Strong but simple: A Baseline for Domain Generalized Dense Perception by CLIP-based Transfer Learning

Supplementary Material

1 Theoretical Motivation

To motivate our fine-tuning setting mathematically, we consider the domain shift from $\mathcal{D}^S$ to $\mathcal{D}^T$ as a bijective map $\mathcal{D}^S \ni \mathbf{x} \mapsto \phi(\mathbf{x}) \in \mathcal{D}^T$. Note that learning such maps is a common computer vision task [119]. Let $\mathcal{D}^S \cup \mathcal{D}^T \ni \mathbf{x} \mapsto T(\mathbf{x}) \in \mathcal{T}$ be the map which provides the text description to the image $\mathbf{x}$. We assume that, with high probability, the text description does not refer to the domain and thus remains valid in the new domain. E.g., if a scene in sunshine $\mathbf{x}$ is transferred to the rainy domain by $\phi(\mathbf{x})$, the text description only changes if there is an explicit mention of weather in either of the scenes, which is assumed to be rare. Mathematically, this is expressed as $T(\mathbf{x}) = T(\phi(\mathbf{x}))$ for $\mathbf{x} \in \mathcal{D}^S$ with probability $p \gg (1-p) \geq 0$. Furthermore, we assume perfect image-to-text feature alignment, i.e. $\mathbf{M}_E^V(\mathbf{x}) = \mathbf{M}_E^L(T(\mathbf{x}))$ for $\mathbf{x} \in \mathcal{D}^S \cup \mathcal{D}^T$. Then we have:

Lemma 1. *Under the above assumptions, the encoder-decoder network* $\mathbf{M} = \mathbf{M}_D^V \circ \mathbf{M}_E^V$ *is domain robust, i.e. provides the same output to* $\mathbf{x}$ *and* $\phi(\mathbf{x})$*, with probability not smaller than* p*.*

Proof. With probability not less than p, we have for $\mathbf{x} \in \mathcal{D}^S$

$$\mathbf{M}_E^V(\phi(\mathbf{x})) = \mathbf{M}_E^L(T(\phi(\mathbf{x}))) = \mathbf{M}_E^L(T(\mathbf{x})) = \mathbf{M}_E^V(\mathbf{x}).$$

Application of $\mathbf{M}_D^V$ to both sides completes the proof.

Let us remark that the task of the decoder network $\mathbf{M}_D^V$ plays no role in the proof of Lemma 1 which is equally valid for semantic segmentation, object detection or other downstream tasks. The mechanics of this lemma is simply based by an alignment of the invariances of the network in the sense of [24, 78] with the direction of the domain shift via feature alignment with an auxiliary modality (language, in our case) with invariance under the given shift already at the level of the data itself. Our practical implementation of this idealized description builds upon large-scale vision-language pre-trainings as done by CLIP [73], or EVA-CLIP [27] with image-to-text feature alignment as a training objective. These produce highly generalized encoder representations for $\mathbf{M}_E^V$. We utilize this pre-trained vision encoder in a simple, transfer learning-only setting for our investigations. In the following, we will introduce the details.

2 ResNet-101 & Synthia Experiments

In Tab. 8, the domain generalization performance of our VLTSeg approach is shown with a `ResNet-101` backbone. When training on GTA5, we perform competitively with previous SOTA in the DG mean, demonstrating the effectiveness

of fine-tuning for convolutional neural networks. It has to be considered that the EVA-02 [90] pre-trained initialization was not available for the `ResNet-101` backbone, so the CLIP [73] initialization was used. Moreover, it is important to note that some recent approaches, like DIDEX [65] or CLOUDS [3], rely on other foundation models like stable diffusion for data generation/augmentation or label refinement by SAM [53]. That creates a significant advantage during training compared to approaches that do not incorporate other foundation models.

We also performed experiments on the Synthia [79] dataset to verify the effectiveness on another synthetic dataset. As shown in Tab. 9 VLTSeg outperforms most of the state-of-the-art approaches and shows only a slightly lower generalization compared to DIDEX [65] which uses additional generated data from a stable diffusion model.

