polynomial learning rate decay (factor 0.9), and polynomial mask annealing (factor 0.9). The number of epochs is set to 12 ($1\times$ schedule [62]) on COCO, 31 [36] on ADE20K, and 107 [36] on Cityscapes.

Evaluation. We use standard evaluation metrics: Panoptic Quality (PQ) for panoptic [39], mean Intersection over Union (mIoU) for semantic [26], and Average Precision (AP) for instance segmentation [43]. We evaluate model efficiency in terms of average inference speed in *frames per second* (FPS) and average number of *floating point operations* (FLOPs) over all images in the validation set, as well

	Method	Params	GFLOPs	FPS	PQ
(0)	ViT-Adapter + Mask2Former	349M	830	29	57.1
(1)	↳ w/o ViT-Adapter	342M	700	36$^{\uparrow 7}$	56.7$^{\downarrow 0.4}$
(2)	↳ w/o Pixel decoder	337M	685	61$^{\uparrow 25}$	56.9$^{\uparrow 0.2}$
(3)	↳ w/o Multi-scale	328M	673	64$^{\uparrow 3}$	56.7$^{\downarrow 0.2}$
(4)	↳ w/o Transformer decoder	316M	828	61$^{\downarrow 3}$	56.2$^{\downarrow 0.5}$
(5)	↳ w/o Masking = **EoMT**	316M	669	128$^{\uparrow 67}$	56.0$^{\downarrow 0.2}$

Table 1. **From ViT-Adapter + Mask2Former to EoMT.** Evaluated on COCO *val2017* [43].

as the number of model parameters. We use an NVIDIA H100 GPU, FlashAttention-2 [18], `torch.compile` [2], and a batch size of 1, unless otherwise specified. The FLOPs are obtained using *fvcore* [53] and reported in terms of GFLOPs (FLOPs $\times 10^9$).

See Appendix A for additional implementation details.

4.2. Main Results

From ViT-Adapter + Mask2Former to EoMT. In Tab. 1, we evaluate the stepwise removal of task-specific components from ViT-Adapter + Mask2Former (M2F) [13, 15] to obtain our proposed EoMT model. We find that removing all task-specific components reduces the PQ only slightly, from 57.1 to 56.0, but increases the prediction speed by a substantial $4.4\times$. Interestingly, this FPS improvement is much larger than the FLOPs improvement. This is because EoMT relies solely on the plain ViT, allowing it to leverage the highly optimized Transformer architecture without being bottlenecked by custom components that contribute little to the segmentation accuracy. Looking at the intermediate steps, we find that removing the ViT-Adapter, pixel decoder, and multi-scale features in steps (1–3) reduce performance by just 0.4 PQ while making the model $2.2\times$ faster. Step (4) temporarily increases FLOPs and reduces FPS, as the mask module is repeatedly applied to generate intermediate segmentation masks for masked attention, with upscaling being the most compute-intensive part. However, in step (5), we remove masked attention entirely, eliminating this overhead. The proposed mask annealing strategy enables masked-attention-free inference, further accelerating the model by $2.1\times$ with minimal impact on the PQ. More detailed results on these steps are provided in Appendix B and Tab. B. Overall, the stepwise removal of task-specific components ultimately yields a model that is significantly faster, remarkably simpler, and nearly as accurate.

Impact of pre-training. Next, we explore the impact of pre-training. Specifically, we consider large-scale self- and weakly-supervised pre-training with DINOv2 [51] and EVA-02 [28], respectively, as well as supervised ImageNet-21K and ImageNet-1K pre-training with DeiT III [56], all with ViT-L. The results in Tab. 2 show that large-scale pre-training allows EoMT to obtain a similar PQ to ViT-Adapter + M2F. For DINOv2 and EVA-02, the overall PQ gap is only 1.1 or 1.2, whereas it increases to 3.9 and 6.1 for ImageNet-21K [22] and ImageNet-1K [54], respectively. These results confirm our hypothesis that large-scale pre-training renders complex task-specific components increasingly redundant. We expect that the *masked image modeling* pre-training objectives of DINOv2 and EVA-02 contribute to this effect, as it enhances the semantic understanding of patches, which is essential for image segmentation [36].

Model	Pre-train	Params	GFLOPs	FPS	PQ
ViT-Adapter + M2F	DINOv2	349M	830	29	57.1
EoMT w/ Masking		316M	828	61$^{\uparrow 32}$	56.2$^{\downarrow 0.9}$
EoMT		316M	669	128$^{\uparrow 99}$	56.0$^{\downarrow 1.1}$
ViT-Adapter + M2F	EVA-02	349M	829	25	56.7
EoMT w/ Masking		316M	826	52$^{\uparrow 27}$	56.0$^{\downarrow 0.7}$
EoMT		316M	667	77$^{\uparrow 52}$	55.5$^{\downarrow 1.2}$
ViT-Adapter + M2F	IN21K	349M	830	29	53.9
EoMT w/ Masking		316M	828	61$^{\uparrow 32}$	51.0$^{\downarrow 2.9}$
EoMT		316M	669	128$^{\uparrow 99}$	50.0$^{\downarrow 3.9}$
ViT-Adapter + M2F	IN1K	349M	830	29	50.4
EoMT w/ Masking		316M	828	61$^{\uparrow 32}$	45.9$^{\downarrow 4.5}$
EoMT		316M	669	128$^{\uparrow 99}$	44.3$^{\downarrow 6.1}$

Table 2. **Pre-training.** EoMT performs significantly better with advanced pre-training, *i.e.*, DINOv2 [51] or EVA-02 [28]. Evaluated on COCO *val2017* [43].

