Exploring the Benefits of Vision Foundation Models for Unsupervised Domain Adaptation

Brunó B. Englert* Fabrizio J. Piva* Tommie Kerssies Daan de Geus Gijs Dubbelman
Eindhoven University of Technology
{b.b.englert, f.j.piva, t.kerssies, d.c.d.geus, g.dubbelman}@tue.nl

Abstract

Achieving robust generalization across diverse data domains remains a significant challenge in computer vision. This challenge is important in safety-critical applications, where deep-neural-network-based systems must perform reliably under various environmental conditions not seen during training. Our study investigates whether the generalization capabilities of Vision Foundation Models (VFMs) and Unsupervised Domain Adaptation (UDA) methods for the semantic segmentation task are complementary. Results show that combining VFMs with UDA has two main benefits: (a) it allows for better UDA performance while maintaining the out-of-distribution performance of VFMs, and (b) it makes certain time-consuming UDA components redundant, thus enabling significant inference speedups. Specifically, with equivalent model sizes, the resulting VFM-UDA method achieves an 8.4× speed increase over the prior non-VFM state of the art, while also improving performance by +1.2 mIoU in the UDA setting and by +6.1 mIoU in terms of out-of-distribution generalization. Moreover, when we use a VFM with 3.6× more parameters, the VFM-UDA approach maintains a 3.3× speed up, while improving the UDA performance by +3.1 mIoU and the out-of-distribution performance by +10.3 mIoU. These results underscore the significant benefits of combining VFMs with UDA, setting new standards and baselines for Unsupervised Domain Adaptation in semantic segmentation. The implementation is available at https://github.com/tue-mps/vfm-uda.

1. Introduction

In machine learning, generalization refers to a model's ability to perform well on data inside, near, and outside the distribution of the data on which it was trained. The challenge of achieving good generalization increases with the distance to the training data distribution. To maximize the chance of good generalization in real-world environments, models are best trained on a diverse dataset having a broad distribution, thus minimizing the likelihood of encountering out-of-distribution data. However, for dense tasks like semantic segmentation, obtaining abundant labeled data can be costly and labor intensive [5], as every pixel has to be labeled. As a result, annotated data is likely to be scarce, and networks trained on limited labeled data suffer from poor generalization due to the lack of exposure to sufficiently diverse training examples. To address the lack of generalization, Vision Foundation Models (VFMs) [7, 9, 14, 23] and Unsupervised Domain Adaptation (UDA) [11, 12, 35, 37] have emerged, amongst other alternatives [18, 24, 41, 42]. UDA methods leverage unlabeled data to adapt a model to a specific target domain, which is typically the domain where the model is to be deployed. By doing so, they do not necessarily aim for wide generalization beyond this target do-

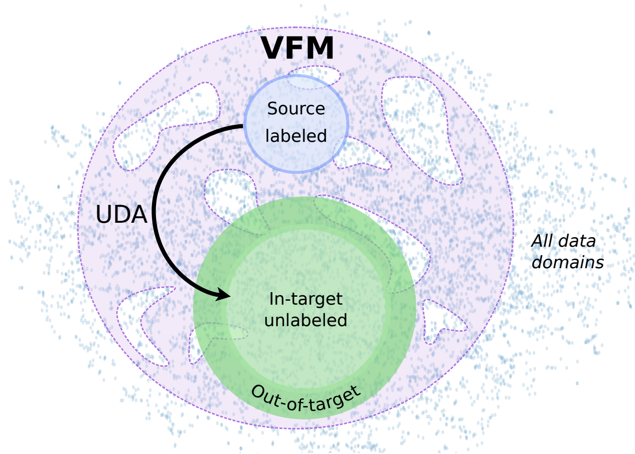

Figure 1. **Generalization capabilities of UDA methods and VFMs.** UDA is designed to adapt a model from a labeled source domain to an unlabeled target domain, whereas VFMs capture a broad spectrum of data distributions, contributing to the overall generalization. The goal of this research is to investigate and leverage the combined in- and out-of-target generalization capabilities of UDA and VFMs.

*Both authors contributed equally.

main. In contrast, VFMs leverage extensive pre-training on large datasets to create models that can be used for efficient fine-tuning on various downstream tasks. Once fine-tuned on (limited) labeled data, these VFMs can generalize better [23, 36] than models that were not as extensively pre-trained. In other words, UDA methods focus on performing well on a specific target domain, whereas VFMs can improve the generalization on domains that are unseen during fine-tuning. Both types of generalization are important, and they are illustrated in Fig. 1. In this work, we study whether the generalization capabilities of UDA and VFMs are complementary.