Table 8: Domain generalization performance in comparison with state-of-the-art approaches. Training was performed on the synthetic GTA5 ($\mathcal{D}^{\text{S}} = \mathcal{D}^{\text{GTA5}}_{\text{train}}$) dataset. All our experiments employed a CLIP [73] pre-trained `ResNet-101` [36] backbone (therefore denoted as VLTSeg-R) with a Mask2Former [15] head. Prior work results are cited from the respective paper, only the values for works marked with ∘ are taken from [69].

	DG Method	mIoU (%) on			
		$\mathcal{D}^{\text{CS}}_{\text{val}}$	$\mathcal{D}^{\text{BDD}}_{\text{val}}$	$\mathcal{D}^{\text{MV}}_{\text{val}}$	**DG mean**
$\mathcal{D}^{\text{S}}$: **GTA5**	Baseline [105]	36.1	36.6	43.8	38.8
	IBN-Net° [67]	37.7	36.7	36.8	37.1
	RobustNet° [16]	37.3	38.7	38.1	38.0
	DRPC [111]	42.5	38.7	38.1	39.8
	SW [68]	36.1	36.6	32.6	35.1
	FSDR [44]	44.8	41.2	43.4	43.1
	SAN+SAW [69]	45.3	41.2	40.8	42.4
	WEDGE [50]	45.2	41.1	48.1	44.8
	GTR [70]	43.7	39.6	39.1	40.8
	SHADE [116]	46.7	43.7	45.5	45.3
	WildNet° [56]	45.8	41.7	47.1	44.9
	TLDR [51]	47.6	44.9	48.8	47.1
	RICA [89]	48.0	45.2	46.3	46.5
	PASTA [9]	45.3	42.3	48.6	45.4
	FAMix [25]	49.5	46.4	52.0	49.3
	CLOUDS [3]	**55.7**	**49.3**	**59.0**	**54.7**
	DIDEX [65]	52.4	40.9	49.2	47.5
	DGinStyle [48]	46.9	42.8	50.2	46.6
	VLTSeg-R (Ours)	49.5	40.0	52.4	47.3

Table 9: Domain generalization performance in comparison with state-of-the-art approaches. Training was performed on the synthetic SYNTHIA and UrbanSyn dataset dataset. Prior work results are cited from the respective paper.

	DG Method	mIoU (%) on			
		$\mathcal{D}_{\text{val}}^{\text{CS}}$	$\mathcal{D}_{\text{val}}^{\text{BDD}}$	$\mathcal{D}_{\text{val}}^{\text{MV}}$	**DG mean**
$\mathcal{D}^{\text{S}}$: SYNTHIA	Baseline [105]	41.4	36.2	42.4	40.0
	ReVT [92]	46.3	40.3	44.8	43.8
	CMFormer [4]	44.6	33.4	43.3	40.4
	PromptFormer [32]	49.3	-	-	-
	IBAFormer [88]	50.9	44.7	50.6	48.7
	CLOUDS [3]	53.4	47.0	55.8	52.1
	DIDEX [65]	**59.8**	47.4	59.5	**55.6**
	VLTSeg	56.8	**51.9**	55.1	54.6
$\mathcal{D}^{\text{S}}$: US	Baseline [105]	63.3	41.3	58.8	54.5
	VLTSeg	**70.1**	**52.4**	**63.0**	**61.8**

3 Decoder Architecture

We also examine the impact of different decoder architectures on the domain generalization performance, as shown in Tab. 10. The DG mean of the model with the Mask2Former [15] decoder is 4.1% higher than the model a Semantic FPN head and also 1.5% better than the ASPP-based decoder from DAFormer [40]. VLTSeg with a Mask2Former [15] decoder performs consistently better across all four real-world datasets than the other decoder architectures. That implies that Mask2Former can leverage the provided visual embeddings from vision-language pre-training more effectively and better focus on domain-invariant features.

Table 10: Ablation study of different decoder architectures and their domain generalization performance (mIoU (%)). Training was performed on the synthetic GTA5 ($\mathcal{D}^{\text{S}} = \mathcal{D}_{\text{train}}^{\text{GTA5}}$) dataset. Evaluation is performed on the four shown real-world datasets.

