How to Benchmark Vision Foundation Models for Semantic Segmentation?

Tommie Kerssies, Daan de Geus, Gijs Dubbelman
Eindhoven University of Technology
{t.kerssies, d.c.d.geus, g.dubbelman}@tue.nl

Abstract

Recent vision foundation models (VFMs) have demonstrated proficiency in various tasks but require supervised fine-tuning to perform the task of semantic segmentation effectively. Benchmarking their performance is essential for selecting current models and guiding future model developments for this task. The lack of a standardized benchmark complicates comparisons. Therefore, the primary objective of this paper is to study how VFMs should be benchmarked for semantic segmentation. To do so, various VFMs are fine-tuned under various settings, and the impact of individual settings on the performance ranking and training time is assessed. Based on the results, the recommendation is to fine-tune the ViT-B variants of VFMs with a 16×16 *patch size and a linear decoder, as these settings are representative of using a larger model, more advanced decoder and smaller patch size, while reducing training time by more than 13 times. Using multiple datasets for training and evaluation is also recommended, as the performance ranking across datasets and domain shifts varies. Linear probing, a common practice for some VFMs, is not recommended, as it is not representative of end-to-end fine-tuning. The benchmarking setup recommended in this paper enables a performance analysis of VFMs for semantic segmentation. The findings of such an analysis reveal that pretraining with promptable segmentation is not beneficial, whereas masked image modeling (MIM) with abstract representations is crucial, even more important than the type of supervision used. The code for efficiently fine-tuning VFMs for semantic segmentation can be accessed through the project page* [1].

1. Introduction

Semantic segmentation is the task of assigning a semantic class to each pixel in an image. Training a model to perform this task requires a large amount of images with semantic mask labels, presenting a significant challenge due to the intensive labor associated with such annotation [7, 12]. Vision Foundation Models (VFMs), *i.e.*, vision models pretrained on broad datasets, offer a solution to the labeling burden. A VFM acquires a fundamental understanding of visual features from the pretraining task, such that it can be used for a large variety of downstream tasks [4]. By transfer learning from the pretraining task to the downstream task of semantic segmentation, the need for extensive annotation is reduced [2, 30]. To gain insight into which VFMs are most effective for the task of semantic segmentation, a standardized benchmarking setup is crucial, but currently lacking. Therefore, the primary objective of this paper is to study how VFMs should be benchmarked for this task.

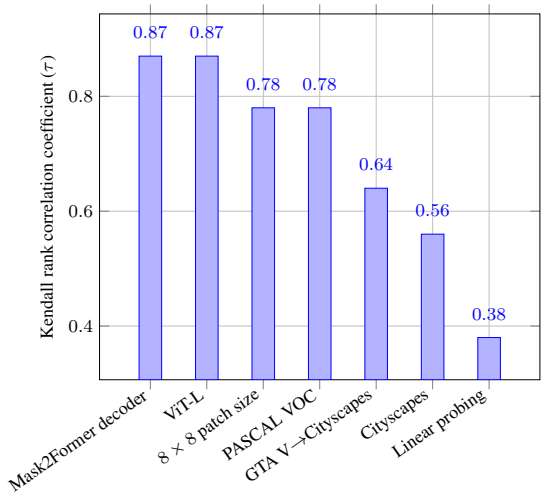

Figure 1. **Performance ranking impact of settings.** Kendall's τ is used to assess ranking similarity between VFMs under default settings (linear decoder, ViT-B, 16×16 patch size, ADE20K, end-to-end fine-tuning) and after changing individual settings, ranging from -1 for a reverse ranking to 1 for an identical ranking.

The Vision Transformer (ViT) [11] is a simple architecture which enables the use of global image context throughout the network. Recent VFMs have mostly adopted the same ViT architecture with various pretraining strategies [13, 14, 16, 19, 22, 28, 29, 32, 37]. These VFMs demonstrate that better pretraining on more data results in better downstream performance for various tasks. However,

[1] https://tue-mps.github.io/benchmark-vfm-ss/

it is unclear which pretraining strategies are most effective for the task of semantic segmentation specifically. Although some VFMs were evaluated for this task through supervised fine-tuning [14, 16, 22, 29, 32], a representative comparison is hindered by the lack of uniform evaluation settings. The inconsistency stems from diverse experimental conditions, including different downstream datasets, parameter freezing practices, number of tokens per image, model dimensions, and decoders. The precise impact of these factors on the performance ranking is not well understood.

This paper examines how variations in evaluation settings impact the performance ranking of VFMs for semantic segmentation and what recommendations can be made for a benchmarking setup that is most efficient while still being representative, *i.e.*, a setup that optimizes training efficiency while accurately reflecting the performance ranking that would emerge under more resource-intensive tuning. By improving efficiency, benchmarking new models on new datasets is made accessible to a wider audience. Moreover, a representative benchmarking setup ensures that performance comparisons are conducted on an equitable basis, facilitating clear, unbiased insights into the efficacy of various pretraining strategies for semantic segmentation.

In this paper, an efficient default benchmarking setup is used to fine-tune various VFMs for semantic segmentation, establishing a baseline performance ranking. Various individual benchmark settings are then changed to observe their impact on the performance ranking, with an overview of the results provided in Figure 1. More details, and the recommendations that can be made for benchmarking, are provided in Section 4.1. Finally, the recommended benchmarking setup enables a performance analysis of VFMs for semantic segmentation, detailed in Section 4.2.

