Adaptive Texture Filtering for Single-Domain Generalized Segmentation

Xinhui Li[1], Mingjia Li[1], Yaxing Wang[2], Chuan-Xian Ren[3], Xiaojie Guo[1]*

[1]Tianjin University, Tianjin, China
[2]Nankai University, Tianjin, China
[3]Sun Yat-sen University, Guangdong, China
{lixinhui, mingjiali}@tju.edu.cn, yaxing@nankai.edu.cn, rchuanx@mail.sysu.edu.cn, xj.max.guo@gmail.com

Abstract

Domain generalization in semantic segmentation aims to alleviate the performance degradation on unseen domains through learning domain-invariant features. Existing methods diversify images in the source domain by adding complex or even abnormal textures to reduce the sensitivity to domain-specific features. However, these approaches depend heavily on the richness of the texture bank, and training them can be time-consuming. In contrast to importing textures arbitrarily or augmenting styles randomly, we focus on the single source domain itself to achieve generalization. In this paper, we present a novel adaptive texture filtering mechanism to suppress the influence of texture without using augmentation, thus eliminating the interference of *domain-specific* features. Further, we design a hierarchical guidance generalization network equipped with structure-guided enhancement modules, which purpose is to learn the *domain-invariant* generalized knowledge. Extensive experiments together with ablation studies on widely-used datasets are conducted to verify the effectiveness of the proposed model, and reveal its superiority over other state-of-the-art alternatives.

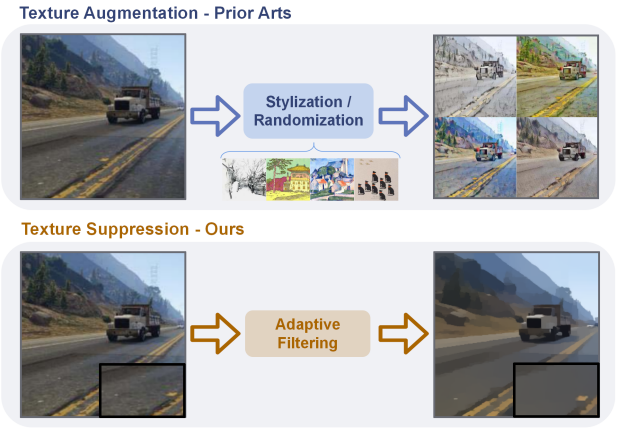

Figure 1: Illustration of existing DG methods (Peng et al. 2021) using texture augmentation, and our proposed approach using texture suppression.

Introduction

The powerful data representation capability endows deep neural networks with remarkable achievements in a series of computer vision tasks, *e.g.*, semantic segmentation, object detection, and scene understanding. These successes are usually based on the assumption that models should be trained and tested on data having the same underlying distribution. However, in realistic scenarios, it is impractical to collect sufficient data that covers the entire space. Even more, some deployed domains are completely inaccessible. For instance, collecting images under all possible conditions and scenarios are impossible in the self-driving system. Therefore, as a representative line, domain generalization (DG) (Carlucci et al. 2019; Qiao, Zhao, and Peng 2020; Peng et al. 2021) is extensively investigated to address a fundamental flaw: the network exhibits obvious performance degradation in the face of out-of-distribution or unseen-domain data. It is also known as the *distribution shift* (Moreno-Torres et al. 2012) issue.

Since *data in the target domain is inaccessible during the training process*, current DG methods advocate alleviating the distribution shift by reducing the domain gap (*e.g.*, appearance, style, or texture). These approaches can be divided into two main categories, *i.e.*, multi-source DG and single-source DG. Multi-source DG schemes (Li et al. 2018; Peng et al. 2019; Matsuura and Harada 2020; Zhu and Li 2021) attempt to align multiple available source domains through minimizing the divergence among domains. However, the alignment is typically based on the accessibility of multiple domain distributions that are not always available. To mitigate the pressure from data, another branch of DG approaches (Tobin et al. 2017; Volpi et al. 2018; Qiao, Zhao, and Peng 2020; Peng et al. 2021; Wang et al. 2021b) using single-source has been studied to achieve the goal. Some of recent methods (Ulyanov, Vedaldi, and Lempitsky 2017; Pan et al. 2018) investigate the normalization to standardize feature distributions, while the strategies (Prakash et al. 2019; Kang et al. 2022; Tjio et al. 2022) focus on the style transformation to diversify image styles. The other works (Tobin et al. 2017; Ulyanov, Vedaldi, and Lempitsky 2017; Yue et al. 2019; Peng et al. 2021) execute the domain randomization for texture synthesis. As shown in Fig. 1(top), some

*Corresponding Author

works mainly focus on diversifying the appearance (*e.g., texture*) (Peng et al. 2021). However, these data augmentation methods are limited by the authenticity and randomness of added textures, and can hardly cover all the appearances that may arise in the target domain. To handle this problem, we investigate the necessity of importing additional textures (or appearances) through the complex transformation. Namely, can we find a straightforward way to learn domain-invariant features by explicitly filtering textures from images?

In this paper, we concentrate on the task of single-domain generalization and introduce a method from a novel perspective. We work on *texture suppression*, as shown in Fig. 1 (bottom), which is more flexible and simpler than the data augmentation operation. We propose an adaptive filtering mechanism (AFM) to screen out the texture component from images, aiming to explicitly mitigate the effect of texture. More concretely, our AFM can break through the deficiency of traditional filtering which requires setting the filtering level manually. The proposed AFM can adaptively generate texture filtering intensity parameters by predicting the style of the input image, and then producing a content-dependent image. Moreover, to further explore the domain-invariant feature representation in the content-dependent image, we design a hierarchical guidance generalization network (HGGN). HGGN is customized with structure-guided enhancement modules, which purpose is to extract domain-invariant features and enhance spatial information learning ability. The major contributions of this work are summarized as follows:

- We propose a novel adaptive filtering mechanism to generate the content-dependent image, aiming to eliminate the interference of domain-specific features.
- We design a hierarchical guidance generalization network, which can implicitly guide the network to learn domain-invariant features.
- Extensive experiments are conducted on various datasets to validate the effectiveness of our method and show the superiority of our approach over other state-of-the-art.