Model	Size	Params	GFLOPS	FPS	PQ
ViT-Adapter + M2F	g	1209M	2510	20	57.7
EoMT w/ Masking		1164M	2689	35$^{\uparrow 15}$	57.2$^{\downarrow 0.5}$
EoMT		1164M	2261	55$^{\uparrow 35}$	57.0$^{\downarrow 0.7}$
ViT-Adapter + M2F	L	349M	830	29	57.1
EoMT w/ Masking		316M	828	61$^{\uparrow 32}$	56.2$^{\downarrow 0.9}$
EoMT		316M	669	128$^{\uparrow 99}$	56.0$^{\downarrow 1.1}$
ViT-Adapter + M2F	B	121M	347	32	54.4
EoMT w/ Masking		93M	286	104$^{\uparrow 72}$	51.5$^{\downarrow 2.9}$
EoMT		93M	216	261$^{\uparrow 229}$	50.6$^{\downarrow 3.8}$
ViT-Adapter + M2F	S	47M	165	33	50.5
EoMT w/ Masking		24M	89	108$^{\uparrow 75}$	46.1$^{\downarrow 4.4}$
EoMT		24M	68	330$^{\uparrow 297}$	44.7$^{\downarrow 5.8}$

Table 3. **Model size.** EoMT performs significantly better as the ViT [23] model size increases. Evaluated on COCO *val2017* [43].

Impact of model size. So far, we have only shown results for one model size, *i.e.*, ViT-L. In Tab. 3, we assess the impact of model size on the importance of task-specific components. The results show that the relative performance of EoMT compared to ViT-Adapter + M2F improves as the size of the model increases. While EoMT lags behind the more complex model by 5.8 PQ for ViT-S, the performance gap narrows significantly as the model scales up, becoming only 0.7 PQ for ViT-g. This shows that increasing the capacity of the ViT, combined with strong pre-training, allows EoMT to better solve the image segmentation task. This further confirms our hypothesis that the necessity for task-specific components decreases as the capacity of the model increases. At the same time, we also observe that replacing the complex ViT-Adapter + M2F model with EoMT significantly improves the prediction speed at all model sizes. Interestingly, this allows EoMT to use larger models and

Method	Backbone	Pre-training	Params	COCO *val2017* [43]				ADE20K *val* [70]			
				Input size	GFLOPs	FPS	PQ	Input size	GFLOPs	FPS	PQ
Mask2Former† [15]	Swin-L [45]	IN21K	216M	800^2	868	24	57.8	640^2	–	33	48.1
kMaX-DeepLab [66]	ConvNext-L [46]	IN21K	232M	1281^2	–	–	58.0	1281^2	1302	–	50.9
OneFormer† [35]	DiNAT-L [30]	IN21K	223M	800^2	736	20	58.0	1280^2	1369	10	51.5
OneFormer† [35]	DiNAT-L [30]	IN21K	223M	–	–	–	–	1280^2	1369	10	53.5[c]
MaskDINO† [40]	Swin-L [45]	IN21K	223M	800^2	1326	14	58.3	–	–	–	–
Mask2Former‡ [15]	ViT-Adapter-L‡ [13]	DINOv2	349M	640^2	830	29	57.1	640^2	830	29	51.8[c]
Mask2Former‡ [15]	ViT-Adapter-L‡ [13]	DINOv2	354M	1280^2	4817	10	59.7	1280^2	4817	10	53.0[c]
Mask2Former‡ [15]	ViT-Adapter-g‡ [13]	DINOv2	1209M	640^2	2510	20	57.7	640^2	2510	20	52.6[c]
Mask2Former‡ [15]	ViT-Adapter-g‡ [13]	DINOv2	1216M	1280^2	13790	6	59.9	1280^2	13790	6	54.2[c]
EoMT (Ours)	ViT-L [23]	DINOv2	316M	640^2	669	128	56.0	640^2	669	128	50.6[c]
EoMT (Ours)	ViT-L [23]	DINOv2	322M	1280^2	4146	30	58.3	1280^2	4146	30	51.7[c]
EoMT (Ours)	ViT-g [51]	DINOv2	1164M	640^2	2261	55	57.0	640^2	2261	55	51.3[c]
EoMT (Ours)	ViT-g [51]	DINOv2	1171M	1280^2	12712	12	59.2	1280^2	12712	12	52.8[c]

Table 4. **EoMT for panoptic segmentation.** †During inference, these models resize the shortest side of images to the indicated scale, while preserving the aspect ratio. ‡Our re-implementation. [c]Models for these ADE20K results are pre-trained for COCO panoptic segmentation.

Method	Backbone	Pre-training	Params	Cityscapes *val* [17]				ADE20K *val* [70]			
				Input size	GFLOPs	FPS	mIoU	Input size	GFLOPs	FPS	mIoU
Mask2Former† [15]	Swin-L [45]	IN21K	216M	1024×2048	–	14	83.3	640^2	–	33	56.1
MaskDINO† [40]	Swin-L [45]	IN21K	223M	–	–	–	–	640^2	–	–	56.6
OneFormer† [35]	ConvNext-XL [46]	IN21K	373M	1024×2048	775	7	83.6	640^2	607	21	57.4
OneFormer† [35]	DiNAT-L [30]	IN21K	223M	1024×2048	450	14	83.1	896^2	678	19	58.1
kMaX-DeepLab [66]	ConvNext-L [46]	IN21K	232M	1025×2049	1673	–	83.5	–	–	–	–
Mask2Former [15]	ViT-L [23]	DINOv2 + DA	–	896×1792	–	–	84.8	896^2	–	–	59.4
Mask2Former‡ [15]	ViT-Adapter-L‡ [13]	DINOv2	351M	1024^2	5200	7	84.5	512^2	910	21	58.9
EoMT (Ours)	ViT-L [23]	DINOv2	319M	1024^2	4350	25	84.2	512^2	721	92	58.4

Table 5. **EoMT for semantic segmentation.** †On ADE20K, these models resize the shortest side of images to the indicated scale during inference, while preserving the aspect ratio. ‡Our re-implementation. ViT-Adapter + Mask2Former and EoMT use windowed inference, dividing each image into multiple crops, and the FLOPs and FPS results account for this. DA is Depth Anything [65].

obtain higher scores than ViT-Adapter + M2F, while being significantly faster. For example, EoMT with ViT-L obtains a PQ of 56.0 at 128 FPS, which is both significantly faster and more accurate than ViT-Adapter + M2F with ViT-B, at a PQ of 54.4 at 32 FPS. As shown in Fig. 1, EoMT obtains a better PQ *vs*. FPS trade-off across all model sizes that we tested. This shows the power of EoMT and its simplicity.