The primary contributions of this paper include:

- An impact analysis of benchmark settings on the performance ranking of VFMs for semantic segmentation.
- A recommended setup for benchmarking VFMs for this task, which balances efficiency with representativeness.
- A performance analysis of VFMs for this task, using the recommended benchmarking setup.
- The code for efficiently fine-tuning VFMs for this task.

2. Related work

(V)FMs. A foundation model (FM) is defined as a model that is pretrained on a broad dataset, such that it can be used for a large variety of downstream tasks [4]. The concept of FMs was popularized in the realm of natural language processing, with models like BERT [10] and GPT [24] demonstrating remarkable capabilities in understanding and generating human language. In computer vision, ImageNet [9] pretraining stands as an earlier example of a VFM, showcasing the power of transfer learning by fine-tuning a broadly pretrained model for diverse applications.

Following the success of ImageNet pretraining, VFMs have evolved to embrace a variety of pretraining strategies. Pretraining strategies for VFMs can be broadly categorized into fully-supervised [19, 29], weakly-supervised [6, 13, 15, 25, 36, 37], and self-supervised [1, 16, 22] learning. Some VFMs are specifically designed for supervised fine-tuning on downstream tasks [14, 16, 22, 29, 32], while others are designed for zero-shot capabilities [6, 13, 15, 25, 36, 37]. Given the inferior performance of zero-shot semantic segmentation methods compared to supervised fine-tuning [21, 34], current VFMs require supervised fine-tuning to perform the task of semantic segmentation effectively. Yet, it remains unclear which pretraining strategies are most effective for this task.

VFMs and semantic segmentation. A representative comparison of the fine-tuned performance of VFMs for this task is hindered by the lack of uniform evaluation settings. An overview of the reported performance in the original papers of a selection of VFMs for IN1K [9] classification and ADE20K [38] semantic segmentation is provided in Table 1. Several data points are missing, as they are not reported. Furthermore, the heterogeneity in evaluation protocols complicates the comparison. It also remains ambiguous whether IN1K classification performance is indicative of ADE20K semantic segmentation performance.

There is one VFM in Table 1 specifically designed for segmentation. SAM [19] solves the task of promptable segmentation and introduces the first large-scale segmentation dataset (SA-1B). However, the masks in this dataset lack semantic information and are not semantically consistent, *i.e.*, they arbitrarily belong to, *e.g.*, whole objects, parts or subparts. Consequently, SAM does not support semantic prompts, and requires supervised fine-tuning to perform semantic segmentation [26, 31]. It is unclear whether SAM would perform better for semantic segmentation than VFMs that were not specifically designed for segmentation.

VFM benchmarking for semantic segmentation. A recent paper [2] investigates the adaptability of various VFMs for few-shot semantic segmentation through linear probing as well as end-to-end fine-tuning. However, the paper is limited to the 1-shot setting, while settings with more downstream data are relevant in many applications. It is unclear how the results for the 1-shot setting would generalize to more downstream data. In another recent paper, AM-RADIO [26] combines multiple VFMs into a single model, and includes a benchmarking process that covers multiple downstream tasks including semantic segmentation. However, the evaluation is limited to linear probing, while end-to-end fine-tuning may yield better performance. It is unclear how representative the results for linear probing are for end-to-end fine-tuning, warranting further investigation.

In summary, it is currently not possible to identify the best VFM for semantic segmentation. This underscores the

	IN1K classification validation accuracy (%)			ADE20K semantic segmentation validation mIoU (%)	
	Zero-shot	Linear probing	End-to-end fine-tuning	Linear probing	End-to-end fine-tuning*
EVA-02 [14]	–	–	87.0 (196), 88.3 (1024)	–	55.3 (1024)
EVA-02-CLIP [28]	74.7 (196)	–	–	–	–
DINOv2 [22]	–	84.5 (256), 86.7 (1024)	88.5 (256), 88.9 (1024)	47.3 (1369)	–
BEiT-3 [32]	–	–	85.4 (196)	–	–
SigLIP [37]	76.2 (196), 79.1 (1024)	–	–	–	–
DFN [13]	76.2 (196)	–	–	–	–
DeiT III (IN21K→IN1K) [29]	–	–	86.7 (576)	–	–
DeiT III (IN1K) [29]	–	–	85.0 (576)	–	–
MAE [16]	–	68.0 (196)	83.6 (196)	–	48.1 (1024)
SAM [19]	–	–	–	–	–

Table 1. **Reported performance of VFMs.** Reported performance in the original papers of a selection of VFMs for IN1K classification and ADE20K semantic segmentation for the ViT-B variants with highest pretraining image size. Parentheses show number of tokens per image for evaluation. – denotes the performance is not reported, * denotes using the UPerNet decoder [35].

critical need for a well-designed, standardized benchmarking setup for the assessment of the performance of VFMs for this task. To address this need, this paper recommends a benchmarking setup that facilitates the comparison between existing VFMs, and guides the development of future VFMs towards enhanced performance for this task.