Related Work

Unsupervised Domain Adaptation

Manually labeling semantic segmentation data is extremely time-consuming and laborious. Thus unsupervised domain adaptation (UDA) (Luo et al. 2022) has drawn growing attention, which can liberate the requirement of annotation. Basically, the manner of UDA is by narrowing the distribution gap between the automatically labeled synthetic data and unlabeled real data to achieve reasonable performance in the real domain. For instance, several works utilize the maximum mean discrepancy (MMD) (Geng, Tao, and Xu 2011; Long et al. 2015) to minimize a certain distance measure between domains and then align the distributions of different datasets. However, because of the insufficient capacity of MMD minimization, domain alignment is not always guaranteed. Other popular approaches leveraging the idea of adversarial learning (Luo et al. 2019; Tsai et al. 2019; Zhang and Wang 2020; Shermin et al. 2021) are applied by adopting domain discriminators. Meanwhile, style transfer and image-to-image translation (Zhu et al. 2017; Huang and Belongie 2017; Lee et al. 2021) methods are also studied to the UDA with the goal of converting the style of the source data and possibly getting closer to the target data. Unfortunately, the distribution discrepancy cannot be estimated from the source domain alone when real data is not accessible during training. In other words, hardly these techniques can cope with domain generalization.

Domain Generalization

Unlike UDA, domain generalization (DG) methods cannot access the target domain. The goal of DG is to train only on the source domain and generalize on target domains. The paradigm of DG methods is basically to learn generalized feature representations. Existing DG methods can be roughly divided into two categories: 1) Multiple-source DG (Matsuura and Harada 2020; Wang et al. 2021a; Liu et al. 2021; Li et al. 2021), and 2) Single-source DG (Volpi et al. 2018; Qiao, Zhao, and Peng 2020; Peng et al. 2021; Wang et al. 2021b; Xu et al. 2022).

Multiple-source DG methods attempt to explore common latent representations among the available source domains to improve the generalization capability. Inspired by the similar idea of feature alignment on UDA, many approaches mainly utilize explicit feature distribution constraints (Peng et al. 2019; Zhu and Li 2021) or feature normalization (Nam and Kim 2018; Jia, Ruan, and Hospedales 2019) to learn domain-invariant feature representations on multiple source domains. However, due to the uncertainty of target domains, the feature alignment method has the risk of over-fitting source domains and leads to poor generalization ability of the unseen domain. Moreover, various methods such as adversarial feature learning (Li et al. 2018), meta-learning (Li et al. 2019; Shu et al. 2021; Kim et al. 2022), and metric learning (Motiian et al. 2017; Dou et al. 2019) are also proposed to boost the generalization performance. In this paper, we focus on single-source DG.

Single-source DG methods strive to solve the generalization issue from the perspective of augmenting source data, such as domain randomization (Yue et al. 2019) or style transformation (Prakash et al. 2019; Peng et al. 2021) to diversify the distribution of single source domain. However, it is inflexible to perform complex adversarial generation or unstable stylized transformation on the source image. Another studies focus on exploiting latent feature representation through feature normalization, such as instance normalization, batch normalization, and many other extended normalization methods (Pan et al. 2018; Choi et al. 2021; Peng et al. 2022). These works show that normalization is effective to solve the DG problem in preventing the over-fitting of training data. Although numerous studies have been proposed to solve the generalization problem, most of the DG methods mentioned above mainly focus on image classification (Kang et al. 2022), and a few researches (Yue et al. 2019; Choi et al. 2021) on semantic segmentation. However, for the field of autonomous driving, semantic segmentation in DG is valuable and practical in the real world. Our work belongs to the single-source DG and focuses on addressing the practical application of DG in semantic segmentation.

