Texture Learning Domain Randomization for Domain Generalized Segmentation

Sunghwan Kim   Dae-hwan Kim   Hoseong Kim*

Agency for Defense Development (ADD)

{ssshwan, dhkim7, hoseongkim}@add.re.kr

Abstract

Deep Neural Networks (DNNs)-based semantic segmentation models trained on a source domain often struggle to generalize to unseen target domains, i.e., a domain gap problem. Texture often contributes to the domain gap, making DNNs vulnerable to domain shift because they are prone to be texture-biased. Existing Domain Generalized Semantic Segmentation (DGSS) methods have alleviated the domain gap problem by guiding models to prioritize shape over texture. On the other hand, shape and texture are two prominent and complementary cues in semantic segmentation. This paper argues that leveraging texture is crucial for improving performance in DGSS. Specifically, we propose a novel framework, coined Texture Learning Domain Randomization (TLDR). TLDR includes two novel losses to effectively enhance texture learning in DGSS: (1) a texture regularization loss to prevent overfitting to source domain textures by using texture features from an ImageNet pretrained model and (2) a texture generalization loss that utilizes random style images to learn diverse texture representations in a self-supervised manner. Extensive experimental results demonstrate the superiority of the proposed TLDR; e.g., TLDR achieves 46.5 mIoU on GTA→Cityscapes using ResNet-50, which improves the prior state-of-the-art method by 1.9 mIoU. The source code is available at `https://github.com/ssssshwan/TLDR`.

1. Introduction

Semantic segmentation is an essential task in computer vision with many real-world applications, such as autonomous vehicles, augmented reality, and medical imaging. Deep Neural Networks (DNNs)-based semantic segmentation models work well when the data distributions are consistent between a source domain and target domains [1,5–7,22,34,55]. However, the performance of the models tends to degrade significantly in practical settings that in-
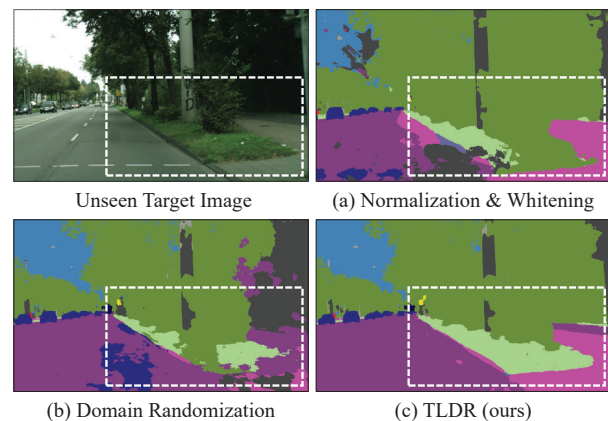
*Corresponding author.

Figure 1. Segmentation results for an image from an unseen domain (*i.e.*, Cityscapes [12]), using models trained on GTA [47] with DGSS methods. **(a-b)** Existing methods [10, 59] have difficulty in distinguishing between *road*, *sidewalk*, and *terrain* which have similar shapes and contexts, as the texture is not sufficiently considered during the training process. **(c)** Our Texture Learning Domain Randomization (TLDR) can distinguish the classes effectively as we utilize the texture as prediction cues.

volve unseen out-of-distribution scenarios, also known as a domain gap problem. Many domain adaptation and generalization methods have been proposed to solve the domain gap problem [10, 24, 28, 42–44, 54, 59, 60]. Domain adaptation assumes accessibility of target domain images, differing from domain generalization. This paper addresses Domain Generalized Semantic Segmentation (DGSS), which aims to train models that can generalize to diverse unseen domains by training on a single source domain.

Existing DGSS methods have attempted to address the domain gap problem by guiding models to focus on shape rather than texture. Given that texture often varies across different domains (*e.g.*, synthetic/real and sunny/rainy/foggy), DNNs are susceptible to domain shift because they tend to be texture-biased [19, 39]. Accordingly, there are two main approaches for the DGSS methods. The first approach is Normalization and Whitening
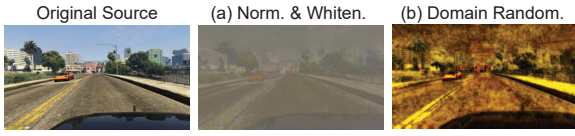
Figure 2. Reconstructed source images from the feature maps of (a) normalization and whitening and (b) domain randomization. Texture features are often omitted in existing DGSS methods.
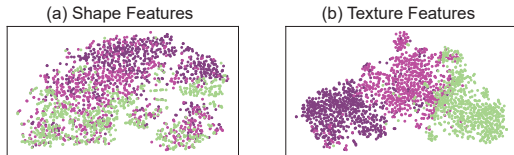

Figure 3. The t-SNE [52] plots for the *road*, *sidewalk*, and *terrain* classes from Cityscapes [12] that have similar shapes. While the shape features (Canny edge [3]) are entangled in (a), the texture features (Gram-matrix) of these classes are clearly separated in (b). The plots are based on an ImageNet pre-trained model.

Figure 4. Visualization of texture for the *road*, *sidewalk*, and *terrain* classes from GTA [47], Cityscapes [12], BDD [58], and Mapillary [40] datasets. For each class, there are commonalities in texture across the datasets.

(NW), which involves normalizing and whitening the features [10, 42–44]. It is possible to remove domain-specific texture features and learn domain-invariant shape features with NW (see Figure 2a). The second approach is Domain Randomization (DR), which trains by transforming source images into randomly stylized images [24,28,45,54,59–61]. The model learns domain-invariant shape features because texture cues are mostly replaced by random styles (see Figure 2b) [38,51].

While the existing methods are effective at making the models focus on shape features, they need to give more consideration to texture features. In addition to utilizing shape features like edges and structures, DNNs also use texture features such as patterns, color variations, and histograms as important cues for prediction [18]. Particularly in semantic segmentation, texture plays a crucial role in accurately maintaining the boundaries between objects [26, 63].

