SoRA: Singular Value Decomposed Low-Rank Adaptation for Domain Generalizable Representation Learning

Seokju Yun Seunghye Chae Dongheon Lee Youngmin Ro[*]
Machine Intelligence Laboratory, University of Seoul, Korea
{wsz871, tmdhey, dslisleedh, youngmin.ro}@uos.ac.kr
https://github.com/ysj9909/DG-SoRA

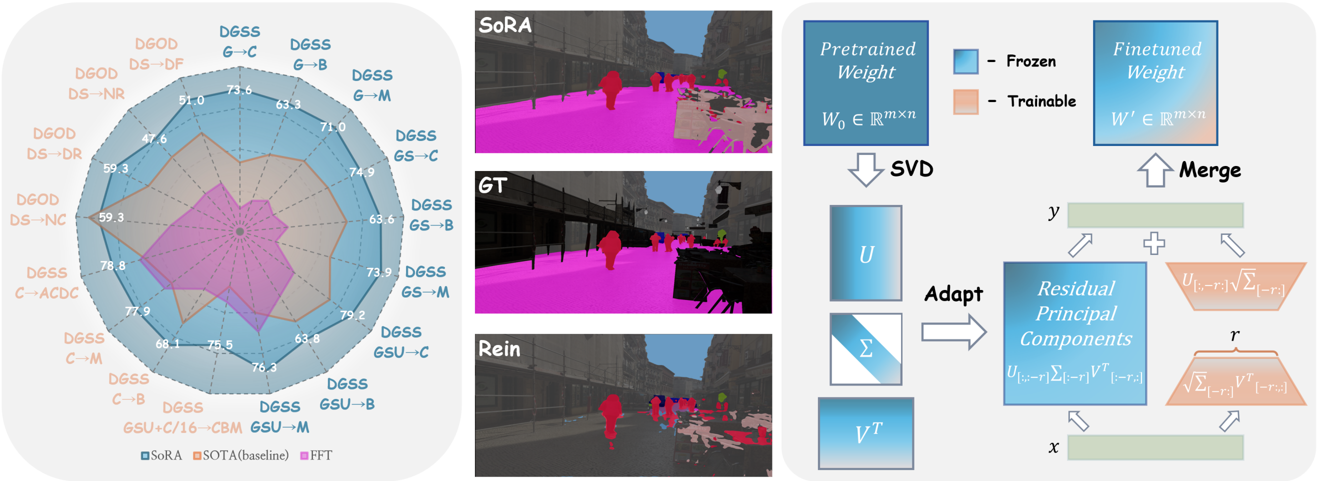


Figure 1. **Overview of SoRA framework.** *Left.* SoRA achieves state-of-the-art results across diverse tasks, ranging from domain-generalized semantic segmentation (DGSS) to object detection (DGOD), and performs well in both synthetic-to-real and real-to-real scenarios. *Middle.* Our method, trained solely on synthetic datasets, demonstrates strong generalization capabilities in complex real-world scenes. *Right.* We present SoRA, a method that applies SVD to the pre-trained weights, decomposing them into r minor singular components $U_{[:,-r:]}\Sigma_{[-r:]}(V^T)_{[-r:,:]}$ and residuals. SoRA selectively tunes the smallest r components, effectively preserving world knowledge of foundation models. Since SoRA shares the same forward architecture as LoRA [33], it adds no extra latency during inference phase.

Abstract

Domain generalization (DG) aims to adapt a model using one or multiple source domains to ensure robust performance in unseen target domains. Recently, Parameter-Efficient Fine-Tuning (PEFT) of foundation models has shown promising results in the context of DG problem. Nevertheless, existing PEFT methods still struggle to strike a balance between preserving generalizable components of the pre-trained model and learning task-specific features. To gain insights into the distribution of generalizable components, we begin by analyzing the pre-trained weights through the lens of singular value decomposition. Building on these insights, we introduce ***S****ingular Value Decomposed* ***L****ow-****R****ank* ***A****daptation (****SoRA****), an approach that selectively tunes minor singular components while keeping the residual parts frozen. SoRA effectively retains the generalization ability of the pre-trained model while efficiently acquiring task-specific skills. Furthermore, we freeze domain-generalizable blocks and employ an annealing weight decay strategy, thereby achieving an optimal balance in the delicate trade-off between generalizability and discriminability. SoRA attains state-of-the-art results on multiple benchmarks that span both domain generalized semantic segmentation to domain generalized object detection. In addition, our methods introduce no additional inference overhead or regularization loss, maintain compatibility with any backbone or head, and are designed to be versatile, allowing easy integration into a wide range of tasks.*

1. Introduction

Deep neural networks (DNNs) have recently made significant strides in dense prediction tasks, including semantic segmentation [13] and object detection [7, 66], both of which are crucial for safety-critical applications, such as autonomous driving. However, DNNs often fail to maintain reliable performance under domain shifts, which may result from diverse lighting conditions, weather changes, or differences in location. Likewise, while synthetic data [27, 67, 69] is employed to avoid the labor-intensive and costly process of constructing real-world datasets, the resulting performance degradation of DNNs in real-world deployment

[*] Corresponding author.

remains a critical issue to be resolved. To tackle these challenges, Domain Generalization (DG) is introduced to design models capable of consistent prediction on arbitrary unseen target domains.

Existing DG methods for dense prediction tasks employ two main strategies to enhance model robustness. The first involves **diversify**ing the training process in the input [10, 62] or feature space [23, 75] using augmentation/perturbation techniques, exposing the model to a broader range of styles to mitigate overfitting to specific domains. The second strategy focuses on imposing **align**ment constraints, such as normalization [63] or regularization losses [1, 14], to facilitate the learning of domain-invariant features. However, these methods mostly utilize outdated backbones (*e.g.*, ResNet [30] or MobileNetV2 [72]) pre-trained on mid-scale datasets, such as ImageNet [18].

Meanwhile, vision foundation models (VFMs) [2], pre-trained on vast, curated datasets with enormous parameter counts, have recently shown unprecedented generalization capabilities [80]. Given this landscape, we propose a paradigm shift—from applying the **diversify & align** approach to classic backbones, to devising methods that **preserve the world knowledge of VFMs while effectively learning task-specific features**. However, naive adaptation of VFMs via *full fine-tuning* (FFT), which involves retraining all model parameters, results in prohibitive costs and a risk of catastrophic forgetting of the pre-trained knowledge.

In light of these challenges, a commonly adopted approach is parameter-efficient fine-tuning (PEFT), where only the injected lightweight adapters [12, 28, 33] or tokens [37, 49] are fine-tuned, while the pre-trained weights remain frozen. Low-Rank Adaptation (LoRA) [33] is one of the most widely used PEFT methods, which substitutes the updates with the product of two low-rank matrices. Interestingly, we found that LoRA exhibits superior generalization ability in DG dense prediction tasks compared to FFT and other PEFT methods (see Tab. 2), aligning with recent findings in LoRA-based literature [6, 34]. Nevertheless, LoRA adapters are built without considering the distribution of generalizable components within the pre-trained weights they modify. As a result, there is still room for enhancing generalizability by preserving the diverse domain knowledge of VFMs.

To uncover the structure in which world knowledge is encoded within the pre-trained weights, we first perform singular value decomposition (SVD) on the learned weight matrices and then analyze the behavior of VFMs after selectively removing specific *singular components*[1]. Our analysis reveals that singular vectors associated with higher singular values tend to extract general features spanning multiple classes in ImageNet, while minor singular components are primarily responsible for capturing context-specific features. Capitalizing on our findings, we introduce **S**ingular Value Decomposed L**o**w-**R**ank **A**daptation (**SoRA**), which starts by performing SVD to decompose the weights into their singular vectors, *i.e.,* $W = U\Sigma V^T$, then fine-tunes only the minor singular components. Specifically, we initialize LoRA adapter using the components with the smallest r singular values, *i.e.,* $U_{[:,-r:]}\Sigma_{[-r:]}(V^T)_{[-r:,:]}$, and the remaining residual components are frozen to maintain generalizability, as illustrated in Fig. 1 *right*.

By extending our analysis from the weight-level to the block-level, we observe and empirically demonstrate that the early blocks of VFMs effectively extract well-localized semantic features and are less affected by style or input-level domain shifts (see Fig. 3). Therefore, we freeze these blocks to preserve generalizable components while reducing the number of trainable parameters. Although our proposed methods are effective, tuning the less-optimized bottom spectral space and freezing early blocks may lead to a lack of discriminability. To mitigate this issue, we introduce an annealing weight decay scheme that gradually reduces the regularization loss incurred by weight decay over the course of training. As shown in Fig. 1, comprehensive evaluations on DG benchmarks demonstrate that our framework outperforms previous state-of-the-art baselines and FFT, training only 0.58% to 1.6% of the parameters. Furthermore, SoRA demonstrates impressive scaling performance with both data and model size.
We make the following contributions:

- We underscore the critical importance of preserving pre-trained knowledge while learning task-specific features in scenarios that involve the use of VFMs.
- We present SoRA, a novel PEFT method specifically tailored for DG problem, which leverages spectral information to retain the generalization capacity of VFMs. In addition, we identify and freeze domain-generalizable blocks while employing an annealing weight decay strategy, thereby achieving an optimal balance between generalizability and discriminability.
- Extensive experiments on various DG for semantic segmentation and object detection tasks reveal that our framework surpasses existing baselines by a substantial margin. Notably, SoRA achieves an mIoU of **80.4%** on *Cityscapes* without accessing real-world datasets, while training only **0.58%** of the parameters. Additionally, under the most challenging *daytime-sunny* → *night-rainy* detection setting, SoRA outperforms the prior SOTA implementation by achieving a **+23.5%** mAP gain.

2. Related Work

Domain generalization for dense predictions. Domain Generalization (DG) methods for dense predictions [5, 9, 19, 35, 39, 42, 46, 50, 61, 73, 79, 82, 85] have recently gar-

[1] In this paper, we refer to the combination of a singular value and its corresponding singular vectors as a *singular component*.

nered considerable attention due to their practical demands. These methods can be categorized along two main axes: i) data augmentation or domain randomization, both of which diversify the training process, and ii) alignment techniques to suppress domain-relevant features.

Methods adopting the first approach at the input level employ various augmentation techniques [10, 41, 62, 91] or, more recently, leverage advances in generative modeling, such as diffusion models [64, 68], to generate data approximating the target domain [3, 38, 57], thereby expanding domain diversity and enriching learned representations. To enable more flexible augmentation, several studies [22, 23, 45, 75, 77] leverage feature stylization to enhance model robustness. On the other hand, alignment-based methods effectively reduce domain-sensitive components by employing feature normalization/whitening techniques [60, 63, 74], or by introducing regularization losses to suppress inconsistencies caused by simulated domain shifts [1, 14, 17, 40, 43, 47, 53, 83, 84, 89, 90]. However, most of the aforementioned approaches experimented with CNN backbones [30] trained on relatively restricted domains. In contrast, when applying vision foundation models (VFMs)—trained on massive datasets using transformer architectures [20, 76] and sophisticated training schemes [58]—to the DG problem, a fundamental shift in approach is imperative.