Decoder	mIoU in %				
	$\mathcal{D}_{\text{val}}^{\text{CS}}$	$\mathcal{D}_{\text{val}}^{\text{BDD}}$	$\mathcal{D}_{\text{val}}^{\text{MV}}$	$\mathcal{D}_{\text{val}}^{\text{ACDC}}$	**DG mean**
Semantic FPN [61]	60.5	57.5	62.2	55.9	59.0
Segformer [105]	58.2	55.0	61.4	56.0	57.7
DAFormer [40]	64.0	57.9	65.0	59.3	61.6
Mask2Former [15]	**65.3**	**58.3**	**66.0**	**62.6**	**63.1**

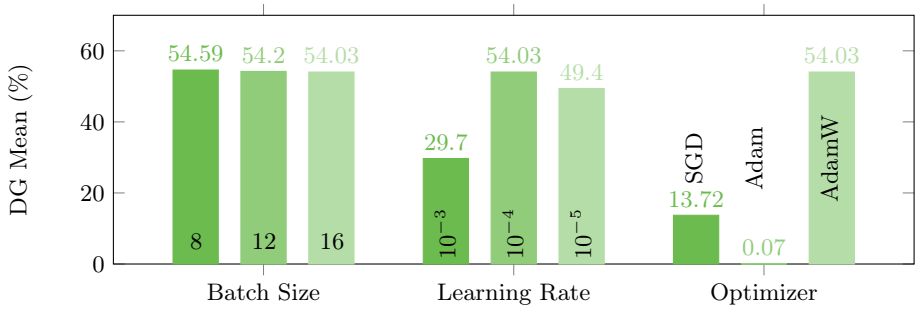
Table 11: Computational complexity of the vision encoders that were employed in our experiments. GFLOPS and Parameters are computed using `MMPreTrain` [17], numbers are rounded to integers.

Encoder	Parameters	GFLOPS
`ViT-B-16`	87	88
`ViT-L-16`	304	311
`SAM-L-16`	308	397
`EVA-02-S-16`	22	32
`EVA-02-B-16`	86	107
`EVA-02-L-14`	303	508

4 Sensitivity Analysis of the Training Configuration

In the analysis with EVA-02-B-CLIP we observe that a change of the learning rate and the optimizer affect the results significantly. Changing the optimizer to

SGD or Adam leads to a collapsed training most likely because of the randomly initialized Mask2Former decoder. While a different batch size does not affect results, as expected, a higher learning rate diminishes DG performance more than a lower one.



5 Qualitative Results

Cityscapes We show predictions on the Cityscapes test set $\mathcal{D}_{\text{test}}^{\text{CS}}$ in Fig. 4 which visualizes the state-of-the-art segmentation quality of our VLTSeg approach when training supervised on Cityscapes.

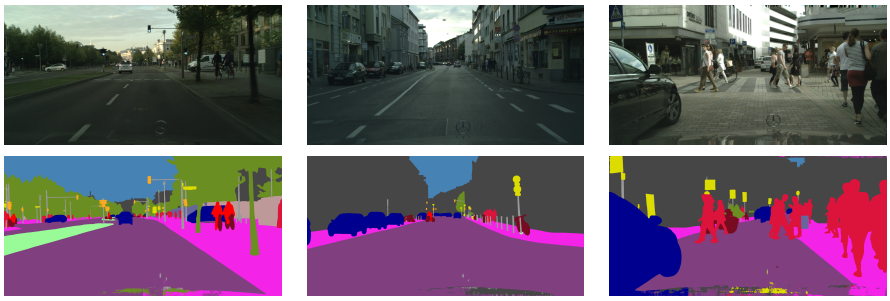


Fig. 4: Predictions on Cityscapes test set $\mathcal{D}_{\text{test}}^{\text{CS}}$. Training and evaluation was conducted as described in Sec. 6.2

ACDC We show predictions on the ACDC val set $\mathcal{D}_{\text{val}}^{\text{ACDC}}$ in Fig. 5. Even though our model has neven seen this domain before during training we can observe that it provides high-quality segmentation maps across challenging adverse weather conditions.