Figure 1 demonstrates the results of predicting an unseen image from Cityscapes [12] using DGSS methods trained on GTA [47]. One can see that the models trained with NW and DR have difficulty distinguishing between the *road*, *sidewalk*, and *terrain* classes, which have similar shapes and contexts. In order to determine these classes accurately, it is necessary to use texture cues. This assertion is further emphasized through t-SNE [52] plots of the classes: the shape features are entangled in Figure 3a, whereas the texture features are clearly separated in Figure 3b. Meanwhile, some textures remain relatively unchanged across domains in DGSS as shown in Figure 4. Based on these observations, we suggest utilizing texture as valuable cues for prediction.

We propose Texture Learning Domain Randomization (TLDR), which enables DGSS models to learn both shape and texture. Since only source domain images are available

in DGSS, texture must be learned from them. To accurately capture the texture, we leverage the source domain images without any modification, which we refer to as the *original source images*. The stylized source images from DR are more focused on learning shape features, and the original source images are more focused on learning texture features (Section 4.1). To further improve texture learning in DGSS, we propose a texture regularization loss and a texture generalization loss. While there are commonalities in texture across different domains, there are also clear texture differences. Thus, if the model overfits source domain textures, it will result in a performance drop. To mitigate this problem, we propose the texture regularization loss to prevent overfitting to source domain textures by using texture features from an ImageNet pre-trained model (Section 4.2). Since only source domain textures alone may not be sufficient for learning general texture representations, we propose the texture generalization loss that utilizes random style images to learn diverse texture representations in a self-supervised manner (Section 4.3).

Our contribution can be summarized into three aspects. *First*, to the best of our knowledge, we are approaching DGSS for the first time from both the shape and texture perspectives. We argue that leveraging texture is essential in distinguishing between classes with similar shapes despite the domain gap. *Second*, to enhance texture learning in DGSS, we introduce two novel losses: the texture regularization loss and the texture generalization loss. *Third*, extensive experiments over multiple DGSS tasks show that our proposed TLDR achieves state-of-the-art performance. Our method attains 46.5 mIoU on GTA→Cityscapes using ResNet-50, surpassing the prior state-of-the-art method by 1.9 mIoU.
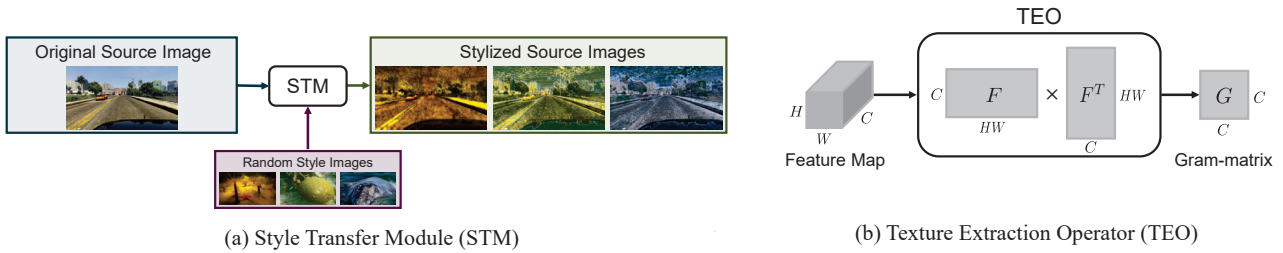
(a) Style Transfer Module (STM)   (b) Texture Extraction Operator (TEO)

Figure 5. **(a)** Visualization of using the Style Transfer Module (STM) to transform an original source image into stylized source images with random style images. **(b)** Illustration of the process of using Texture Extraction Operator (TEO) to extract only texture features, *i.e.*, a Gram-matrix, from a feature map.

2. Related Work

Domain Generalization (DG). DG aims to learn DNNs that perform well on multiple unseen domains [36]. DG has primarily been studied in image classification. A number of studies have been proposed to address DG, including adversarial learning [27, 31, 50], data augmentation [46, 53, 56, 62], meta-learning [2, 14, 33], ensemble learning [57, 62], and self-supervised learning [4, 29].

Recent studies have attempted to train domain generalized models by preserving ImageNet pre-trained feature representations as much as possible. Chen *et al.* defined DG as a life-long learning problem [33] and tried to utilize ImageNet pre-trained weights to prevent catastrophic forgetting [9]. Contrastive learning [8] and attentional pooling [8, 37] were introduced to enhance the capturing of semantic knowledge in ImageNet features. In this paper, we regularize texture representations of DNNs with the ImageNet features. To the best of our knowledge, this is the first attempt to extract a specific semantic concept (*i.e.* texture) from the ImageNet features for regularization in DG.

Domain Generalized Semantic Segmentation (DGSS). DGSS is in its early stages and has yet to receive much research attention. Existing DGSS methods have tried to alleviate a domain gap problem through two main approaches: Normalization and Whitening (NW) and Domain Randomization (DR). NW trains by normalizing the mean and standard deviation of source features and whitening the covariance of source features. This process eliminates domain-specific features, allowing the model to learn domain-invariant features. Pan *et al.* introduced instance normalization to remove domain-specific features [42]. Pan *et al.* proposed switchable whitening to decorrelate features [43], and Choi *et al.* proposed Instance Selective Whitening (ISW) to enhance the ability to whiten domain-specific features [10]. Peng *et al.* tried to normalize and whiten features in a category-wise manner [44].

DR trains by transforming source images into randomly stylized images. The model is guided to capture domain-invariant shape features since texture cues are substituted with random styles. Yue *et al.* presented a method for considering consistency among multiple stylized images [59]. Peng *et al.* distinguished between global and local texture in the randomization process [45]. Huang *et al.* proposed DR in frequency space [24]. Adversarial learning [61] and self-supervised learning [54] have been used as attempts to make style learnable rather than using random style images. It has been shown that utilizing content and style of ImageNet [28] and ImageNet pre-trained features [60] can aid in learning generalized features in DR.