Following this trend, recent studies leverage vision-language models as feature extractors [36], employing text embeddings with domain-invariant semantics as a source for style augmentation [22, 77] or as object queries in transformer-based decoders [59]. Rein [80] injects learnable adapters and tokens to refine feature maps for each instance while keeping the backbone frozen, significantly expanding the performance frontier. FADA [4] and SET [86] introduce frequency-adapted methods to effectively utilize frozen VFM features. In contrast to existing methods that simply freeze all parameters, we perform singular value decomposition on the pre-trained weights, freezing the generalizable principal components and tuning only the minor singular components responsible for context-specific features, thereby achieving superior DG performance.

Vision foundation models (VFMs). Recently, the advent of VFMs with general perception capabilities has laid the foundation for a paradigm in which these models are broadly applicable to numerous downstream tasks. Seminal examples of models advancing this direction include CLIP [65], which demonstrates strong zero-shot generalizability through training on web-scale, weakly supervised image-text pairs; EVA02 [24], which enhances CLIP features with masked image modeling [31]; and DINOv2 [58], which incorporates losses from prior arts like iBoT [92] and DINO [8], and is trained on massive curated datasets, offering strong spatial features for dense prediction tasks. Our SoRA builds on this foundation, harnessing the full potential of VFMs in the context of DG problem.

Parameter-efficient fine-tuning (PEFT). With the remarkable success of PEFT methods in efficiently adapting large-scale foundation models in the realm of natural language processing, there has been a growing interest in extending these approaches to the field of computer vision. For example, VPT [37] prepends prompt tokens to the input sequence of several attention layers. AdaptFormer [12] introduces a lightweight adapter in parallel to the feed-forward network, whereas SSF [51] incorporates scale and shift parameters to modulate features for downstream tasks. LoRA [33] models the incremental updates of pre-trained weights using rank decomposition matrices, achieving performance comparable to full fine-tuning (FFT). However, there still often exists a performance gap between LoRA and FFT. DoRA [52] and PiSSA [55] address this gap by further decomposing the pre-trained weights, initializing the LoRA adapter with directional and principal components, respectively, and fine-tuning accordingly. While these methods aim to match the performance of FFT, our focus is on the preservation of generalizable components.

3. Preliminaries

Low-Rank Adaptation (LoRA). LoRA [33] hypothesizes that the weight changes during fine-tuning exhibit a low-rank structure. Given a pre-trained weight matrix $W_0 \in \mathbb{R}^{m\times n}$, LoRA constrains its update $\Delta W \in \mathbb{R}^{m\times n}$ to a low-rank decomposition $\Delta W = BA$, where $B \in \mathbb{R}^{m\times r}$ and $A \in \mathbb{R}^{r\times n}$ are two low-rank matrices, with an intrinsic rank of $r \ll \min(m, n)$. For $y = W_0 x$, the modified forward pass is as follows:

$$y = (W_0 + \Delta W)x = (W_0 + \underline{BA})x, \tag{1}$$

where W_0 is frozen, and only the underlined low-rank parameters are fine-tuned, significantly reducing the number of trainable parameters. LoRA uses a uniform Kaiming distribution [29] to initialize A, while B is initially set to zero, ensuring $BA = 0$ at the beginning of training. As seen in Eq. 1, the learned matrices BA can be merged with the frozen weight W_0, allowing LoRA-based variants to avoid adding any additional inference burden.

Singular Value Decomposition (SVD) Analysis. We observe that LoRA and existing PEFT methods, by training only a relatively small subset of parameters or by enforcing low-rank-ness of weight updates, deliver promising DG performance without significantly distorting the pre-trained representations. However, many previous studies build adapters without considering how the generalizable components are structured within the weight matrices, potentially interfering with these components in the fine-tuning process. Building on the Eckart–Young–Mirsky theorem [21], which validates that the optimal rank-r approx-

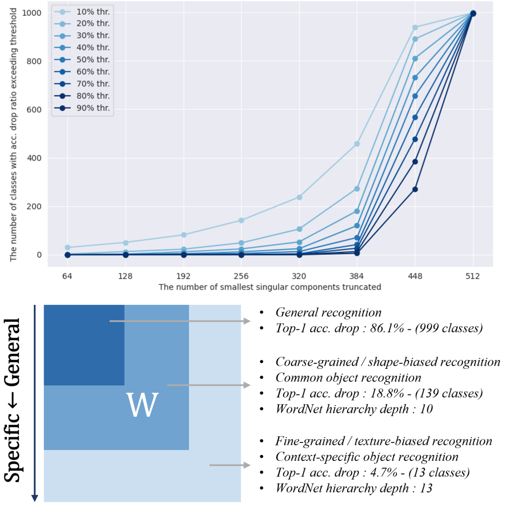


Figure 2. **Distribution of generalizable components.** *Top.* Number of classes exhibiting specific accuracy drops after applying SVD to DINOv2-large weights and reconstructing by truncating the smallest r singular components. *Bottom.* Distinct roles of singular components across levels. Numbers in parentheses represent the count of classes with an accuracy drop ratio exceeding 50%. The average WordNet hierarchy depth of these classes is shown, with higher values indicating greater context specificity.

imation of a matrix W is represented by the sum of its top r singular components, we analyze how the world knowledge of VFMs is structured from an SVD perspective. Using ImageNet-1k [18] as a testbed, our analysis computes the SVD of pre-trained weights across all layers of the DINOv2 [58] ViT-large model, followed by truncating specific singular values and their corresponding singular vectors to investigate the classes that are subsequently misclassified.

As shown in Fig. 2 *top*, as the number of discarded smallest singular components increases, the number of misclassified classes grows exponentially rather than linearly, suggesting that components with larger singular values tend to capture more general features spanning multiple classes. To further examine whether different groups of singular components exhibit distinct behaviors, we consider three groups: the top 8 principal components $U_{[:,:8]}\Sigma_{[:8]}(V^T)_{[:8,:]}$, the middle 160 singular components $U_{[:,384:544]}\Sigma_{[384:544]}(V^T)_{[384:544,:]}$, and the bottom 320 minor components $U_{[:,-320:]}\Sigma_{[-320:]}(V^T)_{[-320:,:]}$. We then truncate each group separately and analyze the classes that the rank-reduced model subsequently fails to classify (see Fig. 2 *bottom*).

According to the Eckart–Young–Mirsky theorem and the heavy-tail distribution of singular values, it is evident that the top singular components encapsulate the core knowledge of VFM and should be preserved to maintain generalizability. Conversely, we note that the bottom-component-truncated model fails to recognize classes that require fine-grained or texture-biased features (e.g., specific dog breeds, wildlife with distinctive textures), as well as those that are context-specific (e.g., cornet, bikini, missile, dock). In contrast, we find that removing middle singular components results in significant performance declines for classes that rely on coarse-grained or shape-biased recognition (e.g., mortarboard, pier, altar, necklace, schooner, horizontal bar). Interestingly, as suggested by differences in the depth of the WordNet hierarchy, higher-value singular components do not function independently but rather interact in a hierarchical and composite manner. For instance, classes with notable accuracy drops can also be clustered into higher-level concepts, such as *functionality* (e.g., wok, hotpot, pitcher, cauldron, pop bottle, cup/ambulance, minivan, passenger car, police van, garbage truck, limousine) or *co-occurrence* (e.g., desktop computer, keyboard, notebook, desk). Additionally, almost all of these objects are commonly found in everyday scenes. *Consequently, it can be seen that generalizable components are hierarchically and compositionally structured according to the magnitude of singular values.*

4. Proposed Method

In this section, we present our parameter-efficient adaptation methods specifically tailored for DG dense prediction tasks, inspired by the insights from our SVD analysis. Subsequently, we propose a block freeze strategy and an annealing weight decay scheme to strike an optimal trade-off between generalizability and discriminability.

4.1. Singular Value Decomposed Low-Rank Adaptation

To effectively **preserve** the integrity of generalizable representations in pre-trained weights, we propose Singular value decomposed Low-Rank Adaptation (SoRA). Firstly, SoRA performs singular value decomposition (SVD) on the pre-trained weight matrices, including those from the attention layers, multi-layer perceptron (MLP) layers, and more. The weight matrix $W \in \mathbb{R}^{m\times n}$ is decomposed using SVD as follows:

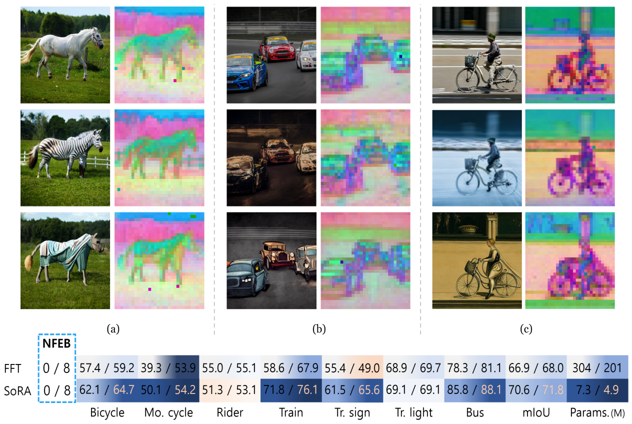
$$W = U\Sigma V^T = \sum_{i=1}^{R} \sigma_i u_i v_i^T, \tag{2}$$

where $U = [u_1, u_2, \cdots, u_m] \in \mathbb{R}^{m\times m}$ and $V = [v_1, v_2, \cdots, v_n] \in \mathbb{R}^{n\times n}$ are the left and right singular vectors, whose columns form an orthonormal basis for $\mathbb{R}^m$ and $\mathbb{R}^n$ respectively, $\Sigma \in \mathbb{R}^{m\times n}$ is a diagonal matrix whose entries σ_i are singular values arranged in descending order, and R denotes the rank of W, with $R \leq \min(m, n)$.

