RobustNet: Improving Domain Generalization in Urban-Scene Segmentation via Instance Selective Whitening

Sungha Choi[*1,3] Sanghun Jung[*2] Huiwon Yun[4] Joanne T. Kim[3]
Seungryong Kim[3] Jaegul Choo[2]

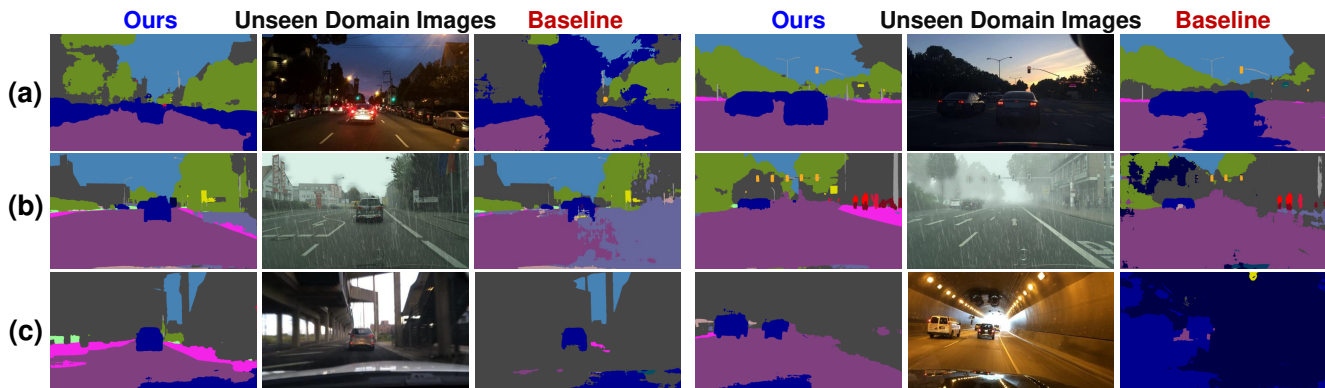[1]LG AI Research [2]KAIST [3]Korea University [4]Sogang University

Figure 1. Segmentation results on *unseen* domains (*i.e.,* BDD-100K [60] and RainCityscapes [19]) with the models trained on Cityscapes [9]. Note that Cityscapes does not contain the following types of images: (a) low-illuminated, (b) rainy, and (c) unexpected scenes. Our method makes reasonable predictions in these three cases, while the baseline [5] model completely fails on them.

Abstract

Enhancing the generalization capability of deep neural networks to unseen domains is crucial for safety-critical applications in the real world such as autonomous driving. To address this issue, this paper proposes a novel instance selective whitening loss to improve the robustness of the segmentation networks for unseen domains. Our approach disentangles the domain-specific style and domain-invariant content encoded in higher-order statistics (i.e., feature covariance) of the feature representations and selectively removes only the style information causing domain shift. As shown in Fig. 1, our method provides reasonable predictions for (a) low-illuminated, (b) rainy, and (c) unseen structures. These types of images are not included in the training dataset, where the baseline shows a significant performance drop, contrary to ours. Being simple yet effective, our approach improves the robustness of various backbone networks without additional computational cost. We conduct extensive experiments in urban-scene segmentation and show the superiority of our approach to existing work. Our code is available at this link[1].

---

* indicates equal contribution
[1] https://github.com/shachoi/RobustNet.

1. Introduction

When deploying deep neural networks (DNNs) trained on a *given* dataset (*i.e.,* source domain) in real-world *unseen* data (*i.e.,* target domain), DNNs often fail to perform properly due to the domain shift. Overcoming this issue is crucial, especially for safety-critical applications such as autonomous driving. In particular, real-world data consist of unexpected and unseen samples, for example, those images taken under diverse illumination, adverse weather conditions, or from different locations. It is generally impossible to model such a full data distribution with limited training data, so reducing the domain gap between source and target domains has been a long-standing problem in computer vision.

Domain adaptation (DA) is an approach to mitigate the performance degradation caused by such a domain gap [3, 12, 18, 11, 65, 39, 58, 41, 50]. Generally, DA focuses on adapting the source domain distribution to that of the target domain, but it requires access to the samples in the target domain, which limits their applicability. When we set the entire real world as a target domain, it is difficult in pactice to obtain data samples that fully cover the target domain.