Recently, Vision Foundation Models (VFMs) have made significant contributions by offering pre-trained models that excel in generalization, requiring minimal fine-tuning on downstream tasks [7, 9, 23, 27]. Contrary to the traditional approach of pre-training models on labeled datasets like ImageNet [30] or MSCOCO [17], Vision Foundation Models (VFMs) stand out by utilizing extensive pre-training on labeled and/or unlabeled datasets. Training on unlabeled data is done using different self-supervised techniques such as masked image modeling [39] or self-training [3]. Specifically in the context of semantic segmentation, VFMs have shown promising results in improving the performance to domains never seen during fine-tuning [36].

Alternatively, Unsupervised Domain Adaptation (UDA) methods continue to make progress in enabling models to adapt to any unlabeled target domain [11, 12, 35, 37]. To adapt a model to a target domain, UDA methods use a labeled source domain, consisting of either synthetic images [28, 29] or real images. The goal is to bridge the gap between the source and target domains, transferring what the model has learned from the source to perform as well as possible in the target domain. UDA methods are typically only evaluated on this target domain, but this does not reflect their generalization performance. Therefore, Piva *et al*. [25] proposed to evaluate these models on an additional, unseen dataset, and they show that UDA methods can also improve the performance in this out-of-target setting.

Despite significant individual advancements by Vision Foundation Models (VFMs) and Unsupervised Domain Adaptation (UDA) methods, both have been studied in isolation, and it remains an open question to what extent they are complementary to each other. To address this gap in research, this work investigates the integration of VFM in UDA to obtain increased in- and out-of-target performance. For this purpose, we incorporate VFMs into a representative state-of-the-art UDA method, MIC [12]. We conduct ablations over components, image resolution, and self-training strategies, and assess the impact of VFM size and pre-training strategy. Based on these results, we adopt the best combination of VFM and UDA components which we refer to as the VFM-UDA method. The experimental results across synthetic-to-real and real-to-real scenarios demonstrate that VFMs can have a very positive influence on a UDA method's ability to perform well on both in- and out-of-target domains. This highlights the potential of their future use in combination with UDA methods.

In summary, the contributions of this work are:

- A careful investigation of the complementarity of VFMs and UDA, resulting in new UDA standards and baselines for the VFM era.
- A detailed ablation over the necessity of UDA components when used in combination with VFMs, and an investigation of the influence of VFM model size and pre-training strategy.
- A broad experimental validation of VFM-UDA and comparison to non-VFM UDA methods on synthetic-to-real and real-to-real dataset combinations.

2. Related Work

Vision Foundation Models (VFMs) have brought notable advancements in generalization within computer vision, being trained on large-scale data and adaptable for multiple downstream tasks. For instance, CLIP [27] learns high-quality visual representations through contrastive learning [4] with large-scale image-text pairs. MAE [9] utilizes a masked image modeling framework for image pixel reconstruction. SAM [14], trained on a large-scale segmentation dataset, extracts features from images and prompts to predict single or multiple segmentation masks. EVA02 [7] applies masked image modeling to a CLIP model's visual features, offering a unique approach to visual representation learning. DINOv2 [23], on the other hand, is pre-trained on carefully curated datasets without explicit supervision, showcasing its self-supervised learning strength. Most VFMs currently rely on the plain Vision Transformer (ViT) [6] architecture, which outputs single-scale features, posing a design challenge when integrating them with UDA, as is explained in the next section.

Unsupervised Domain Adaptation (UDA) methods aim to increase the performance of a model on a known target domain. This domain usually represents the environment where the model is likely to be deployed in the real world. These models can leverage unlabeled target data and labeled data from a source domain to increase a model's performance on a target domain. UDA methods leverage techniques like feature alignment [10, 21, 40], self-supervised learning [10–12, 25, 35] and data augmentation [16, 25] to minimize the discrepancy between source and target domain distributions. Current UDA methods typically use hierarchical encoders, consisting of either Convolutional [8, 31] or Transformer [19, 38] blocks that yield multi-scale features to obtain optimal performance on small-scale objects, whereas VFMs produce single-scale features. As such, state-of-the-art UDA methods are not directly compatible with VFMs in an optimal manner. This work focuses on bridging this incompatibility and applying these UDA techniques to VFMs, assessing them outside the standard practice of initializing on ImageNet [30], and evaluating their effectiveness in adaptation settings that consider both in-target as well as out-of-target performance.