3. Benchmarking setup

3.1. Models

To study how VFMs should be benchmarked for semantic segmentation, a diverse set of VFMs is selected for their representation of the latest advancements in the field, with varying training data sources, objectives, supervision methods, and pretraining image sizes, as shown in Table 2. The models share the same fundamental ViT [11] architecture and the variants with the highest pretraining image size are selected for benchmarking.

3.2. Settings

Freezing the encoder. By freezing the encoder and using a linear layer as decoder, commonly referred to as linear probing, the encoder is used as a fixed feature extractor, while only the parameters of the linear layer are learned. In contrast, end-to-end fine-tuning allows both the encoder and decoder to be updated. Linear probing would be ideal for adapting a VFM to the task of semantic segmentation, as a VFM should have learned a rich set of features in pretraining that need no fine-tuning. As such, some VFMs solely evaluate semantic segmentation performance through linear probing [1, 22]. To assess whether freezing the encoder impacts the performance ranking, the analysis compares linear probing to end-to-end fine-tuning. If the performance ranking is similar between both methodologies, this suggests it is sufficient to use linear probing for benchmarking, while being representative of the performance ranking with end-to-end fine-tuning.

Changing the decoder. In semantic segmentation, the decoder maps the encoded features to semantic classes for each pixel in the image. The commonly used Mask2Former decoder [5] achieves state-of-the-art performance for multiple image segmentation tasks by performing mask classification instead of pixel-wise classification, using a transformer decoder with masked cross-attention. To assess whether this more advanced decoder impacts the performance ranking, the analysis compares using a simple linear layer to Mask2Former. If the performance ranking is similar between these decoders, this suggests it is sufficient to use a linear decoder for benchmarking, while being representative of the performance ranking with Mask2Former.

Scaling the model. ViTs have been scaled up to billions of parameters [8]. The smallest size available for the models in Table 2 is ViT-B, with approximately 86 million parameters. Additionally, the models have a ViT-L counterpart, with approximately 304 million parameters. To assess whether increasing the model size impacts the performance ranking, the analysis compares the ViT-B variants to the ViT-L counterparts. If the performance ranking is similar between these model sizes, this suggests it is sufficient to use ViT-B for benchmarking, while being representative of the performance ranking with ViT-L.

Varying the patch size. The ViT processes images by dividing them into non-overlapping patches, which are then linearly embedded to generate the input sequence for the model. Smaller patch sizes lead to an increase in the number of tokens for the same image, enhancing the model's capacity to discern fine-grained details, while requiring more compute. The models are pretrained with a certain number of tokens per image (see Table 2), and it is unclear to what extent the models are effective when fine-tuned with a different number of tokens. To assess whether increasing the number of tokens impacts the performance ranking, the analysis compares the commonly used 16×16 patch size to a smaller 8×8 patch size, which quadruples the number of tokens. If the performance ranking is similar between these patch sizes, this suggests it is sufficient to use a 16×16 patch size for benchmarking, while being representative of the performance ranking with an 8×8 patch size.

Name	Data	Objective	Supervision	Tokens
EVA-02 [14]	IN-21K [9] (B) / Merged-38M [14] (L)	MIM	CLIP teacher	196
EVA-02-CLIP [28]	IN-21K [9] (B) / Merged-38M [14] (L)→Merged-2B [28]	MIM→CLIP	CLIP teacher→texts	196→196
DINOv2 [22]	LVD-142M [22]	MIM, discrimination	–	256→1369
BEiT-3 [32]	IN-21K [9], image-text [32], text [32]	MIM, MLM	CLIP teacher, texts	196
SigLIP [37]	WebLI [17]	LIP	Texts	1024 (B) / 576 (L)
DFN [13]	DFN-2B [13]	CLIP	Texts	196
DeiT III (IN21K→IN1K) [29]	IN-21K [9]→IN1K [9]	Classification	Classes	576
DeiT III (IN1K) [29]	IN1K [9]	Classification	Classes	576
MAE [16]	IN1K [9]	MAE	–	196
SAM [19]	IN1K [9]→SA-1B [19]	MAE→promptable segmentation	–→class-agnostic masks	196→4096

Table 2. **Overview of VFMs.** Comparison by pretraining data source, learning objective, supervision type, and number of tokens per image. →: transfer learning.

Changing the downstream dataset. Current VFMs require supervised fine-tuning to perform the task of semantic segmentation effectively. The most commonly used dataset for this task is ADE20K [38], which consists of complex scenes, annotations for 150 semantic classes, and an average image size of around 512×512 pixels. Secondly, another commonly used dataset is PASCAL VOC [12], which consists of object-centric scenes, annotations for 21 semantic classes, and an average image size of around 512×512 pixels. Lastly, another commonly used dataset is Cityscapes [7], which consists of homogeneous urban scenes, fine-grained annotations for 19 semantic classes, and a consistent image size of 2048×1024 pixels. Together, these datasets encompass a selection of varying scene types, classes, and image sizes. To assess how the choice of downstream dataset impacts the performance ranking, the analysis compares training and evaluating on ADE20K to training and evaluating on PASCAL VOC as well as Cityscapes. If the performance ranking is similar between these datasets, this suggests it is sufficient to use one of the datasets for benchmarking, as the ranking for this dataset is representative of the other ones.