EoMT on different benchmarks. To demonstrate EoMT's versatility across image segmentation tasks and datasets, we evaluate its performance on multiple benchmarks. For panoptic segmentation, as shown in Tab. 4, EoMT achieves a significantly better PQ *vs*. FPS trade-off than ViT-Adapter + M2F [13, 15] on COCO [43], and a similar trade-off on ADE20K [70], while being significantly simpler. Moreover, on COCO, EoMT achieves a PQ that is on par with existing state-of-the-art methods while being up to $2.1\times$ faster, highlighting the strength of our simplified design.

For semantic segmentation on Cityscapes [17] and ADE20K, Tab. 5 shows that EoMT performs comparably to the more complex ViT-Adapter + M2F baseline in terms of mIoU, while being considerably faster, *i.e*., up to $4.4\times$. Compared to other state-of-the-art methods, EoMT again obtains competitive performance, even though it is a much simpler and more efficient model. This shows that EoMT is also highly effective for semantic segmentation, providing a strong balance between speed and accuracy.

For instance segmentation on COCO, we see similar positive results in Tab. 6. Although the overall accuracy drop is slightly higher than for the other tasks, EoMT still achieves a better AP *vs*. FPS trade-off than ViT-Adapter + M2F, *e.g*., 48.8 AP at 30 FPS *vs*. 47.6 AP at 29 FPS. Overall, these results demonstrate the strength and general applicability of EoMT, as it performs effectively across a variety of segmentation tasks and datasets.

Importance of mask annealing. In Tab. 1, we observed that mask annealing allows EoMT to remove masked attention during inference to improve efficiency, while keeping the PQ roughly the same. In Tab. 7, we compare mask annealing to alternative strategies. The first alternative approach of training with masked attention and simply disabling it during inference causes a severe performance drop, showing that using a different strategy at training and inference time is ineffective. The second alternative approach of disabling masked attention during both training and inference does not fail catastrophically, but it results in a PQ that is still significantly lower than when using masked attention. In contrast, mask annealing enables EoMT to leverage masked attention during early training stages to improve convergence and roughly maintain the PQ, while eliminating the need for masking during inference, thereby more

Method	Backbone	PT	Params	Input	GFLOPs	FPS	AP
OneFormer† [35]	DiNAT-L [30]	I	223M	800^2	736	20	49.2
Mask2Former† [15]	Swin-L [45]	I	216M	800^2	868	24	50.1
MaskDINO† [40]	Swin-L [45]	I	223M	800^2	1326	14	52.3
Mask2Former‡ [15]	ViT-Adapter-L‡ [13]	D	349M	640^2	830	29	47.6
Mask2Former‡ [15]	ViT-Adapter-L‡ [13]	D	354M	1280^2	4817	10	51.4
EoMT (Ours)	ViT-L [23]	D	316M	640^2	669	128	45.2
EoMT (Ours)	ViT-L [23]	D	322M	1280^2	4146	30	48.8

Table 6. **EoMT for instance segmentation.** Evaluated on COCO *val2017* [43]. †During inference, these models resize the shortest side of images to the indicated scale while preserving the aspect ratio. ‡Our re-implementation. **PT** indicates pre-training (**I** for ImageNet-21K [22] and **D** for DINOv2 [51]).

Training	Inference	GFLOPs	FPS	PQ
✓ Masking	✓ Masking	828	61	56.2
✓ Masking	✗ w/o Masking	669	128	27.4$^{\downarrow 28.8}$
✗ w/o Masking	✗ w/o Masking	669	128	53.2$^{\downarrow 3.0}$
✓→✗ *Mask annealing*	✗ w/o Masking	669	128	56.0$^{\downarrow 0.2}$

Table 7. ***Mask annealing.*** Effectively removes masked attention during inference. When never masking, intermediate masks are not predicted or supervised. Evaluated on COCO *val2017* [43].

than doubling inference speed. As such, it is a key component of EoMT. More results on the general applicability of mask annealing are provided in Appendix B and Tab. C.

4.3. Further Experiments

Out-of-distribution generalization. Since EoMT is ViT-based, it supports initialization with vision foundation model (VFMs) like DINOv2 [51], which achieve state-of-the-art out-of-distribution (OOD) generalization [25, 37]. In contrast, prior segmentation models typically rely on ConvNeXt [46], Swin [45], or other non-plain ViT backbones, which cannot leverage VFM pre-training due to their incompatible architectures. We evaluate OOD generalization on the BRAVO [58] benchmark, by training on Cityscapes and evaluating on multiple OOD datasets. Tab. 8 shows that DINOv2-based models demonstrate superior OOD generalization, outperforming the Swin-based model by more than 7.8 mIoU despite similar in-distribution performance on Cityscapes. Importantly, EoMT leverages the strong OOD performance of VFMs while being significantly more efficient than existing VFM-based methods, which is essential for real-world applications. Further analysis of OOD confidence estimation, showing that EoMT produces significantly more reliable confidence scores than ViT-Adapter + M2F, is provided in Appendix C.

