SoMA: Singular Value Decomposed Minor Components Adaptation for Domain Generalizable Representation Learning

Supplementary Material

For a comprehensive understanding of our proposed SoMA framework, we have provided this supplementary material. The following table of contents gives a concise overview and directs readers to specific sections of interest.

Contents

A. Implementation Details

We utilize the MMSegmentation [10] and MMDetection [6] codebase for Domain Generalized Semantic Segmentation (DGSS) and Domain Generalized Object Detection (DGOD) implementations, respectively, and leverage the training scripts developed by HuggingFace [45] for subject personalization experiments.

A.1. DGSS Settings

The experimental settings for all studies conducted in the main paper are outlined in Tab. 1. Unless otherwise specified, Mask2Former [8] is utilized as the default decode head, and following Rein [46], we adopt only the basic data augmentation used in Mask2Former. Additionally, EMA is selectively employed to ensure stable training. To avoid potential overfitting due to the large model (dimension) size when employing DINOv2-giant as the backbone, we opt to lower the SoMA rank from 16 to 8.

A.2. DGOD Settings

The DGOD settings are detailed in the rightmost two columns of Tab. 1. When applying SoMA to convolution-based backbones such as ResNet [17], we linearize both the patch-level convolution and its weights. Specifically, a single convolution operation can be represented as a linear layer, $y = Wx$, where $x \in \mathbb{R}^{(n\times h\times w)\times 1}$, $y \in \mathbb{R}^m$, and $W \in \mathbb{R}^{m\times(n\times h\times w)}$. We then apply SoMA as described in Eq. 1 of the main paper. For ResNet backbones, which possess a much narrower pre-trained knowledge compared to VFMs, we use extensive image corruption techniques to simulate domain shifts, following DivAlign [11]. In contrast, when using DINOv2 [35] as the backbone, we simply utilize basic data augmentation used in Co-DETR [54].

A.3. Subject Personalization Settings

We conduct experiments on the DreamBooth dataset [41], which consists of 30 subjects with 4–6 images per subject. In all experiments the SoMA weights are trained using Adam optimizer for 500 iterations with a learning rate of $5e-5$. We set the adapter rank to $r = 32$ and only use a center crop for data augmentation. Inspired by recent findings [16] that the first 10 attention layers of `up_blocks.0` in SDXL [40] are pivotal for preserving image content, we fine-tune only these layers. Furthermore, to fully leverage pre-trained image-text joint representations, we freeze the cross-attention modules and apply SoMA solely to the self-attention modules.

B. Detailed Ablations

B.1. Component Analysis

In this subsection, we conduct detailed ablation studies under multiple settings: *GTAV* $\rightarrow$ *Mapillary* DGSS and *Daytime-Sunny* $\rightarrow$ {*Dusk-Rainy, Daytime-Foggy*} DGOD scenarios. In Tables 2 and 3, we systematically evaluate the effectiveness of each component within the SoMA framework based on class-wise IoU/AP (%). All proposed components enhance overall generalization performance without adding any additional training or inference costs.

As illustrated in Tab. 2, *freezing early blocks* not only substantially reduces the number of trainable parameters but also significantly improves performance for classes that are infrequently observed in the source dataset (e.g., bicycle, motorcycle, train). Additionally, *tuning minor singular components* maximizes the retention of VFM's world knowledge during task adaptation, leading to superior generalization performance over tuning principal components (PiSSA [33]) for most classes. For an in-depth comparison of our methods with PiSSA, please refer to Sec. E. Lastly, *annealing weight decay* proves especially beneficial for classes requiring fine-detail discrimination (e.g., road *vs.* sidewalk, traffic light *vs.* traffic sign, car *vs.* truck *vs.* bus

Hyperparameters	DGSS						DGOD	
Setting	*G*→{*C, B, M*}	*G+S*→{*C, B, M*}	*G+S+U*→{*C, B, M*}		*C*→{*B, M*}/*ACDC*	$\frac{1}{16}$*C*→{*C, B, M*}	*DS*→{*NC, DR, NR, DF*}	
Backbone	DINOv2-L/EVA02-L	DINOv2-L	DINOv2-L	DINOv2-G	DINOv2-L	DINOv2-L	DINOv2-L	RN101
rank r	16	16	16	8	16	16	16	24
NFEB	8	8	8	12	8	8	8	21
optimizer	AdamW							
lr scheduler	Linear	Linear	Linear	Linear	Linear	Linear	MultiStep	MultiStep
AWD scheduler	Cosine							
learning rate	1e-4	1e-4	1e-4	1e-4	1e-4	1e-5	2e-4	2e-4
backbone lr mult.	0.5							
weight decay	5e-2/3e-2	5e-2	5e-2	5e-2	5e-2	5e-2	5e-2	1e-3
batch size	4	4	4	4	8	8	8	4
warmup iters	0	0	0	0	1.5k/10k	0	1.5k	1.5k
iters	40k	40k	40k	40k	40k	4k	40k	40k
EMA	✓	✓	✓	✓	✓	✓	✓	✓

Table 1. **DGSS/DGOD hyperparameter configurations.** "NFEB" denotes the number of frozen early blocks.

Methods	Params.	road	side.	build.	wall	fence	pole	light	sign	vege.	terr.	sky	pers.	rider	car	truck	bus	train	motor.	bicy.	mIoU
Full fine-tuning (baseline)	304.2M	92.1	64.5	87.8	49.0	**56.4**	58.8	66.1	57.3	**82.9**	**53.9**	**95.0**	79.3	**63.2**	91.8	65.3	75.9	50.9	64.7	54.1	68.9
∟ + Freezing early blocks	201.6M	91.6	65.0	87.6	46.9	55.0	57.0	66.6	52.9	81.5	52.5	94.5	77.9	56.4	91.6	64.5	75.8	60.1	68.8	57.3	68.6
∟ + Tuning principal components	4.9M	93.1	69.4	88.3	48.4	53.9	59.0	67.4	57.5	81.9	53.1	94.8	79.9	61.4	90.0	65.6	80.8	44.7	70.7	55.3	69.2
∟ + Tuning minor components	4.9M	93.4	70.6	88.2	**52.4**	55.0	**59.1**	68.3	60.4	81.8	52.3	94.9	**79.9**	61.0	91.2	69.7	84.5	60.2	70.9	59.3	71.2
∟ + Annealing weight decay	4.9M	**93.6**	**71.4**	**88.3**	52.3	54.9	**59.1**	**69.4**	**62.7**	81.9	52.8	94.9	79.4	57.7	**91.8**	**72.9**	**85.3**	**61.9**	**71.6**	**60.1**	**71.7**

Table 2. **Effect of the proposed components under *GTAV* → *Mapillary* DGSS setting**. We highlight the **best** and second-best for each column.