Existing DR methods have yet to comprehensively address DGSS from both shape and texture perspectives. Several studies [28, 59] have stated that the success of DR is attributed to its ability to learn various styles, resulting in improved performance in generalized domains. However, from a different perspective, we consider that the effectiveness of DR is due to the model becoming more focused on shape, as discussed in recent DG methods [38, 51]. Therefore, we assume that the model needs to learn texture for further performance improvement in DR. We propose a novel approach for learning texture in DGSS without overfitting to source domain textures.

3. Preliminaries

3.1. Domain Randomization with Style Transfer

We adopt DR as a baseline DGSS method. We use a neural style transfer method at each epoch to transform each original source image into a different random style. If edge information is lost during the style transfer process, it may cause a mismatch with the semantic label, leading to a decrease in performance. We utilize photoWCT [30] known as an edge-preserving style transfer method. Random style images are sampled from ImageNet [13] validation set. Figure 5a is a visualization of using the Style Transfer Module (STM) to transform an original source image into stylized source images with random style images.
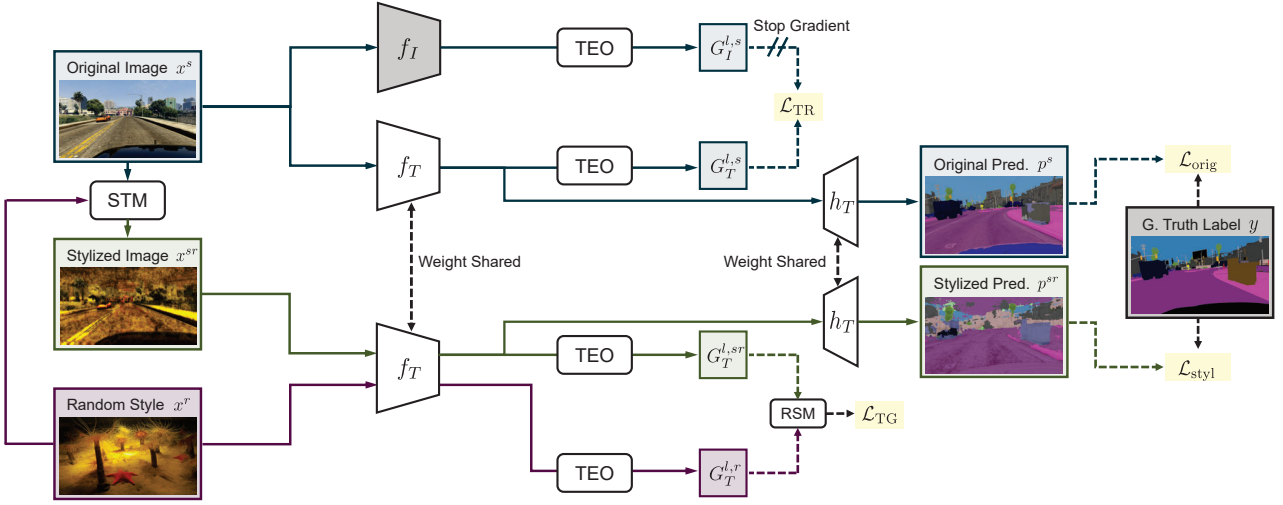
Figure 6. An overview of our proposed Texture Learning Domain Randomization (TLDR) framework. The stylized source image x^{sr} (green) is obtained by stylizing the original source image x^s (blue) with the random style image x^r (purple). The stylized task loss $\mathcal{L}_{\text{styl}}$ focuses on learning shape from x^{sr}, and the original task loss $\mathcal{L}_{\text{orig}}$ focuses on learning texture from x^s. The texture regularization loss $\mathcal{L}_{\text{TR}}$ enforces the consistency between the Gram-matrices of the ImageNet model f_I and the task model f_T for x^s. The texture generalization loss $\mathcal{L}_{\text{TG}}$ enforces the consistency between the Gram-matrices of the task model f_T for x^{sr} and x^r. Random Style Masking (RSM) selects only the random style features when applying the texture generalization loss.

3.2. Texture Extraction with Gram-matrix

Texture is a regional descriptor that can offer measurements for both local structural (*e.g.*, pattern) and global statistical (*e.g.*, overall distribution of colors) properties of an image [21]. It has been shown that texture can be represented by pairwise correlations between features extracted by DNNs, also known as a Gram-matrix [16, 17, 32].

We use the Gram-matrix to extract only texture features from a feature map. The Gram-matrix $G \in \mathbb{R}^{C \times C}$ for the vectorized matrix $F \in \mathbb{R}^{C \times HW}$ of the feature map is defined as Equation 1. C, H, and W denote the channel, height, and width of the feature map, respectively.

$$G_{i,j} = F_i \cdot F_j, \tag{1}$$

where $\cdot$ represents a dot product, F_i and F_j are the ith and j^{th} row vectors of F, respectively. $G_{i,j}$ is the entry at the ith row and j^{th} column of G. Each entry of the Gram-matrix indicates a pairwise correlation between the features corresponding to a texture feature. In this paper, the operator extracting texture features is called Texture Extraction Operator (TEO). Figure 5b illustrates the process of TEO.

4. Approach

This section describes our proposed Texture Learning Domain Randomization (TLDR) framework. TLDR learns texture features in addition to learning shape features through domain randomization. TLDR consists of four losses: a stylized task loss, an original task loss, a tex-

ture regularization loss, and a texture generalization loss. The stylized task loss focuses on learning shape, and the original task loss focuses on learning texture. The texture regularization loss and the texture generalization loss prevent overfitting to source domain textures. Figure 6 is an overview of TLDR. The frozen ImageNet pre-trained encoder is denoted as f_I, *i.e.*, the ImageNet model. The training encoder is denoted as f_T, *i.e.*, the task model. The training semantic segmentation decoder is denoted as h_T.