Token merging. EoMT benefits from ongoing ViT advancements by using the plain ViT architecture. One such advancement is token merging [7, 48, 50], which improves efficiency by merging semantically redundant tokens while preserving segmentation accuracy. As shown in Tab. 9, EoMT is compatible with ALGM [50], the state-of-the-art token merging method for semantic segmentation. With ALGM, EoMT's throughput increases by up to 31% without affecting mIoU. In contrast, while ViT-Adapter + M2F reduces in FLOPs, it sees no throughput gain, as it is bottlenecked by its additional components and the overhead of 'unmerging' tokens for ViT-Adapter interaction. This highlights the benefit of keeping EoMT close to the plain ViT.

Method	Backbone	Pre-training	mIoU_{ID}	mIoU_{OOD}
M2F [15]	Swin-L [45]	IN21K	83.3	69.4
M2F‡ [15]	ViT-Adapter-L‡ [13]	DINOv2	84.5	78.0
EoMT (Ours)	ViT-L [23]	DINOv2	84.2	77.2

Table 8. **Out-of-distribution generalization.** Despite similar in-distribution performance on Cityscapes *val* (mIoU_{ID}), DINOv2-based models generalize significantly better out-of-distribution (mIoU_{OOD}). Trained on Cityscapes *train* [17], evaluated on BRAVO [58]. ‡Our re-implementation.

Method	Token merging	GFLOPs	Throughput	mIoU
ViT-Adapter + M2F	✗	5200	9	84.5
	✓	3031	9$^{\uparrow 0\%}$	84.3
EoMT (Ours)	✗	4350	29	84.2
	✓	1183	38$^{\uparrow 31\%}$	84.2

Table 9. **Token merging.** EoMT's ViT-only design is no longer bottlenecked by additional task-specific components, enabling ViT optimizations like ALGM [50]. Throughput is in *images per second*, with a batch size of 32. Evaluated on Cityscapes *val* [17].

5. Conclusion

In this paper, we show that task-specific components for image segmentation with Vision Transformers (ViTs) become increasingly redundant as model size and pre-training are scaled up. By removing all such components, we introduce the Encoder-only Mask Transformer (EoMT), a segmentation model that purely uses a plain ViT, revealing that *your ViT is secretly an image segmentation model*. EoMT delivers both high accuracy and impressive speed, with a significantly simpler design than existing models. Our findings indicate that compute resources should not be spent on adding architectural complexity, but rather on scaling the ViT and pre-training, as we found that these factors are critical in improving performance. As a simple and scalable approach, EoMT provides a solid foundation for next-generation segmentation models that readily adapts to advances in the rapidly evolving fields of Transformers and foundation models.

Acknowledgements. This work was supported by Chips Joint Undertaking (Chips JU) in EdgeAI "Edge AI Technologies for Optimised Performance Embedded Processing" project, grant agreement no. 101097300. Niccolò Cavagnero acknowledges travel support from the European Union's Horizon 2020 research and innovation program, grant agreement no. 951847. Giuseppe Averta was supported by FAIR - Future Artificial Intelligence Research, Next-GenEU (PNRR – MISS. 4 COMP. 2, INV. 1.3 – D.D. 1555 11/10/2022, PE00000013). We also acknowledge the CINECA award under the ISCRA initiative and the Dutch national e-infrastructure with the support of the SURF Cooperative, grant agreement no. EINF-9663 and EINF-11151, financed by the Dutch Research Council (NWO), for the availability of high-performance computing resources and support.

References

[1] Ibrahim Alabdulmohsin, Xiaohua Zhai, Alexander Kolesnikov, and Lucas Beyer. Getting ViT in Shape: Scaling Laws for Compute-Optimal Model Design. In *NeurIPS*, 2023. 1

[2] Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, C. K. Luk, Bert Maher, et al. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *ASPLOS*, 2024. 6, 13

[3] Ali Athar, Alexander Hermans, Jonathon Luiten, Deva Ramanan, and Bastian Leibe. TarViS: A Unified Approach for Target-Based Video Segmentation. In *CVPR*, 2023. 2

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer Normalization. *arXiv preprint arXiv:1607.06450*, 2016. 3

[5] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT Pre-Training of Image Transformers. In *ICLR*, 2022. 1, 5

[6] Lucas Beyer, Pavel Izmailov, Alexander Kolesnikov, Mathilde Caron, Simon Kornblith, Xiaohua Zhai, Matthias Minderer, Michael Tschannen, Ibrahim Alabdulmohsin, and Filip Pavetic. FlexiViT: One Model for All Patch Sizes. In *CVPR*, 2023. 12

[7] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token Merging: Your ViT But Faster. In *ICLR*, 2023. 2, 8, 13

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers. In *ECCV*, 2020. 2, 3

[9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *ICCV*, 2021. 3

[10] Niccolò Cavagnero, Gabriele Rosi, Claudia Cuttano, Francesca Pistilli, Marco Ciccone, Giuseppe Averta, and Fabio Cermelli. PEM: Prototype-based Efficient MaskFormer for Image Segmentation. In *CVPR*, 2024. 2

[11] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *ECCV*, 2018. 2

[12] Wuyang Chen, Xianzhi Du, Fan Yang, Lucas Beyer, Xiaohua Zhai, Tsung-Yi Lin, Huizhong Chen, Jing Li, Xiaodan Song, Zhangyang Wang, et al. A Simple Single-Scale Vision Transformer for Object Localization and Instance Segmentation. In *ECCV*, 2022. 3

[13] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision Transformer Adapter for Dense Predictions. In *ICLR*, 2023. 1, 3, 6, 7, 8, 14, 15