To the best of our knowledge, leveraging the combined generalization capabilities of VFMs and UDA has not been explored, and filling this gap is what we aim for in our work.

3. Methodology

This section outlines the adaptations made to align UDA strategies with Vision Transformer (ViT) architectures to enable the integration of UDA with VFMs, and introduces the motivation and design of our experiments.

3.1. VFM-UDA

UDA baseline. As a baseline, we start from MIC [12], a state-of-the-art Unsupervised Domain Adaptation (UDA) method. MIC utilizes a student-teacher framework [1, 11, 12, 15, 35, 37] with some additional UDA components. In the student-teacher framework, the teacher network generates pseudo labels for the target domain which are then used to supervise the student network. The student network uses a vanilla cross-entropy loss on the labeled source domain and on the unlabeled target domain using the pseudo labels generated by the teacher network. The teacher network is updated with an exponential moving average (EMA) using the student model's parameters. Below, we specify the additional components of this UDA method. In Sec. 4.2, we assess the effectiveness of each of these components in combination with a VFM. For the final VFM-UDA model, we keep only the components that remain effective.

Feature Distance (FD) [10] is a UDA strategy that adds a Mean Squared Error (MSE) loss between the student's encoder output and those of a frozen pre-trained encoder. Minimizing this MSE loss encourages the student model to retain the features learned during pre-training, balancing the adaptation to new domain-specific features with the preservation of essential general features.

Masked Image Consistency (MIC) [12] is a UDA strategy that introduces an asymmetry between the teacher and student models by masking out parts of the original images for the student in the target dataset. This is achieved by randomly generating a patch mask and masking out different parts of each image. This masking forces the student model to infer from contextual information from the unmasked regions.

HRDA [11] is a model architectural change that is aimed at making high-resolution segmentation predictions in both UDA and conventional supervised learning [13]. This is achieved by conducting semantic segmentation on both high-resolution (HR) and low-resolution (LR) crops of an image. The resulting segmentation predictions for the LR and HR crops are fused by a learned scale-attention head. This fusion approach leverages the detailed information from HR crops and the broader context from LR crops, with the added drawback of having to do multiple forward passes for one image.

VFM encoder. We choose the DINOv2 [23] VFM as the primary encoder on top of which we conduct UDA, but we also evaluate alternative VFMs in our experiments in Sec. 4.3. The UDA baseline method uses the MiT-B5 encoder, which produces multi-scale features. However, because all performant VFMs use single-scale VIT-based encoders, we need to make an adaption to the MIC baseline. This adaption is performed in the decoder, as described in the next section. To ensure a fair comparison of the VFM-UDA method to the baseline, we use encoders with a roughly equal number of learnable parameters. Specifically, we use the ViT-B/14 encoder with 86M parameters, while MiT-B5 has 81M parameters.

VFM decoder. MIC and other well-performing UDA methods use decoders that are designed for encoders that output multi-scale features. However, ViTs only output single-resolution features. This difference motivates us to use a different, yet much simpler decoder architecture that is specifically designed for ViTs. Our decoder, depicted in Fig. 2, is inspired by the Segment Anything Model's (SAM) [14] upsampling stage but is slightly modified for our use case. Compared to the SAM model's upsampling stage, we introduce an additional 3×3 Conv2D before the final output. While a more complex and larger decoder head could be used, DINOv2 already performs well with a simple linear decoder and a frozen encoder [23]. This suggests that a large decoder is not necessary for VFMs due to their extensive pre-training. The more efficient decoder for the VFM-UDA approach contains 1.8M parameters, in contrast to the MIC model's decoder, which has 5.2M parameters.

VFM masking. The baseline UDA method, MIC, uses direct image masking for the image masking consistency loss. In our approach, instead of applying a mask directly to the image, we mask the patch tokens and replace them with a learnable token, similarly to how BEiT is trained [2]. This adjustment acknowledges the architectural differences in ViT models and optimizes the process for token-based architectures.

3.2. Experimental set-up

We conduct several experiments to thoroughly assess the combined VFM-UDA method and support our design choices.