4.1. Task Losses

Stylized task loss. We denote an original source image x^s and its semantic label y. x^{sr} is a stylized source image obtained by stylizing an original source image x^s with a random style image x^r. The prediction result for x^{sr} of the model is $p^{sr} = h_T(f_T(x^{sr}))$. Then the stylized task loss $\mathcal{L}_{\text{styl}}$ is given by Equation 2.

$$\mathcal{L}_{\text{styl}} = \text{CE}(p^{sr}, y), \tag{2}$$

where $\text{CE}(\cdot)$ represents the categorical cross-entropy loss. The stylized task loss encourages the model to focus on shape features during training since the texture cues are mostly replaced by random styles [38, 51].

Original task loss. The model struggles to learn texture from the stylized source images as the texture cues are mostly substituted with random styles. To accurately capture the source domain textures, the model is trained on the original source images. The prediction result for x^s of the

model is $p^s = h_T(f_T(x^s))$. The original task loss $\mathcal{L}_{\text{orig}}$ is given by Equation 3.

$$\mathcal{L}_{\text{orig}} = \text{CE}(p^s, y). \tag{3}$$

DNNs tend to prioritize texture cues without restrictions [19, 39]. The original task loss guides the model to concentrate on texture features during training.

4.2. Texture Regularization Loss

If the model is trained on the original source images without regularization, it will overfit the source domain textures [19, 39]. We regularize texture representations of the task model using the ImageNet model, which encodes diverse feature representations [8]. However, it is important to note that ImageNet features include not only texture features, but also other semantic features such as shape features. We therefore assume that regularizing the entire feature may interfere with the task model learning texture. To address this issue, we propose to apply TEO to extract only texture features from ImageNet features and regularize the task model with the extracted texture features (see Table 5 for ablation).

Let $F_I^{l,s}$ and $F_T^{l,s}$ denote the vectorized feature maps in layer l of the ImageNet model f_I and the task model f_T for x^s, respectively. The Gram-matrices $G_I^{l,s}$ and $G_T^{l,s}$ from $F_I^{l,s}$ and $F_T^{l,s}$ are the texture features of the original source image as seen by f_I and f_T, respectively. The contribution of the lth layer to the texture regularization loss is $\|G_I^{l,s} - G_T^{l,s}\|_2$. The total texture regularization loss is given by Equation 4.

$$\mathcal{L}_{\text{TR}} = \sum_{l=1}^{L} \frac{u_l}{C_l H_l W_l} \|G_I^{l,s} - G_T^{l,s}\|_2, \tag{4}$$

where L is the number of feature map layers, and u_l is a weighting factor for the contribution of the lth layer to $\mathcal{L}_{\text{TR}}$. C_l, H_l, and W_l denote the channel, height and width of the lth layer feature map, respectively. We reduce the value of u_l as l increases, considering that fewer texture features are encoded in feature maps as layers become deeper [25].

4.3. Texture Generalization Loss

We supplement texture learning from the random style images for more diverse texture representations. Since the random style images are unlabeled, the texture representations should be learned self-supervised. Note that the random style image x^r and the stylized source image x^{sr} share some texture features. To encourage learning of diverse texture representations, we induce the texture features of x^{sr} to become more similar to those of x^r as much as possible while preserving source texture features.

Let $F_T^{l,r}$ and $F_T^{l,sr}$ denote the vectorized feature maps in layer l of f_T for the random style image x^r and the stylized source image x^{sr}, respectively. $G_T^{l,r}$ and $G_T^{l,sr}$ are the
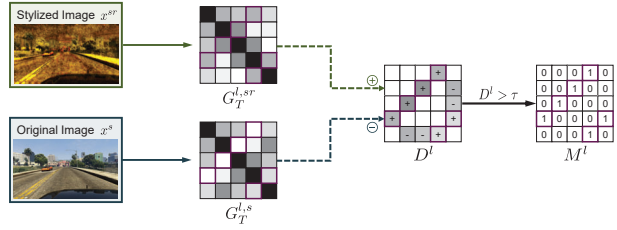

Figure 7. Random Style Masking (RSM) masks only the entries corresponding to the random style features (purple) in the Gram-matrices. The difference between $G_T^{l,sr}$ and $G_T^{l,s}$ is calculated as D^l. The entries with values greater than a certain threshold τ in D^l are masked as M^l.

corresponding Gram-matrices. Our goal is basically to set an objective that makes $G_T^{l,r}$ and $G_T^{l,sr}$ consistent. However, $G_T^{l,sr}$ includes both random style features and remaining source texture features. Applying the constraints to the entire $G_T^{l,sr}$ also imposes the objective on the source texture features in the stylized source image. To select only the random style features when enforcing the consistency between $G_T^{l,r}$ and $G_T^{l,sr}$, we propose Random Style Masking (RSM), inspired by ISW [10]. Figure 7 is an illustration of RSM. We assume that the entries corresponding to the random style features are activated in $G_T^{l,sr}$ but deactivated in $G_T^{l,s}$. Considering $D^l = G_T^{l,sr} - G_T^{l,s}$, the entries corresponding to the random style features are expected to be larger than a certain threshold τ. We denote the mask for the entries representing the random style features as M^l (see Equation 5).

$$M_{i,j}^l = \begin{cases} 1, & \text{if } D_{i,j}^l > \tau \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where i and j are the row and column indices in each matrix, respectively. The threshold τ is determined empirically (see Table 6 for ablation). We only apply the objective to selected random style features by RSM. The contribution of the lth layer to the texture generalization loss is $\|(G_T^{l,r} - G_T^{l,sr}) \odot M^l\|_2$, where $\odot$ represents an element-wise product. The total texture generalization loss is given by Equation 6.

$$\mathcal{L}_{\text{TG}} = \sum_{l=1}^{L} \frac{v_l}{C_l H_l W_l} \|(G_T^{l,r} - G_T^{l,sr}) \odot M^l\|_2, \tag{6}$$

where v_l is a weighting factor for the contribution of the lth layer to $\mathcal{L}_{\text{TG}}$. We also reduce the value of v_l as l increases for the same reason as u_l in Equation 4.