[14] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-Pixel Classification is Not All You Need for Semantic Segmentation. In *NeurIPS*, 2021. 1, 2, 3, 4

[15] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention Mask Transformer for Universal Image Segmentation. In *CVPR*, 2022. 1, 2, 3, 4, 5, 6, 7, 8, 12, 13, 14, 15

[16] Jack Choquette. NVIDIA Hopper H100 GPU: Scaling Performance. *IEEE Micro*, 43(3):9–17, 2023. 2

[17] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *CVPR*, 2016. 5, 7, 8, 12, 14

[18] Tri Dao. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. In *ICLR*, 2024. 2, 6

[19] Tri Dao, Daniel Y Fu, Stefano Ermon, Atri Rudra, and Christopher Re. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. In *NeurIPS*, 2022. 2

[20] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision Transformers Need Registers. In *ICLR*, 2024. 13

[21] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme Ruiz, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd Van Steenkiste, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Collier, Alexey A. Gritsenko, Vighnesh Birodkar, Cristina Nader Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetic, Dustin Tran, Thomas Kipf, Mario Lucic, Xiaohua Zhai, Daniel Keysers, Jeremiah J. Harmsen, and Neil Houlsby. Scaling Vision Transformers to 22 Billion Parameters. In *ICML*, 2023. 1

[22] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 3, 6, 8, 13

[23] 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 *ICLR*, 2021. 1, 3, 6, 7, 8, 13, 14

[24] Anne C. Elster and Tor A. Haugdahl. Nvidia Hopper GPU and Grace CPU Highlights. *Computing in Science & Engineering*, 24(2):95–100, 2022. 2

[25] Brunó B. Englert, Fabrizio J. Piva, Tommie Kerssies, Daan De Geus, and Gijs Dubbelman. Exploring the Benefits of Vision Foundation Models for Unsupervised Domain Adaptation. In *CVPR Workshops*, 2024. 8

[26] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *IJCV*, 2010. 5

[27] Yuxin Fang, Bencheng Liao, Xinggang Wang, Jiemin Fang, Jiyang Qi, Rui Wu, Jianwei Niu, and Wenyu Liu. You Only

Look at One Sequence: Rethinking Transformer in Vision through Object Detection. In *NeurIPS*, 2021. 3

[28] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. EVA-02: A Visual Representation for Neon Genesis. *Image and Vision Computing*, 2024. 1, 2, 3, 6, 13

[29] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation. In *CVPR*, 2021. 12

[30] Ali Hassani and Humphrey Shi. Dilated Neighborhood Attention Transformer. *arXiv preprint arXiv:2209.15001*, 2022. 3, 7, 8, 13

[31] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood Attention Transformer. In *CVPR*, 2023. 3

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 3

[33] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 2

[34] Jie Hu, Linyan Huang, Tianhe Ren, Shengchuan Zhang, Rongrong Ji, and Liujuan Cao. You Only Segment Once: Towards Real-Time Panoptic Segmentation. In *CVPR*, 2023. 4

[35] Jitesh Jain, Jiachen Li, Mang Tik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. OneFormer: One Transformer To Rule Universal Image Segmentation. In *CVPR*, 2023. 1, 2, 4, 7, 8, 13

[36] Tommie Kerssies, Daan De Geus, and Gijs Dubbelman. How to Benchmark Vision Foundation Models for Semantic Segmentation? In *CVPR Workshops*, 2024. 2, 3, 5, 6, 13

[37] Tommie Kerssies, Daan de Geus, and Gijs Dubbelman. First Place Solution to the ECCV 2024 BRAVO Challenge: Evaluating Robustness of Vision Foundation Models for Semantic Segmentation. *arXiv preprint arXiv:2409.17208*, 2024. 8

[38] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollar. Panoptic Feature Pyramid Networks. In *CVPR*, 2019. 2

[39] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic Segmentation. In *CVPR*, 2019. 1, 2, 5

[40] Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M. Ni, and Heung-Yeung Shum. Mask DINO: Towards a Unified Transformer-Based Framework for Object Detection and Segmentation. In *CVPR*, 2023. 2, 4, 7, 8

[41] Xiangtai Li, Haobo Yuan, Wei Li, Henghui Ding, Size Wu, Wenwei Zhang, Yining Li, Kai Chen, and Chen Change Loy. Omg-seg: Is one model good enough for all segmentation? In *CVPR*, 2024. 2

[42] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring Plain Vision Transformer Backbones for Object Detection. In *ECCV*, 2022. 4

[43] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014. 1, 5, 6, 7, 8, 13, 14, 15

[44] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *CVPR*, 2017. 3

[45] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *ICCV*, 2021. 3, 7, 8, 13, 14

[46] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *CVPR*, 2022. 3, 7, 8

[47] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *ICLR*, 2019. 5

[48] Chenyang Lu, Daan de Geus, and Gijs Dubbelman. Content-Aware Token Sharing for Efficient Semantic Segmentation With Vision Transformers. In *CVPR*, 2023. 2, 8

[49] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In *3DV*, 2016. 12

[50] 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 *CVPR*, 2024. 2, 8, 13

[51] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning Robust Visual Features without Supervision. *TMLR*, 2024. 1, 2, 3, 5, 6, 7, 8, 13, 15

[52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, 2021. 3

[53] Meta Research. fvcore, 2023. 6

[54] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. 6

[55] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for Semantic Segmentation. In *ICCV*, 2021. 2

[56] Hugo Touvron, Matthieu Cord, and Hervé Jégou. DeiT III: Revenge of the ViT. In *ECCV*, 2022. 6, 13