Introducing a domain shift. Recent work [33] has shown that large VFMs, when fine-tuned for semantic segmentation, exhibit superior robustness to domain shifts, surpassing specialized domain generalization methods. To assess whether the performance ranking is consistent across domain shifts, the analysis compares training on the Cityscapes training set to training on the synthetic GTA V [27] dataset, by evaluating both on the Cityscapes validation set. By training on synthetic data while evaluating on real data, a significant domain shift is introduced. If the performance ranking is similar with and without a domain shift, this suggests it is sufficient to only perform in-distribution evaluation on Cityscapes for benchmarking, as the ranking is representative of training on data from GTA V and evaluating on out-of-distribution data from Cityscapes.

Default setup. The default benchmarking setup for the impact analysis constitutes end-to-end fine-tuning with a linear decoder of the ViT-B variants with a 16×16 patch size on ADE20K

3.3. Evaluation metrics

Following common practice, the mean intersection over union (mIoU) is used as the evaluation metric for semantic segmentation, which measures the overlap between predicted and ground truth masks. The mIoU is calculated as the average of the IoU scores for each class, with higher values indicating better performance.

To assess how changes in settings affect the performance ranking, the Kendall rank correlation coefficient [18] is used. This coefficient measures the similarity between two ranking sequences, ranging from -1 for a reverse ranking to 1 for an identical ranking. It is given by:

$$\tau = \frac{2}{n(n-1)} \sum_{i<j} \text{sign}(x_i - x_j) \cdot \text{sign}(y_i - y_j) \qquad (1)$$

where, in this paper, n is the number of models, x_i and x_j are the ranks of the i-th and j-th model in one ranking, and y_i and y_j are their ranks in the other ranking. The choice of this coefficient is motivated by its sensitivity to changes in rank order, making it effective at identifying how changes in settings affect the relative performance of models.

3.4. Implementation details

The patch embeddings are fixed to the same patch size for all models, by uniform resizing with the FlexiViT [3] method. Likewise, the positional embeddings are fixed to the same size, by uniform resizing with bicubic interpolation. For ADE20K and PASCAL VOC, the crop size is fixed to 512×512 pixels, while for Cityscapes and GTA V, it is fixed to 1024×1024 pixels. In accordance with Mask2Former [5], training images undergo horizontal flipping, color jittering, resizing between 0.5 and 2.0 times the original size, padding if needed, and random cropping. For inference, images have the shortest side resized to the fixed crop size, with forward passes performed using a sliding window over the proportionally adjusted longer side. Overlaps in the sliding windows are averaged to combine the logits. The combined logits are resized back to the original image size with bilinear interpolation. The argmax function is used to convert the logits to pixel-wise class predictions.

Setting	Training time	Trainable parameters (M)
Default	1.0	86.6
Linear probing	0.6	0.1 (-86.5)
Mask2Former decoder	4.1	101 (+14.4)
ViT-L	1.8	304 (+217.4)
8×8 patch size	1.8	88.5 (+1.9)

Table 3. **Training efficiency of benchmark settings.** Training time and number of trainable parameters for changed settings compared to the default setup (end-to-end fine-tuning, linear decoder, ViT-B, 16×16 patch size).

In accordance with Mask2Former, AdamW [20] is used as the optimizer, with a weight decay of 0.05 and a learning rate of 1e-5 for the encoder and 1e-4 for the decoder, polynomially decayed with a power of 0.9. The batch size is set to 1 to make training more accessible by reducing the memory requirements. Through gradient accumulation the effective batch size is increased to 16 to align with Mask2Former. The number of training steps is set to 40,000 for ADE20K, and 20,000 for PASCAL VOC, Cityscapes and GTA V. Experiments are repeated thrice with different random seeds, and the average performance is reported along with the standard deviation. The code leverages `torch.compile` where possible to improve training efficiency. The default setup requires approximately five hours to train a single model once on one NVIDIA A100 GPU.

Integrating Mask2Former, designed for multi-scale features from hierarchical encoders, with the single-scale outputs of a standard ViT, necessitates two simplifications. Firstly, the pixel decoder, designed to handle multi-scale features from a hierarchical encoder, is substituted with a straightforward linear layer that projects patch tokens from the ViT's final layer to the decoder embedding dimension. Secondly, the multi-scale features and level embeddings, used for efficiency specifically for hierarchical encoders, are replaced by the same single-scale features across the decoder layers. The ablation study in the Mask2Former paper indicates single-scale feature usage for cross-attention does not compromise performance [5].

4. Results

4.1. Impact of settings

Default setup. The results of end-to-end fine-tuning with a linear decoder of the ViT-B variants with a 16×16 patch size on ADE20K are shown in Figure 2. The subsequent paragraphs detail the impact of changing individual settings on the performance ranking and training efficiency, with an overview in Figure 1 and Table 3, respectively.

Freezing the encoder. The results comparing linear probing to the default experiment of end-to-end fine-tuning are shown in Figure 3. All models show large drops in performance and high variability in the size of these drops, while training time is reduced by 0.6 times. Current VFMs may not acquire enough spatial semantic understanding in pretraining to perform semantic segmentation effectively without end-to-end fine-tuning. Furthermore, a low correlation coefficient of 0.38 suggests freezing the encoder significantly changes the performance ranking. Therefore, even though linear probing is efficient, end-to-end fine-tuning is recommended for a representative benchmark.