We then divide the singular components into two groups according to their singular values: the minor singular components with the smallest r singular values $U_{[:,-r:]}\Sigma_{[-r:]}(V^T)_{[-r:,:]}$ and the residual singular components $U_{[:,:-r]}\Sigma_{[:-r]}(V^T)_{[:-r,:]}$. As illustrated in Fig. 1 *right*, we employ the QR-type reconstruction of minor singular components to initialize the adapter for fine-tuning:

$$B = U_{[:,-r:]}\sqrt{\Sigma}_{[-r:]} \in \mathbb{R}^{m\times r}, \tag{3}$$

$$A = \sqrt{\Sigma}_{[-r:]}(V^T)_{[-r:,:]} \in \mathbb{R}^{r\times n}. \tag{4}$$



	NFEB	Bicycle	Mo. cycle	Rider	Train	Tr. sign	Tr. light	Bus	mIoU	Params.(M)
FFT	0 / 8	57.4 / 59.2	39.3 / 53.9	55.0 / 55.1	58.6 / 67.9	55.4 / 49.0	68.9 / 69.7	78.3 / 81.1	66.9 / 68.0	304 / 201
SoRA	0 / 8	62.1 / 64.7	50.1 / 54.2	51.3 / 53.1	71.8 / 76.1	61.5 / 65.6	69.1 / 69.1	85.8 / 88.1	70.6 / 71.8	7.3 / 4.9

Figure 3. **The inherent generalization capabilities of the early blocks of VFM.** *Top.* We apply PCA on the extracted intermediate features (8th block of DINOv2-Large) and visualize the top three leading components. *Bottom.* Class-wise IoU comparison for rare classes [32] under the *GTAV→Cityscapes* setting, focusing on the impact of freezing the first eight blocks. "NFEB" stands for the Number of Frozen Early Blocks. A darker shade of blue indicates enhanced generalization performance relative to the baseline.

In the fine-tuning process, the frozen residual matrix W_{res} is defined as $W - BA$ to prevent numerical errors introduced during the SVD step, while maintaining the pre-trained generalization capability at the beginning of fine-tuning. After fine-tuning, the learned low-rank matrices B' and A' can be merged into W_{res}, as in LoRA [33], ensuring no additional computation overhead during inference:

$$y = (W \underbrace{-BA + B'A'}_{\Delta W_{SoRA}})x = (W_{res} + B'A')x = W'x. \quad (5)$$

Importantly, SoRA offers several key advantages for tackling the DG problem: i) SoRA initializes the adapter using components orthogonal to the principal components, thus minimizing interference with pre-trained representations. For the batched version of the forward pass described in Eq. 1 (with $X \in \mathbb{R}^{n\times b}$ and $Y \in \mathbb{R}^{m\times b}$), the gradients of B and A are $\frac{\partial \mathcal{L}}{\partial B} = \frac{\partial \mathcal{L}}{\partial Y} X^T A^T$ and $\frac{\partial \mathcal{L}}{\partial A} = B^T \frac{\partial \mathcal{L}}{\partial Y} X^T$. Therefore, unlike LoRA, which randomly initializes A, our approach facilitates convergence along singular directions that preserve the integrity of the generalizable components. ii) Consistent with prior DG methods [14, 43, 45, 91] that guide models to prioritize shape over texture, thereby achieving superior out-of-distribution performance, SoRA effectively preserves the domain generalizable shape-biased features of VFMs by tuning only the minor singular components (see Fig. 2 *bottom* and Tab. 1).

Discussion. To assess whether the proposed adaptation matrix ΔW_{SoRA} minimally correlates with the top singular directions of the pre-trained weight matrix W_0, we project ΔW onto the singular vectors of W_0. Specifically, we quantify the extent of interference by ΔW on each i-th singular direction through a *singular modulation ratio* (SMR), which is formally defined as follows:

$$\mathrm{SMR}_i = \left|\frac{u_i^{\mathsf{T}} \Delta W v_i}{\sigma_i}\right|. \quad (6)$$

SMR	Block	i	ΔW_{SoRA}	ΔW^*_{SoRA}	ΔW_{LoRA}
$\lvert\frac{u_i^{\mathsf{T}} \Delta W v_i}{\sigma_i}\rvert$	12th	0–255	0.075	0.084 (↓ 17%)	0.101
		256–511	0.094	0.094 (↓ 43%)	0.166
		512–767	0.157	0.165 (↓ 40%)	0.275
		768–1023	4.104	6.152 (↑ 21%)	5.095
	24th	0–255	0.097	0.095 (↓ 24%)	0.125
		256–511	0.105	0.108 (↓ 35%)	0.166
		512–767	0.206	0.189 (↓ 41%)	0.323
		768–1023	2.781	3.620 (↑ 26%)	2.865

Table 1. **Singular modulation ratio comparison.** * indicates the adaptation matrix trained using the annealing weight decay strategy. The weight matrices are taken from the query projection layers of the 12th and 24th attention blocks in DINOv2-Large [58].

For ease of analysis, we partition the 1024 singular directions into four equal groups, calculate the average SMR for each group, and then compare the resulting values between SoRA and LoRA. As shown in Tab. 1, the SoRA initialization scheme proves more effective than LoRA in minimizing interference with singular components with higher singular values. In summary, our method effectively preserves the structure of the VFM's generalizable representations while adapting to downstream tasks, resulting in significant improvements on DG benchmarks compared to SOTA baselines, as confirmed by our experimental results.

4.2. Freezing Early Blocks

In this section, we explore the distribution of generalizable components at the block level. Drawing on recent domain adaptation methods [48, 88], which find that the early blocks of a model primarily engage in domain-specific feature extraction and should therefore be updated, we conjecture that in the context of leveraging VFM for DG, the early blocks should instead remain unchanged. Intuitively, these initial blocks of VFMs are adept at projecting low-level features from diverse domains into domain-invariant semantic representations. In other words, while **adjusting early blocks** may help bridge input-level domain gaps in domain adaptation, **freezing the early blocks of VFMs** is essential for DG to prevent overfitting to the source domain and to retain the model's generalization capability.

To validate our hypothesis, we perform PCA analysis on the features extracted from the early blocks of VFM and compare the performance on rare classes when these blocks are frozen, as illustrated in Fig. 3. Despite considerable domain shifts in style and content, the principal components align well with the image layout and maintain consistent semantic information (e.g., the horses have similar colors). Furthermore, simply freezing the early blocks yields a notable improvement in IoU for rare classes, indicating that updating these blocks can severely disrupt the learned representations of various classes in VFMs. Thus, by default, we opt to freeze all blocks up to the one that generates the feature map fed as the highest resolution input to the segmentation/detection head (e.g., the first eight blocks). Detailed ablations of varying the number of frozen early blocks are presented in the *Supplemental*.

4.3. Annealing Weight Decay

Our methods are specialized in preserving the pre-trained knowledge of VFMs, but this may come at the cost of diminished discriminability. One can remove weight decay or set its coefficient to a very small value to enhance discriminability. However, doing so risks increased interference of the tuned parameters with the principal components of the pre-trained weights. Therefore, based on the observation that regularization primarily affects the early phases of learning by guiding a model towards having good generalization properties [26], we leverage an Annealing Weight Decay (AWD) scheme. Specifically, AWD starts with a relatively large weight decay coefficient and progressively reduces it to zero as training progresses. By default, we use a cosine schedule for AWD. As shown in Tab. 1 (ΔW_{SoRA} *vs.* ΔW^{*}_{SoRA}) and Tab. 8, AWD integrates seamlessly with SoRA, improving discriminability *without compromising SoRA's ability to preserve pre-trained knowledge*.

5. Experiments

We conduct a variety of experiments to showcase the efficacy of SoRA on DG benchmarks including domain generalized semantic segmentation (DGSS) and domain generalized object detection (DGOD). We then perform an ablation study on SoRA, emphasizing the contribution of each component to the DG performance. Lastly, we extend our approach beyond dense predictions to generative modeling.

5.1. Experimental Setup

Datasets. For DGSS, we use three synthetic datasets (GTAV [67], SYNTHIA [69], UrbanSyn [27]), and four real-world datasets (Cityscapes [16], BDD100K [87], Mapillary [56], ACDC [71]). Specifically, GTAV, a large-scale dataset generated from the game engine, comprises 24,966 images. SYNTHIA consists of 9,400 photo-realistic synthetic images. UrbanSyn provides 7,539 driving scene images. Cityscapes is an autonomous driving dataset containing 2,975 training and 500 validation images (2048×1024). BDD and Mapillary have 1,000 (1280×720) and 2,000 (1920×1080) validation images, respectively. ACDC is a driving scene dataset under adverse conditions, including night, snow, fog, and rain. For DGOD, we use the urban scene detection dataset introduced by [82]. The dataset consists of five different weather conditions: Daytime-Sunny (DS), Night-Clear (NC), Night-Rainy (NR), Dusk-Rainy (DR), and Daytime-Foggy (DF). Daytime-Sunny serves as the source domain, offering 26,518 images, with 8,313 used for in-domain evaluation, while the remaining adverse weather conditions are employed as unseen target domains.

Synthetic-to-Real Generalization			Test Domains (mIoU in %)			
Methods	Backbone	Params.*	→*Citys.*	→*BDD*	→*Map.*	Avg.
Single-source DGSS Trained on GTAV						
○ CLOUDS [3]	CLIP-CN-L	0.0M	60.20	57.40	67.00	61.50
○ VLTSeg [36]	EVA02-L	304.2M	65.30	58.30	66.00	63.20
○ Rein [80]	EVA02-L	3.0M	65.30	60.50	64.90	63.60
○ FADA [4]	EVA02-L	11.7M	66.70	61.90	66.10	64.90
○ tqdm [59]	EVA02-L	304.2M	68.88	59.18	70.10	66.05
○ SoRA (Ours)	EVA02-L	5.1M	68.05	60.81	68.33	65.73
● SoRA (Ours)	EVA02-L	5.1M	69.94	62.48	68.33	66.92
○ DoRA [52]	DINOv2-L	7.5M	66.12	59.31	67.07	64.17
○ VPT [37]	DINOv2-L	3.7M	68.75	58.64	68.32	65.24
○ SET [86]	DINOv2-L	6.1M	68.06	61.64	67.68	65.79
○ FADA [4]	DINOv2-L	11.7M	68.23	61.94	68.09	66.09
○ AdaptFormer [12]	DINOv2-L	6.3M	70.10	59.81	68.77	66.23
○ SSF [51]	DINOv2-L	0.5M	68.97	61.30	68.77	66.35
○ LoRA [33]	DINOv2-L	7.3M	70.13	60.13	70.42	66.89
● Rein† [80]	DINOv2-L	3.0M	70.68	62.51	69.61	67.60
○ SoRA (Ours)	DINOv2-L	4.9M	71.82	61.31	**71.67**	68.27
● SoRA (Ours)	DINOv2-L	4.9M	**73.63**	**63.33**	70.98	**69.31**
Multi-source DGSS Trained on GTAV + SYNTHIA						
○ Rein† [80]	DINOv2-L	3.0M	72.17	61.53	70.69	68.13
○ SoRA (Ours)	DINOv2-L	4.9M	73.16	61.90	72.73	69.26
● SoRA (Ours)	DINOv2-L	4.9M	**74.85**	**63.59**	**73.92**	**70.79**
Multi-source DGSS Trained on GTAV + SYNTHIA + UrbanSyn						
○ FFT	DINOv2-L	304.2M	75.90	60.93	72.80	69.88
○ SoRA (Ours)	DINOv2-L	4.9M	77.33	62.78	74.93	71.68
○ FFT‡	DINOv2-L	307.3M	77.06	61.81	75.09	71.32
○ Rein‡ [80]	DINOv2-L	3.0M	78.42	62.20	74.49	71.70
○ SoRA‡ (Ours)	DINOv2-L	4.9M	79.22	63.84	**76.30**	73.12
○ Freeze	DINOv2-G	0.0M	76.08	61.98	72.23	70.10
○ FFT	DINOv2-G	1.1B	76.90	61.69	73.53	70.71
○ SoRA (Ours)	DINOv2-G	6.6M	78.39	63.75	75.16	72.43
● SoRA (Ours)	DINOv2-G	6.6M	**80.37**	**65.67**	76.18	**74.07**

Table 2. Comparison of the proposed SoRA with existing DGSS ○ and PEFT ○ methods under various **synthetic-to-real settings**.