		Daytime-Sunny→*Dusk-Rainy*								*Daytime-Sunny*→*Daytime-Foggy*							
Methods	Params.	bus	bike	car	motor	person	rider	truck	mAP	bus	bike	car	motor	person	rider	truck	mAP
Full fine-tuning (baseline)	307.3M	61.2	44.9	80.7	45.2	56.0	41.0	67.2	56.6	46.4	37.3	65.1	42.9	49.2	48.9	40.9	47.2
∟ + Freezing early blocks	201.6M	63.0	46.5	**81.2**	46.0	58.2	39.5	68.0	57.5	47.5	39.5	67.2	45.6	50.3	50.0	40.6	48.7
∟ + Tuning principal components	4.9M	64.3	49.2	80.5	46.3	57.9	41.6	68.2	58.3	48.5	40.1	67.5	47.5	50.3	51.0	45.3	50.0
∟ + Tuning minor components	4.9M	65.6	**50.6**	80.8	46.9	58.4	**42.7**	69.4	59.2	**50.6**	39.7	67.7	48.3	50.9	51.6	**46.1**	50.7
∟ + Annealing weight decay	4.9M	**65.7**	50.5	81.1	**48.1**	**59.0**	41.0	**69.5**	**59.3**	50.3	**40.9**	**67.9**	**48.6**	**51.5**	**52.3**	45.7	**51.0**

Table 3. **Effect of the proposed components under *Daytime-Sunny* → *Dusk-Rainy* and *Daytime-Sunny* → *Daytime-Foggy* DGOD settings**. We highlight the **best** and second-best for each column.

vs. train, motorcycle *vs.* bicycle). Likewise, all components clearly improve recognition for the majority of classes under adverse weather detection settings (see Tab. 3).

B.2. Freezing Scheme

While freezing the initial blocks of VFM is effective in preserving its generalization ability during task adaptation, freezing too many blocks can lead to a reduction in discriminability. To better understand this trade-off, we explore the effects of varying the number of frozen blocks. Tab. 4 shows that freezing up to the first 8 blocks progressively enhances performance, but freezing beyond this point results in a decline. Considering that feature maps from multiple blocks (e.g., the 8th, 12th, 16th, and 24th blocks in large-sized backbones) serve as inputs to the segmentation/detection head, using more than one frozen VFM features as head input significantly undermines task adaptability (*i.e.* discriminability). Furthermore, since the first input feature map of the decode head is directly incorporated into the final mask prediction in Mask2Former [8], freezing the blocks that generate this feature map allows the full utilization of the generalization capacity of the early blocks in VFMs (see Fig. 3 in the main paper).

# frozen early blocks	0	4	8	12	16
Citys. perf. (mIoU in %)	70.62	71.51	**71.82**	70.71	70.47
Params.*	7.3M	6.1M	4.9M	3.7M	2.4M

Table 4. Performance comparison with **varying numbers of frozen early blocks** under *GTAV* → *Cityscapes* DGSS setting.

Backbone Ablation			Test Domains (mIoU in %)			
Backbones	Methods	Params.*	→*Citys.*	→*BDD*	→*Map.*	Avg.
Single-source DGSS Trained on GTAV						
DINOv2-L [35]	FFT	304.2M	66.93	57.34	68.89	64.39
	SoMA	4.9M	**71.82**	**61.31**	**71.67**	**68.27**
DINOv2-B [35]	FFT	86.5M	60.84	52.98	62.12	58.65
	SoMA	2.3M	**66.71**	**57.48**	**67.34**	**63.84**
DINOv2-S [35]	FFT	22.0M	53.71	49.03	58.10	53.61
	SoMA	1.0M	**57.58**	**52.95**	**62.48**	**57.67**
ConvNeXt V2-L [47]	FFT	196.4M	55.93	50.71	60.79	55.81
	SoMA	12.1M	**60.12**	**53.36**	**61.46**	**58.31**
Swin-L [32]	FFT	195.2M	54.40	49.85	60.05	54.77
	SoMA	5.4M	**56.91**	**51.98**	**60.73**	**56.54**
ResNet101 [17]	FFT	42.3M	**41.29**	44.29	48.79	44.79
	SoMA	2.5M	41.23	**45.57**	**49.71**	**45.50**

Table 5. **Results across various backbones and model sizes.**

SemFPN Results			Test Domains (mIoU in %)			
Backbones	Methods	Params.*	→*Citys.*	→*BDD*	→*Map.*	Avg.
Single-source DGSS Trained on GTAV						
DINOv2-L [35]	Rein [46]	2.5M	63.60	59.00	63.70	62.10
	SoMA	4.9M	**67.81**	**60.12**	**68.95**	**65.63**
EVA02-L [15]	Rein [46]	2.5M	61.40	**58.50**	62.00	60.70
	SoMA	5.1M	**64.91**	57.54	**65.33**	**62.59**

Table 6. DGSS evaluation results with **SemFPN head** [28].

C. Additional Experiments

C.1. Results on Various Backbones

Tab. 5 showcases the versatility of SoMA across a wide range of backbones, ranging from isotropic Vision Transformers (ViTs) to ConvNets and hierarchical ViT, as well as models trained under various approaches, such as ImageNet [12] supervision and MAE [18, 47] pre-training. SoMA consistently outperforms FFT across diverse backbone architectures. Notably, the improvements brought by SoMA become increasingly pronounced with larger model sizes and more extensive, high-quality data during pre-training, highlighting the superior ability of our method to preserve pre-trained knowledge.

C.2. Results on SemFPN Head

While Mask2Former [8] is predominantly employed as decode head in all DGSS experiments, SoMA is compatible with any decode head. To assess its robustness across different heads, we employ the lightweight SemFPN [28] head to benchmark its performance against Rein [46]. Our experimental results (Tab. 6) indicate that SoMA integrates seamlessly with diverse backbones and heads, consistently surpassing the SOTA baseline.