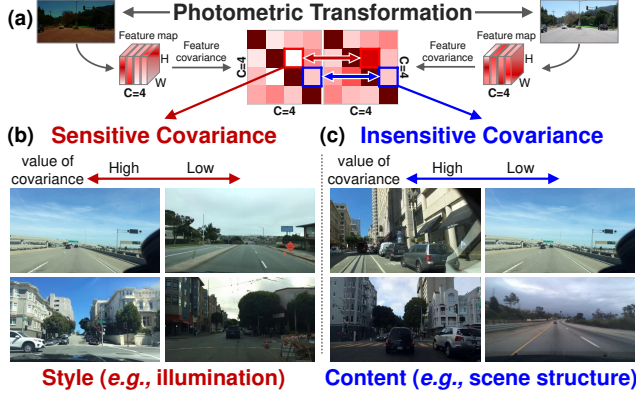Domain generalization (DG) overcomes this limitation

Figure 2. **Overview of our motivation.** (a) We first identify the feature covariance sensitive to the photometric transformation and examine the tendency of the images in each group. (b) Sensitive covariances: Illumination (*i.e.,* style) tends to significantly vary. (c) Insensitive covariances: Sensitive to scene structure differences (*i.e.,* content) but unaffected by the photometric transformation. Accordingly, we aim to *selectively* remove only the style-sensitive covariances that may cause the domain shift.

by improving the robustness of DNNs to arbitrary unseen domains. In general, most DG methods [27, 53, 10, 37, 38, 15, 25, 2, 26, 31] accomplish this through the learning of a shared representation across multiple source domains. However, collecting such multi-domain datasets is costly and labor-intensive, and furthermore, the performance highly depends on the number of source datasets.

A recent study [42] has shown that the DG problem can be addressed by exploiting instance normalization layers [57] instead of relying on multiple source domains, leading to a simple and cost-effective training process. The instance normalization just standardizes features while not considering the correlation between channels. However, a number of studies [13, 14, 28, 54, 49, 34, 7, 43, 55] claim that feature covariance contains domain-specific style such as texture and color. This implies that applying instance normalization to the networks may not be sufficient for domain generalization, because the feature covariance is not considered. A whitening transformation is a technique that removes feature correlation and makes each feature have unit variance. It has been proven that the feature whitening effectively eliminates domain-specific style information as shown in image translation [7], style transfer [28], and domain adaptation [43, 55, 49], and thus it may improve the generalization ability of the feature representation, but not yet fully explored in DG. However, simply adopting the whitening transformation to improve the robustness of DNNs is not straightforward, since it may eliminate domain-specific style and domain-invariant content at the same time. Decoupling the two factors and selectively removing the domain-specific style is the main scope of this paper.

In this paper, we present an instance *selective* whitening

loss that alleviates the limitations of the existing whitening transformation for domain generalization, by selectively removing information that causes a domain shift while maintaining a discriminative power of feature within DNNs. Our method does not rely on an explicit *closed-form* whitening transformation, but implicitly encourage the networks to learn such a whitening transformation through the proposed loss function, thus requiring negligible computational cost. As illustrated in Fig. 2, our method selectively removes only those feature covariances that respond sensitively to photometric augmentation such as color transformation. Our experiments on urban-scene segmentation in DG settings, performed using several backbone networks, show evidence that our approach consistently boosts the DG performance.

The main contributions include the following:

- We propose an instance selective whitening loss for domain generalization, which disentangles domain-specific and domain-invariant properties from higher-order statistics of the feature representation and selectively suppresses domain-specific ones.

- Our proposed loss can easily be used in existing models and significantly improves the generalization ability with negligible computational cost.

- We apply the proposed loss to urban-scene segmentation in a DG setting and show the superiority of our approach over existing approaches in both a qualitative and quantitative manner.

2. Related Work

Domain adaptation and generalization It is well known that significant labeling efforts are required so as to ensure the reliable performance of various tasks such as semantic segmentation [33, 1, 4, 64, 8]. To tackle this challenge, domain adaptation (DA) methods were proposed to transfer the knowledge learned from abundant labeled data (*i.e.,* a source domain) to a target domain where labeled data are scarce. In contrast to DA, domain generalization (DG) methods assume that the model cannot access the target domain during training and aim to improve the generalization ability to perform well in an unseen target domain. Various approaches such as meta-learning [25, 2, 26, 31], adversarial training [27, 30, 44], autoencoder [15, 27], metric learning [10, 37], data augmentation [61, 16, 63] have been proposed to learn domain-agnostic feature representations. Recently, several studies [42, 53] have shown the effectiveness of exploiting both batch normalization (BN) [23] and instance normalization (IN) [57] within DNNs to solve the DG problem. These studies show that BN improves discriminative ability on features, while IN prevents overfitting on training data, so that generalization performance is improved on unseen domains by combining BN and IN. Especially, IBN-Net [42] shows a significant performance improvement with the marginal architectural modification