Implementation details. Since our method focuses on preserving pre-trained knowledge, we mainly evaluate its effectiveness using VFMs like EVA02 [24] and DINOv2 [58] as the backbone. To further elevate performance, we leverage state-of-the-art decode heads such as Mask2Former [13] and Co-DETR [93]. Extensive experiments with various backbones and heads are provided in the *Supplemental*. By default, SoRA is applied to all linear layers within the self-attention and MLP, with a rank of 16. Detailed training settings are presented in the *Supplemental*.

5.2. Comparison with State-of-the-Arts

We compare SoRA with previous SOTA baselines, achieving the highest performance across all DG scenarios *without adding inference latency or regularization loss*. In our tables, $*$ indicates trainable parameters in the backbones. Marker ● refers to models tested with multi-scale flip augmentation. † represents re-implemented test results using official checkpoints for a fair comparison, and ‡ denotes

Real-to-Real Generalization			Test Domains (mIoU in %)		
Methods	Backbone	Params.*	→*BDD*	→*Map.*	Avg.
Single-source DGSS Trained on Cityscapes					
○ HGFormer [19]	Swin-L	196.0M	61.50	72.10	66.80
○ CMFormer [5]	Swin-L	196.0M	62.60	73.60	68.10
○ tqdm [59]	EVA02-L	304.2M	64.72	76.15	70.44
○ FADA [4]	DINOv2-L	11.7M	65.12	75.86	70.49
○ Rein† [80]	DINOv2-L	3.0M	66.53	75.18	70.86
○ SoRA (Ours)	DINOv2-L	4.9M	67.02	76.45	71.74
● SoRA (Ours)	DINOv2-L	4.9M	**68.08**	**77.87**	**72.98**

Table 3. **Real-to-real DGSS comparison.**

Clear-to-Adverse Weather	ACDC [71] Test Domains (mIoU in %)				
Methods	→*Night*	→*Snow*	→*Fog*	→*Rain*	All
Single-source DGSS Trained on Cityscapes					
○ HGFormer [19]	52.7	68.6	69.9	72.0	67.2
○ SoRA (Ours)	61.7	77.3	74.7	77.8	74.4
○ VLTSeg‡ [36]	-	-	-	-	77.9
○ Rein‡ [80]	70.6	79.5	76.4	79.4	77.6
○ FFT‡	68.9	**80.3**	**76.9**	79.7	77.7
○ SoRA‡ (Ours)	**73.2**	79.8	76.8	**80.2**	**78.8**

Table 4. Results on **Cityscapes** → **ACDC <u>test</u> set.**

Data Efficiency			Test Domains (mIoU in %)			
Methods	Backbone	Params.*	→*Citys.*	→*BDD*	→*Map.*	Avg.
DGSS Pre-trained on GTAV + SYNTHIA + UrbanSyn → $\frac{1}{16}$ *of Cityscapes*						
○ FFT‡	DINOv2-L	307.3M	81.53	65.22	75.73	74.16
○ Rein‡ [80]	DINOv2-L	3.0M	**82.58**	64.76	73.73	73.69
○ SoRA‡ (Ours)	DINOv2-L	4.9M	82.50	**66.99**	**77.02**	**75.50**

Table 5. **Real-world data efficiency.**

training on images cropped to 1024×1024; otherwise, the default is 512×512. A comprehensive comparison with earlier works can be found in the *Supplemental*.

Synthetic-to-real DGSS. In Tab. 2, we adopt *GTAV* → {*Citys.*, *BDD*, *Map.*} as the basic setting, and demonstrate scalability of SoRA by incorporating additional synthetic datasets into the training process. We report the mean Intersection over Union (mIoU) and benchmark our results against SOTA methods. SoRA consistently outperforms VFM-based DGSS methods and PEFT approaches using both vision-language (EVA02) and vision-only (DINOv2) VFM backbones. Notably, SoRA achieves superior results over LoRA by minimally interfering with the hierarchical structure of VFM's pre-trained knowledge (see Tab. 1). Moreover, SoRA exhibits superior scalability across data, model, and input size, with improvements becoming more pronounced when test-time augmentation is applied.

Real-to-real DGSS. In Tables 3 and 4, we conduct experiments under the *Citys.* → {*BDD*, *Map.*} / *ACDC* settings. When combined with DINOv2-Large backbone, SoRA consistently gains the best results across all settings. This highlights the effectiveness of our method as a strong solution for real-world deployment scenarios, where domain shifts caused by geographic or weather variations are common.

Clear-to-Adverse Weather		S-DGOD [82] Test Domains (mAP@0.5 in %)					
Methods	Params.*	DS	→*NC*	→*DR*	→*NR*	→*DF*	Avg.
Single-source DGOD Trained on Daytime-Sunny (DS)							
Backbone : ResNet101 [30] / Head : Faster R-CNN [66]							
S-DGOD [82]	42.3M	56.1	36.6	28.2	16.6	33.5	28.7
CLIP-Gap [77]	42.3M	51.3	36.9	32.3	18.7	38.5	31.6
OA-DG [47]	42.3M	55.8	38.0	33.9	16.8	38.3	31.8
PDOC [50]	42.3M	53.6	38.5	33.7	19.2	39.1	32.6
UFR [53]	42.3M	58.6	40.8	33.2	19.2	**39.6**	33.2
DivAlign [17]	42.3M	52.8	**42.5**	**38.1**	24.1	37.2	35.5
SoRA (Ours)	3.1M	49.3	41.9	37.9	**24.5**	38.2	**35.6**
Backbone : DINOv2-L [58] / Head : Co-DETR [93]							
Freeze	0.0M	65.0	54.2	55.0	42.8	46.9	49.7
FFT	307.3M	68.2	57.1	56.6	43.1	47.2	51.0
DoRA [52]	5.8M	69.0	58.7	58.0	45.0	48.9	52.7
AdaptFormer [12]	6.3M	68.9	58.8	58.3	44.4	49.8	52.8
LoRA [33]	5.5M	69.6	**59.6**	58.1	46.1	49.5	53.3
SoRA (Ours)	4.9M	69.4	59.3	**59.3**	**47.6**	**51.0**	**54.3**

Table 6. **Domain generalized object detection.**

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
SoRA $_{\text{large}}$	**9.0**	12.7G	**56.4 / 79.7**	**4.4G / 40.9G**
FFT $_{\text{giant}}$	27.9	45.3G	21.6 / -	10.6G / OOM
SoRA $_{\text{giant}}$	**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.

Real-world data efficiency. Following [80], we evaluate models under constrained conditions using a limited set of real-world images. Specifically, models pre-trained on synthetic datasets *GTAV* + *SYNTHIA* + *UrbanSyn* (marked with ‡ in Tab. 2) are fine-tuned on 1/16 of the *Cityscapes* training set. As shown in Tab. 5, SoRA achieves performance on par with Rein in the source domain, while offering greater robustness on unseen domains. This underscores SoRA's data-efficient adaptability to real-world scenes, while effectively avoiding overfitting to the source domain.

Clear-to-adverse weather DGOD. In Tab. 6, we compare SoRA with existing DGOD methods for classic backbone-head setting and with PEFT methods for recent foundation architectures. As expected, SoRA demonstrates only minor improvements in the classic setting; however, in the VFM setting—where there is considerably more knowledge to retain—it shows substantial performance gains. As illustrated by the visualization results in Fig. 4, SoRA makes robust predictions even under diverse challenging scenes.

Model efficiency. As detailed in Tab. 7, SoRA exhibits higher throughput than adapter- and VPT-based methods like Rein [80] and SET [86], as it incurs no additional latency. This advantage is especially significant in online inference settings, where the batch size is typically as small

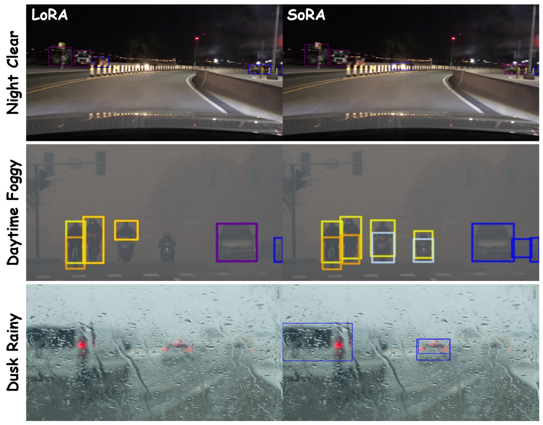


Figure 4. **DGOD qualitative results.**

Methods	Params.*	DGSS Avg.	DGOD Avg.
Full fine-tuning (baseline)	304.2M	64.4	51.0
└ + Freezing early blocks	201.6M	65.0 (↑ 0.6)	51.4 (↑ 0.4)
└ + Tuning principal components	4.9M	66.1 (↑ 1.1)	53.0 (↑ 1.6)
└ + Tuning minor components	4.9M	67.7 (↑ 2.7)	53.8 (↑ 2.4)
└ + Annealing weight decay	4.9M	68.3 (↑ 0.6)	54.3 (↑ 0.5)

Table 8. **Effect of our changes** evaluated on DG benchmarks. See the full ablation study in the *Supplemental*.

$\{W_q,W_k,W_v,W_o\}$	$\{W_{up},W_{down}\}$	Params.*	→*Citys.*	→*BDD*	→*Map.*	Avg.
	✓	2.7M	68.76	**61.50**	70.00	66.75
✓		2.2M	70.82	61.36	69.86	67.35
✓	✓	4.9M	**71.82**	61.31	**71.67**	**68.27**

Table 9. DGSS performance after applying SoRA to **different types of modules (Self-attention / MLP)** in DINOv2-Large.

Rank r	4	8	16	32	64
DGSS avg. (mIoU in %)	66.91	67.71	**68.27**	67.76	67.59
Params.*	1.3M	2.5M	4.9M	9.6M	19.0M

Table 10. DGSS performance with **different rank** r.

as one [33]. Furthermore, in scenarios involving DINOv2-giant exceeding 1B parameters, SoRA can drastically reduce training costs compared to FFT. SoRA initialization is completed within 30 seconds for large-sized models, which is a negligible cost given the improved performance.