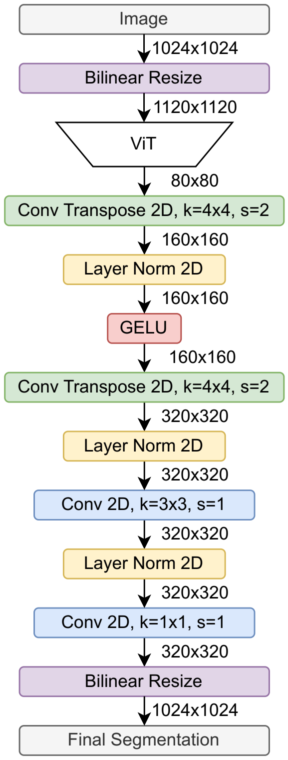

Figure 2. **Decoder head architecture.**

Domain adaptation setup. To assess the UDA capabilities of models, we evaluate the performance in synthetic-to-real and real-to-real adaptation scenarios. The synthetic-to-real scenario allows us to assess how well the model can bridge the gap between computer-generated images and real-world images, representing an extreme case of domain shift. On the other hand, real-to-real experiments test the model's ability to adapt between different real-world conditions, reflecting more subtle variations and complexities found in natural settings. This dual approach ensures a thorough evaluation of the VFM-UDA integration, highlighting its adaptability and performance across different visual domains.

In- and out-of-target evaluation. To truly assess a model's generalization capabilities, it should also be evaluated beyond the domain of its target dataset. Therefore, similarly to Piva *et al*. [26], we additionally evaluate each model on another dataset that falls outside of the distribution of the target domain. In other words, to measure a UDA method's performance both in-target and out-of-target, we use two completely separate evaluation datasets. This additional out-of-target dataset is never used during UDA training, to ensure there is no data leakage. Training and validation splits are made for both of these evaluation datasets. The training split for the out-of-target dataset is only used to determine the oracle performance.

Baseline, UDA, and oracle. Our benchmark compares the UDA performance with respect to a *baseline* and an *oracle*. The baseline is trained on only the source domain, in a supervised manner, representing the model's performance before domain adaptation. The oracle is trained on only the labeled target data, also in a supervised manner, and reflects the empirical upper bound of a model's performance on the target domain. Both the baseline and the oracle rely only on supervised learning, meaning they do not use unlabeled data. In contrast, in the experiments with the UDA methods, we train on the source domain with labeled data and try to adapt to the target domain with unlabeled data, using different pre-training configurations and model sizes.

Datasets. To assess the in-target performance, following previous state-of-the-art methods, we use GTA5 [28] $\rightarrow$ Cityscapes [5] as the synthetic-to-real scenario. For the real-to-real adaptation scenarios, we use Cityscapes $\rightarrow$ Mapillary [22]. To assess the out-of-target performance, we use WildDash2 [43], a completely separate dataset from the source and target datasets. We choose WildDash2 because it includes city, highway, and rural scenes under various weather conditions, and because the images are captured in more than 100 countries, providing diverse and challenging imagery.

Implementation details. The encoder is a vanilla ViT-B/14 [6] with DINOv2 [23] pre-training. The learning rate for the decoder is 1.4×10^{-4} and for the encoder it is 1.4×10^{-5}. We train for 40,000 iterations with a batch size of 8, and use the AdamW optimizer [20]. We use a linear learning rate warmup of 1,500 iterations and linear decay afterwards. During training, the source dataset is sampled using rare-class sampling [10] to address class imbalances.

When training UDA methods, we use a student-teacher setup, where the student's weights are aggregated during training into an EMA teacher model. The running weight for the EMA model is $\alpha = 0.999$. This EMA teacher model is never backpropagated and is only used for pseudo-label generation. When creating the pseudo labels, the target images are not augmented, but we use horizontal flip aggregation to create the final pseudo label to reduce labeling noise. The threshold on the softmax outputs to generate the final pseudo labels is $\rho = 0.968$. We use a mask ratio of $r = 0.7$, like in the original MIC [12]. However, diverging from MIC's strategy of masking image regions directly, our adaptation involves masking patch tokens when using a ViT encoder. The target images are mixed with the source images using DACS [34] data augmentation. The final VFM-UDA method does not use the FD loss or HRDA, see Sec. 4.2.