that incorporates the IN layers through training on a single source domain, unlike most DG methods that require multiple source domains. This normalization based DG method is attractive because it can be applied as a complement to other DG methods based on multiple source domains.

Semantic segmentation in DG   Based on the synthetic data such as GTAV [45] and SYNTHIA [47], numerous DA studies [41, 58, 51, 6, 65, 18, 56, 35, 62] have been proposed in semantic segmentation, but only a few DG studies [61, 42] address semantic segmentation, as the majority of the DG methods mainly focused on image classification. DA, which can access the target domains, generally has better performance than DG, but DG methods that can handle an arbitrary unseen domain without access to the target domain are mandatory in the real world. This paper focuses on the DG method practically helpful in semantic segmentation where various conditions exist such as adverse weather, diverse illumination, location differences, and so on.

Feature covariance   The seminal studies [13, 14] have demonstrated that feature correlations (*i.e.,* a gram matrix or covariance matrix) take style information of images. Since then, numerous studies exploit the feature correlation in style transfer [28], image-to-image translation [7], domain adaptation [49, 55] and networks architecture [34, 43, 20, 54]. Especially, the whitening transformation that removes feature correlation and makes each feature have unit variance, has been known to help to remove the style information from the feature representations [28, 43, 7]. Our work explores the whitening transformation to improve domain generalization performance. To the best of our knowledge, this is the first attempt to apply whitening to DG.

3. Preliminaries

Whitening transformation (WT)   Let $\mathbf{X} \in \mathbb{R}^{C \times HW}$ denote the intermediate feature map, where C is the number of channels, H and W are the spatial dimensions of the feature map, height and width, respectively. WT is a linear transformation that makes the variance term of each channel equal to one and the covariances between each pair of channels equal to zero. A whitening-transformed feature map $\tilde{\mathbf{X}}$ from $\mathbf{X}$ satisfies that $\tilde{\mathbf{X}} \cdot \tilde{\mathbf{X}}^\top = (HW) \cdot \mathbf{I} \in \mathbb{R}^{C \times C}$, where $\mathbf{I}$ denotes the identity matrix, and can be computed as

$$\tilde{\mathbf{X}} = \boldsymbol{\Sigma}_\mu^{-\frac{1}{2}} \left(\mathbf{X} - \boldsymbol{\mu} \cdot \mathbf{1}^\top \right), \tag{1}$$

where $\mathbf{1} \in \mathbb{R}^{HW}$ is a column vector of ones, and $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}_\mu$ are the mean vector and the covariance matrix, respectively, *i.e.,*

$$\boldsymbol{\mu} = \frac{1}{HW} \mathbf{X} \cdot \mathbf{1} \in \mathbb{R}^{C \times 1}, \tag{2}$$

$$\boldsymbol{\Sigma}_\mu = \frac{1}{HW} \left(\mathbf{X} - \boldsymbol{\mu} \cdot \mathbf{1}^\top \right) \left(\mathbf{X} - \boldsymbol{\mu} \cdot \mathbf{1}^\top \right)^\top \in \mathbb{R}^{C \times C}. \tag{3}$$

Since the covariance matrix $\boldsymbol{\Sigma}_\mu$ can be further eigendecomposed such that $\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\top$, where $\mathbf{Q} \in \mathbb{R}^{C \times C}$ is the

orthogonal matrix of eigenvectors, and $\boldsymbol{\Lambda} \in \mathbb{R}^{C \times C}$ is the diagonal matrix that contains each eigenvalue of the corresponding eigenvector from $\mathbf{Q}$, we can calculate an inverse square root of the covariance matrix $\boldsymbol{\Sigma}_\mu^{-\frac{1}{2}}$ as

$$\boldsymbol{\Sigma}_\mu^{-\frac{1}{2}} = \mathbf{Q}\boldsymbol{\Lambda}^{-\frac{1}{2}}\mathbf{Q}^\top. \tag{4}$$

It has been known that WT can effectively remove style information by being applied to each instance in style transfer [28].