5.3. Ablation Study

Component Analysis. A thorough examination of each component of SoRA in the DGOD and DGSS basic setting, as summarized in Tab. 8, demonstrates that all components of SoRA incrementally improve DG performance *without adding computational costs*. Notably, the marked performance gap with tuning principal singular components [55] suggests that SoRA better preserves the integrity of pre-trained representations while learning task-specific features.

Tuning granularity. Interestingly, as shown in Tables 9 and 10, our method consistently achieves competitive results compared to SOTA baselines across all tuning granularities, indicating that SoRA can be flexibly configured according to the training budget, task difficulty, or domain gap. The best results are achieved with the default setting.

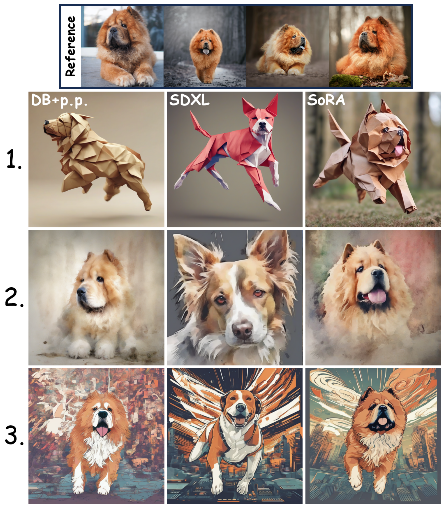


Figure 5. **Subject Personalization.** 1. A *dog* gracefully leaping in origami style, 2. A *dog* in watercolor painting style, and 3. A *dog* soaring through a digital landscape in vector illustration style.

5.4. Subject Personalization

DreamBooth (DB) [70] introduces subject personalization, which fine-tunes a pre-trained Text-to-Image (T2I) model on a few reference images to enable it to generate new visual concepts. DB updates all parameters within specific blocks, potentially impairing the pre-trained capability to generate images across various domains. Therefore, we explore whether SoRA can be seamlessly integrated into the DB framework to preserve pre-trained knowledge. For this task, we utilize Stable Diffusion XL (SDXL) [64] for the T2I diffusion model. Fig. 5 presents a visual comparison between DB and SoRA, using identical hyperparameter settings and sample seeds. As seen, SoRA, similar to the original SDXL model, can generate high-quality images of diverse visual domains (e.g., style, texture, pose) and exhibits superior identity preservation compared to DB with prior preservation (p.p.) loss [70]. More results with varied prompts and subjects are provided in the *Supplemental*.

6. Conclusion

This paper presents a multifaceted exploration of the distribution of generalizable components in VFMs—examining them at the weight level, block level, and within training dynamics—and introduces the SoRA framework as an effective means to preserve generalizable components. Contrary to recent approaches that passively refine frozen VFM features, SoRA is integrated into all linear layers to effectively facilitate task adaptation, while minimizing interference with generalizable components by initializing the low-rank adapter with minor singular components. Through extensive experiments, we observe that SoRA significantly outperforms the state-of-the-art methods in DGSS, DGOD, and even subject-driven image generation tasks.

SoRA: Singular Value Decomposed Low-Rank Adaptation for Domain Generalizable Representation Learning

Supplementary Material

For a comprehensive understanding of our proposed SoRA 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 [15] and MMDetection [11] codebase for Domain Generalized Semantic Segmentation (DGSS) and Domain Generalized Object Detection (DGOD) implementations, respectively, and leverage the training scripts developed by HuggingFace [78] for subject personalization experiments.

A.1. DGSS Settings

The experimental settings for all studies conducted in the main paper are outlined in Tab. 11. Unless otherwise specified, Mask2Former [13] is utilized as the default decode head, and following Rein [80], we adopt only the basic data augmentation used in Mask2Former. Additionally, EMA is selectively employed to ensure stable training.

A.2. DGOD Settings

The DGOD settings are detailed in the rightmost two columns of Tab. 11. When applying SoRA to convolution-based backbones such as ResNet [30], 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 SoRA as described in Eq. 1. 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 [17]. In contrast, when using DINOv2 [58] as the backbone, we simply utilize basic data augmentation used in Co-DETR [93].

A.3. Subject Personalization Settings

We conduct experiments on the DreamBooth dataset [70], which consists of 30 subjects with 4–6 images per subject. In all experiments the SoRA 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 [25] that the first 10 attention layers of `up_blocks.0` in SDXL [64] 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 SoRA 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* → *Mapillary* DGSS and *Daytime-Sunny* → {*Dusk-Rainy, Daytime-Foggy*} DGOD scenarios. In Tables 12 and 13, we systematically evaluate the effectiveness of each component within the SoRA 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. 12, *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 for most classes. 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 *vs.* train, motorcycle *vs.* bicycle). Likewise, all components clearly improve recognition for the majority of classes under adverse weather detection settings (see Tab. 13).

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 11. **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 12. **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 13. **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.

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. 14 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 [13], 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).

# 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 14. 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 [58]	FFT	304.2M	66.93	57.34	68.89	64.39
	SoRA	4.9M	**71.82**	**61.31**	**71.67**	**68.27**
DINOv2-B [58]	FFT	86.5M	60.84	52.98	62.12	58.65
	SoRA	2.3M	**66.71**	**57.48**	**67.34**	**63.84**
DINOv2-S [58]	FFT	22.0M	53.71	49.03	58.10	53.61
	SoRA	1.0M	**57.58**	**52.95**	**62.48**	**57.67**
ConvNeXt V2-L [81]	FFT	196.4M	55.93	50.71	60.79	55.81
	SoRA	12.1M	**60.12**	**53.36**	**61.46**	**58.31**
Swin-L [54]	FFT	195.2M	54.40	49.85	60.05	54.77
	SoRA	5.4M	**56.91**	**51.98**	**60.73**	**56.54**
ResNet101 [30]	FFT	42.3M	**41.29**	44.29	48.79	44.79
	SoRA	2.5M	41.23	**45.57**	**49.71**	**45.50**

Table 15. **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 [58]	Rein [80]	2.5M	63.60	59.00	63.70	62.10
	SoRA	4.9M	**67.81**	**60.12**	**68.95**	**65.63**
EVA02-L [24]	Rein [80]	2.5M	61.40	**58.50**	62.00	60.70
	SoRA	5.1M	**64.91**	57.54	**65.33**	**62.59**

Table 16. DGSS evaluation results with **SemFPN head** [44].

C. Additional Experiments

C.1. Results on Various Backbones

Tab. 15 showcases the versatility of SoRA 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 [18] supervision and MAE [31, 81] pre-training. SoRA consistently outperforms FFT across diverse backbone architectures. Notably, the improvements brought by SoRA 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 [13] is predominantly used as decode head in all DGSS experiments, SoRA is compatible with any decode head. To assess its robustness across different heads, we employ the lightweight SemFPN [44] head to benchmark its performance against Rein [80]. Our experimental results (Tab. 16) indicate that SoRA integrates seamlessly with diverse backbones and heads, consistently surpassing the SOTA baseline.

D. Additional Comparison

In Tables 17, 18, and 19, we present an exhaustive comparison with existing methods to illustrate the broader research landscape across multiple DGSS settings. Additionally, Figures 6, 7, and 8 depict DGSS prediction results on unseen domains for Cityscapes, BDD100k, and Mapillary, respectively, while Fig. 9 provides detection results under various adverse conditions. As evident from the visual comparisons above, SoRA 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.

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 10 and 11, integrating the SoRA 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.**

E. 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 [60]	RN50	DL-V3+	33.85	32.30	37.75	34.63
○ RobustNet [14]	RN50	DL-V3+	36.58	35.20	40.33	37.37
○ DRPC [89]	RN101	FCN	42.53	38.72	38.05	39.77
○ SiamDoGe [83]	RN50	DL-V3+	42.96	37.54	40.64	40.38
○ DIRL [84]	RN50	DL-V3+	41.04	39.15	41.60	40.60
○ GTR [62]	RN101	-	43.70	39.60	39.10	40.80
○ AdvStyle [91]	RN101	DL-V3+	43.44	40.32	41.96	41.91
○ PintheMem [42]	RN101	DL-V2	44.90	39.71	41.31	41.97
○ MRFP+ [75]	RN50	DL-V3+	42.40	39.55	44.93	42.29
○ SAN-SAW [63]	RN101	DL-V3+	45.33	41.18	40.77	42.43
○ SPC [35]	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 [45]	RN101	DL-V3+	45.79	41.73	47.08	44.87
○ SHADE [90]	RN101	DL-V3+	46.66	43.66	45.50	45.27
○ PASTA [10]	RN101	DL-V3+	45.33	42.32	48.60	45.42
○ SoRA (Ours)	RN101	M2F	41.23	45.57	49.71	45.50
○ MoDify [39]	RN101	DL-V2	48.80	44.20	47.50	46.80
○ TLDR [43]	RN101	DL-V3+	47.58	44.88	48.80	47.09
○ FAMix [22]	CLIP RN101	DL-V3+	49.47	46.40	51.97	49.28
○ CMFormer [5]	Swin-L	-	55.31	49.91	60.09	55.10
○ SoRA (Ours)	Swin-L	M2F	56.91	51.98	60.73	56.54
○ DGInStyle [38]	MiT-B5	HRDA	58.63	52.25	62.47	57.78
○ DIDEX [57]	MiT-B5	DAFormer	62.00	54.30	63.00	59.70
○ CLOUDS [3]	CLIP CN-L	M2F	60.20	57.40	67.00	61.50
○ VLTSeg [36]	EVA02-L	M2F	65.30	58.30	66.00	63.20
○ Rein [80]	EVA02-L	M2F	65.30	60.50	64.90	63.60
○ FADA [4]	EVA02-L	M2F	66.70	61.90	66.10	64.90
○ tqdm [59]	EVA02-L	M2F	68.88	59.18	70.10	66.05
○ SoRA (Ours)	EVA02-L	M2F	68.05	60.81	68.33	65.73
● SoRA (Ours)	EVA02-L	M2F	69.94	62.48	68.33	66.92
○ DoRA [52]	DINOv2-L	M2F	66.12	59.31	67.07	64.17
○ VPT [37]	DINOv2-L	M2F	68.75	58.64	68.32	65.24
○ SET [86]	DINOv2-L	M2F	68.06	61.64	67.68	65.79
○ FADA [4]	DINOv2-L	M2F	68.23	61.94	68.09	66.09
○ AdaptFormer [12]	DINOv2-L	M2F	70.10	59.81	68.77	66.23
○ SSF [51]	DINOv2-L	M2F	68.97	61.30	68.77	66.35
○ LoRA [33]	DINOv2-L	M2F	70.13	60.13	70.42	66.89
○ Rein† [80]	DINOv2-L	M2F	69.19	60.01	69.06	66.09
● Rein† [80]	DINOv2-L	M2F	70.68	62.51	69.61	67.60
○ SoRA (Ours)	DINOv2-L	M2F	71.82	61.31	**71.67**	68.27
● SoRA (Ours)	DINOv2-L	M2F	**73.63**	**63.33**	70.98	**69.31**
Multi-source DGSS Trained on GTAV + SYNTHIA						
○ RobustNet [14]	RN50	DL-V3+	37.69	34.09	38.49	36.76
○ AdvStyle [91]	RN50	DL-V3+	39.29	39.26	41.14	39.90
○ DIGA [73]	RN101	DL-V2	46.43	33.87	43.51	41.27
○ PintheMem [42]	RN50	DL-V3+	44.51	38.07	42.70	41.76
○ MRFP+ [75]	RN50	DL-V3+	46.18	41.13	45.28	44.24
○ SHADE [90]	RN50	DL-V3+	47.43	40.30	47.60	45.11
○ TLDR [43]	RN50	DL-V3+	48.83	42.58	47.80	46.40
○ SPC [35]	RN101	DL-V3+	47.93	43.62	48.79	46.78
○ FAMix [22]	CLIP RN50	DL-V3+	49.41	45.51	51.61	48.84
○ Rein† [80]	DINOv2-L	M2F	72.17	61.53	70.69	68.13
○ SoRA (Ours)	DINOv2-L	M2F	73.16	61.90	72.73	69.26
● SoRA (Ours)	DINOv2-L	M2F	**74.85**	**63.59**	**73.92**	**70.79**