Efficiency	Training (bs = 4)		Inference (bs = 1 / 32)	
Methods	Time (hrs)	Memory	Throughput (imgs/s)	Memory
SET $_{\text{large}}$	9.2	12.5G	20.0 / -	5.5G / OOM
Rein $_{\text{large}}$	9.3	**12.2G**	33.6 / 64.3	4.7G / 48.2G
SoMA $_{\text{large}}$	**9.0**	12.7G	**56.4 / 79.7**	**4.4G / 40.9G**
FFT $_{\text{giant}}$	27.9	45.3G	21.6 / -	10.6G / OOM
SoMA $_{\text{giant}}$	**18.9**	**25.6G**	21.6 / -	10.6G / OOM

Table 7. **DGSS model efficiency.** Inference statistics are measured only for the backbone on image crops of 512×512, and are measured with warmup and averaged over multiple runs. We use an NVIDIA RTX A6000. "bs" denotes batch size.

Synthetic-to-Real Generalization			Test Domains (mIoU in %)			
Methods	Backbone	Params.*	→*Citys.*	→*BDD*	→*Map.*	Avg.
Single-source DGSS Trained on GTAV						
PEGO [20]	DINOv2-L	2.6M	68.86	**61.44**	68.61	66.30
PiSSA$_{r16}$ [33]	DINOv2-L	7.3M	69.43	60.62	69.44	66.50
SoMA (Ours)	DINOv2-L	4.9M	**71.82**	61.31	**71.67**	**68.27**

Table 8. Performance Comparison of our **SoMA against PEGO and PiSSA** under the basic DGSS setting. The reported performance of PEGO indicates the best result achieved within the hyperparameter space proposed in the original paper.

D. Model Efficiency Comparison

As detailed in Tab. 7, SoMA exhibits higher throughput than adapter- and VPT-based methods like Rein [46] and SET [50], as it incurs no additional latency. This advantage is especially significant in online inference settings, where the batch size is typically as small as one [19]. Furthermore, in scenarios involving DINOv2-giant exceeding 1B parameters, SoMA can drastically reduce training costs compared to FFT. SoMA initialization is completed within 30 seconds for large-sized models, which is a negligible cost given the improved performance.

E. Additional Comparison

In Tables 9, 10, and 11, we present an exhaustive comparison with existing methods to illustrate the broader research landscape across multiple DGSS settings.

Comparing SoMA with PiSSA [33] **and PEGO** [20]. PiSSA optimizes parameter efficiency by selectively adjusting the principal singular direction, which is the most stretched direction of the weight matrix. Also PEGO enhances domain generalization by enforcing strict orthogonality between the LoRA adapter and every direction of the pre-trained weights. In stark contrast to PiSSA, SoMA tunes minor singular components, effectively preserving the integrity of pre-trained knowledge. Importantly, our method accounts not only for component orthogonality but also for how pre-trained knowledge is structured within the weight matrix. Thus, although both methods maintain similar orthogonality between tuned and frozen com-

ponents, SoMA, unlike PiSSA—which directly tunes principal components—robustly adjusts the hierarchical world knowledge structure (see Fig. 2 in the main paper), achieving superior DGSS performance, as shown in Tab. 8. These findings are consistent with the observations presented in the Ablation Sec. B.1.

Furthermore, unlike PEGO, our SoMA leverages spectral information to initialize the LoRA adapter, allowing it to naturally preserve the pre-trained knowledge structure without relying on explicit regularization loss. As evidenced by its superior performance relative to PEGO in Tab. 8, we argue that imposing strict orthogonality constraints on all directions of the pre-trained weights may excessively restrict task adaptation, potentially compromising discriminability. Lastly, whereas PEGO and PiSSA explore adaptation solely at the weight level, our SoMA framework extends its analysis to both the block level and training dynamics.

DGSS and DGOD Qualitative Comparison. Figures 1, 2, and 3 depict DGSS prediction results on unseen domains for Cityscapes, BDD100k, and Mapillary, respectively, while Fig. 4 provides detection results under various adverse conditions. As evident from the visual comparisons above, SoMA demonstrates remarkable robustness to domain shifts resulting from diverse attributes (e.g., translucency, lighting conditions, road features, geographic variations, weather differences), while also excelling in fine-detail recognition compared to the selected baselines.

Subject Personalization. Domain generalized recognition requires consistent processing of inputs from diverse domains, whereas domain generalized generation involves generating outputs across a range of domains. Although large-scale Text-to-Image (T2I) models have convincingly demonstrated this ability, it can be compromised in subject personalization tasks involving fine-tuning. As shown in Figures 5 and 6, integrating the SoMA framework in this case enables T2I models to fully leverage their generalization capability to synthesize target subjects in new domains.

In summary, our proposed methods effectively facilitate domain-generalizable representation learning by maximally preserving pre-trained knowledge across diverse domains while learning task-specific features.

F. Discussion and Limitations

Our adaptation approach introduces SVD as an interpretable tool applied to raw weight matrices, offering a fresh perspective on domain generalization. Within this perspective, we focus on tuning the minor singular components to preserve the integrity of generalizable components with minimal interference. However, achieving further performance improvements will require a more structured and nuanced design space. Questions such as whether focusing solely on the lowest spectral space is optimal, or how to identify and adjust specific singular components for particular tasks, remain as avenues for future exploration. Additionally, we plan to investigate design choices such as setting different ranks for each block or examining whether the low-rank matrices A and B play distinct roles, analyzing how these decisions influence generalization performance. Extending these comprehensive analyses to other domains where foundation models are primarily employed, such as LLM benchmarks and audio applications, would also be an exciting direction for future work.