Our experimental setup aims to investigate the following performance aspects:

- **VFM-UDA vs. existing UDA methods:** We assess the performance of VFM-UDA against current UDA methods in synthetic, real, in-target, and out-of-target settings, focusing on segmentation quality. This wide range of test scenarios gives insights into each UDA method's generalization capabilities, crucial for real-world deployment.
- **Ablation on UDA components:** We conduct an in-depth evaluation of the individual impact and contributions of various UDA components within the VFM-UDA framework. This includes examining the effects of resolution adjustments, masking strategies, FD, and HRDA.
- **Efficiency analysis:** We compare the inference speed of VFM-UDA to that of existing UDA methods.
- **Exploring various VFMs:** We assess the effect of using different VFMs to find the relative advantages of various VFM pre-training strategies on the in- and out-of-target performance.
- **Impact of model size and pre-training:** We study how scaling up models with ImageNet and VFM pre-training affects the in- and out-of-target performance. This setup provides insights into how we can further scale UDA with larger model sizes.

4. Results

In this section, we present the results and discuss the five experiments that we introduced in Sec. 3.

4.1. Generalization of UDA with VFMs

Overall findings. The results of both the synthetic-to-real and real-to-real adaptation scenarios can be seen in Tab. 1 and Tab. 2, respectively. In this experiment, we only consider the VFM-UDA method with ViT-B/14, since it has a similar parameter count as MIC. On both UDA benchmarks, VFM-UDA demonstrates superior in-target and out-of-target performance compared to the current state-of-the-art UDA method, MIC. In the synthetic-to-real scenario, VFM-UDA adapts better than MIC by +1.2 mIoU points and generalizes better by +6.1 points. In the real-to-real one, the integration surpasses MIC even more, with differences of +5.8 mIoU in-target and +7.8 out-of-target.

These results show that the generalization capabilities of VFMs and UDA methods are complementary, as the VFM-UDA combination achieves better UDA performance than the state-of-the-art UDA methods, while maintaining – or even slightly improving – the out-of-target performance of the VFM.

Effect of model size. When integrating UDA with a significantly larger model, ViT-L/14, the adaptation and generalization capabilities of the model increase even more, outperforming the state-of-the-art UDA method by larger margins. In the real-to-real scenario, it is interesting to note that the combination VFM-UDA yields only a minor performance increase compared to its VFM baseline, both in-target and out-of-target. Essentially, when the VFM is large, the added benefits from UDA to the model's overall performance become marginal. This suggests that the performance improvements obtained by scaling VFMs might limit the additional generalization benefits achievable by pairing with UDA techniques, especially in simpler settings like real-to-real scenarios. For a more in-depth analysis of the scalability of VFM-UDA with different pre-trainings, we refer to Sec. 4.4.

Next, we will also demonstrate that these larger models can be faster than the smaller state-of-the-art UDA method.

4.2. Evaluation of UDA components

To investigate how each UDA component affects the overall adaptation performance when integrated with VFMs, we use the synthetic-to-real adaptation scenario. The baseline UDA method only applies supervised learning to the source domain and self-training to the target domain. Using this baseline, we try to incrementally improve it by introducing the following configurations:

- **Incorporation of masking:** adding Mask Image Consistency (MIC) when performing self-training, either at image or token level.
- **Feature Distance (FD):** adding the FD loss to prevent the model from forgetting the pre-training.
- **Multi-resolution training:** applying multi-resolution training by fusing high-resolution and low-resolution predictions, as proposed in HRDA [11].

Findings. Our analysis, detailed in Tab. 3, reveals nuanced performance impacts for each UDA component. Token Masking, as opposed to the original Image Masking used in MIC, yields a slightly better performance. When we use either the FD loss or HRDA on top of Token Masking, there is a noticeable decline in mIoU. This suggests that these components may not translate as effectively to ViT-based encoders, which lack hierarchical features that are present in the MiT-B5-encoder-based UDA methods. Although the full combination of Token Masking, the FD loss, and HRDA shows some improvement over using them separately, the combined effect still does not exceed the performance of the Token Masking alone. These findings imply that the FD and the HRDA components may be redundant when using ViT-based VFMs. Therefore, we do not incorporate them in the final VFM-UDA method.