6 Implementation Details

6.1 Training settings

We provide an extensive list of our hyperparameters and detailed settings in Tab. 12 to make our experiments reproducible. For the Mask2Former [15] decoder we used the default settings as provided by the authors. For the `EVA-02-S-16`

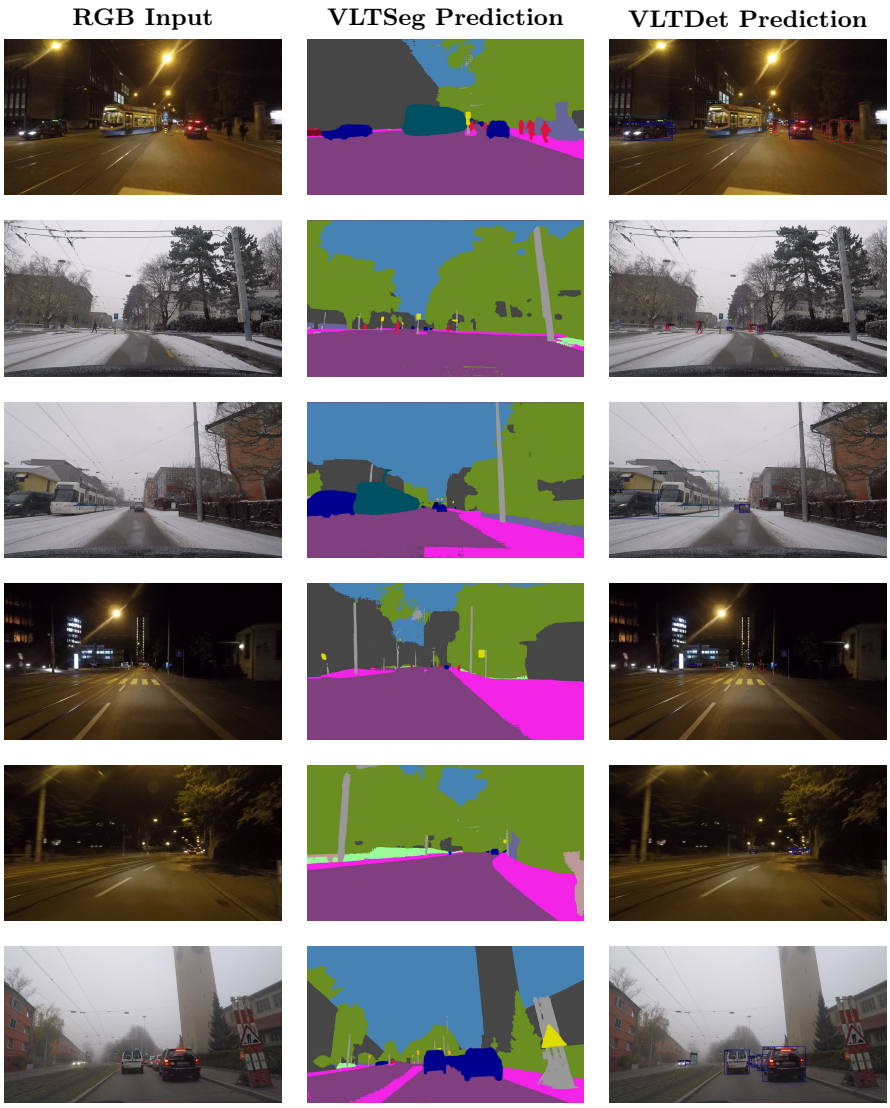


Fig. 5: Predictions on the ACDC val set $\mathcal{D}_{\text{val}}^{\text{ACDC}}$. Training on $\mathcal{D}_{\text{train}}^{\text{CS}}$ and evaluation was conducted as described in Sec. 6.2. Best viewed digital for the predictions.

model we interpolated the text encoder output projection in bilinear mode to match the vision encoder output shape of 384. Similar to DenseCLIP [74] we used FPN projection layers between Encoder and Decoder.