[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *NeurIPS*, 2017. 2, 3

[58] Tuan-Hung Vu, Eduardo Valle, Andrei Bursuc, Tommie Kerssies, Daan de Geus, Gijs Dubbelman, Long Qian, Bingke Zhu, Yingying Chen, Ming Tang, Jinqiao Wang, Tomáš Vojíř, Jan Šochman, Jiří Matas, Michael Smith, Frank Ferrie, Shamik Basu, Christos Sakaridis, and Luc Van Gool. The BRAVO Semantic Segmentation Challenge Results in UNCV2024. In *ECCV Workshops*, 2024. 8, 14

[59] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. MaX-DeepLab: End-to-End Panoptic Segmentation with Mask Transformers. In *CVPR*, 2021. 2, 3

[60] Ross Wightman. PyTorch Image Models, 2019. 12, 13

[61] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-Art Natural Language Processing. In *EMNLP Demos*, 2020. 12

[62] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2, 2019. 5

[63] Chunlong Xia, Xinliang Wang, Feng Lv, Xin Hao, and Yifeng Shi. ViT-CoMer: Vision Transformer with Convolutional Multi-scale Feature Interaction for Dense Predictions. In *CVPR*, 2024. 1, 3

[64] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In *NeurIPS*, 2021. 3

[65] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data. In *CVPR*, 2024. 7

[66] Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. k-means Mask Transformer. In *ECCV*, 2022. 1, 2, 4, 7

[67] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling Vision Transformers. In *CVPR*, 2022. 1

[68] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid Loss for Language Image Pre-Training. In *ICCV*, 2023. 1

[69] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid Scene Parsing Network. In *CVPR*, 2017. 2

[70] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene Parsing through ADE20K Dataset. In *CVPR*, 2017. 5, 7, 12, 13

[71] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *ICLR*, 2021. 3

Appendix

Table of contents:

A. Implementation Details

A.1. Models

Visualizations of model configurations. In Sec. 3.2, we explain how we gradually remove task-specific components. We visualize the architectures of the resulting intermediate configurations in Fig. A. Here, the subscript $\boldsymbol{F}_i$ indicates that the features have a resolution of $\frac{1}{i}$ of the input image. The visualized model numbers correspond to those reported in Tab. 1.

Libraries. For Mask2Former [15], we use the implementation of `Huggingface Transformers` [61]. For pre-trained models, we use `timm` [60].

Pre-trained models. In Tab. A, we specify the `timm` model weights that we use for the experiments in this work. To support a patch size of 16×16 and different input sizes, we resize the patch embedding kernel and positional embeddings of pre-trained models following the FlexiViT [6] implementation of `timm`. Specifically, the patch embedding kernel is resized to a 16×16 patch size by approximately inverting the effect of patch resizing. The positional embeddings are resized to the required token grid size by using bicubic interpolation. The patch embedding kernel and positional embeddings are resized prior to fine-tuning, and keep the same size during fine-tuning.

Queries. In accordance with Mask2Former [15], the models for panoptic and instance segmentation use $K = 200$ queries, while the models for semantic segmentation use $K = 100$ queries. For ViT-S and ViT-B we use $L_2 = 3$, for ViT-L we use $L_2 = 4$, and for ViT-g we use $L_2 = 5$. For EoMT, adding 200 tokens to a model that processes 640×640 images with a 16×16 patch size results in an increase of 12.5% of the tokens processed by a ViT block, *but only for the last L_2 ViT blocks*. As $L_1 = 20$ and $L_2 = 4$ for ViT-L, the total number of tokens processed in the entire ViT increases by only 2.1%.

A.2. Training

Augmentation. During training, we apply the same data augmentation techniques as used by Mask2Former [15]. Specifically, training images undergo random horizontal flipping, random scale jittering, padding if necessary, and random cropping. Random color jittering is additionally applied for ADE20K [70] and Cityscapes [17]. For panoptic and instance segmentation, we use large-scale jitter [29]

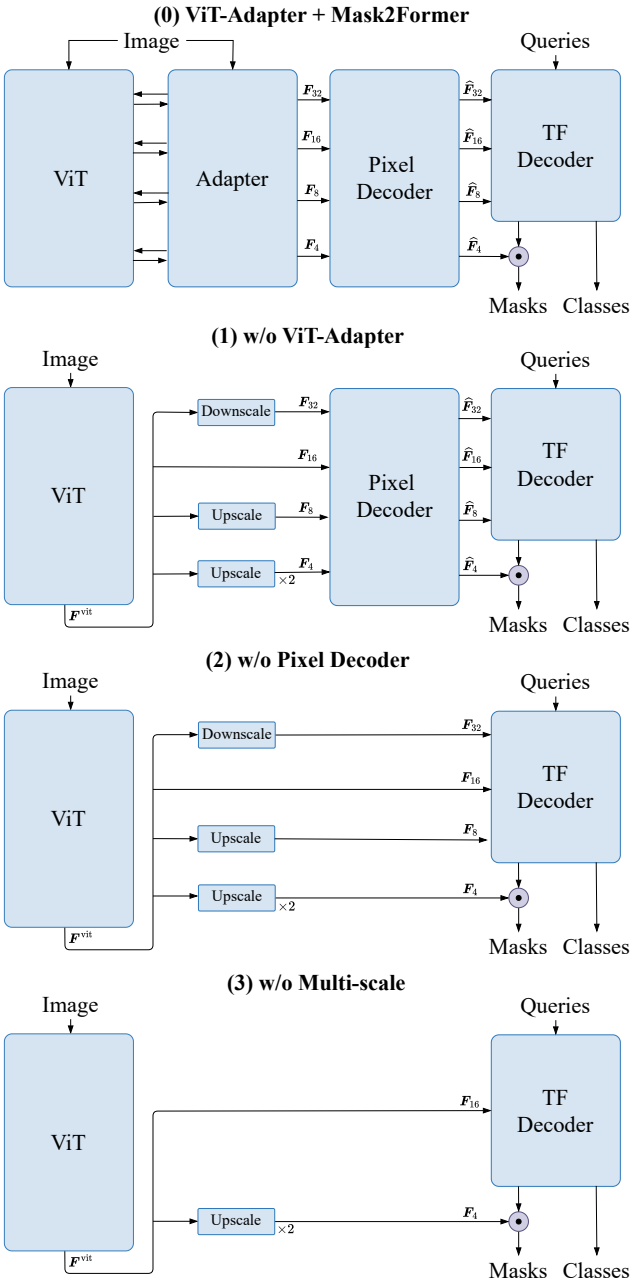


Figure A. **Removing task-specific components.** We visualize the architectures of the resulting intermediate configurations.