Changing the decoder. The results comparing Mask2Former to the default linear decoder are shown in

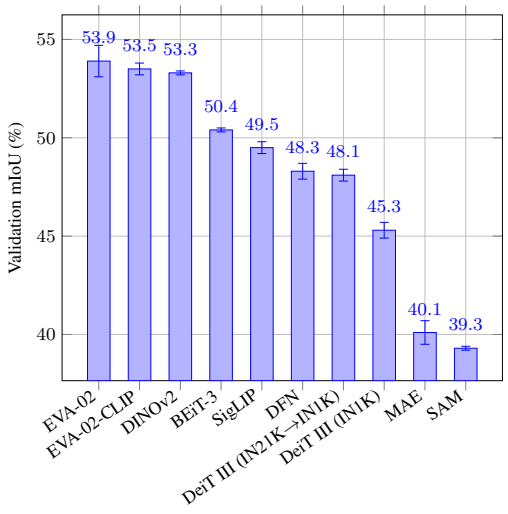

Figure 2. **Default setup results.** End-to-end fine-tuning with a linear decoder of the ViT-B variants with a 16×16 patch size on ADE20K.

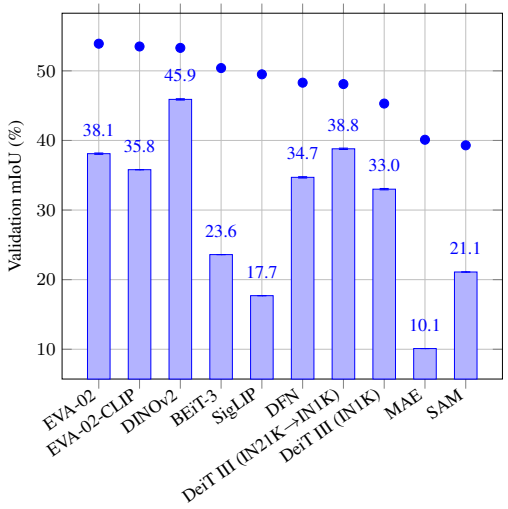

Figure 3. **Linear probing results.** Freezing the encoder results in a correlation coefficient of 0.38 and reduces training time by 0.6 times compared to end-to-end fine-tuning (blue dots).

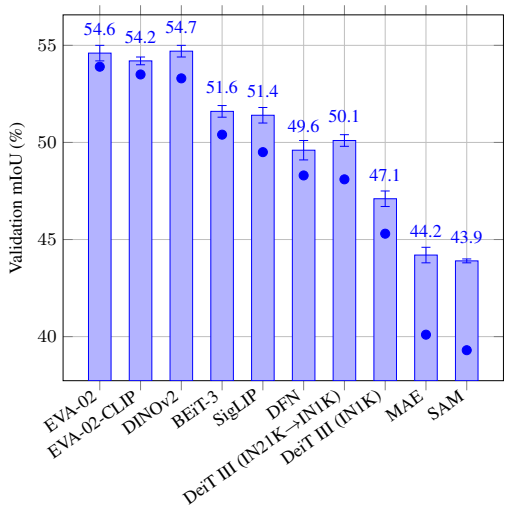

Figure 4. **Mask2Former results.** Using the Mask2Former decoder results in a correlation coefficient of 0.87 and increases training time by 4.1 times compared to a linear decoder (blue dots).

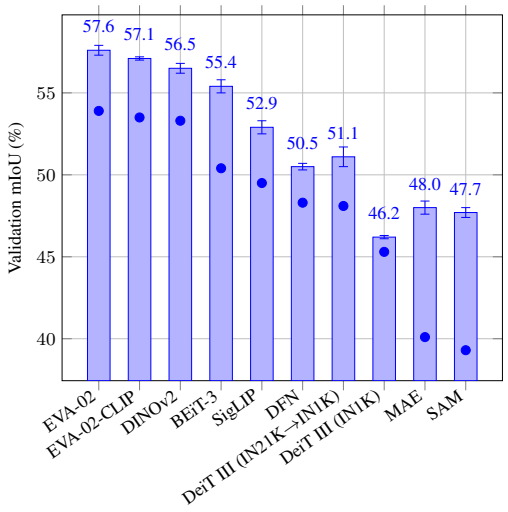

Figure 5. **ViT-L results.** Using the ViT-L counterparts results in a correlation coefficient of 0.87 and increases training time by 1.8 times compared to the ViT-B variants (blue dots).

Figure 4. Mask2Former enhances the performance across all models, at the cost of a training time increase of 4.1 times. A high correlation coefficient of 0.87 suggests Mask2Former does not significantly change the performance ranking. The models that did change in ranking did so by a small margin. Although their ranking remains the same, the lower-performing models MAE and SAM benefit more, likely because these models require more adaptation, where Mask2Former provides more capacity for adaptation compared to a simple linear layer. As the performance ranking remains largely the same with Mask2Former, while requiring significantly longer training times, the linear decoder is recommended for an efficient benchmark.