Synthetic-to-Real Generalization			Test Domains (mIoU in %)			
Methods	Backbone	Head	→*Citys.*	→*BDD*	→*Map.*	Avg.
Single-source DGSS Trained on GTAV						
○ IBN-Net [37]	RN50	DL-V3+	33.85	32.30	37.75	34.63
○ RobustNet [9]	RN50	DL-V3+	36.58	35.20	40.33	37.37
○ DRPC [51]	RN101	FCN	42.53	38.72	38.05	39.77
○ SiamDoGe [48]	RN50	DL-V3+	42.96	37.54	40.64	40.38
○ DIRL [49]	RN50	DL-V3+	41.04	39.15	41.60	40.60
○ GTR [38]	RN101	-	43.70	39.60	39.10	40.80
○ AdvStyle [53]	RN101	DL-V3+	43.44	40.32	41.96	41.91
○ PintheMem [26]	RN101	DL-V2	44.90	39.71	41.31	41.97
○ MRFP+ [44]	RN50	DL-V3+	42.40	39.55	44.93	42.29
○ SAN-SAW [39]	RN101	DL-V3+	45.33	41.18	40.77	42.43
○ SPC [21]	RN50	DL-V3+	44.10	40.46	45.51	43.36
○ BlindNet [1]	RN50	DL-V3+	45.72	41.32	47.08	44.71
○ WildNet [29]	RN101	DL-V3+	45.79	41.73	47.08	44.87
○ SHADE [52]	RN101	DL-V3+	46.66	43.66	45.50	45.27
○ PASTA [5]	RN101	DL-V3+	45.33	42.32	48.60	45.42
○ SoMA (Ours)	RN101	M2F	41.23	45.57	49.71	45.50
○ MoDify [25]	RN101	DL-V2	48.80	44.20	47.50	46.80
○ TLDR [27]	RN101	DL-V3+	47.58	44.88	48.80	47.09
○ FAMix [14]	CLIP RN101	DL-V3+	49.47	46.40	51.97	49.28
○ CMFormer [4]	Swin-L	-	55.31	49.91	60.09	55.10
○ SoMA (Ours)	Swin-L	M2F	56.91	51.98	60.73	56.54
○ DGInStyle [24]	MiT-B5	HRDA	58.63	52.25	62.47	57.78
○ DIDEX [34]	MiT-B5	DAFormer	62.00	54.30	63.00	59.70
○ CLOUDS [2]	CLIP CN-L	M2F	60.20	57.40	67.00	61.50
○ VLTSeg [22]	EVA02-L	M2F	65.30	58.30	66.00	63.20
○ Rein [46]	EVA02-L	M2F	65.30	60.50	64.90	63.60
○ FADA [3]	EVA02-L	M2F	66.70	61.90	66.10	64.90
○ tqdm [36]	EVA02-L	M2F	68.88	59.18	70.10	66.05
○ SoMA (Ours)	EVA02-L	M2F	68.05	60.81	68.33	65.73
● SoMA (Ours)	EVA02-L	M2F	69.94	62.48	68.33	66.92
○ DoRA [31]	DINOv2-L	M2F	66.12	59.31	67.07	64.17
○ VPT [23]	DINOv2-L	M2F	68.75	58.64	68.32	65.24
○ SET [50]	DINOv2-L	M2F	68.06	61.64	67.68	65.79
○ FADA [3]	DINOv2-L	M2F	68.23	61.94	68.09	66.09
○ AdaptFormer [7]	DINOv2-L	M2F	70.10	59.81	68.77	66.23
○ SSF [30]	DINOv2-L	M2F	68.97	61.30	68.77	66.35
○ LoRA [19]	DINOv2-L	M2F	70.13	60.13	70.42	66.89
○ Rein† [46]	DINOv2-L	M2F	69.19	60.01	69.06	66.09
● Rein† [46]	DINOv2-L	M2F	70.68	62.51	69.61	67.60
○ SoMA (Ours)	DINOv2-L	M2F	71.82	61.31	**71.67**	68.27
● SoMA (Ours)	DINOv2-L	M2F	**73.63**	**63.33**	70.98	**69.31**
Multi-source DGSS Trained on GTAV + SYNTHIA						
○ RobustNet [9]	RN50	DL-V3+	37.69	34.09	38.49	36.76
○ AdvStyle [53]	RN50	DL-V3+	39.29	39.26	41.14	39.90
○ DIGA [43]	RN101	DL-V2	46.43	33.87	43.51	41.27
○ PintheMem [26]	RN50	DL-V3+	44.51	38.07	42.70	41.76
○ MRFP+ [44]	RN50	DL-V3+	46.18	41.13	45.28	44.24
○ SHADE [52]	RN50	DL-V3+	47.43	40.30	47.60	45.11
○ TLDR [27]	RN50	DL-V3+	48.83	42.58	47.80	46.40
○ SPC [21]	RN101	DL-V3+	47.93	43.62	48.79	46.78
○ FAMix [14]	CLIP RN50	DL-V3+	49.41	45.51	51.61	48.84
○ Rein† [46]	DINOv2-L	M2F	72.17	61.53	70.69	68.13
○ SoMA (Ours)	DINOv2-L	M2F	73.16	61.90	72.73	69.26
● SoMA (Ours)	DINOv2-L	M2F	**74.85**	**63.59**	**73.92**	**70.79**

Table 9. Comparison of the proposed SoMA with existing DGSS ○ and PEFT ○ methods under various **synthetic-to-real settings**.

Real-to-Real Generalization			Test Domains (mIoU in %)		
Methods	Backbone	Head	→*BDD*	→*Map.*	Avg.
Single-source DGSS Trained on Cityscapes					
○ RobustNet [9]	RN50	DL-V3+	50.73	58.64	54.69
○ WildNet [29]	RN50	DL-V3+	50.94	58.79	54.87
○ SiamDoGe [48]	RN50	DL-V3+	51.53	59.00	55.27
○ SHADE [52]	RN50	DL-V3+	50.95	60.67	55.81
○ BlindNet [1]	RN50	DL-V3+	51.84	60.18	56.01
○ FAMix [14]	CLIP RN50	DL-V3+	54.07	58.72	56.40
○ SAN-SAW [39]	RN101	DL-V3+	54.73	61.27	58.00
○ HGFormer [13]	Swin-L	-	61.50	72.10	66.80
○ CMFormer [4]	Swin-L	-	62.60	73.60	68.10
○ tqdm [36]	EVA02-L	M2F	64.72	76.15	70.44
○ FADA [3]	DINOv2-L	M2F	65.12	75.86	70.49
○ Rein† [46]	DINOv2-L	M2F	66.53	75.18	70.86
○ SoMA (Ours)	DINOv2-L	M2F	67.02	76.45	71.74
● SoMA (Ours)	DINOv2-L	M2F	**68.08**	**77.87**	**72.98**