(between $0.1\times$ and $2.0\times$), and for semantic segmentation we use normal-scale jitter (between $0.5\times$ and $2.0\times$).

Loss function. To supervise our models, we adopt the same loss function as Mask2Former [15]. Specifically, across all tasks and datasets, we use the cross-entropy (CE) loss for the class logits, and the binary-cross entropy (BCE) and the Dice loss [49] for the mask logits. The individual losses are weighted using scalars, resulting in the total loss function:

$$\mathcal{L}_{\text{tot}} = \lambda_{\text{bce}}\mathcal{L}_{\text{bce}} + \lambda_{\text{dice}}\mathcal{L}_{\text{dice}} + \lambda_{\text{ce}}\mathcal{L}_{\text{ce}}, \qquad (2)$$

Model	Pre-training	`timm` model
ViT-g	DINOv2 [20, 51]	`vit_giant_patch14_reg4_dinov2`
ViT-L	DINOv2 [20, 51]	`vit_large_patch14_reg4_dinov2`
ViT-B	DINOv2 [20, 51]	`vit_base_patch14_reg4_dinov2`
ViT-S	DINOv2 [20, 51]	`vit_small_patch14_reg4_dinov2`
ViT-L	EVA-02 [28]	`eva02_large_patch14_224.mim_m38m`
ViT-L	DeiT-III (ImageNet-21K) [22, 56]	`deit3_large_patch16_384.fb_in22k_ft_in1k`
ViT-L	DeiT-III (ImageNet-1K) [22, 56]	`deit3_large_patch16_384.fb_in1k`

Table A. **Model specification.** For each ViT backbone [23] used in this work, we specify the `timm` model [60] that we use.

	Method	Params	GFLOPs	FPS	Panoptic Quality (PQ)			Average Precision (AP)			
					All	Things	Stuff	All	Large	Medium	Small
(0)	ViT-Adapter + Mask2Former	349M	830	29	57.1	62.7	48.7	47.6	73.2	53.4	23.4
(1)	↳ w/o ViT-Adapter	342M	700	36	56.7	62.3	48.3	46.9	72.7	52.9	22.7
(2)	↳ w/o Pixel decoder	337M	685	62	56.9	62.3	48.6	46.8	73.1	52.6	22.1
(3)	↳ w/o Multi-scale	328M	673	64	56.7	62.2	48.4	46.2	73.1	52.3	21.4
(4)	↳ w/o Transformer decoder	316M	828	61	56.2	61.4	48.4	45.6	72.1	51.4	20.8
(5)	↳ w/o Masking = **EoMT**	316M	669	128	56.0	61.2	48.2	45.2	72.2	51.0	20.3

Table B. **From ViT-Adapter + Mask2Former to EoMT in detail.** Evaluated on COCO *val2017*.

where λ_{bce}, λ_{dice}, and λ_{ce} are set to 5.0, 5.0, and 2.0, respectively, following Mask2Former [15].

Learning rate warm-up. We use a two-stage linear learning rate warm-up for all models. In practice, we first warm-up the randomly initialized parameters for 500 iterations, while keeping the pre-trained parameters frozen. After 500 iterations, we warm-up the pre-trained parameters for 1000 iterations. In both cases, the initial learning rate is set to 0.

A.3. Evaluation

Image processing. For panoptic and instance segmentation, we use padded inference, resizing the longer side of the image to the input size, and padding the shorter side with zeros to create a square image. For semantic segmentation, we apply windowed inference, resizing the shorter side of the image to the input size, and processing the image through the model in several proportionally spaced square crops, in a sliding-window manner [36].

Efficiency measurements. For existing works, we report FLOPs from the respective papers but measure FPS the same way that we measure it for our models, on the same hardware. For ViT-Adapter + M2F and our models, we calculate the FLOPs ourselves. When measuring FPS, `torch.compile` [2] is disabled for Mask2Former [15] with Swin-L [45] on ADE20K [70] due to compilation errors. On COCO [43], `torch.compile` only yields a small speedup for this model ($< 10\%$). Additionally, mixed precision is not supported for OneFormer [35] with DiNAT-L [30], thus we use full precision here.

Token merging. For our token merging experiment in Sec. 4.3 and Tab. 9, we evaluate the throughput of the model in *images per second*, following existing work for token merging [7, 50]. This means that we use a batch size of 32, apply ALGM [50] for token merging, and report the number of images that are processed per second, averaged over the entire validation set. ALGM adaptively determines the number of tokens that should be merged per image, based on image complexity. To allow batch processing, we identify the lowest number of mergeable tokens per image across the batch according to the ALGM token merging criterion, and use that number of merged tokens for all images in the batch.