Scaling the model. The results comparing the ViT-L counterparts to the default ViT-B variants of the models are shown in Figure 5. Increasing model capacity enhances the performance across all models, while training time is increased by 1.8 times. A high correlation coefficient of 0.87 suggests increasing model size does not significantly change the performance ranking. The lower-performing models MAE and SAM benefit more from a larger model. However, an outlier is DeiT III (IN1K), which exhibits a minimal performance improvement. On the other hand, DeiT III (IN21K→IN1K) benefits significantly more, implying supervised pretraining on the smaller IN1K dataset yields limited benefits from scaling the model beyond ViT-B for downstream semantic segmentation performance. While MAE is pretrained on IN1K, its improvement may be attributed to the better scaling properties of the self-supervised MAE objective compared to supervised learning [16]. As the performance ranking remains largely the same with the ViT-L counterparts, the ViT-B variants of the models are recommended for an initial benchmark. However, it is important to consider that scaling the model may reach a plateau in case of limited pretraining data, depending on the learning objective.

Varying the patch size. The results comparing an 8×8 patch size to the default 16×16 patch size are shown in Figure 6. Most models benefit a small amount from a smaller patch size, while training time is increased by 1.8 times. A correlation coefficient of 0.78 indicates a small change in the performance ranking. However, the models that changed in ranking only did so by a small margin. The number of pretraining tokens per image (see Table 2) might affect how models respond to patch size changes, although results are inconsistent. Specifically, EVA-02-CLIP shows a greater improvement with a smaller patch size compared to EVA-02, despite identical number of pretraining tokens per image. Additionally, an outlier is MAE, being the only model that shows a decrease in performance with a smaller patch size. Despite small ranking changes, a smaller patch size does not consistently advantage all models, with only marginal improvements observed in those that do benefit. Hence, the most commonly used 16×16 patch size is recommended for an efficient benchmark.

Changing the downstream dataset. The results for PASCAL VOC are shown in Figure 7. A correlation coefficient of 0.78 compared to ADE20K indicates a small change in the performance ranking. The DeiT III models, however, stand out by exhibiting better relative performance on PASCAL VOC. This enhanced performance is likely due to their supervised pretraining on the object-centric ImageNet dataset, which aligns well with the object-centric scenes in PASCAL VOC.

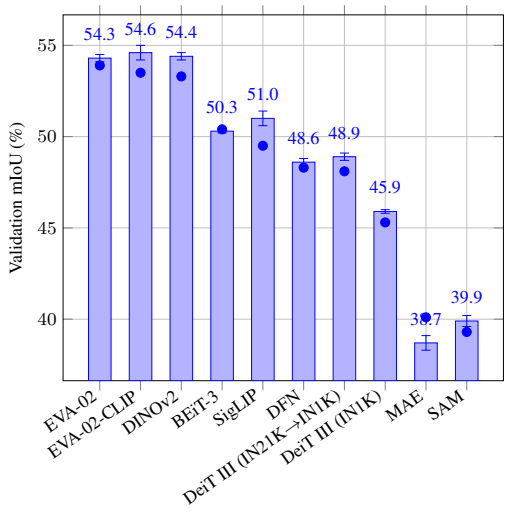

Figure 6. **8 × 8 patch size results.** Using an 8 × 8 patch size results in a correlation coefficient of 0.78 and increases training time by 1.8 times compared to a 16 × 16 patch size (blue dots).

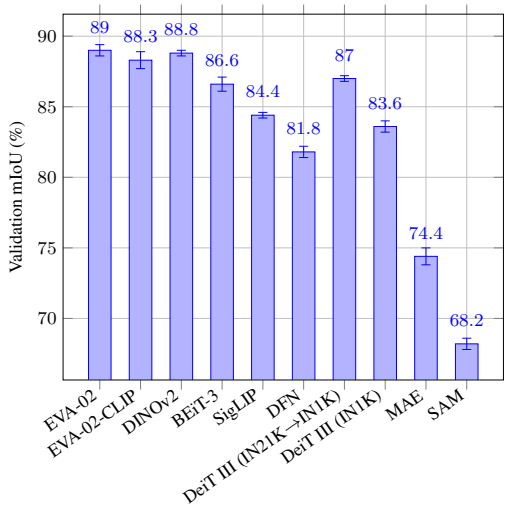

Figure 7. **PASCAL VOC results.** Using PASCAL VOC results in a correlation coefficient of 0.78 compared to ADE20K.

The results for Cityscapes are shown in Figure 8. A correlation coefficient of 0.56 compared to ADE20K indicates the performance ranking is even more dissimilar. Notably, DINOv2 and SAM exhibit higher relative performance on Cityscapes compared to the other datasets. These models are likely better at capturing the fine-grained details in the Cityscapes dataset due to their high resolution pretraining in combination with a patch-level objective, where none of the other models have both of these properties.

Given the impact of scene similarity and granularity between pretraining and downstream data on the performance ranking, benchmarking across multiple datasets is recommended to gain a comprehensive understanding of model performance in various scenarios.