Table 10. **Real-to-real DGSS comparison.**

Clear-to-Adverse Weather	ACDC [42] Test Domains (mIoU in %)				
Methods	→*Night*	→*Snow*	→*Fog*	→*Rain*	Avg.
Single-source DGSS Trained on Cityscapes					
○ CMFormer [4]	33.7	64.3	77.8	67.6	60.9
○ SET [50]	57.3	73.7	80.1	74.8	71.5
○ FADA [3]	**57.4**	73.5	80.2	75.0	71.5
○ SoMA (Ours)	52.4	**74.6**	**84.1**	**75.5**	**71.7**

Table 11. Results on **Cityscapes → ACDC validation set.**

References

[1] Woo-Jin Ahn, Geun-Yeong Yang, Hyun-Duck Choi, and Myo-Taeg Lim. Style blind domain generalized semantic segmentation via covariance alignment and semantic consistence contrastive learning. In *CVPR*, 2024. 5

[2] Yasser Benigmim, Subhankar Roy, Slim Essid, Vicky Kalogeiton, and Stéphane Lathuilière. Collaborating foundation models for domain generalized semantic segmentation. In *CVPR*, 2024. 5

[3] Qi Bi, Jingjun Yi, Hao Zheng, Haolan Zhan, Yawen Huang, Wei Ji, Yuexiang Li, and Yefeng Zheng. Learning frequency-adapted vision foundation model for domain generalized semantic segmentation. In *NeurIPS*, 2024. 5

[4] Qi Bi, Shaodi You, and Theo Gevers. Learning content-enhanced mask transformer for domain generalized urban-scene segmentation. In *AAAI*, 2024. 5

[5] Prithvijit Chattopadhyay, Kartik Sarangmath, Vivek Vijaykumar, and Judy Hoffman. Pasta: Proportional amplitude spectrum training augmentation for syn-to-real domain generalization. In *ICCV*, 2023. 5

[6] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDe-

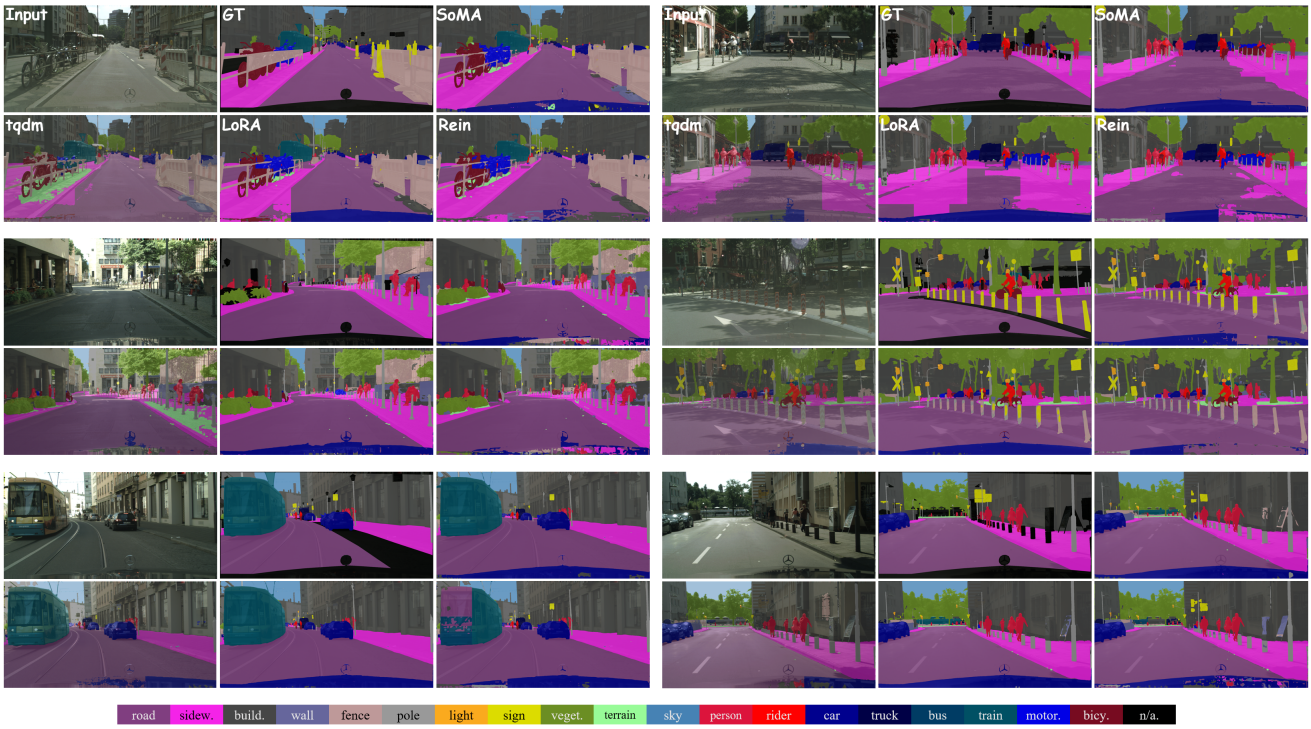

Figure 1. Segmentation results of SoMA on the Cityscapes. The model is trained on GTAV with DINOv2-L backbone.