Limitations of WT   We can compute the whitening transformation matrix $\boldsymbol{\Sigma}_\mu^{-\frac{1}{2}}$ analytically through Eq. (4), but eigenvalue decomposition is computationally expensive, leading to slow training and inference speed and prevents the gradient back-propagation [20, 7]. To alleviate these problems, previous studies have shown that the goal of WT can be achieved without the eigen-decomposition through the whitening loss [7] or approximating the whitening transformation matrix using Newton's iteration [21, 20, 43].

Especially, GDWCT [7] proposes the deep whitening transformation (DWT) that implicitly makes the covariance matrix $\boldsymbol{\Sigma}_\mu$ close to the identity matrix $\mathbf{I}$ by means of the loss defined as

$$\mathcal{L}_{\text{DWT}} = \mathbb{E}[\|\boldsymbol{\Sigma}_\mu - \mathbf{I}\|_1], \tag{5}$$

where $\mathbb{E}$ denotes the arithmetic mean. GDWCT applies this loss to image-to-image translation for more significant style changes than other methods [22, 24] of aligning only the first-order statistics (*i.e.,* channel-wise mean and variance). However, applying these alternative methods of WT to DG is not straightforward. Whitening all covariance elements may diminish feature discrimination [43, 59] and distort the boundary of an object [29, 28] because domain-specific style and domain-invariant content are simultaneously encoded in the covariance of the feature map.

4. Proposed Method

This section presents our approach to solve the domain generalization problem through whitening the feature representation by mitigating undesirable effects of a whitening transformation. Our method disentangles the covariance into the encoded style and content so that only the style information can be selectively removed, thus increasing the domain generalization ability. We firstly propose an instance whitening and instance-relaxed loss in Section 4.1 and then finally propose our novel instance selective whitening loss in Section 4.3.

4.1. Instance Whitening Loss

This subsection describes a series of steps to transform the input feature into the whitening transformed feature as shown in Fig. 3. Note that our method is applied to each instance, not to a mini-batch. Let $\boldsymbol{\Sigma}_{\mu\,(i,i)}$ denote a diagonal
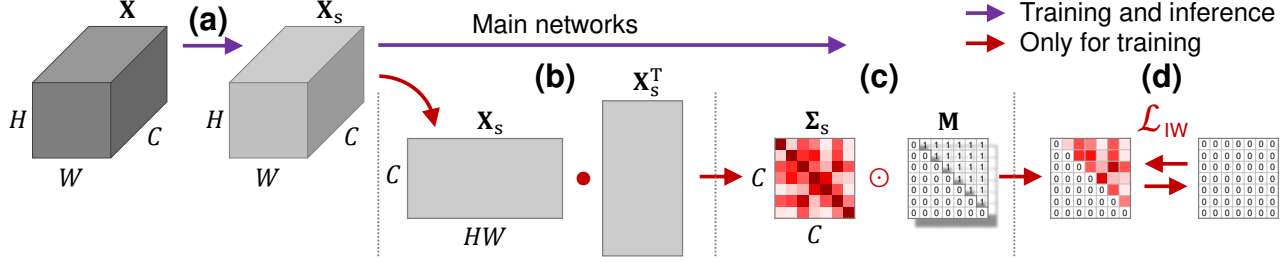
Figure 3. **Overall process of our proposed method.** (a) Instance standardization. (b) Deriving a covariance matrix from a standardized feature map. (c) Leaving only the covariance to which the whitening loss is applied. (d) Applying the criterion that measures the mean absolute error between the remaining covariance values and zero. No additional computation is required for inference as the operations in red are used only for training. Notations; $\mathbf{X}$: intermediate feature map, $\mathbf{X_s}$: standardized feature map, $\boldsymbol{\Sigma_s}$: covariance matrix of the standardized feature map, $\mathbf{M}$: matrix for masking, $\mathcal{L}_{\text{IW}}$: our proposed instance whitening loss.