4.4. Full Objective

As the training progresses, the task losses decrease while the texture regularization loss and the texture generalization loss remain relatively constant. We add a Linear Decay Factor (LDF) to the texture regularization and generalization losses to balance their scale with the task losses

Method	Encoder	C	B	M	S
DRPC [59]		37.42	32.14	34.12	-
RobustNet [10]		36.58	35.20	40.33	28.30
SAN-SAW [44]		39.75	37.34	41.86	30.79
SiamDoGe [54]	ResNet-50	42.96	37.54	40.64	28.34
WildNet [28]		44.62	38.42	46.09	31.34
SHADE [60]		44.65	39.28	43.34	-
TLDR (ours)		**46.51**	**42.58**	**46.18**	**36.30**
DRPC [59]		42.53	38.72	38.05	29.67
GTR [45]		43.70	39.60	39.10	29.32
FSDR [24]		44.80	41.20	43.40	-
SAN-SAW [44]	ResNet-101	45.33	41.18	40.77	31.84
WildNet [28]		45.79	41.73	47.08	32.51
SHADE [60]		46.66	43.66	45.50	-
TLDR (ours)		**47.58**	**44.88**	**48.80**	**39.35**

Table 1. Comparison of mIoU (%; higher is better) between DGSS methods trained on GTA and evaluated on C, B, M, and S. The best and second best results are **highlighted** and underlined, respectively. Our method is marked in gray .

Method	Encoder	C	B	M	G
DRPC [59]		35.65	31.53	32.74	28.75
SAN-SAW [44]	ResNet-50	38.92	**35.24**	34.52	29.16
TLDR (ours)		**41.88**	34.35	**36.79**	**35.90**
DRPC [59]		37.58	34.34	34.12	29.24
GTR [45]		39.70	35.30	36.40	28.71
FSDR [24]	ResNet-101	40.80	**39.60**	37.40	-
SAN-SAW [44]		40.87	35.98	37.26	30.79
TLDR (ours)		**42.60**	35.46	**37.46**	**37.77**

Table 2. Comparison of mIoU (%; higher is better) between DGSS methods trained on SYNTHIA and evaluated on C, B, M, and G. The best and second best results are **highlighted** and underlined, respectively. Our method is marked in gray .

(see Table 6 for ablation). The LDF at iteration t is set $w(t) = (1 - t/t_{\text{total}})$, where t_{total} denotes the total number of iterations. Our full objective is given by Equation 7.

$$\mathcal{L}_{\text{total}} = \alpha_{\text{orig}}\mathcal{L}_{\text{orig}} + \alpha_{\text{styl}}\mathcal{L}_{\text{styl}} + w(t)\mathcal{L}_{\text{TR}} + w(t)\mathcal{L}_{\text{TG}}, \tag{7}$$

where α_{orig} and α_{styl} are the weights for the original task loss and the stylized task loss, respectively.

5. Experiments

5.1. Implementation Details

Datasets. As synthetic datasets, GTA [47] consists of 24,966 images with a resolution of 1914×1052. It has 12,403, 6,382, and 6,181 images for training, validation, and test sets. SYNTHIA [48] contains 9,400 images with a resolution of 1280×760. It has 6,580 and 2,820 images for training and validation sets, respectively.

As real-world datasets, Cityscapes [12] consists of 2,975 training images and 500 validation images with a resolution of 2048×1024. BDD [58] contains 7,000 training images and 1,000 validation images with a resolution of 1280×720. Mapillary [40] involves 18,000 training images and 2,000 validation images with diverse resolutions. For brevity, we denote GTA, SYNTHIA, Cityscapes, BDD, and Mapillary as G, S, C, B, and M, respectively.

Network architecture. We conduct experiments using ResNet [23] as an encoder architecture and DeepLabV3+ [7] as a semantic segmentation decoder architecture. In all experiments, encoders are initialized with an ImageNet [13] pre-trained model.

Training. We adopt an AdamW [35] optimizer. An initial learning rate is set to 3×10^{-5} for the encoder and 3×10^{-4} for the decoder, $40k$ training iterations, a batch size of 4. A weight decay is set to 0.01, with a linear warmup [20] over t_{warm}=$1k$ iterations, followed by a linear decay. We use random scaling in the range [0.5, 2.0] and random cropping with a size of 768×768. We apply additional data augmentation techniques, including random flipping and color jittering. We set the texture regularization parameters as u_l=$5 \times 10^{-l-2}$, and the texture generalization parameters as v_l=$5 \times 10^{-l-2}$. The original task loss and the stylized task loss weights are set to α_{orig}=0.5 and α_{styl}=0.5, respectively. We set the RSM threshold τ=0.1.

5.2. Comparison with DGSS methods

To measure generalization capacity in unseen domains, we train on a single source domain and evaluate multiple unseen domains. We conduct experiments on two settings, (1) G→{C, B, M, S} and (2) S→{C, B, M, G}. We repeat each benchmark three times, each time with a different random seed, and report the average results. We evaluate our method using ResNet-50 and ResNet-101 encoders. We use mean Intersection over Union (mIoU) [15] as the evaluation metric. The best and second best results are **highlighted** and underlined in tables, respectively.

Tables 1 and 2 show the generalization performance of models trained on GTA [47] and SYNTHIA [48], respectively. Our TLDR generally outperforms other DGSS methods in most benchmarks. In particular, we improve the G→C benchmark by +1.9 mIoU and +0.9 mIoU on ResNet-50 and ResNet-101 encoders, respectively.