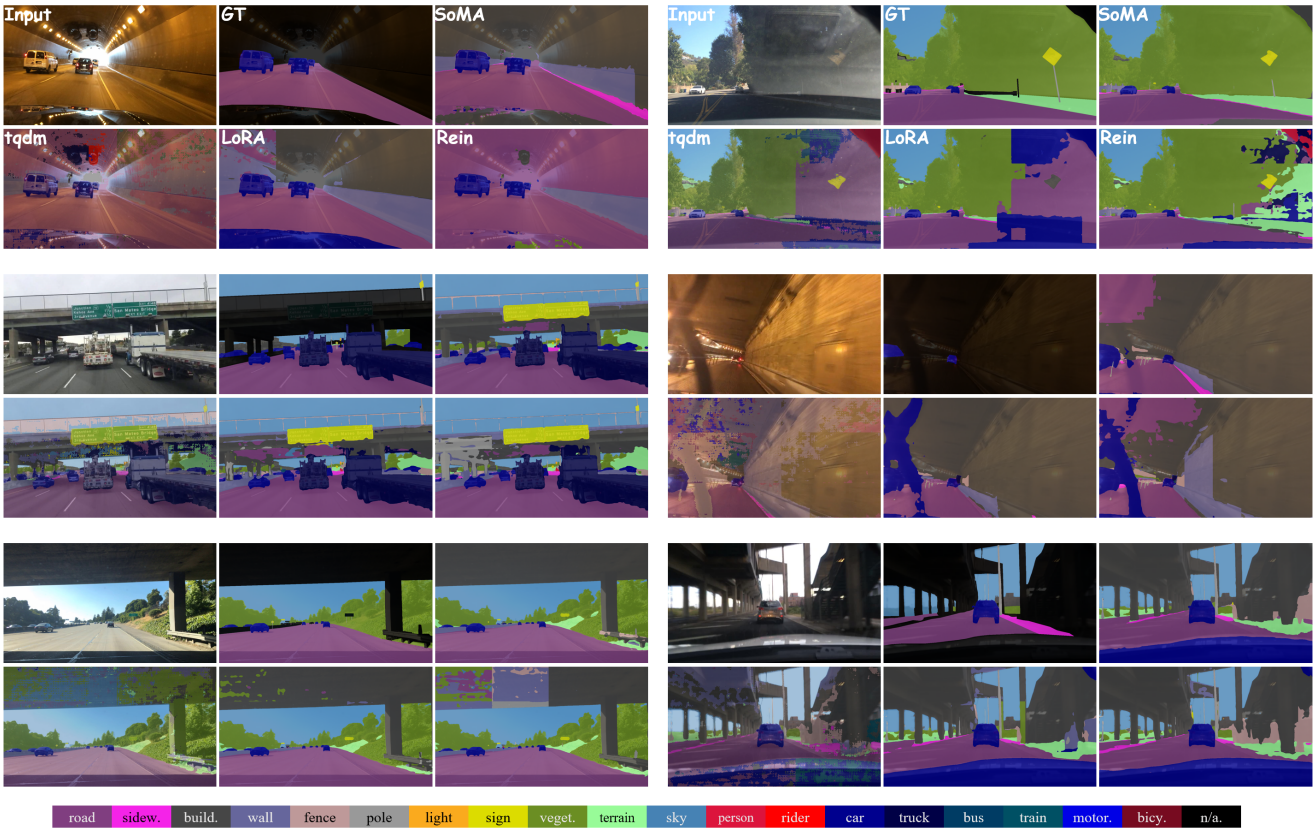

Figure 2. Segmentation results of SoMA on the BDD100k. The model is trained on GTAV with DINOv2-L backbone.

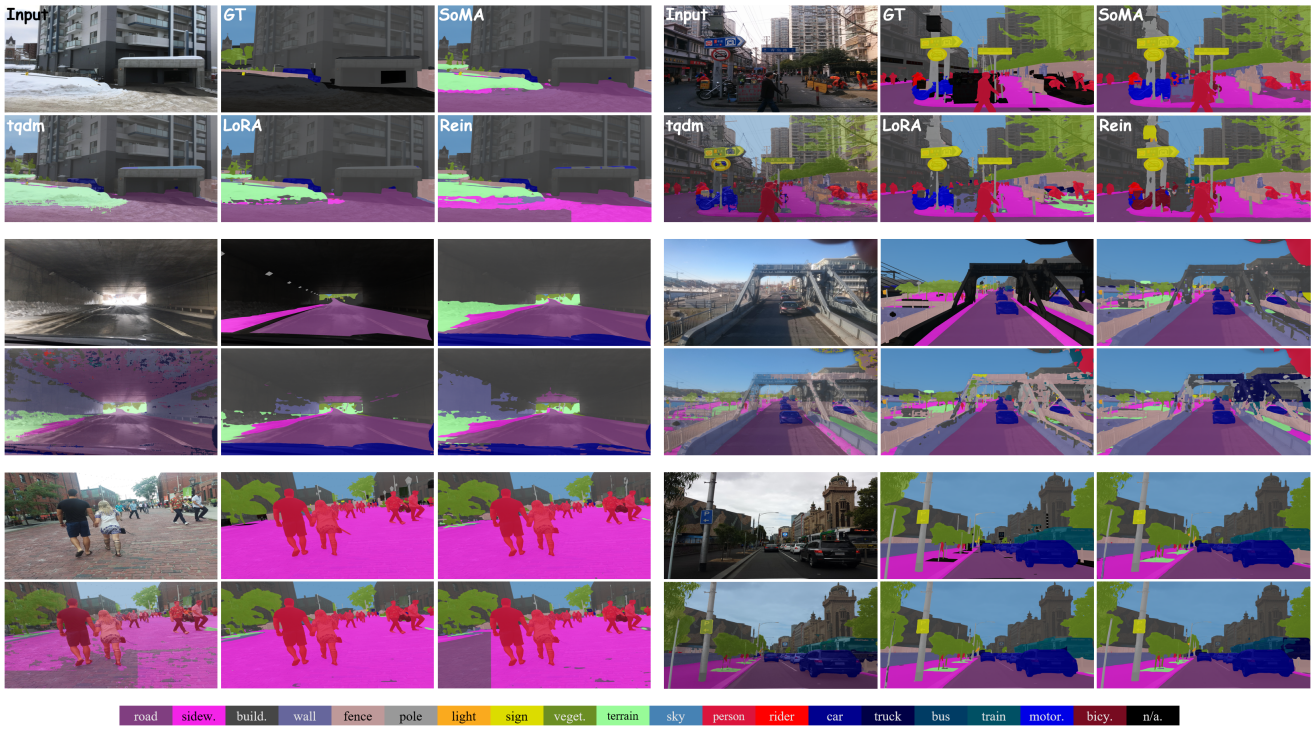

Figure 3. Segmentation results of SoMA on the Mapillary. The model is trained on GTAV with DINOv2-L backbone.