Table 17. Comparison of the proposed SoRA 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 [14]	RN50	DL-V3+	50.73	58.64	54.69
○ WildNet [45]	RN50	DL-V3+	50.94	58.79	54.87
○ SiamDoGe [83]	RN50	DL-V3+	51.53	59.00	55.27
○ SHADE [90]	RN50	DL-V3+	50.95	60.67	55.81
○ BlindNet [1]	RN50	DL-V3+	51.84	60.18	56.01
○ FAMix [22]	CLIP RN50	DL-V3+	54.07	58.72	56.40
○ SAN-SAW [63]	RN101	DL-V3+	54.73	61.27	58.00
○ HGFormer [19]	Swin-L	-	61.50	72.10	66.80
○ CMFormer [5]	Swin-L	-	62.60	73.60	68.10
○ tqdm [59]	EVA02-L	M2F	64.72	76.15	70.44
○ FADA [4]	DINOv2-L	M2F	65.12	75.86	70.49
○ Rein† [80]	DINOv2-L	M2F	66.53	75.18	70.86
○ SoRA (Ours)	DINOv2-L	M2F	67.02	76.45	71.74
● SoRA (Ours)	DINOv2-L	M2F	**68.08**	**77.87**	**72.98**

Table 18. **Real-to-real DGSS comparison.**

Clear-to-Adverse Weather	ACDC [71] Test Domains (mIoU in %)				
Methods	→*Night*	→*Snow*	→*Fog*	→*Rain*	Avg.
Single-source DGSS Trained on Cityscapes					
○ CMFormer [5]	33.7	64.3	77.8	67.6	60.9
○ SET [86]	57.3	73.7	80.1	74.8	71.5
○ FADA [4]	**57.4**	73.5	80.2	75.0	71.5
○ SoRA (Ours)	52.4	**74.6**	**84.1**	**75.5**	**71.7**

Table 19. 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. 2, 3, 12

[2] Muhammad Awais, Muzammal Naseer, Salman Khan, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Foundational models defining a new era in vision: A survey and outlook. *arXiv:2307.13721*, 2023. 2

[3] Yasser Benigmim, Subhankar Roy, Slim Essid, Vicky Kalogeiton, and Stéphane Lathuilière. Collaborating foundation models for domain generalized semantic segmentation. In *CVPR*, 2024. 3, 6, 12

[4] 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. 3, 6, 7, 12

[5] Qi Bi, Shaodi You, and Theo Gevers. Learning content-enhanced mask transformer for domain generalized urban-scene segmentation. In *AAAI*, 2024. 2, 7, 12

[6] Dan Biderman, Jose Gonzalez Ortiz, Jacob Portes, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, et al. Lora learns less and forgets less. *TMLR*, 2024. 2

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas

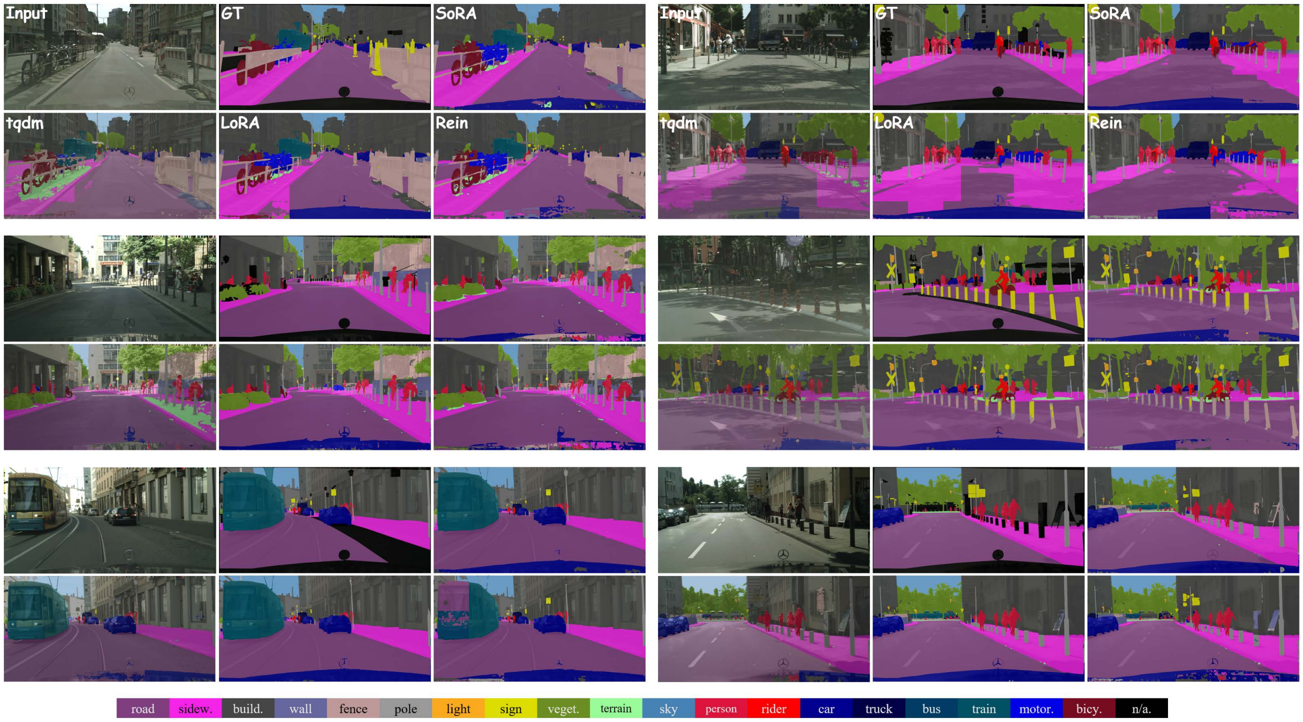


Figure 6. Segmentation results of SoRA on the Cityscapes. The model is trained on GTAV with DINOv2-L backbone.

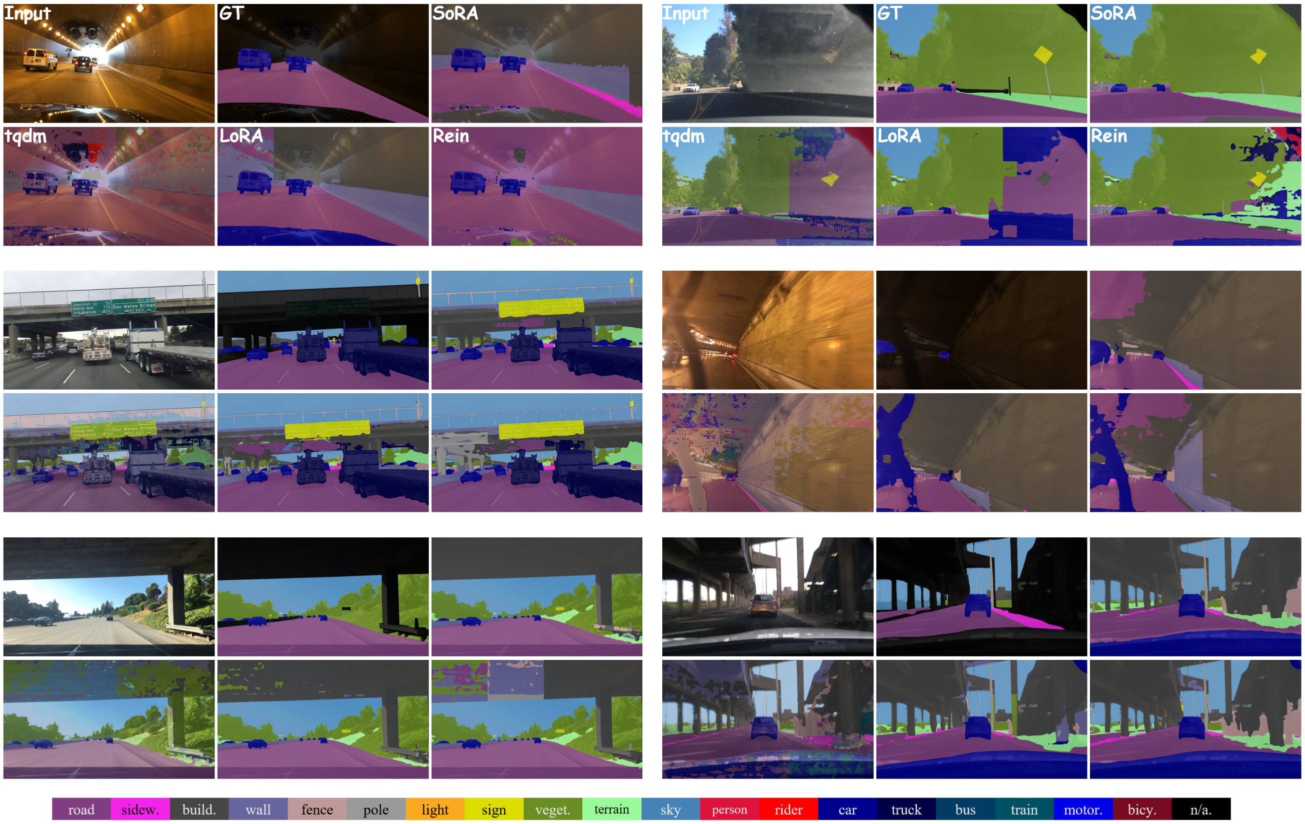


Figure 7. Segmentation results of SoRA on the BDD100k. The model is trained on GTAV with DINOv2-L backbone.