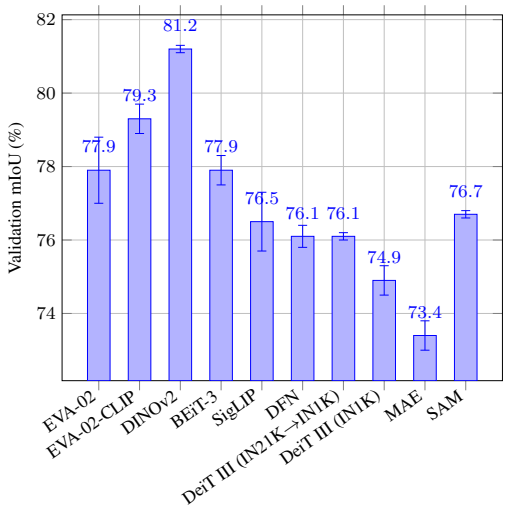

Figure 8. **Cityscapes results.** Using Cityscapes results in a correlation coefficient of 0.56 compared to ADE20K.

Introducing a domain shift. The results for training on GTA V and evaluating on Cityscapes are shown in Figure 9. All models experience a significant decrease in performance by introducing the synthetic-to-real domain shift. A correlation coefficient of 0.73 compared to the oracle experiment of using Cityscapes for both training and evaluation indicates a small change in the performance ranking when a domain shift is introduced. Notably, SAM and MAE show more substantial declines, with SAM experiencing the most drastic drop in ranking, falling from 5th to 9th place. This decline may be attributed to the lack of semantic understanding required for the pretraining tasks of these models [31]. Given the low semantic complexity of the homogeneous scenes in the GTA V and Cityscapes datasets [23], these models initially perform well in-distribution. However, when faced with a domain shift, the lack of semantic understanding becomes apparent. Further investigation shows similar findings under real-to-real domain shifts. Therefore, if generalization under domain shifts is important, it is recommended to include domain shifts in the benchmarking setup, as in-distribution performance is not necessarily representative of out-of-distribution performance.

4.2. Analysis of model performance

The benchmarking setup recommended in this paper enables a performance analysis of VFMs for semantic segmentation.

Surprisingly, SAM, despite being the only model pretrained with mask labels, performs poorly across most experiments. SAM is pretrained with a promptable segmenta-

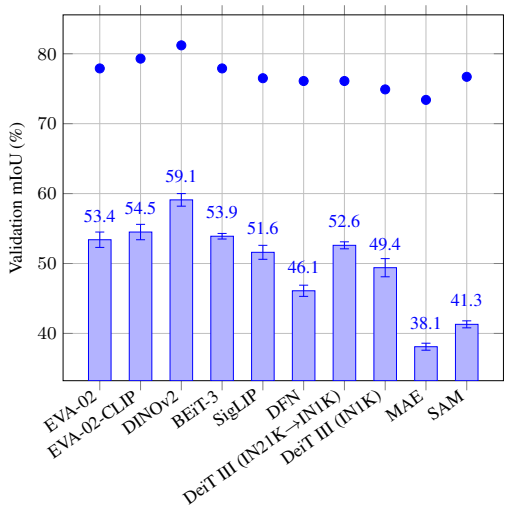

Figure 9. **GTA V to Cityscapes results.** Using GTA V for training and Cityscapes for evaluation results in a correlation coefficient of 0.73 compared to using Cityscapes for both training and evaluation (blue dots) and 0.64 compared to ADE20K.

tion objective and initialized with the parameters of MAE. Intriguingly, MAE outperforms SAM in some settings. Further investigation leads to the observation that, while mask labels arbitrarily belonging to, *e.g.*, whole objects, parts or subparts, are beneficial for promptable segmentation, their semantic inconsistency may hinder the learning of effective features for semantic segmentation. These findings indicate that fine-tuning a model pretrained for promptable segmentation has limited benefits for semantic segmentation, and can even lead to negative transfer to this task.

The highest performing models across all experiments performed in this paper–EVA-02, EVA-02-CLIP, DINOv2 and BEiT-3–have the pretraining objective of masked image modeling (MIM) with abstract representations in common. While EVA-02-CLIP directly and EVA-02 and BEiT-3 indirectly rely on weak supervision from text labels, DINOv2 is pretrained without any labels. EVA-02-CLIP is pretrained with a language-image learning objective and initialized with the parameters of EVA-02. Although EVA-02-CLIP did not benefit from additional language-image learning in the default setup, it outperforms all other language-image models across all experiments performed in this paper, while it performs significantly worse than SigLIP and DFN in zero-shot classification on IN1K (see Table 1). Thus, this shows that zero-shot evaluation of CLIP models is not indicative of their semantic segmentation performance downstream, highlighting the necessity of the established benchmark. Moreover, these findings suggest that MIM with abstract representations is a crucial pretraining objective for semantic segmentation, even more important than the type of supervision used.

5. Conclusion

This paper presents a study on how to benchmark VFMs for semantic segmentation. The paper includes an analysis on how varying benchmark settings impact the performance ranking and training efficiency of VFMs for this task. The results lead to a recommended benchmarking setup of fine-tuning the ViT-B variants of VFMs with a 16×16 patch size and a linear decoder. Leveraging multiple datasets for training and evaluation is also recommended, as the performance ranking across datasets and domain shifts varies. Linear probing is not recommended, as it is not representative of end-to-end fine-tuning. Finally, the recommended benchmarking setup enables a performance analysis of VFMs for semantic segmentation, challenging the value of promptable segmentation pretraining and highlighting the crucial role of MIM with abstract representations.