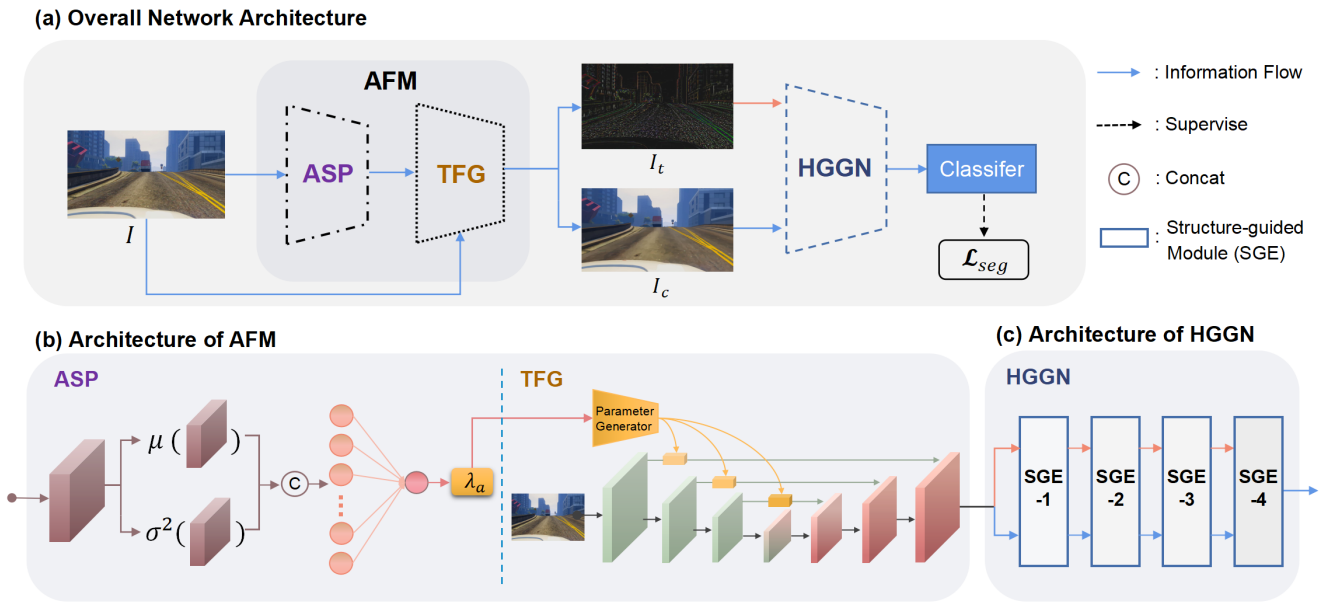

Figure 2: (a) Network architecture of the proposed method, including the adaptive filtering mechanism (AFM) and hierarchical guidance generalization network (HGGN). The AFM consists of the adaptive strength predictor (ASP) module and the texture filtering generator (TFG) module. (b) ASP predicts the filtering intensity parameter λ_a by calculating the mean μ and variance σ^2 of the extracted features. TFG is responsible for generating the content-dependent image I_c and texture-dependent image I_t. (c) HGGN composed of structure-guided enhancement modules (SGE) guides the network to learn generalized representations.

Methodology

This work focuses on how to solve the single-domain generalization problem: a model is trained on the source domain, expecting to generalize well on many unseen real-world domains. Due to the effect of domain-specific textures in generalization, as previously analyzed, our method aims to alleviate the influence of domain-specific component through texture filtering. For this purpose, we propose a novel adaptive filtering mechanism to obtain the content-dependent component. This is a straightforward and efficient way for the network to learn feature representation without the interference of texture. In addition, we design a hierarchical guidance generalization network to implicitly guide the network to learn domain-invariant features.

Overall Architecture

The overall architecture of our method is depicted in Fig. 2. Given an input image I, we first filter its texture by the adaptive filtering mechanism (AFM). Within it, the adaptive strength predictor (ASP) estimates a filtering intensity parameter λ_a determining the strength of the texture filtering generator (TFG). The TFG produces the texture-dependent image I_t and content-dependent image I_c. The hierarchical guidance generalization network (HGGN) stacked by structure-guided enhancement (SGE) modules is linked after the AFM to learn domain-invariant feature representations under contour supervision.

Adaptive Filtering Mechanism

The key to ensuring the generalization performance is the ability to generate the content-dependent component that allows the network to learn domain-invariant features directly. To this end, the adaptive filtering mechanism is proposed to separate the image into content-dependent component I_c and texture-dependent component I_t, which contains the adaptive strength predictor (ASP) and texture filtering generator (TFG).

Adaptive Strength Predictor (ASP) ASP aims to generate the filter strength value based on the style of the input image. Specifically, we implement a feature extraction module containing three convolution layers and then calculate the mean μ and variance σ^2 of the extracted features across the spatial dimension of each channel, respectively. Due to the effectiveness of the statistics in representing image information (Huang and Belongie 2017; Zhou et al. 2021), the value of mean and variance are concatenated to represent the style of the input image. Then, the value of filter strength λ_a can be obtained through a fully connected layer, as shown in Fig. 2(b). For more detail, in the training period, we add the λ_a with an error term ϵ, whose value is obtained through the sampling of the normal distribution with a mean of 0 and a variance of 1, and truncated at 1.5. This operation can effectively improve the robustness of the network during training and is removed in the testing. Finally, after the processing of ASP, the filter strength value λ_a required by TFG can be automatically generated based on the style of the input image.

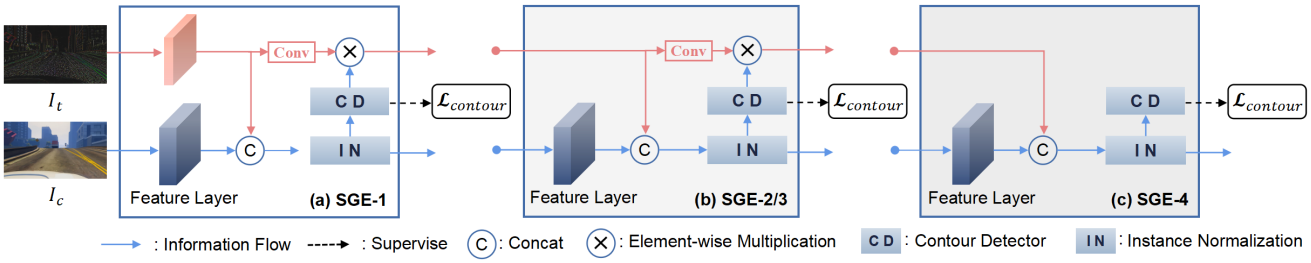

Figure 3: Network architecture of Structure-guided Enhancement Modules (SGE), which constitute the HGGN.

Texture Filtering Generator (TFG) TFG is designed to filter the texture component from the image while keeping the structure. Motivated by the effectiveness of image filtering or smoothing (Guo, Li, and Ma 2020; Li et al. 2022) on texture removal, we adopt smoothing operation as texture filter. As shown in Fig. 2(b), we utilize a U-shape network as the backbone of our smoothing generator. As noticed by previous art (Chen et al. 2020), the smoothing strength can be manipulated by inserting different convolution modules in the skip-connection. According to the smoothing strength λ_m given we first generate convolution kernels, and then use the generated convolution kernel to perform structure-preserving smoothing. Note that our TFG is pre-trained with labels from the training split of the GTA5 dataset, and then the parameters of TFG is fixed when training the segmentation network.