Inference speed findings. Tab. 4 shows the inference speed of the methods compared in Tab. 1. The VFM-UDA approach shows a large improvement in inference

Method	Backbone	Pre-training	# Parameters	In-target mIoU			Out-of-target mIoU		
				Baseline	UDA	Oracle	Baseline	UDA	Oracle
DaFormer [10]	MiT-B5	ImageNet-1K [30]	85.2M	47.1	68.3	76.6	41.9	50.1	66.8
SePiCo [37]	MiT-B5	ImageNet-1K [30]	85.2M	46.5	70.3	78.3	37.7	47.5	64.8
HRDA [11]	MiT-B5	ImageNet-1K [30]	85.7M	47.3	73.8	80.8	40.6	50.8	66.9
MIC [12]	MiT-B5	ImageNet-1K [30]	85.7M	47.3	75.9	80.8	40.6	55.2	66.9
VFM-UDA	ViT-B/14	DINOv2 [23]	88.4M	62.9	77.1	82.4	60.4	61.3	74.8
VFM-UDA	ViT-L/14	DINOv2 [23]	307.6M	68.7	**79.0**	83.4	64.8	**65.5**	76.3

Table 1. **Semantic segmentation performance for synthetic-to-real scenario.** We use the GTA5 → Cityscapes setup, a common benchmark for UDA methods. This scenario evaluates the performance of the UDA methods when there is a significant domain gap between the source and the target domain. The out-of-target dataset is WildDash2.

Method	Backbone	Pre-training	# Parameters	In-target mIoU			Out-of-target mIoU		
				Baseline	UDA	Oracle	Baseline	UDA	Oracle
DaFormer [10]	MiT-B5	ImageNet-1K [30]	85.2M	60.1	62.1	70.7	51.7	52.2	66.8
SePiCo [37]	MiT-B5	ImageNet-1K [30]	85.2M	60.8	62.5	70.9	49.6	53.8	64.8
HRDA [11]	MiT-B5	ImageNet-1K [30]	85.7M	64.4	69.9	77.4	46.3	60.7	66.9
MIC [12]	MiT-B5	ImageNet-1K [30]	85.7M	64.4	73.3	77.4	46.3	61.9	66.9
VFM-UDA	ViT-B/14	DINOv2 [23]	88.4M	75.7	**79.1**	80.8	66.7	69.7	74.8
VFM-UDA	ViT-L/14	DINOv2 [23]	307.6M	78.3	78.7	82.5	69.9	**70.2**	76.3

Table 2. **Semantic segmentation performance for the real-to-real scenario.** We use the Cityscapes → Mapillary setup, where both the source and the target domain contain real-world images, to analyze the performance of the UDA methods when the domain gap is relatively small. The out-of-target dataset is WildDash2.

Image masking	Token masking	FD	HRDA	mIoU
–	–	–	–	72.8
✓	–	–	–	76.8
–	✓	–	–	**77.1**
–	✓	✓	–	75.1
–	✓	–	✓	75.1
–	✓	✓	✓	76.9

Table 3. **Analysis of VFM with different UDA components.** The model is initialized with DINOv2 pretraining, adapting GTA5 → Cityscapes. Using masking improves performance, while the multi-resolution training proposed in HRDA [11] has no positive effect.

Method	Image size (px)	mIoU	Time (ms)	Speedup
MIC (MiT-B5)	1024×2048	75.9	1072	1.0×
DaFormer (MiT-B5)	512×1024	68.3	110	9.7×
SePiCo (MiT-B5)	640×1280	70.3	116	9.3×
HRDA (MiT-B5)	1024×2048	73.8	1072	1.0×
MIC (MiT-B5)	1024×2048	75.9	1072	1.0×
VFM-UDA (ViT-B/14)	1024×2048	77.1	128	8.4×
VFM-UDA (ViT-L/14)	1024×2048	79.0	323	3.3×

Table 4. **Inference runtime analysis.** After adapting GTA5 → Cityscapes, the performance of all UDA methods is measured on an Nvidia A6000 GPU with 16-bit mixed precision. The VFM-UDA combination benefits from higher inference speed compared to its baseline MIC, even when using ViT-L/14 which has 3.6× more parameters.

speed Specifically, using the ViT-B/14 model achieves an 8.4× speed increase over the prior state-of-the-art method MIC [12]. It obtains this speed because – unlike the HRDA and MIC methods – it does not use the HRDA technique that requires multiple inference passes. Even when scaling up to a ViT-L/14 model, which has 3.6× more parameters, the VFM-UDA method still maintains a significant advantage, operating 3.3× faster than HRDA-based approaches. The experiments are run on a Nvidia A6000 GPU with a batch size of 1, and we report the average inference time per image.