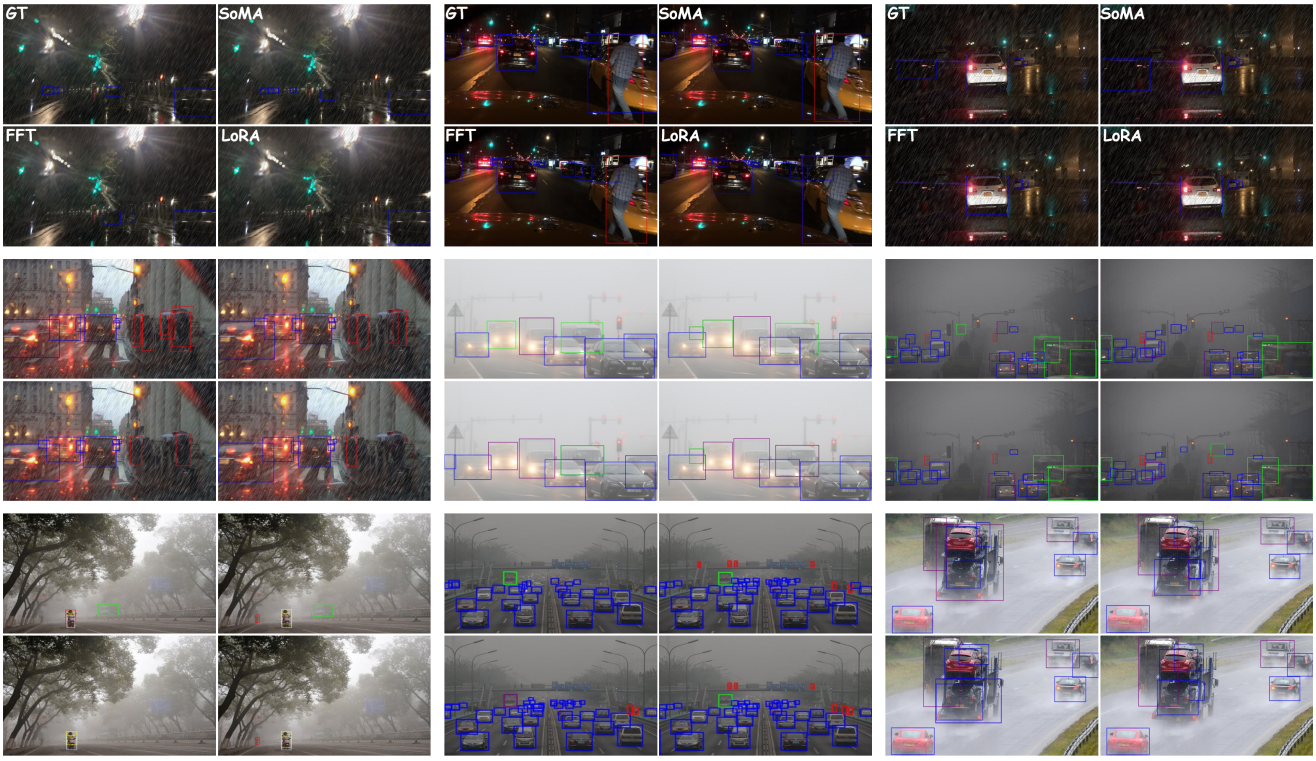

Figure 4. Detection results of SoMA on the adverse scene. The model is trained on Daytime-Sunny with DINOv2-L backbone.

Figure 5. **Multiple subject-consistent synthesis results with prompts describing various domains.** SoMA effectively preserves SDXL's ability to generate images across diverse domains while learning new visual concepts. As a result, simply using prompts from multiple domains allows us to generate an image set of different domains that share the same subject.

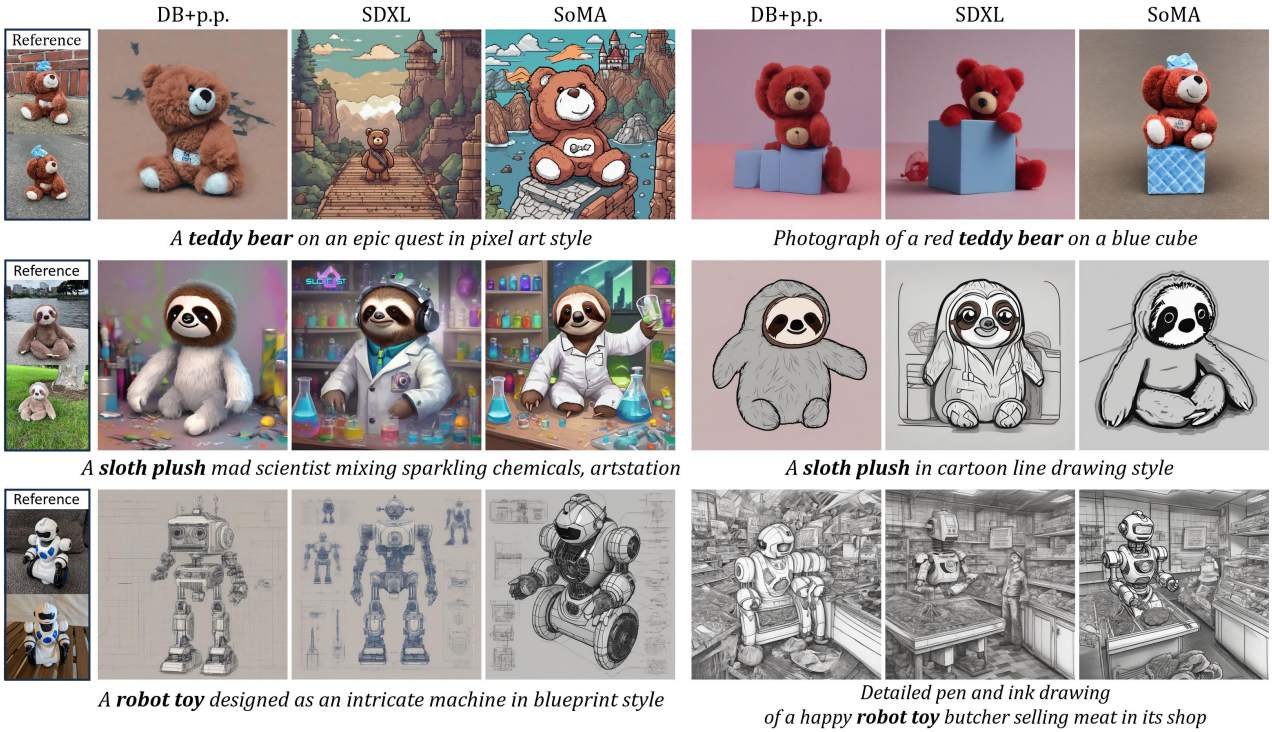

Figure 6. **Qualitative comparison to DreamBooth** [41] **with prior preservation (p.p.) loss.**

tection: Open mmlab detection toolbox and benchmark. *arXiv:1906.07155*, 2019. 1

[7] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. In *NeurIPS*, 2022. 5

[8] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 1, 2, 3

[9] Sungha Choi, Sanghun Jung, Huiwon Yun, Joanne T Kim, Seungryong Kim, and Jaegul Choo. Robustnet: Improving domain generalization in urban-scene segmentation via instance selective whitening. In *CVPR*, 2021. 5