We extract semantic boundaries from segmentation annotations as training guidance to protect semantically meaningful edges, which is significant for segmentation. Given input image I and semantic boundary G, the loss function of TFG can be expressed as follows:

$$\mathcal{L}_{smooth} = \|I - S\|_2^2 + \lambda_m \left\| \frac{\nabla S}{\nabla I + G + \epsilon} \right\|_2^2, \quad (1)$$

where S is the smoothed output, $\|\cdot\|_2$ represents the ℓ_2 norm, $\nabla\cdot$ represents the gradient of an image, ϵ is a small constant to avoid zeros in the denominator (set to 0.005 empirically) and λ_m is the smoothing strength. When training the TFG, we randomly sample $\lambda_m \in [0, 4]$ from a uniform distribution as the smoothing strength. The first term in $\mathcal{L}_{smooth}$ regularizes the outputs to be consistent with the input image, while the second term tends to penalize the texture. That is to say, the higher λ_m is, the more smooth the output will be. Finally, through the cooperation of ASP and TFG, we regard the the smoothed image S without texture as the content-dependent image I_c. Meanwhile, the texture-dependent image I_t is expressed as $I_t = I - I_c$. In Fig. 4, we show some samples of content-dependent images and texture-dependent images.

Hierarchical Guidance Generalization Network

To learn the generalized representation of the content-dependent image I_c effectively, we propose a hierarchical guidance generalization network (HGGN) including structure-guided enhancement modules (SGE), as shown in Fig. 3. Firstly, we utilize the I_c generated from the AFM as the input of the HGGN. Although the filtering operation

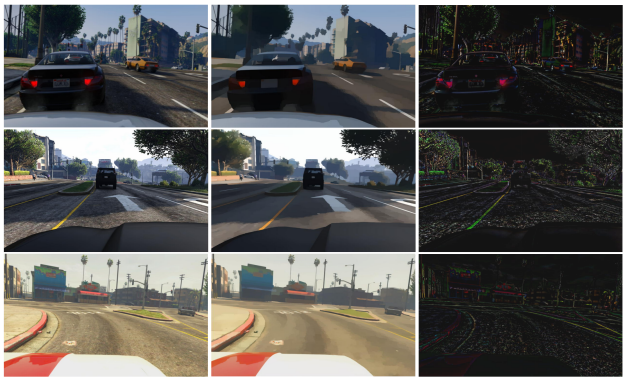

Figure 4: Samples of content-dependent images I_c and texture-dependent images I_t generated from the AFM in the training period. Original images are from the GTA5 dataset.

can preserve the contour of objects, some internal structure information of the object would be removed during this processing. However, the internal structure information is also helpful for the segmentation task (Ronneberger, Fischer, and Brox 2015). To this end, we also extract the internal structure from the texture-dependent image I_t. In Fig. 4, it can be clearly observed that I_t contains some internal structure information of the object. Therefore, we design SGE modules to fuse the extracted feature from I_c and I_t, which can better guide the network to learn the spatial feature representation.

More specifically, there are four SGE modules in the HGGN, as shown in Fig. 3. To ensure that internal structure information in I_t is better preserved, we adopt concatenate to fuse the features. Subsequently, we normalize the fused features through instance normalization (IN) to extract the style-normalized feature representations. Many previous works like (Ulyanov, Vedaldi, and Lempitsky 2017; Jin et al. 2020) have proved the effectiveness of IN on style normalization, but the spatial feature representation might be weakened during IN. However, in the DG semantic segmentation task, the spatial features after IN need to be highlighted due to the similar structural information among different domains. Specifically, for this purpose, we design a contour detector (CD). CD generates the predicted contour map y through three convolution layers and then calculates the contour loss $\mathcal{L}_{contour}$ with the contour ground truth, which is obtained from the semantic label. Meanwhile, the contour

map y is also utilized to enhance the spatial feature representation of the extracted feature from I_t by element-wise multiplication. A convolution layer is used to change the channel dimension, namely the "Conv" in Fig. 3. The supervised training of contour detection can explicitly assist the model in learning the domain-invariant (shape and spatial) information. We use the class-balanced cross-entropy loss as the contour loss to supervise the predicted contour map. For a predicted contour map y, the contour loss ℓ_{contour} can be written as:

$$\begin{aligned}\ell_{\text{contour}} = &- \beta \sum_{j \in Y_+} \log \sigma \left(y_j = 1\right) \\ &- (1-\beta) \sum_{j \in Y_-} \log \sigma \left(y_j = 0\right),\end{aligned} \tag{2}$$

where $\sigma(\cdot)$ is the sigmoid function. $\beta = |Y_-| \,/\, (|Y_-| + |Y_+|)$, where $|Y_-|$ and $|Y_+|$ denote the contour and non-contour in the ground truth Y. The total contour loss $\mathcal{L}_{\text{contour}}$ is obtained by the sum of ℓ_{contour} from four SGE modules.

In addition to the SGE, the classifier is responsible for generating semantic segmentation results. The segmentation loss $\mathcal{L}_{\text{seg}}$ given by standard cross-entropy loss is defined as:

$$\mathcal{L}_{\text{seg}} = -\sum_{h,w} \sum_{c=1}^{C} y_s^{(h,w,c)} \log p_s^{(h,w,c)}, \tag{3}$$

where $p_s^{(h,w,c)}$ denotes the predicted semantic segmentation result. $y_s^{(h,w,c)}$ is the semantic ground truth label and C is the number of classes. Combining the above contour loss term $\mathcal{L}_{\text{contour}}$ yields our final objective function:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{seg}} + \alpha \cdot \mathcal{L}_{\text{contour}}, \tag{4}$$

where we use hyper-parameter α to balance the importance of semantic segmentation loss and contour loss. Here, α is empirically set to 2.5.

Experiments

In this section, we first describe the experimental setup in detail and then reveal the advance of our design in comparison with other state-of-the-art on different real-world datasets. Finally, the effectiveness of our designed modules is analyzed in ablation studies.

Dataset Description

To verify the generalization ability of our method, we conduct experiments on extensive datasets following the common protocol adopted by prior works. There are two synthetic datasets (*e.g.*, GTA5 and SYNTHIA) and three real-world datasets (*e.g.*, Cityscapes, BDD-100K, and Mapillary) in the experiments.