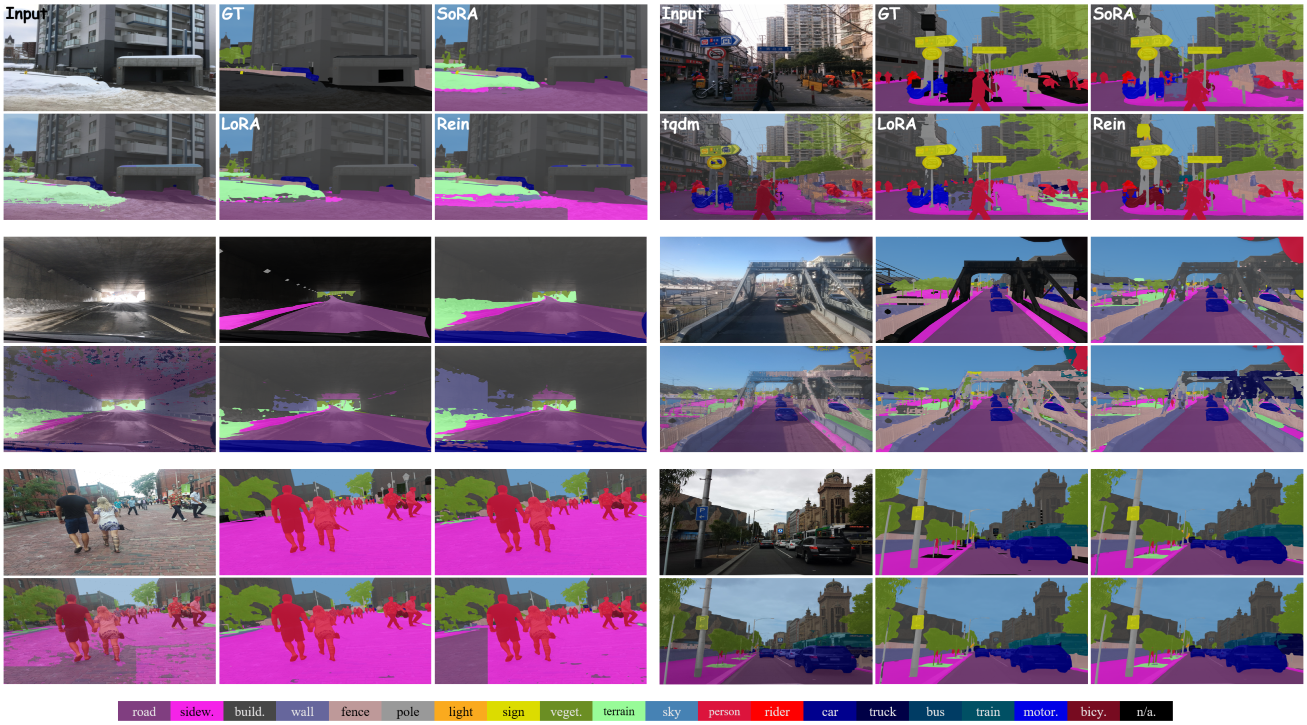


Figure 8. Segmentation results of SoRA on the Mapillary. The model is trained on GTAV with DINOv2-L backbone.

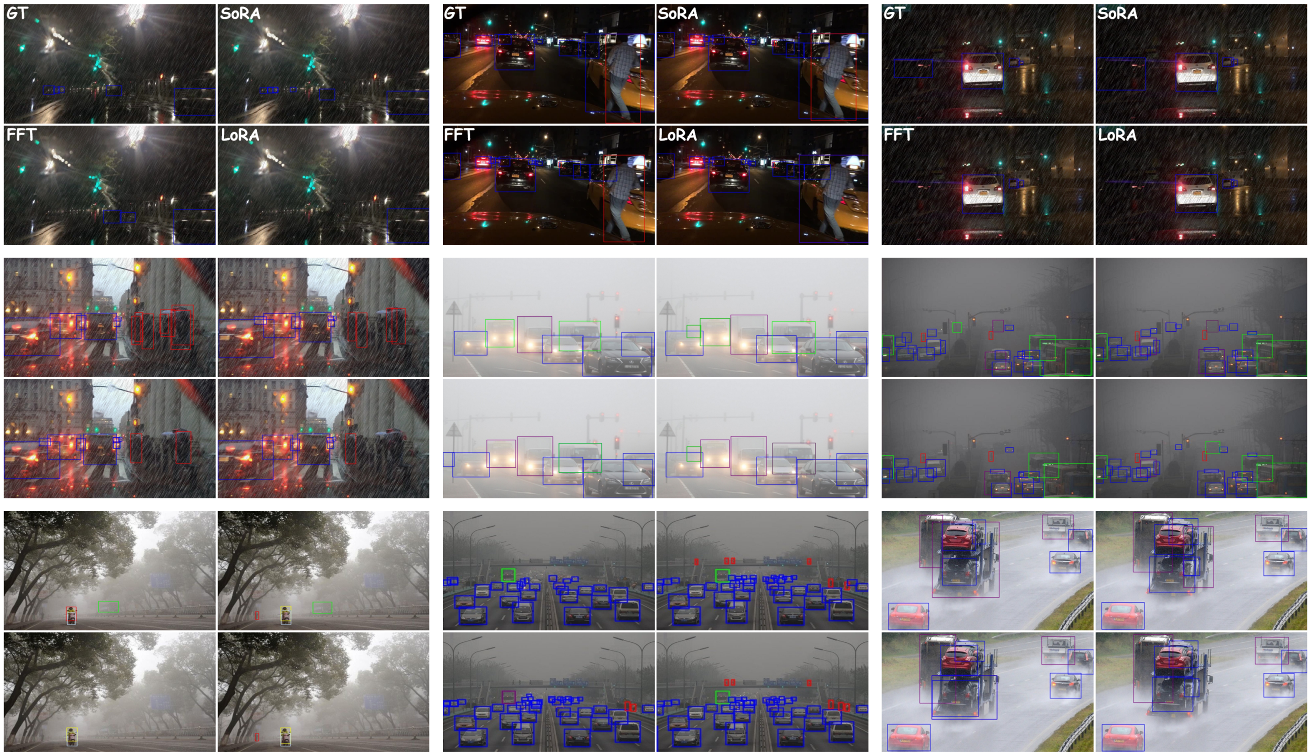


Figure 9. Detection results of SoRA on the adverse scene. The model is trained on Daytime-Sunny with DINOv2-L backbone.

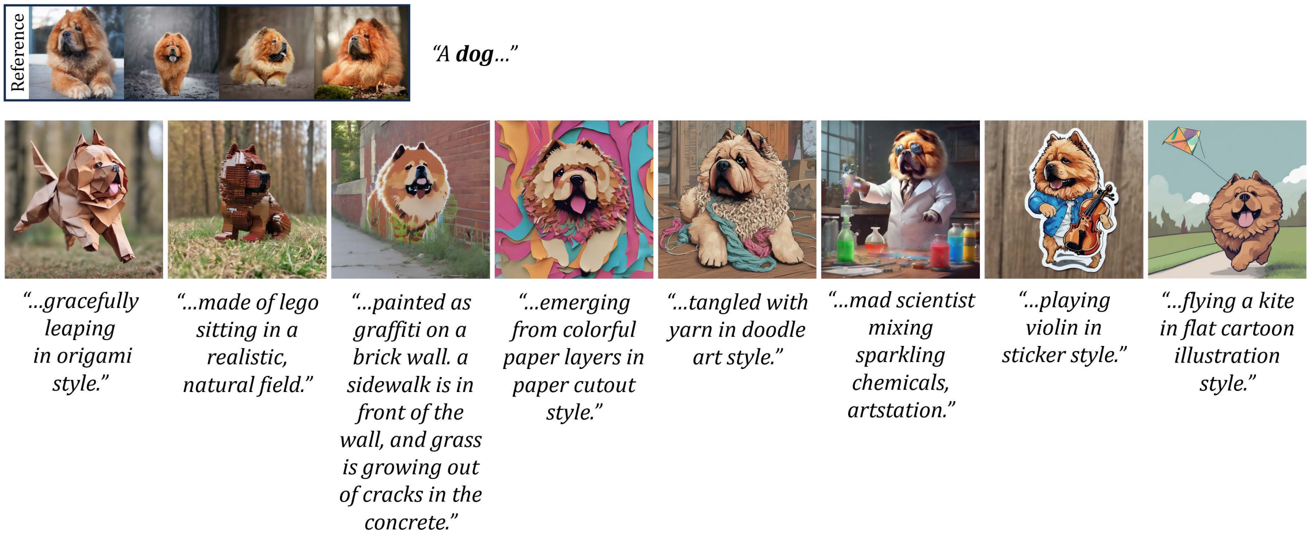

Reference
"A **dog**..."
"...gracefully leaping in origami style."
"...made of lego sitting in a realistic, natural field."
"...painted as graffiti on a brick wall. a sidewalk is in front of the wall, and grass is growing out of cracks in the concrete."
"...emerging from colorful paper layers in paper cutout style."
"...tangled with yarn in doodle art style."
"...mad scientist mixing sparkling chemicals, artstation."
"...playing violin in sticker style."
"...flying a kite in flat cartoon illustration style."

Figure 10. **Multiple subject-consistent synthesis results with prompts describing various domains.** SoRA 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.