element (i, i) and $\boldsymbol{\Sigma}_{\mu\,(i,j)}$ denote an off-diagonal element (i, j) of the covariance matrix $\boldsymbol{\Sigma}_\mu$ of the intermediate feature map, where $0 \le i, j < C$, $i \ne j$. The DWT loss in Eq. (5) can be decomposed as

$$\left\| \boldsymbol{\Sigma}_{\mu\,(i,i)} - 1 \right\|_1 = \left\| \frac{\mathbf{x}_i^\top \cdot \mathbf{x}_i}{HW} - 1 \right\|_1 = \left\| \frac{|\mathbf{x}_i||\mathbf{x}_i|\cos 0^\circ}{HW} - 1 \right\|_1 \quad (6)$$

$$\left\| \boldsymbol{\Sigma}_{\mu\,(i,j)} \right\|_1 = \left\| \frac{\mathbf{x}_i^\top \cdot \mathbf{x}_j}{HW} \right\|_1 = \left\| \frac{|\mathbf{x}_i||\mathbf{x}_j|\cos\theta}{HW} \right\|_1, \quad (7)$$

where $\mathbf{x}_i \in \mathbb{R}^{HW}$ denotes the i-th channel of the intermediate feature map $\mathbf{X} \in \mathbb{R}^{C \times HW}$. Note that Eq. (6) applies to the diagonal elements, and Eq. (7) applies to the off-diagonal elements of the covariance matrix. The optimization process for the whitening loss should minimize both Eq. (6) and Eq. (7) simultaneously, but there exists a limitation on it. The scale of each channel (*i.e.,* $|\mathbf{x}_i|$) is forced to increase to the value of $\sqrt{HW}$ by Eq. (6) and decrease to zero by Eq. (7). Therefore, forcing the diagonal and off-diagonal of the covariance matrix to be one and zero, respectively, conflicts with each other, so it is difficult to optimize both at the same time.

To address this issue, the feature map $\mathbf{X}$ can first be standardized into $\mathbf{X_s}$ through an instance normalization [57]:

$$\mathbf{X_s} = (\text{diag}(\boldsymbol{\Sigma}_\mu))^{-\frac{1}{2}} \odot (\mathbf{X} - \boldsymbol{\mu} \cdot \mathbf{1}^\top), \quad (8)$$

where $\odot$ is an element-wise multiplication, and $\text{diag}(\boldsymbol{\Sigma}_\mu) \in \mathbb{R}^{C \times 1}$ denotes the column vector consisting of diagonal elements in the covariance matrix. Note that each diagonal element is copied along with the spatial dimension HW for element-wise multiplication. Since the scale of each feature vector is already fixed as the unit value after the instance standardization, the whitening loss only affects the $\cos\theta$ term in Eq. (7). In the end, this approach fits the purpose of the whitening transformation to decorrelate the features.

After standardization of the intermediate feature map, the covariance matrix is calculated as

$$\boldsymbol{\Sigma_s} = \frac{1}{HW}(\mathbf{X_s})(\mathbf{X_s})^\top \in \mathbb{R}^{C \times C}, \quad (9)$$

where $\mathbf{X_s}$ is the standardized feature map. Thanks to the standardization process, diagonal elements of the covariance matrix are already set as unit values. Thus, we only

need to make the off-diagonals of the covariance matrix close to zero, which makes it easy to optimize for the whitening process, and the aforementioned conflict can thus be resolved. Since the covariance matrix is symmetric, the loss can be applied only to the strict upper triangular part. Our instance whitening (IW) loss is formulated as

$$\mathcal{L}_{\text{IW}} = \mathbb{E}[\|\boldsymbol{\Sigma_s} \odot \mathbf{M}\|_1], \quad (10)$$

where $\mathbb{E}$ denotes the arithmetic mean and $\mathbf{M} \in \mathbb{R}^{C \times C}$ denotes a strict upper triangular matrix, *i.e.,*

$$\mathbf{M}_{i,j} = \begin{cases} 0, & \text{if } i \ge j \\ 1, & \text{otherwise} \end{cases} \quad 0 \le i, j < C. \quad (11)$$

4.2. Margin-based relaxation of whitening loss

The instance whitening loss (Eq. (10)) suppresses all covariance elements to zero, so it can adversely affect the discriminative power of features within DNNs. To address this issue, we propose an instance-*relaxed* whitening (IRW) loss to sustain the covariance elements essential in maintaining the discriminative power. The IRW loss is designed so that the expected value of the total covariance lies within a specified margin δ rather than being close to zero, *i.e.,*

$$\mathcal{L}_{\text{IRW}} = \max(\mathbb{E}[\|\boldsymbol{\Sigma_s} \odot \mathbf{M}\|_1] - \delta, 0) \quad (12)$$

The loss $\mathcal{L}_{\text{IRW}}$ allows the covariance to have a certain level of values, so it gives room to keep discriminative features intact. The empirical effect of the IRW loss can be found in Section 5.2.1. It shows better performance compared to the IW loss not including margin δ (Eq. (10)). Nonetheless, it may not be sufficient because we cannot guarantee that only the covariance useful for generalization performance remains through the margin relaxation.