Source Domain Datasets For source domain datasets, GTA5 (Richter et al. 2016) and SYNTHIA (Ros et al. 2016) are used with automatically generated annotations during training, respectively. GTA5 is a large synthetic dataset containing 24966 urban scene images of size 1914×1052. It is rendered by Grand Theft Auto V game engine and automatically annotated into different semantic categories by pixel. SYNTHIA is a synthetic dataset with pixel-level semantic annotation. We use the SYNTHIA-RAND-CITYSCAPES subset in our experiments, which contains 9,400 synthetic images with a high resolution of 1280×760.

Target Domain Datasets To evaluate the generalization capability, three real-world datasets: Cityscapes (Cordts et al. 2016), BDD-100k (Yu et al. 2020), and Mapillary (Neuhold et al. 2017) are adopted during testing. We only utilize the validation set of these datasets to test the performance of our model for comparison with other approaches. Cityscapes is a large-scale semantic segmentation dataset collected from 50 different cities in street scenarios. It contains a training set with 2,975 images and a validation set with 500 images. BDD-100k is an urban driving scene dataset collected from various locations in the US, which has 7000 training images and 1000 validation images. Mapillary is a diverse street view dataset with annotations of 66 classes, and we only use the overlapped classes with the synthetic datasets. The training and validation sets contain 18,000 and 2,000 images, respectively.

Implementation Details

Following the current state-of-the-art, we train our model on the synthetic datasets and then test on the unseen real-world datasets. ResNet-50 and ResNet-101 (He et al. 2016) are used as the backbone, respectively. The baseline model is a segmentation network (Chen et al. 2018) which follows the same architecture as prior works (Choi et al. 2021). We use pre-trained parameters on ImageNet to initialize the model except for the classifier. In the training phase, we use Stochastic Gradient Descent (SGD) optimizer with a batch size of 2, a momentum of 0.9, and a weight decay of 0.0005. The initial learning rate is set to 2.5e-4 and follows the poly learning rate policy with the power of 0.9. The training iteration is set to 200,000 for all involved models. We implement our method with PyTorch and use a single NVIDIA RTX3090 with 24 GB memory. The PASCAL VOC Intersection over Union (IoU) (Everingham et al. 2015) is used as the evaluation metric to measure the segmentation performance, and mIoU is the mean value of IoUs across all categories. During the training, we use random cropping and flipping. Meanwhile, we also adopt augmentations by randomly changing color, brightness, sharpness, and contrast.

Comparison with State-of-the-Art

We compare our method with state-of-the-art in domain generalization for semantic segmentation. To evaluate the effectiveness of our method, extensive experiments are conducted to show the generalization capability in the unseen domains. Table 1 summarizes the comparison results with recent state-of-the-art on the semantic segmentation task including IBN-Net (Pan et al. 2018), DRPC (Yue et al. 2019), ISW (Choi et al. 2021), GTR (Peng et al. 2021), and SAN-SAW (Peng et al. 2022). Benefiting from the texture suppression in AFM and domain-invariant representa-

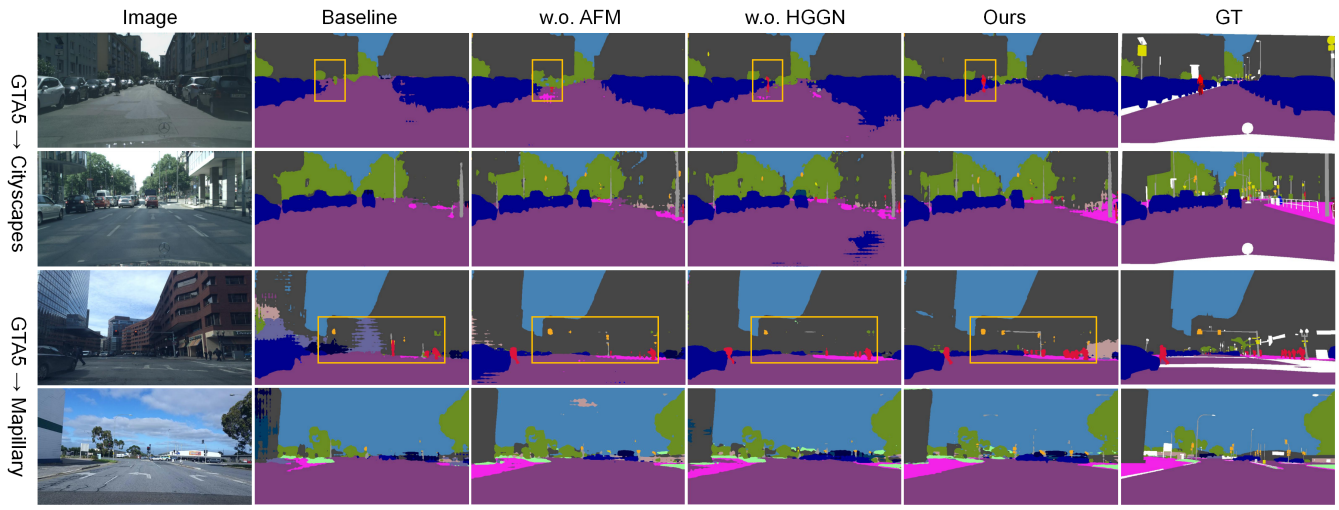

Figure 5: Qualitative domain generalization results on semantic segmentation. The model is trained on GTA5 (Richter et al. 2016) and then generalized to Cityscapes (Cordts et al. 2016) and Mapillary (Neuhold et al. 2017) with ResNet-101.