5.3. Texture Awareness of Model

We design experiments to verify whether the performance improvement of TLDR is actually due to being aware of texture. In the experiments, we compare TLDR to plain Domain Randomization (DR), which is only trained with the stylized task loss. We train on GTA using a ResNet-101 encoder. For convenience, we refer to the models trained
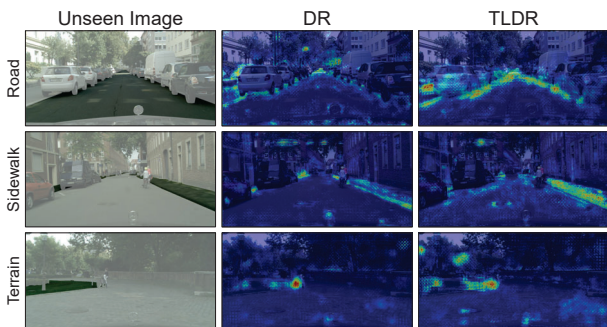
Figure 8. Visualization of class activation maps for different classes using DR and our TLDR. The classes are *road*, *sidewalk*, and *terrain*. Only the classes are displayed as opaque for better visualization in the unseen images. The TLDR model tends to have activation throughout broader areas than the DR model, suggesting it can rely on more texture cues when making predictions.
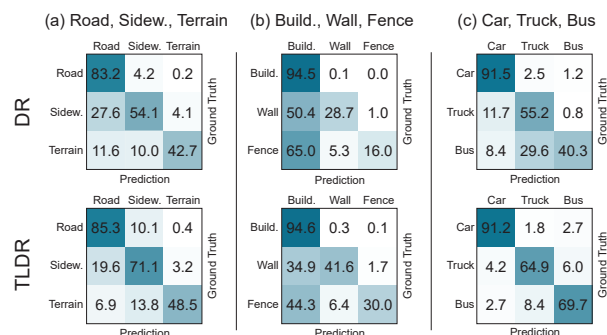

Figure 9. Comparison of the confusion matrices between DR and our TLDR for classes with similar shapes. The TLDR model tends to have less confusion and higher accuracy than the DR model.
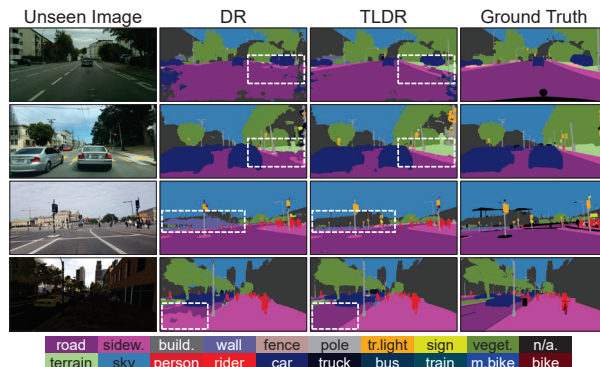

Figure 10. Qualitative results of DR and our TLDR on unseen images. The unseen images are from Cityscapes, BDD, Mapillary, and SYNTHIA in row order. The TLDR model provides better prediction results for shape-similar classes than the DR model, as seen in the white boxes.

Layer	DR		TLDR	
	Texture	Shape	Texture	Shape
$l = 1$	66.5%	29.4%	66.7%	29.2%
$l = 2$	56.6%	39.8%	57.3%	39.1%
$l = 3$	32.0%	63.5%	33.8%	61.9%
$l = 4$	19.1%	54.3%	20.1%	52.5%

Table 3. Layer-wise dimensionality of texture and shape in the latent representations of DR and our TLDR, evaluated in percentages (%). The TLDR model stores more texture information than the DR model across all the layers.

with DR and TLDR as the DR model and the TLDR model, respectively. The models are evaluated on Cityscapes in the experiments if no specific dataset is mentioned.

Class activation map. To validate whether the models utilize texture cues for prediction, we generate class activation maps and analyze the contribution of texture features to the predictions. We apply Grad-CAM [49] to *road*, *sidewalk*, and *terrain* on the DR and TLDR models. Figure 8 displays the attention heat maps generated for each class, highlighting the regions the models focus on during prediction. Our analysis reveals that the DR model tends to have high activation in edge regions, while the TLDR model tends to have activation throughout broader areas. One can infer that these broader areas include texture cues, which the TLDR model uses as valuable prediction cues.

Confusion matrix. We compare confusion matrices between the two models. Figure 9 shows the confusion matrices of the DR model and the TLDR model for shape-similar classes. The TLDR model has lower false positive rates in classifying *sidewalk* and *terrain* as *road*, with rates of

19.6% and 6.9%, respectively, compared to the DR model, which has rates of 27.6% and 11.6% (see Figure 9a). Also, there is an increase in the accuracy of each class. There are also clear reductions in confusion between *building*, *wall* & *fence* and *car*, *truck* & *bus* (see Figures 9b and 9c).

Qualitative results. Figure 10 shows the qualitative results of the DR and TLDR models in various domains. The TLDR model provides better prediction results for shape-similar classes than the DR model (see white boxes).

Dimensionality analysis. We investigate whether the TLDR model encodes additional texture information compared to the DR model. We use the method proposed by Islam *et al.* to quantify the dimensions corresponding to texture and shape in latent representations [25]. The method estimates the dimensionality of a semantic concept by computing mutual information between feature representations of input pairs that exhibit the same semantic concept. A detailed explanation of the experiment is in Appendix A. Table 3 shows the estimated dimensions (%) for texture and shape in each layer of the DR and TLDR models. At every layer, it is apparent that the TLDR model encodes more texture information than the DR model.

The experimental results demonstrate that the TLDR

	$\mathcal{L}_{\text{orig}}$	$\mathcal{L}_{\text{styl}}$	$\mathcal{L}_{\text{TR}}$	$\mathcal{L}_{\text{TG}}$	C	B	M	S
1	✓	-	-	-	36.12	31.38	30.89	37.43
2	-	✓	-	-	39.48	39.21	43.45	31.96
3	✓	✓	-	-	43.62	43.94	44.06	38.75
4	✓	✓	✓	-	46.85	**46.07**	46.52	38.64
5	✓	✓	-	✓	46.36	44.01	47.49	38.10
6	✓	✓	✓	✓	**47.58**	44.88	**48.80**	**39.35**

Table 4. Ablation experiment on each loss in TLDR. The model is trained on GTA using ResNet-101. The best results are **highlighted**. The default setting is marked in gray .