6. Discussion

The analysis on how varying benchmark settings impact the performance ranking of VFMs for semantic segmentation leads to several novel observations on the performance of specific models in various settings. This facilitates future work to further investigate the causes of these observations. In addition to its contributions, this paper has some specific limitations. Firstly, the hyperparameters used in the experiments mostly align with Mask2Former and the training steps were determined such that the best performing models converge, but hyperparameter tuning was not performed. Secondly, although individual modifications to benchmark settings were investigated, interactions between settings were not considered, as this would have resulted in a combinatorial explosion of experiments. Thirdly, the analysis was limited to specific settings, and the results may not generalize to other settings, such as even larger models, more limited downstream data, other fine-tuning strategies, or other types of downstream supervision. Likewise, the impact of different settings observed in this paper may change for future models, as the field of VFMs is rapidly evolving. By demonstrating how to evaluate the representativeness of a benchmarking setup, this paper facilitates future work to re-evaluate the benchmarking setup for the next generation of VFMs.

Acknowledgements. This paper was supported by Key Digital Technologies Joint Undertaking (KDT JU) in EdgeAI "Edge AI Technologies for Optimised Performance Embedded Processing" project, grant agreement No 101097300. This paper made use of the Dutch national e-infrastructure with the support of the SURF Cooperative using grant no. EINF-5838, which is financed by the Dutch Research Council (NWO).

References

[1] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *CVPR*, 2023. 2, 3

[2] Reda Bensaid, Vincent Gripon, François Leduc-Primeau, Lukas Mauch, Ghouthi Boukli Hacene, and Fabien Cardinaux. A novel benchmark for few-shot semantic segmentation in the era of foundation models. *arXiv preprint arXiv:2401.11311*, 2024. 1, 2

[3] 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. 4

[4] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 1, 2

[5] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 3, 4, 5

[6] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *CVPR*, 2023. 2

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 1, 4

[8] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *ICML*, 2023. 3

[9] 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. 2, 4

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 2

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 3

[12] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 1, 4

[13] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. Data filtering networks. In *NeurIPS*, 2023. 1, 2, 3, 4

[14] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *arXiv preprint arXiv:2303.11331*, 2023. 1, 2, 3, 4

[15] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. In *NeurIPS*, 2024. 2

[16] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 1, 2, 3, 4, 6

[17] Ahmet Iscen, Alireza Fathi, and Cordelia Schmid. Improving image recognition by retrieving from web-scale image-text data. In *CVPR*, 2023. 4

[18] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 1938. 4

[19] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023. 1, 2, 3, 4

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 5

[21] Muhammad Ferjad Naeem, Yongqin Xian, Xiaohua Zhai, Lukas Hoyer, Luc Van Gool, and Federico Tombari. Silc: Improving vision language pretraining with self-distillation. *arXiv preprint arXiv:2310.13355*, 2023. 2

[22] 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. In *TMLR*, 2023. 1, 2, 3, 4

[23] Fabrizio J Piva, Daan de Geus, and Gijs Dubbelman. Empirical generalization study: Unsupervised domain adaptation vs. domain generalization methods for semantic segmentation in the wild. In *WACV*, 2023. 7

[24] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *OpenAI blog*, 2018. 2

[25] 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. 2

[26] Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Am-radio: Agglomerative model–reduce all domains into one. *arXiv preprint arXiv:2312.06709*, 2023. 2

[27] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016. 4

[28] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023. 1, 3, 4

[29] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In *ECCV*, 2022. 1, 2, 3, 4

[30] Raviteja Vemulapalli, Hadi Pouransari, Fartash Faghri, Sachin Mehta, Mehrdad Farajtabar, Mohammad Rastegari, and Oncel Tuzel. Label-efficient training of small task-specific models by leveraging vision foundation models. *arXiv preprint arXiv:2311.18237*, 2023. 1

[31] Haoxiang Wang, Pavan Kumar Anasosalu Vasu, Fartash Faghri, Raviteja Vemulapalli, Mehrdad Farajtabar, Sachin Mehta, Mohammad Rastegari, Oncel Tuzel, and Hadi Pouransari. Sam-clip: Merging vision foundation models towards semantic and spatial understanding. *arXiv preprint arXiv:2310.15308*, 2023. 2, 7

[32] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. In *CVPR*, 2023. 1, 2, 3, 4

[33] Zhixiang Wei, Lin Chen, Yi Jin, Xiaoxiao Ma, Tianle Liu, Pengyang Lin, Ben Wang, Huaian Chen, and Jinjin Zheng. Stronger, fewer, & superior: Harnessing vision foundation models for domain generalized semantic segmentation. In *CVPR*, 2024. 4

[34] Monika Wysoczańska, Oriane Siméoni, Michaël Ramamonjisoa, Andrei Bursuc, Tomasz Trzciński, and Patrick Pérez. Clip-dinoiser: Teaching clip a few dino tricks. *arXiv preprint arXiv:2312.12359*, 2023. 2

[35] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 3

[36] Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying clip data. In *ICLR*, 2024. 2

[37] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023. 1, 2, 3, 4

[38] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 2, 4