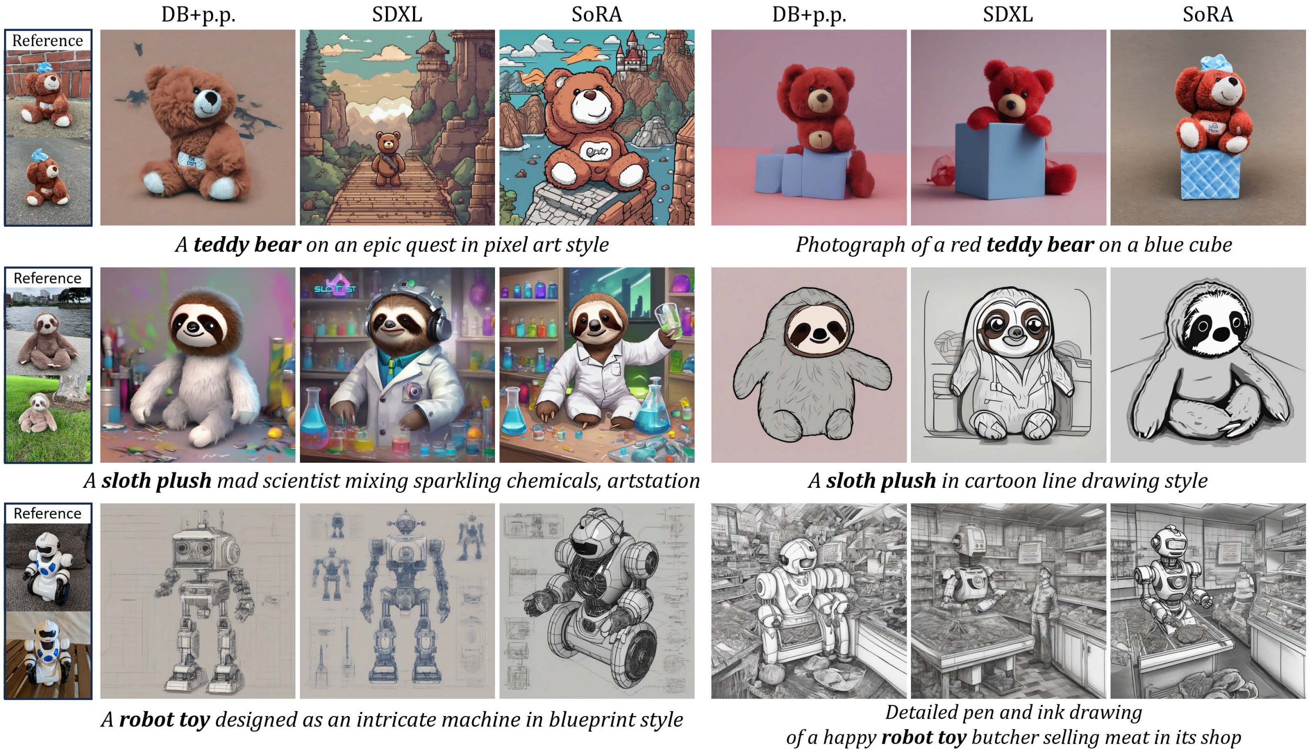

DB+p.p.
SDXL
SoRA
DB+p.p.
SDXL
SoRA
Reference
A **teddy bear** on an epic quest in pixel art style
Photograph of a red **teddy bear** on a blue cube
Reference
A **sloth plush** mad scientist mixing sparkling chemicals, artstation
A **sloth plush** in cartoon line drawing style
Reference
A **robot toy** designed as an intricate machine in blueprint style
Detailed pen and ink drawing of a happy **robot toy** butcher selling meat in its shop

Figure 11. **Qualitative comparison to DreamBooth** [70] **with prior preservation (p.p.) loss.**

Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1

[8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 3

[9] Gyusam Chang, Jiwon Lee, Donghyun Kim, Jinkyu Kim, Dongwook Lee, Daehyun Ji, Sujin Jang, and Sangpil Kim. Unified domain generalization and adaptation for multi-view 3d object detection. In *NeurIPS*, 2024. 2

[10] Prithvijit Chattopadhyay, Kartik Sarangmath, Vivek Vijaykumar, and Judy Hoffman. Pasta: Proportional amplitude

spectrum training augmentation for syn-to-real domain generalization. In *ICCV*, 2023. 2, 3, 12

[11] 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. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv:1906.07155*, 2019. 9

[12] 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. 2, 3, 6, 7, 12

[13] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 1, 6, 9, 10, 11

[14] 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. 2, 3, 5, 12

[15] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 9

[16] 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. 6

[17] 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. 3, 7, 9

[18] 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, 11

[19] Jian Ding, Nan Xue, Gui-Song Xia, Bernt Schiele, and Dengxin Dai. Hgformer: Hierarchical grouping transformer for domain generalized semantic segmentation. In *CVPR*, 2023. 2, 7, 12

[20] 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. 3

[21] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1936. 3

[22] 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. 3, 12

[23] Qi Fan, Mattia Segu, Yu-Wing Tai, Fisher Yu, Chi-Keung Tang, Bernt Schiele, and Dengxin Dai. Towards robust object detection invariant to real-world domain shifts. In *ICLR*, 2023. 2, 3

[24] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *IVC*, 2024. 3, 6, 11

[25] Yarden Frenkel, Yael Vinker, Ariel Shamir, and Daniel Cohen-Or. Implicit style-content separation using b-lora. In *ECCV*, 2024. 9

[26] Aditya Sharad Golatkar, Alessandro Achille, and Stefano Soatto. Time matters in regularizing deep networks: Weight decay and data augmentation affect early learning dynamics, matter little near convergence. In *NeurIPS*, 2019. 6

[27] Jose L Gómez, Manuel Silva, Antonio Seoane, Agnès Borrás, Mario Noriega, Germán Ros, Jose A Iglesias-Guitian, and Antonio M López. All for one, and one for all: Urbansyn dataset, the third musketeer of synthetic driving scenes. *arXiv:2312.12176*, 2023. 1, 6

[28] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *ICLR*, 2022. 2

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 3

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 3, 7, 9, 11

[31] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 3, 11

[32] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In *CVPR*, 2022. 5

[33] 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. 1, 2, 3, 5, 6, 7, 8, 12

[34] 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. 2

[35] 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. 2, 12

[36] 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. 3, 6, 7, 12

[37] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, 2022. 2, 3, 6, 12

[38] 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. 3, 12

[39] Xueying Jiang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Domain generalization via balancing training difficulty and model capability. In *ICCV*, 2023. 2, 12

[40] Mengmeng Jing, Xiantong Zhen, Jingjing Li, and Cees GM Snoek. Order-preserving consistency regularization for domain adaptation and generalization. In *ICCV*, 2023. 3

[41] Christoph Kamann and Carsten Rother. Increasing the robustness of semantic segmentation models with painting-by-numbers. In *ECCV*, 2020. 3

[42] Jin Kim, Jiyoung Lee, Jungin Park, Dongbo Min, and Kwanghoon Sohn. Pin the memory: Learning to generalize semantic segmentation. In *CVPR*, 2022. 2, 12

[43] Sunghwan Kim, Dae-hwan Kim, and Hoseong Kim. Texture learning domain randomization for domain generalized segmentation. In *ICCV*, 2023. 3, 5, 12

[44] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, 2019. 11

[45] Suhyeon Lee, Hongje Seong, Seongwon Lee, and Euntai Kim. Wildnet: Learning domain generalized semantic segmentation from the wild. In *CVPR*, 2022. 3, 5, 12

[46] Sohyun Lee, Taeyoung Son, and Suha Kwak. Fifo: Learning fog-invariant features for foggy scene segmentation. In *CVPR*, 2022. 2

[47] Wooju Lee, Dasol Hong, Hyungtae Lim, and Hyun Myung. Object-aware domain generalization for object detection. In *AAAI*, 2024. 3, 7

[48] Yoonho Lee, Annie S Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. Surgical fine-tuning improves adaptation to distribution shifts. In *ICLR*, 2023. 5

[49] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *EMNLP*, 2021. 2

[50] Deng Li, Aming Wu, Yaowei Wang, and Yahong Han. Prompt-driven dynamic object-centric learning for single domain generalization. In *CVPR*, 2024. 2, 7

[51] Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. In *NeurIPS*, 2022. 3, 6, 12

[52] 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. 3, 6, 7, 12

[53] Yajing Liu, Shijun Zhou, Xiyao Liu, Chunhui Hao, Baojie Fan, and Jiandong Tian. Unbiased faster r-cnn for single-source domain generalized object detection. In *CVPR*, 2024. 3, 7

[54] 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. 11

[55] Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. In *NeurIPS*, 2024. 3, 8

[56] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017. 6

[57] 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. 3, 12

[58] 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. 3, 4, 5, 6, 7, 9, 11

[59] Byeonghyun Pak, Byeongju Woo, Sunghwan Kim, Daehwan Kim, and Hoseong Kim. Textual query-driven mask transformer for domain generalized segmentation. In *ECCV*, 2024. 3, 6, 7, 12

[60] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *ECCV*, 2018. 3, 12

[61] Yuwen Pan, Rui Sun, Naisong Luo, Tianzhu Zhang, and Yongdong Zhang. Exploring reliable matching with phase enhancement for night-time semantic segmentation. In *ECCV*, 2024. 2

[62] Duo Peng, Yinjie Lei, Lingqiao Liu, Pingping Zhang, and Jun Liu. Global and local texture randomization for synthetic-to-real semantic segmentation. *TIP*, 2021. 2, 3, 12

[63] Duo Peng, Yinjie Lei, Munawar Hayat, Yulan Guo, and Wen Li. Semantic-aware domain generalized segmentation. In *CVPR*, 2022. 2, 3, 12

[64] 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. 3, 8, 9

[65] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 3

[66] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 1, 7

[67] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016. 1, 6

[68] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 3

[69] 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. 1, 6

[70] 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. 8, 9, 15

[71] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Acdc: The adverse conditions dataset with correspondences for se-

mantic driving scene understanding. In *ICCV*, 2021. 6, 7, 12

[72] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 2

[73] 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. 2, 12

[74] Zhiqiang Tang, Yunhe Gao, Yi Zhu, Zhi Zhang, Mu Li, and Dimitris N Metaxas. Crossnorm and selfnorm for generalization under distribution shifts. In *ICCV*, 2021. 3

[75] 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. 2, 3, 12

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

[77] Vidit Vidit, Martin Engilberge, and Mathieu Salzmann. Clip the gap: A single domain generalization approach for object detection. In *CVPR*, 2023. 3, 7

[78] 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. 9

[79] Zhixiang Wei, Lin Chen, Tao Tu, Pengyang Ling, Huaian Chen, and Yi Jin. Disentangle then parse: Night-time semantic segmentation with illumination disentanglement. In *ICCV*, 2023. 2

[80] 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, 3, 6, 7, 9, 11, 12

[81] 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. 11

[82] Aming Wu and Cheng Deng. Single-domain generalized object detection in urban scene via cyclic-disentangled self-distillation. In *CVPR*, 2022. 2, 6, 7

[83] Zhenyao Wu, Xinyi Wu, Xiaoping Zhang, Lili Ju, and Song Wang. Siamdoge: Domain generalizable semantic segmentation using siamese network. In *ECCV*, 2022. 3, 12

[84] 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. 3, 12

[85] Liwei Yang, Xiang Gu, and Jian Sun. Generalized semantic segmentation by self-supervised source domain projection and multi-level contrastive learning. In *AAAI*, 2023. 2

[86] 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, 6, 7, 12

[87] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, 2020. 6

[88] Yeonguk Yu, Sungho Shin, Seunghyeok Back, Mihwan Ko, Sangjun Noh, and Kyoobin Lee. Domain-specific block selection and paired-view pseudo-labeling for online test-time adaptation. In *CVPR*, 2024. 5

[89] 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. 3, 12

[90] 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. 3, 12

[91] Zhun Zhong, Yuyang Zhao, Gim Hee Lee, and Nicu Sebe. Adversarial style augmentation for domain generalized urban-scene segmentation. In *NeurIPS*, 2022. 3, 5, 12

[92] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv:2111.07832*, 2021. 3

[93] Zhuofan Zong, Guanglu Song, and Yu Liu. Detrs with collaborative hybrid assignments training. In *ICCV*, 2023. 6, 7, 9