	$\mathcal{L}_{\text{TR}}$	$\mathcal{L}_{\text{TG}}$	TEO	C	B	M	S
1	✓	-	-	44.95	44.09	44.24	**39.07**
2	✓	-	✓	**46.85**	**46.07**	**46.52**	38.64
3	-	✓	-	44.25	43.53	46.90	**38.96**
4	-	✓	✓	**46.36**	**44.01**	47.49	38.10
5	✓	✓	-	45.70	44.13	46.20	38.57
6	✓	✓	✓	**47.58**	**44.88**	**48.80**	**39.35**

Table 5. Ablation experiments on TEO. The model is trained on GTA using ResNet-101. The best results are **highlighted**. The default setting is marked in gray .

	Case	C	B	M	S
1	w/o RSM	46.57	**44.89**	47.17	38.87
2	w/ RSM (τ=0.01)	46.92	44.01	47.75	38.61
3	w/ RSM (τ=0.1)	**47.58**	44.88	**48.80**	**39.35**
4	w/o LDF	46.77	44.84	46.87	**39.72**
5	w/ LDF	**47.58**	**44.88**	**48.80**	39.35

Table 6. Ablation experiments on the design choices of TLDR. The model is trained on GTA using ResNet-101. The best results are **highlighted**. The default setting is marked in gray .

	Random Style	C	B	M	S
1	WikiArt [41]	45.09	44.58	45.16	39.52
2	DTD [11]	45.65	**45.10**	45.29	**39.78**
3	ImageNet val [13]	**47.58**	44.88	**48.80**	39.35

Table 7. Ablation experiment on different random style datasets. The model is trained on GTA using ResNet-101. The best results are **highlighted**. The default setting is marked in gray .

model can effectively distinguish between classes with similar shapes by encoding and using texture as additional discriminative cues.

5.4. Ablation Study

In ablation experiments, we use ResNet-101 as the encoder, train on GTA, and evaluate on C, B, M, and S. The best results are **highlighted** in tables.

Loss components. We investigate how each loss component contributes to overall performance. Table 4 shows the mIoU performance change with respect to the ablation of loss components. There is a large performance improvement when the stylized task loss and the original task loss are used simultaneously (cf. row 3). It seems to be the effect of using shape and texture cues as complementary. There is an additional performance improvement when the texture regularization loss and the texture generalization loss are added separately (cf. rows 4 and 5). The overall performance is the best when both losses are used (cf. row 6).

Texture Extraction Operator (TEO). We use TEO to extract only texture features from feature maps. We conduct experiments to verify whether TEO leads to performance improvement in the texture regularization loss and the texture generalization loss. In the absence of TEO, $\mathcal{L}_{\text{TR}}$ and $\mathcal{L}_{\text{TG}}$ refer to the calculation of direct consistency in the feature maps without any alterations. Table 5 shows the results of TEO ablation experiments. The results show that the performance is improved in the presence of TEO (cf. rows 2, 4, and 6) in both $\mathcal{L}_{\text{TR}}$ and $\mathcal{L}_{\text{TG}}$ compared to the absence of

TEO (cf. rows 1, 3, and 5).

Random Style Masking (RSM). We conduct an experiment to validate the effectiveness of RSM. As shown in Table 6, the model with RSM (cf. row 3) performs better than the model without RSM (cf. row 1). Also, when the threshold of RSM is decreased to τ=0.01 (cf. row 2), the performance is higher than without RSM (cf. row 1) but lower than the default setting τ=0.1 (cf. row 3). The experiment suggests that maintaining an appropriate RSM threshold can aid in learning random styles.

Linear Decay Factor (LDF). When analyzing the texture regularization and generalization losses, we observe that the losses do not naturally decrease as much as the task losses (see Appendix B). The texture regularization loss remains relatively constant likely due to the frozen ImageNet model. The texture generalization loss is also relatively stable likely due to the different combinations of source images and random style images used in each epoch. We design LDF to match the scale of the losses. As shown in Table 6, the model with LDF (cf. row 5) performs better than the model without LDF (cf. row 4).

Random style dataset. We analyze how performance changes when training on various random style datasets. Table 7 shows the performance when using WikiArt [41] dataset, Describable Texture Dataset (DTD) [11] and ImageNet [13] validation set as the random style dataset. The overall performance is the best when using the ImageNet validation set as the random style dataset (cf. row 3). The performances remain consistently demonstrated even using the other two datasets (cf. rows 1 and 2). The performance improvement when using ImageNet is probably due to the superiority of the ImageNet pre-trained model to extract the texture present in ImageNet.

6. Conclusion

Texture often contributes to the domain gap in DGSS. Existing DGSS methods have attempted to eliminate or randomize texture features. However, this paper argues that texture remains supplementary prediction cues for shape despite the domain gap. Accordingly, we proposed TLDR to learn texture features without overfitting to source domain textures in DGSS. TLDR includes novel texture regularization and generalization losses, using a Gram-matrix as a key component. We conducted a diverse set of experiments to demonstrate that TLDR effectively differentiates shape-similar classes by leveraging texture as a prediction cue. The experiments on multiple DGSS tasks show that our TLDR achieves state-of-the-art performance.