Methods (GTA5)	Mapillary	BDD-100k	Cityscapes	Avg.
ResNet-101				
Baseline	31.50	27.85	30.06	29.80
IBN-Net (ECCV'18)	37.94	38.10	37.23	37.76
DRPC (ICCV'19)	38.05	38.72	42.53	39.77
ISW (CVPR'21)	39.05	38.53	42.87	40.15
GTR (TIP'21)	39.10	39.60	43.70	40.80
SAN-SAW (CVPR'22)	40.77	41.18	**45.33**	42.43
Ours	**45.59**	**42.31**	44.83	**44.24**
ResNet-50				
Baseline	29.28	25.31	28.97	27.85
IBN-Net (ECCV'18)	37.75	32.30	33.85	34.63
DRPC (ICCV'19)	34.12	32.14	37.42	34.56
ISW (CVPR'21)	40.33	35.20	36.58	37.37
GTR (TIP'21)	34.52	33.75	37.53	35.27
SAN-SAW (CVPR'22)	41.86	37.34	39.75	39.65
Ours	**42.82**	**38.79**	**39.91**	**40.51**

Table 1: The comparison in mIoU (%) with other DG methods of the ResNet-101 and ResNet-50. The model is trained on the GTA5 dataset and generalized to real-world datasets.

Methods (SYN)	Mapillary	BDD-100k	Cityscapes	Avg.
ResNet-101				
Baseline	21.84	25.01	23.85	23.57
IBN-Net (ECCV'18)	36.19	36.63	34.18	35.67
DRPC (ICCV'19)	34.12	34.34	37.58	35.35
ISW (CVPR'21)	35.86	33.98	37.21	35.68
GTR (TIP'21)	36.40	35.30	39.70	37.13
SAN-SAW (CVPR'22)	37.26	35.98	40.87	38.04
Ours	**39.10**	**36.87**	**41.32**	**39.10**
ResNet-50				
Baseline	21.79	24.50	23.18	23.16
IBN-Net (ECCV'18)	32.16	30.57	32.04	31.60
DRPC (ICCV'19)	32.74	31.53	35.65	33.31
ISW (CVPR'21)	30.84	31.62	35.83	32.76
GTR (TIP'21)	32.89	32.02	36.84	33.92
SAN-SAW (CVPR'22)	34.52	35.24	38.92	36.23
Ours	**37.14**	**36.07**	**39.48**	**37.56**

Table 2: The comparison in mIoU (%) with other DG methods of the ResNet-101 and ResNet-50 trained on the SYNTHIA (SYN) and generalized to real-world datasets.

tion learning in HGGN, our method outperforms most methods on real-world datasets. Furthermore, we also adopt a more lightweight backbone network ResNet-50 to verify the generalization performance. It should be highlighted some other methods encounter lower improvement when switching the backbone network from ResNet-50 to ResNet-101, and even obtain degraded performance on the Mapillary dataset. This indicates that the alternatives cannot overcome the generalization problem on the heavier model, while ours still maintains significant improvements. In addition, Table 2 shows the superior results trained on another synthetic dataset (SYNTHIA) and then tested on the same real-world datasets. Extensive experiments demonstrate that our method can provide robust representations by using our proposed adaptive filtering mechanism and hierarchical guidance generalization network.

Ablation Study

We investigate the proposed modules including AFM and HGGN to find out how they contribute to the generalization

Settings	Mapillary	BDD-100k	Cityscapes	Avg.
w/o AFM	39.72	35.49	38.24	37.82
Fixed level (λ_a=1)	42.34	41.02	42.96	42.11
Fixed level(λ_a=2)	41.13	39.28	40.27	40.23

Table 3: Ablation study of adaptive filtering mechanism (AFM). The model is trained on the GTA5 with ResNet-101 and validated on three real-world datasets.

Settings	Mapillary	BDD-100k	Cityscapes	Avg.
w/o HGGN	39.13	37.94	38.91	38.66
w/ 1-level SGE	41.04	38.61	40.73	40.13
w/ 2-level SGE	42.72	40.10	42.33	41.72
w/ 3-level SGE	43.96	41.03	43.25	42.75
w/ 4-level SGE	45.59	42.31	44.83	44.24

Table 4: Ablation study of HGGN with SGE modules. The model is trained on the GTA5 with ResNet-101 and validated on three real-world datasets.

ability of segmentation. Moreover, we study the sensitivity of the hyper-parameter α in the objective function.

Effect of Adaptive Filtering Mechanism To verify the effectiveness of the proposed AFM, we conduct the ablation experiment by removing AFM ("w/o AFM"), as shown in the Table 3. In addition, we evaluate the importance of adaptive generation of filter strength in AFM by fixing the filter strength λ_a as 1 and 2, respectively. We can observe that model with fixed λ_a performs worse than the model with an adaptive generation mechanism. This demonstrates that the network can benefit from the adaptive generation of texture filtering strength according to the style of the input image.

Effect of Hierarchical Guidance Generalization Network HGGN is designed to extract the domain-invariant feature representations. To better understand the extracted domain-invariant feature, we visualize the features under the "w/ HGGN" and "w/o HGGN". As shown in Fig. 6, it is obvious that the features of "w/ HGGN" focus more on the structure features. In Table 4, we verify the effectiveness of the HGGN by conducting the setting of "w/o HGGN". Besides, we deploy different numbers of SGE in the network. This means that in the absence of SGE, we adopt the corresponding feature layer from the backbone network. Compared with 4-level SGE, the performance of 1-level SGE (without SGE-2/3/4), 2-level SGE (without SGE-3/4), and 3-level SGE (without SGE-4) on segmentation results are decreased, which indicates that the spatial guidance design after each feature layer can greatly promote generalization. According to the quantitative evaluation in Table 4 and qualitative results in Fig. 5, the HGGN with SGE modules can obtain better generalization performance.