4.3. Analysis with different VFMs

To determine the most effective Vision Foundation Model (VFM) for UDA, we initially selected DINOv2 due to its robust performance across a range of downstream tasks [23]. To validate this choice and explore alternatives, we extend

Method	In-target (mIoU)	Out-of-target (mIoU)
EVA-02 [7]	65.8	57.3
EVA-02-CLIP [32]	72.3	58.4
DINOv2 [23]	77.1	61.3

Table 5. **Effect of using different VFMs**. When adapting GTA5 → Cityscapes, on the ViT-B/14 encoder, DINOv2 [23] provides the best in-target and out-of-target performance.

Method	In-target (mIoU)		Out-of-target (mIoU)	
	ImageNet-1K	DINOv2	ImageNet-1K	DINOv2
ViT-S/14	66.9	69.7	46.0	56.2
ViT-B/14	68.1	77.1	51.8	61.3
ViT-L/14	67.3	79.0	54.6	65.5

Table 6. **Effect of model size with different pre-training strategies.** We use an ImageNet pre-trained ViT as our non-VFM model and DINOv2 as the VFM. Adapting GTA5 → Cityscapes and scaling with model size, the ImageNet pre-trained model is unable to scale in performance, while the DINOv2 shows a consistent increase for both in- and out-of-target.

our analysis to include two other VFMs, EVA02 [7] and EVA02-CLIP [32]. These models were chosen for their close performance to DINOv2, making them suitable candidates for comparison [23]. We evaluated these VFMs in a synthetic-to-real adaptation scenario, specifically adapting GTA5 to Cityscapes, and assessed both in-target and out-of-target performance to study their adaptation and generalization capabilities.

Findings. Tab. 5 shows that DINOv2 consistently outperforms EVA-02 and EVA-02-CLIP with a significant margin of +4.8 mIoU points in terms of in-target performance and +2.9 points in terms of out-of-target performance compared to EVA-02-CLIP. While EVA-02 and EVA-02-CLIP yield similar results in out-of-target scenarios, EVA-02-CLIP surpasses EVA-02 in terms of in-target performance. This experiment underscores DINOv2's superior adaptability and generalization, supporting its selection for the VFM-UDA method.

4.4. Effect of model size with different pre-training strategies

In our examination of model scaling with ImageNet pre-training versus VFM pre-training, we compare the adaptation and generalization capabilities at varying sizes: small (ViT-S/14), base (ViT-B/14), and large (ViT-L/14). We use DeiT III [33] for the ViTs pre-trained on ImageNet-1K.

Findings. Tab. 6 shows that models pre-trained on ImageNet do not exhibit improved performance with increased model size in the in-target setting. In contrast, models pre-trained with DINOv2 show a consistent improvement in performance as model size increases, for both in-target and out-of-target evaluations. This shows the benefit of using VFMs for UDA, as they have the potential to benefit from increased model scale, to achieve superior generalization.

5. Conclusions

In this work, we explore whether the generalization capabilities of UDA and VFMs are complementary, to obtain models that can excel at both adaptation to a specific target domain and generalization beyond this target domain. Due to architectural differences between VFMs and encoders previously used in UDA, we made the necessary adjustments for the combined model to work in multiple configurations. From the experiments, we found that at equivalent model sizes, the combined VFM-UDA model can (a) adapt better to target domains than current state-of-the-art UDA methods, while (b) maintaining – or even slightly improving – the out-of-distribution generalization performance of VFMs. Moreover, we found that the VFM-UDA combination benefits from increased model scale, as larger VFMs yield higher in- and out-of-target performance. This study sets new standards and baselines for UDA for target-specific adaptation and out-of-distribution generalization, and offers practical guidelines for integrating VFMs into UDA to harness their joint benefits.

References

[1] Nikita Araslanov and Stefan Roth. Self-supervised Augmentation Consistency for Adapting Semantic Segmentation. In *CVPR*, 2021. 3

[2] Hangbo Bao, Li Dong, and Furu Wei. BEiT: BERT Pre-Training of Image Transformers. In *ICLR*, 2021. 3

[3] 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. 2

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML*, 2020. 2

[5] 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

[6] 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. 2, 4

[7] 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, 7

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 2

[9] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked Autoencoders Are Scalable Vision Learners. In *CVPR*, 2022. 1, 2

[10] Lukas Hoyer, Dengxin Dai, and Luc Gool. DAFormer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In *CVPR*, 2022. 2, 3, 4, 6