[10] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 1

[11] Muhammad Sohail Danish, Muhammad Haris Khan, Muhammad Akhtar Munir, M Saquib Sarfraz, and Mohsen Ali. Improving single domain-generalized object detection: A focus on diversification and alignment. In *CVPR*, 2024. 1

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 3

[13] Jian Ding, Nan Xue, Gui-Song Xia, Bernt Schiele, and Dengxin Dai. Hgformer: Hierarchical grouping transformer for domain generalized semantic segmentation. In *CVPR*, 2023. 5

[14] Mohammad Fahes, Tuan-Hung Vu, Andrei Bursuc, Patrick Pérez, and Raoul de Charette. A simple recipe for language-guided domain generalized segmentation. In *CVPR*, 2024. 5

[15] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *IVC*, 2024. 3

[16] Yarden Frenkel, Yael Vinker, Ariel Shamir, and Daniel Cohen-Or. Implicit style-content separation using b-lora. In *ECCV*, 2024. 1

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 3

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

[19] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. 3, 5

[20] Jiajun Hu, Jian Zhang, Lei Qi, Yinghuan Shi, and Yang Gao. Learn to preserve and diversify: Parameter-efficient group with orthogonal regularization for domain generalization. In *ECCV*, 2024. 3

[21] Wei Huang, Chang Chen, Yong Li, Jiacheng Li, Cheng Li, Fenglong Song, Youliang Yan, and Zhiwei Xiong. Style projected clustering for domain generalized semantic segmentation. In *CVPR*, 2023. 5

[22] Christoph Hümmer, Manuel Schwonberg, Liangwei Zhong, Hu Cao, Alois Knoll, and Hanno Gottschalk. Vltseg: Simple transfer of clip-based vision-language representations for domain generalized semantic segmentation. In *ACCV*, 2024. 5

[23] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, 2022. 5

[24] Yuru Jia, Lukas Hoyer, Shengyu Huang, Tianfu Wang, Luc Van Gool, Konrad Schindler, and Anton Obukhov. Dginstyle: Domain-generalizable semantic segmentation with image diffusion models and stylized semantic control. In *ECCV*, 2024. 5

[25] Xueying Jiang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Domain generalization via balancing training difficulty and model capability. In *ICCV*, 2023. 5

[26] Jin Kim, Jiyoung Lee, Jungin Park, Dongbo Min, and Kwanghoon Sohn. Pin the memory: Learning to generalize semantic segmentation. In *CVPR*, 2022. 5

[27] Sunghwan Kim, Dae-hwan Kim, and Hoseong Kim. Texture learning domain randomization for domain generalized segmentation. In *ICCV*, 2023. 5

[28] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, 2019. 3

[29] Suhyeon Lee, Hongje Seong, Seongwon Lee, and Euntai Kim. Wildnet: Learning domain generalized semantic segmentation from the wild. In *CVPR*, 2022. 5

[30] Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. In *NeurIPS*, 2022. 5

[31] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. In *ICML*, 2024. 5

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 3

[33] Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. In *NeurIPS*, 2024. 1, 3

[34] Joshua Niemeijer, Manuel Schwonberg, Jan-Aike Termöhlen, Nico M Schmidt, and Tim Fingscheidt. Generalization by adaptation: Diffusion-based domain extension for domain-generalized semantic segmentation. In *WACV*, 2024. 5

[35] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *TMLR*, 2023. 1, 3

[36] Byeonghyun Pak, Byeongju Woo, Sunghwan Kim, Dae-hwan Kim, and Hoseong Kim. Textual query-driven mask transformer for domain generalized segmentation. In *ECCV*, 2024. 5

[37] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *ECCV*, 2018. 5

[38] Duo Peng, Yinjie Lei, Lingqiao Liu, Pingping Zhang, and Jun Liu. Global and local texture randomization for synthetic-to-real semantic segmentation. *TIP*, 2021. 5

[39] Duo Peng, Yinjie Lei, Munawar Hayat, Yulan Guo, and Wen Li. Semantic-aware domain generalized segmentation. In *CVPR*, 2022. 5

[40] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *ICLR*, 2024. 1

[41] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, 2023. 1, 8

[42] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *ICCV*, 2021. 5

[43] Fengyi Shen, Akhil Gurram, Ziyuan Liu, He Wang, and Alois Knoll. Diga: Distil to generalize and then adapt for domain adaptive semantic segmentation. In *CVPR*, 2023. 5

[44] Sumanth Udupa, Prajwal Gurunath, Aniruddh Sikdar, and Suresh Sundaram. Mrfp: Learning generalizable semantic segmentation from sim-2-real with multi-resolution feature perturbation. In *CVPR*, 2024. 5

[45] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 1

[46] 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. 1, 3, 5

[47] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *CVPR*, 2023. 3

[48] Zhenyao Wu, Xinyi Wu, Xiaoping Zhang, Lili Ju, and Song Wang. Siamdoge: Domain generalizable semantic segmentation using siamese network. In *ECCV*, 2022. 5

[49] Qi Xu, Liang Yao, Zhengkai Jiang, Guannan Jiang, Wenqing Chu, Wenhui Han, Wei Zhang, Chengjie Wang, and Ying Tai. Dirl: Domain-invariant representation learning for generalizable semantic segmentation. In *AAAI*, 2022. 5

[50] Jingjun Yi, Qi Bi, Hao Zheng, Haolan Zhan, Wei Ji, Yawen Huang, Yuexiang Li, and Yefeng Zheng. Learning spectral-decomposited tokens for domain generalized semantic segmentation. In *ACMMM*, 2024. 3, 5

[51] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *ICCV*, 2019. 5

[52] Yuyang Zhao, Zhun Zhong, Na Zhao, Nicu Sebe, and Gim Hee Lee. Style-hallucinated dual consistency learning for domain generalized semantic segmentation. In *ECCV*, 2022. 5

[53] Zhun Zhong, Yuyang Zhao, Gim Hee Lee, and Nicu Sebe. Adversarial style augmentation for domain generalized urban-scene segmentation. In *NeurIPS*, 2022. 5

[54] Zhuofan Zong, Guanglu Song, and Yu Liu. Detrs with collaborative hybrid assignments training. In *ICCV*, 2023. 1