Table 12: Detailed Experimental Settings

Hyperparameter		Synthetic-to-Real	Real-to-Real
	crop size	512	1024
	stride size	426	768
	iterations	5k	20k
	batch size	16	8
Optimizer	type	AdamW	
	default lr	1e-04	
	backbone lr	1e-05	
	weight_decay	0.05	
	eps	1e-08	
	betas	(0.9, 0.999)	
LR Schedule	type	PolyLR	
	eta_min	0	
	power	0.9	
	begin	500	
Text	context length	13	
	embedding size	768	
	transformer heads	12	
	transformer width	768	
	transformer layers	12	
Encoder	in_channels	3	
	patch size	14	
	embedding size	1024	
	depth	24	
	indices	[9, 14, 19, 23]	
	output_dim	768	
Decoder	in_channels	[1024,1024,1024,1024]	
	stride	[4, 8, 16, 32]	
	feat_channels	256	
	out_channels	256	
	num_classes	19	
	num_queries	100	
	num_transformer_feat_level	3	
	positional_encoding	128	

6.2 Test set evaluation

For the evaluation on the Cityscapes [20] and ACDC [81] test set we followed the corresponding common practice.

For Cityscapes test set evaluation, we first trained VLTSeg on Mapillary for 20k iterations as also done by previous state-of-the-art approaches [6, 7, 99]. Afterwards, 40k fine-tuning iterations on the official $\mathcal{D}^{\text{CS}}_{\text{train}}$ dataset with 2975 images were conducted. In contrast to the other works, no additional data like the coarse annotations were used. Both trainings used 1024×1024 resolution. We used multi-scale evaluation with $[1.0, 1.25, 1.5, 1.75, 2.0, 2.25]$ image ratios and random flip as test-time augmentation during inference. Test set evaluation was done on the full $2048\times$ 1024 resolution.For the ACDC test set evaluation, we trained for 20k iterations on 1024×1024 crops only on Cityscapes $\mathcal{D}^{\text{CS}}_{\text{train}}$ without using any ACDC data. The test set inference was similar to Cityscapes with a multi-scale evaluation with $[1.0, 1.25, 1.5, 1.75]$ image ratios and on the full 1920×1080 resolution.

7 Computational complexity of vision encoders

We show the GFLOPS and parameters of the vision encoders employed in our experiments in Tab. 11. We used three variants of the `ViT` with base, large, and SAM and a patch size of 16. Moreover, we employed three complexities of EVA: small, base, and large. Due to the architectural modifications, EVA has more GFLOPS than ViT at a similar number of parameters.

8 Feature Space Analysis

We conducted a feature space analysis for the other real-world datasets for Cityscapes [19] before and after fine-tuning on the synthetic GTA5 [75] dataset. The results are shown in Fig. 6.
We observe that real-world embeddings are well divided for the real-world target dataset after the synthetic source-only training. That implies that fine-tuning improves the feature space of vision-text alignment by separating classes more clearly and offering strong generalization capabilities across several different real-world domains.

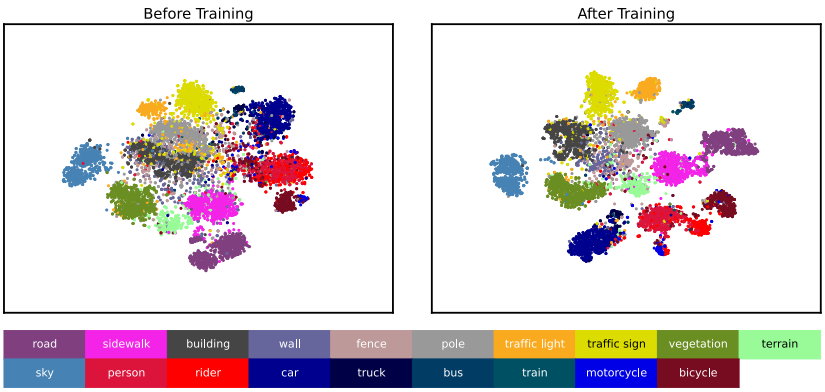


Fig. 6: t-SNE feature space analysis on the real-world dataset Cityscapes. We sampled 500 images from the real-world validation set and extracted the visual embeddings of our best performing VLTSeg network. From the plot, we can see that real target class clusters are well separated (right image) after our VLTSeg training on source synthetic GTA5 dataset. Best viewed digitally.