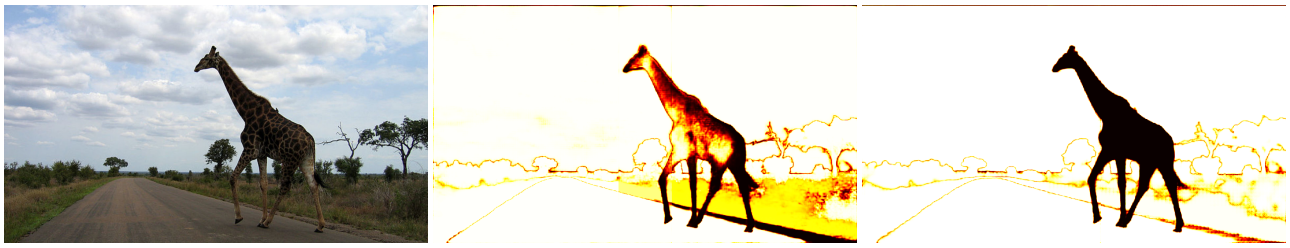
Importantly, ALGM is applied only during inference. Thus, the throughput improvement in Tab. 9 is achieved simply by applying ALGM to EoMT and processing batches of images, with no additional training required.

B. Detailed Experimental Analysis

From ViT-Adapter + M2F to EoMT in detail. In Tab. B, we provide more detailed results on the impact of the removal of task-specific components on both panoptic and instance segmentation on COCO [43]. For panoptic segmentation, we not only report the overall Panoptic Quality (PQ), but also separately the PQ for countable thing classes (PQ^{th}) and uncountable stuff classes (PQ^{st}). Similarly, for instance segmentation, we separately report AP for large (AP^{L}), medium (AP^{M}), and small objects (AP^{S}).

General applicability of mask annealing. In Tab. C, we assess the effect of our mask annealing strategy for both EoMT and the ViT-Adapter + M2F baseline. The results demonstrate the general applicability of mask annealing, as it is also effective for ViT-Adapter + M2F.

Number of blocks that process queries. In Tab. D, we examine the impact of varying L_2, *i.e.*, the number of ViT blocks in EoMT that process queries as well as patch tokens. EoMT demonstrates stable performance across different configurations, with the highest PQ for ViT-L observed around $L_2 = 4$, while the prediction speed in FPS is not



(1) Input image (2) ViT-Adapter + M2F [13, 15] confidence (3) **EoMT** (Ours) confidence

Figure B. **Qualitative comparison of out-of-distribution (OOD) confidence estimation.** EoMT reliably assigns low confidence to the full OOD object, while ViT-Adapter + M2F only does so partially. Darker colors indicate lower confidence. Trained on Cityscapes *train* [17], evaluated on BRAVO [58].

Training	Inference	Panoptic Quality (PQ)	
		EoMT	ViT-Ad. + M2F
✓ Masking	✓ Masking	56.2	57.1
✗ w/o Masking	✗ w/o Masking	53.2$^{\downarrow 3.0}$	54.0$^{\downarrow 3.1}$
✓→✗ *Mask annealing*	✗ w/o Masking	56.0$^{\downarrow 0.2}$	56.8$^{\downarrow 0.3}$

Table C. ***Mask annealing***. Effective for both EoMT and ViT-Adapter + M2F [13, 15]. When never masking, intermediate masks are not predicted or supervised. Evaluated on COCO *val2017* [43].

# Blocks (L_2)	Params	GFLOPs	FPS	PQ
9	316	688	126	55.7
6	316	676	127	55.7
4	316	669	128	**56.0**
2	316	660	128	55.4

Table D. **Number of blocks that process queries.** The model with $L_2 = 4$ achieves the best PQ, while FPS is not significantly affected by changing L_2. Evaluated on COCO *val2017* [43].

Method	Backbone	Pre-training	$\text{AUPRC}_{\text{OOD}}$
M2F [15]	Swin-L [45]	IN21K	56.8
M2F‡ [15]	ViT-Adapter-L‡ [13]	DINOv2	68.7
EoMT (Ours)	ViT-L [23]	DINOv2	**89.7**

Table E. **Quantitative comparison of out-of-distribution (OOD) confidence estimation.** EoMT achieves the highest $\text{AUPRC}_{\text{OOD}}$, demonstrating its superior confidence estimation. Trained on Cityscapes *train* [17], evaluated on BRAVO [58]. ‡Our re-implementation.

significantly affected by changing L_2. Consequently, we set $L_2 = 4$ as the default configuration for ViT-L.

C. Out-of-distribution Confidence Estimation

In Sec. 4.3, we discuss the out-of-distribution (OOD) generalization capabilities of EoMT. There, we show that DINOv2-based models, such as EoMT, significantly outperform non-ViT-based models such as Swin [45] in OOD generalization despite similar in-distribution (ID) performance.

Next, we also assess how well different models distinguish OOD regions from ID regions with their confidence scores. OOD regions, as defined in the BRAVO [58] benchmark, refer to novel object classes that were not present in the training data. We report the $\text{AUPRC}_{\text{OOD}}$ metric, which quantifies the model's ability to assign lower confidence to these unseen objects, ensuring they can be correctly identified as OOD.

As shown in Tab. E, EoMT achieves an $\text{AUPRC}_{\text{OOD}}$ of 89.7, significantly outperforming ViT-Adapter + M2F [13, 15] with a score of 68.7 and Swin [45] + M2F with a score of 56.8. The visualization in Fig. B further highlights that EoMT consistently assigns low confidence to the OOD object while maintaining high confidence for ID regions. In contrast, ViT-Adapter + M2F [13, 15] fails to reliably assign low confidence to all OOD pixels.

D. Qualitative Examples

In Fig. C we visualize predictions of ViT-Adapter + M2F [13, 15] and EoMT for panoptic segmentation on COCO [43].

(1) Input images

(2) Ground-truth annotations

(3) ViT-Adapter + M2F [13, 15] predictions

(4) **EoMT** (Ours) predictions

Figure C. **Qualitative examples for panoptic segmentation on COCO [43].** Using DINOv2-g [51] and a 1280 × 1280 input size.