Limitation. During the experiments, we found that texture differences between domains vary class-wise. Learning more source domain textures for classes with small texture differences may help with generalization, but it may be advantageous for classes with high texture differences to learn less of them. In our method, there is no class-wise prescription for different texture differences. We leave it as an interesting future work.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(12):2481–2495, 2017. 1

[2] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018. 3

[3] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, (6):679–698, 1986. 2

[4] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2229–2238, 2019. 3

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(4):834–848, 2017. 1

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 1

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 801–818, 2018. 1, 6

[8] Wuyang Chen, Zhiding Yu, Shalini De Mello, Sifei Liu, Jose M Alvarez, Zhangyang Wang, and Anima Anandkumar. Contrastive syn-to-real generalization. *arXiv preprint arXiv:2104.02290*, 2021. 3, 5

[9] Wuyang Chen, Zhiding Yu, Zhangyang Wang, and Animashree Anandkumar. Automated synthetic-to-real generalization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1746–1756, 2020. 3

[10] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11580–11590, 2021. 1, 2, 3, 5, 6

[11] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3606–3613, 2014. 8

[12] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016. 1, 2, 6

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 3, 6, 8

[14] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019. 3

[15] Mark Everingham, SM Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision (IJCV)*, 111(1):98–136, 2015. 6

[16] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 28, 2015. 4

[17] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016. 4

[18] Yunhao Ge, Yao Xiao, Zhi Xu, Xingrui Wang, and Laurent Itti. Contributions of Shape, Texture, and Color in Visual Recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 369–386, 2022. 2

[19] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018. 1, 5

[20] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 6

[21] Robert M Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, 1979. 4

[22] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017. 1

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6

[24] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Fsdr: Frequency space domain randomization for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6891–6902, 2021. 1, 2, 3, 6

[25] Md Amirul Islam, Matthew Kowal, Patrick Esser, Sen Jia, Bjorn Ommer, Konstantinos G Derpanis, and Neil Bruce. Shape or texture: Understanding discriminative features in cnns. *arXiv preprint arXiv:2101.11604*, 2021. 5, 7

[26] Deyi Ji, Haoran Wang, Mingyuan Tao, Jianqiang Huang, Xian-Sheng Hua, and Hongtao Lu. Structural and Statistical Texture Knowledge Distillation for Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16876–16885, 2022. 2

[27] Yunpei Jia, Jie Zhang, Shiguang Shan, and Xilin Chen. Single-side domain generalization for face anti-spoofing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8484–8493, 2020. 3

[28] Suhyeon Lee, Hongje Seong, Seongwon Lee, and Euntai Kim. WildNet: Learning Domain Generalized Semantic Segmentation from the Wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9936–9946, 2022. 1, 2, 3, 6

[29] Lei Li, Ke Gao, Juan Cao, Ziyao Huang, Yepeng Weng, Xiaoyue Mi, Zhengze Yu, Xiaoya Li, and Boyang Xia. Progressive domain expansion network for single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 224–233, 2021. 3

[30] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 453–468, 2018. 3

[31] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 624–639, 2018. 3

[32] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. *arXiv preprint arXiv:1701.01036*, 2017. 4

[33] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(12):2935–2947, 2017. 3

[34] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015. 1

[35] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[36] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 10–18, 2013. 3

[37] Gilhyun Nam, Gyeongjae Choi, and Kyungmin Lee. GCISG: Guided Causal Invariant Learning for Improved Syn-to-Real Generalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 656–672, 2022. 3

[38] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8690–8699, 2021. 2, 3, 4

[39] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:23296–23308, 2021. 1, 5

[40] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4990–4999, 2017. 2, 6

[41] Kiri Nichol. Painter by numbers, wikiart. *Kiri Nichol*, 2016. 8

[42] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 464–479, 2018. 1, 2, 3

[43] Xingang Pan, Xiaohang Zhan, Jianping Shi, Xiaoou Tang, and Ping Luo. Switchable whitening for deep representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1863–1871, 2019. 1, 2, 3

[44] Duo Peng, Yinjie Lei, Munawar Hayat, Yulan Guo, and Wen Li. Semantic-aware domain generalized segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2594–2605, 2022. 1, 2, 3, 6

[45] Duo Peng, Yinjie Lei, Lingqiao Liu, Pingping Zhang, and Jun Liu. Global and local texture randomization for synthetic-to-real semantic segmentation. *IEEE Transactions on Image Processing*, 30:6594–6608, 2021. 2, 3, 6

[46] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12556–12565, 2020. 3

[47] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Proceedings of the European conference on computer vision (ECCV)*, pages 102–118, 2016. 1, 2, 6

[48] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3234–3243, 2016. 6

[49] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 618–626, 2017. 7

[50] Rui Shao, Xiangyuan Lan, Jiawei Li, and Pong C Yuen. Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10023–10031, 2019. 3

[51] Nathan Somavarapu, Chih-Yao Ma, and Zsolt Kira. Frustratingly simple domain generalization via image stylization. *arXiv preprint arXiv:2006.11207*, 2020. 2, 3, 4

[52] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research (JMLR)*, 9(11), 2008. 2

[53] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018. 3

[54] Zhenyao Wu, Xinyi Wu, Xiaoping Zhang, Lili Ju, and Song Wang. SiamDoGe: Domain Generalizable Semantic Segmentation Using Siamese Network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 603–620, 2022. 1, 2, 3, 6

[55] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:12077–12090, 2021. 1

[56] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. A fourier-based framework for domain generaliza-tion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14383–14392, 2021. 3

[57] Zheng Xu, Wen Li, Li Niu, and Dong Xu. Exploiting low-rank structure from latent domains for domain generalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 628–643, 2014. 3

[58] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2636–2645, 2020. 2, 6

[59] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2100–2110, 2019. 1, 2, 3, 6

[60] Yuyang Zhao, Zhun Zhong, Na Zhao, Nicu Sebe, and Gim Hee Lee. Style-hallucinated dual consistency learning for domain generalized semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 535–552, 2022. 1, 2, 3, 6

[61] Zhun Zhong, Yuyang Zhao, Gim Hee Lee, and Nicu Sebe. Adversarial style augmentation for domain generalized urban-scene segmentation. *arXiv preprint arXiv:2207.04892*, 2022. 2, 3

[62] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. *arXiv preprint arXiv:2104.02008*, 2021. 3

[63] Lanyun Zhu, Deyi Ji, Shiping Zhu, Weihao Gan, Wei Wu, and Junjie Yan. Learning statistical texture for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12537–12546, 2021. 2