[11] Lukas Hoyer, Dengxin Dai, and Luc Gool. HRDA: Context-aware high-resolution domain-adaptive semantic segmentation. In *ECCV*, 2022. 1, 2, 3, 5, 6

[12] Lukas Hoyer, Dengxin Dai, Haoran Wang, and Luc Van Gool. MIC: Masked Image Consistency for Context-Enhanced Domain Adaptation. In *CVPR*, 2023. 1, 2, 3, 4, 6

[13] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Domain Adaptive and Generalizable Network Architectures and Training Strategies for Semantic Image Segmentation. *IEEE TPAMI*, 46(1):220–235, 2024. 3

[14] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloé Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment Anything. In *ICCV*, 2023. 1, 2, 3

[15] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017. 3

[16] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional Learning for Domain Adaptation of Semantic Segmentation. In *CVPR*, 2019. 2

[17] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014. 2

[18] Xiaofeng Liu, Chae Hwa Yoo, Fangxu Xing, Hyejin Oh, Georges El Fakhri, Je-Won Kang, and Jonghye Woo. Deep Unsupervised Domain Adaptation: A Review of Recent Advances and Perspectives. *arXiv preprint arXiv:2208.07422*, 2022. 1

[19] 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. 2

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 4

[21] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a Closer Look at Domain Shift: Category-Level Adversaries for Semantics Consistent Domain Adaptation. In *CVPR*, 2019. 2

[22] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In *ICCV*, 2017. 4

[23] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael G. Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *TMLR*, 2024. 1, 2, 3, 4, 6, 7

[24] Duo Peng, Yinjie Lei, Munawar Hayat, Yulan Guo, and Wen Li. Semantic-Aware Domain Generalized Segmentation. In *CVPR*, 2022. 1

[25] Fabrizio J. Piva and Gijs Dubbelman. Exploiting Image Translations via Ensemble Self-Supervised Learning for Unsupervised Domain Adaptation. *CVIU*, 234:103745, 2023. 2

[26] 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. 4

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, 2021. 2

[28] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for Data: Ground Truth from Computer Games. In *ECCV*, 2016. 2, 4

[29] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In *CVPR*, 2016. 2

[30] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211 – 252, 2015. 2, 3, 6

[31] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015. 2

[32] 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. 7

[33] Hugo Touvron, Matthieu Cord, and Hervé Jégou. DeiT III: Revenge of the ViT. In *ECCV*, 2022. 7

[34] Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, and Lennart Svensson. DACS: Domain Adaptation via Cross-Domain Mixed Sampling. In *WACV*, 2021. 4

[35] Midhun Vayyat, Jaswin Kasi, Anuraag Bhattacharya, Shuaib Ahmed, and Rahul Tallamraju. CLUDA: Contrastive Learning in Unsupervised Domain Adaptation for Semantic Segmentation. *arXiv preprint arXiv:2208.14227*, 2022. 1, 2, 3

[36] Zhixiang Wei, Lin Chen, Yi Jin, Xiaoxiao Ma, Tianle Liu, Pengyang Ling, Ben Wang, Huaian Chen, and Jinjin Zheng. Stronger, Fewer, & Superior: Harnessing Vision Foundation Models for Domain Generalized Semantic Segmentation. In *CVPR*, 2024. 2

[37] Binhui Xie, Shuang Li, Mingjia Li, Chi Harold Liu, Gao Huang, and Guoren Wang. SePiCo: Semantic-Guided

Pixel Contrast for Domain Adaptive Semantic Segmentation. *IEEE TPAMI*, 45(07):9004–9021, 2023. 1, 2, 3, 6

[38] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, José M. Álvarez, and Ping Luo. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In *NeurIPS*, 2021. 2

[39] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. SimMIM: a Simple Framework for Masked Image Modeling. In *CVPR*, 2022. 2

[40] Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. Mind the Class Weight Bias: Weighted Maximum Mean Discrepancy for Unsupervised Domain Adaptation. In *CVPR*, 2017. 2

[41] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A Survey on Deep Semi-Supervised Learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):8934–8954, 2023. 1

[42] Yang Yuan. On the Power of Foundation Models. In *ICML*, 2023. 1

[43] Oliver Zendel, Katrin Honauer, Markus Murschitz, Daniel Steininger, and Gustavo Fernández Domínguez. WildDash - Creating Hazard-Aware Benchmarks. In *ECCV*, 2018. 4