Sensitivity to Hyper-parameter As described above, there is a hyper-parameter α to trade off the importance of the segmentation loss and contour loss in the objective function. To evaluate the impact of α, we set different values to

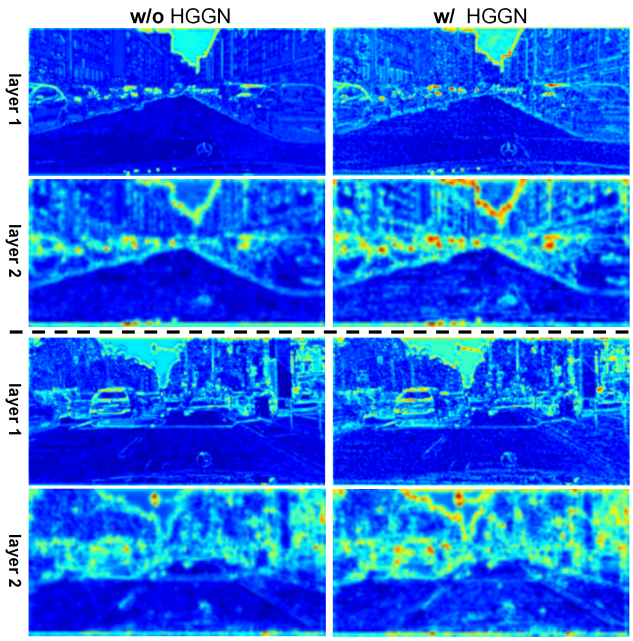

Figure 6: Visual comparison of "w/o HGGN" and "w/ HGGN". The right column shows visualized features corresponding to the first and second output features of SGE, while the left column is without SGE.

Model (GTA5)	Mapillary	BDD-100k	Cityscapes	Avg.
α=1.5	41.89	40.72	42.38	41.66
α=2.0	43.77	41.66	43.61	43.01
α=2.5	**45.59**	**42.31**	**44.83**	**44.24**
α=3.0	43.94	41.75	43.70	43.13

Table 5: Comparison of mIoU (%) with different α values. The model is trained on the GTA5 with ResNet-101 and validated on three unseen datasets.

indicate the influence of hyper-parameter on segmentation performance, as shown in Table 5. The experiments show that the network can obtain better segmentation results when α is set to 2.5.

Conclusion

This paper studied how to eliminate domain-specific feature influence and seek domain-invariant feature representation in single-domain generalization. To alleviate the reliance on domain-specific texture features, we proposed an adaptive filtering mechanism to achieve texture suppression and boost the generalization ability. Moreover, a hierarchical guidance generalization network has been designed to guide domain-invariant feature learning, such as spatial and shape information. Extensive experiments have been presented to reveal the effectiveness of our method. Our method shows its superiority over other state-of-the-art on the semantic segmentation task.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant no. 62072327.

References

Carlucci, F. M.; D'Innocente, A.; Bucci, S.; Caputo, B.; and Tommasi, T. 2019. Domain Generalization by Solving Jigsaw Puzzles. In *CVPR*, 2224–2233.

Chen, D.; Fan, Q.; Liao, J.; Avilés-Rivero, A. I.; Yuan, L.; Yu, N.; and Hua, G. 2020. Controllable Image Processing via Adaptive FilterBank Pyramid. *IEEE TIP*, 29: 8043–8054.

Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; and Adam, H. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 801–818.

Choi, S.; Jung, S.; Yun, H.; Kim, J. T.; Kim, S.; and Choo, J. 2021. Robustnet: Improving domain generalization in urban-scene segmentation via instance selective whitening. In *CVPR*, 11580–11590.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *CVPR*, 3213–3223.

Dou, Q.; Coelho de Castro, D.; Kamnitsas, K.; and Glocker, B. 2019. Domain generalization via model-agnostic learning of semantic features. In *NeurIPS*, 6447–6458.

Everingham, M.; Eslami, S.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2015. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1): 98–136.

Geng, B.; Tao, D.; and Xu, C. 2011. DAML: Domain Adaptation Metric Learning. *IEEE TIP*, 20: 2980–2989.

Guo, X.; Li, Y.; and Ma, J. 2020. Mutually Guided Image Filtering. *IEEE TPAMI*, 42: 694–707.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Huang, X.; and Belongie, S. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 1501–1510.

Jia, J.; Ruan, Q.; and Hospedales, T. M. 2019. Frustratingly Easy Person Re-Identification: Generalizing Person Re-ID in Practice. In *BMVC*, 117.

Jin, X.; Lan, C.; Zeng, W.; Chen, Z.; and Zhang, L. 2020. Style normalization and restitution for generalizable person re-identification. In *CVPR*, 3143–3152.

Kang, J.; Lee, S.; Kim, N.; and Kwak, S. 2022. Style Neophile: Constantly Seeking Novel Styles for Domain Generalization. In *CVPR*, 7130–7140.

Kim, J. Y.; Lee, J.; Park, J.; Min, D.; and Sohn, K. 2022. Pin the Memory: Learning to Generalize Semantic Segmentation. In *CVPR*, 4350–4360.

Lee, S.; Hyun, J.; Seong, H.; and Kim, E. 2021. Unsupervised Domain Adaptation for Semantic Segmentation by Content Transfer. In *AAAI*, 8306–8315.

Li, D.; Yang, J.; Kreis, K.; Torralba, A.; and Fidler, S. 2021. Semantic Segmentation with Generative Models: Semi-Supervised Learning and Strong Out-of-Domain Generalization. In *CVPR*, 8296–8307.

Li, D.; Zhang, J.; Yang, Y.; Liu, C.; Song, Y.-Z.; and Hospedales, T. M. 2019. Episodic training for domain generalization. In *ICCV*, 1446–1455.

Li, H.; Pan, S. J.; Wang, S.; and Kot, A. C. 2018. Domain generalization with adversarial feature learning. In *CVPR*, 5400–5409.

Li, M.; Fu, Y.; Li, X.; and Guo, X. 2022. Deep Flexible Structure Preserving Image Smoothing. In *ACM MM*.

Liu, Q.; Chen, C.; Qin, J.; Dou, Q.; and Heng, P.-A. 2021. FedDG: Federated Domain Generalization on Medical Image Segmentation via Episodic Learning in Continuous Frequency Space. In *CVPR*, 1013–1023.

Long, M.; Cao, Y.; Wang, J.; and Jordan, M. I. 2015. Learning Transferable Features with Deep Adaptation Networks. In *ICML*, 97–105.

Luo, X.; Zhang, J.; Yang, K.; Roitberg, A.; Peng, K.; and Stiefelhagen, R. 2022. Towards Robust Semantic Segmentation of Accident Scenes via Multi-Source Mixed Sampling and Meta-Learning. In *CVPR*, 4429–4439.

Luo, Y.; Zheng, L.; Guan, T.; Yu, J.; and Yang, Y. 2019. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *CVPR*, 2507–2516.

Matsuura, T.; and Harada, T. 2020. Domain Generalization Using a Mixture of Multiple Latent Domains. In *AAAI*, 11749–11756.

Moreno-Torres, J. G.; Raeder, T.; Alaíz-Rodríguez, R.; Chawla, N.; and Herrera, F. 2012. A unifying view on dataset shift in classification. *Pattern Recognit*, 45: 521–530.

Motiian, S.; Piccirilli, M.; Adjeroh, D. A.; and Doretto, G. 2017. Unified deep supervised domain adaptation and generalization. In *ICCV*, 5715–5725.

Nam, H.; and Kim, H.-E. 2018. Batch-Instance Normalization for Adaptively Style-Invariant Neural Networks. In *NeurIPS*, 2563–2572.

Neuhold, G.; Ollmann, T.; Bulò, S. R.; and Kontschieder, P. 2017. The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In *ICCV*, 5000–5009.

Pan, X.; Luo, P.; Shi, J.; and Tang, X. 2018. Two at Once: Enhancing Learning and Generalization Capacities via IBN-Net. In *ECCV*, 484–500.

Peng, D.; Lei, Y.; Hayat, M.; Guo, Y.; and Li, W. 2022. Semantic-aware domain generalized segmentation. In *CVPR*, 2594–2605.

Peng, D.; Lei, Y.; Liu, L.; Zhang, P.; and Liu, J. 2021. Global and Local Texture Randomization for Synthetic-to-Real Semantic Segmentation. *IEEE TIP*, 30: 6594–6608.

Peng, X.; Bai, Q.; Xia, X.; Huang, Z.; Saenko, K.; and Wang, B. 2019. Moment Matching for Multi-Source Domain Adaptation. In *ICCV*, 1406–1415.

Prakash, A.; Boochoon, S.; Brophy, M.; Acuna, D.; Cameracci, E.; State, G.; Shapira, O.; and Birchfield, S. 2019. Structured domain randomization: Bridging the reality gap by context-aware synthetic data. In *ICRA*, 7249–7255.

Qiao, F.; Zhao, L.; and Peng, X. 2020. Learning to Learn Single Domain Generalization. In *CVPR*, 12553–12562.

Richter, S. R.; Vineet, V.; Roth, S.; and Koltun, V. 2016. Playing for Data: Ground Truth from Computer Games. In *ECCV*, 102–118.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 234–241.

Ros, G.; Sellart, L.; Materzynska, J.; Vázquez, D.; and López, A. M. 2016. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In *CVPR*, 3234–3243.

Shermin, T.; Lu, G.; Teng, S. W.; Murshed, M. M.; and Sohel, F. 2021. Adversarial Network With Multiple Classifiers for Open Set Domain Adaptation. *IEEE TMM*, 23: 2732–2744.

Shu, Y.; Cao, Z.; Wang, C.; Wang, J.; and Long, M. 2021. Open Domain Generalization with Domain-Augmented Meta-Learning. In *CVPR*, 9619–9628.

Tjio, G.; Liu, P.; Zhou, J. T.; and Goh, R. S. M. 2022. Adversarial Semantic Hallucination for Domain Generalized Semantic Segmentation. In *WACV*, 3849–3858.

Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; and Abbeel, P. 2017. Domain randomization for transferring deep neural networks from simulation to the real world. In *IROS*, 23–30.

Tsai, Y.-H.; Sohn, K.; Schulter, S.; and Chandraker, M. 2019. Domain adaptation for structured output via discriminative patch representations. In *ICCV*, 1456–1465.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. S. 2017. Improved Texture Networks: Maximizing Quality and Diversity in Feed-Forward Stylization and Texture Synthesis. In *CVPR*, 4105–4113.

Volpi, R.; Namkoong, H.; Sener, O.; Duchi, J. C.; Murino, V.; and Savarese, S. 2018. Generalizing to unseen domains via adversarial data augmentation. In *NeurIPS*, 5339–5349.

Wang, J.; Lan, C.; Liu, C.; Ouyang, Y.; and Qin, T. 2021a. Generalizing to Unseen Domains: A Survey on Domain Generalization. In *IJCAI*, 4627–4635.

Wang, Z.; Luo, Y.; Qiu, R.; Huang, Z.; and Baktashmotlagh, M. 2021b. Learning to diversify for single domain generalization. In *ICCV*, 834–843.

Xu, Q.; Yao, L.; Jiang, Z.; Jiang, G.; Chu, W.; Han, W.; Zhang, W.; Wang, C.; and Tai, Y. 2022. DIRL: Domain-Invariant Representation Learning for Generalizable Semantic Segmentation. In *AAAI*, 2884–2892.

Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; and Darrell, T. 2020. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In *CVPR*, 2633–2642.

Yue, X.; Zhang, Y.; Zhao, S.; Sangiovanni-Vincentelli, A.; Keutzer, K.; and Gong, B. 2019. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *ICCV*, 2100–2110.

Zhang, Y.; and Wang, Z. 2020. Joint adversarial learning for domain adaptation in semantic segmentation. In *AAAI*, 6877–6884.

Zhou, K.; Yang, Y.; Qiao, Y.; and Xiang, T. 2021. Domain generalization with mixstyle. In *ICLR*.

Zhu, J.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *ICCV*, 2242–2251.

Zhu, R.; and Li, S. 2021. Self-supervised Universal Domain Adaptation with Adaptive Memory Separation. In *ICDM*, 1547–1552.