4.3. Separating Covariance Elements

To further improve our approach, we need to separate the covariance terms into two groups: domain-specific style and domain-invariant content. We propose to selectively suppress only the style-encoded covariances that cause the
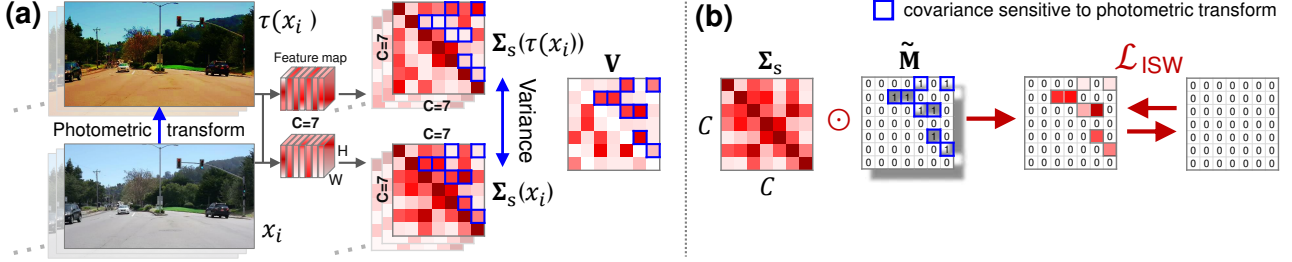
Figure 4. **Instance selective whitening loss**. (a) The variance matrix $\mathbf{V}$ is computed out of the covariance matrices of the i-th image x_i and its photometric transformed image $\tau(x_i)$ to identify those elements sensitive to the transformation (blue boxes). Note that these matrices are symmetric. (b) The covariance matrix $\mathbf{\Sigma_s}$ is masked by the matrix $\tilde{\mathbf{M}}$ to selectively suppress style-sensitive covariances by $\mathcal{L}_{\text{ISW}}$.

domain shift. Assuming that the domain shift includes changes in color and blurriness, we simulate the domain shift through photometric augmentation such as color jittering and Gaussian blurring.

First, we add only the instance standardization layer into the networks (Fig. 3(a)) and train them during the n initial epochs without the whitening loss to get the pure statistics of the covariance matrices from training images. n is a hyper-parameter, which we empirically set to 5. Afterwards, we extract two covariance matrices by inferring from two input images, namely an original and a photometric-transformed image, and calculate the variance matrix from the differences between two different covariance matrices. Formally, the variance matrix $\mathbf{V} \in \mathbb{R}^{C \times C}$ is defined as

$$\mathbf{V} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\sigma}_i^2, \tag{13}$$

from mean $\boldsymbol{\mu}_{\mathbf{\Sigma}_i}$ and variance $\boldsymbol{\sigma}_i^2$ for each element from two different covariance matrices of the i-th image, *i.e.*,

$$\boldsymbol{\mu}_{\mathbf{\Sigma}_i} = \frac{1}{2} \left(\mathbf{\Sigma_s}(x_i) + \mathbf{\Sigma_s}(\tau(x_i)) \right) \tag{14}$$

$$\boldsymbol{\sigma}_i^2 = \frac{1}{2} \left(\left(\mathbf{\Sigma_s}(x_i) - \boldsymbol{\mu}_{\mathbf{\Sigma}_i} \right)^2 + \left(\mathbf{\Sigma_s}(\tau(x_i)) - \boldsymbol{\mu}_{\mathbf{\Sigma}_i} \right)^2 \right) \tag{15}$$

where N denotes the number of image samples, x_i is the i-th image sample, τ is a photometric transformation, and $\mathbf{\Sigma_s}(\cdot)$ extracts the covariance matrix of the intermediate feature map from an input image. As a result, $\mathbf{V}$ consists of elements of the variance of each covariance element across various photometric transformations.

We assume that the variance matrix $\mathbf{V}$ implies the sensitivity of the corresponding covariance to the photometric transformation. This means that the covariance elements with high variance value contain the domain-specific style such as color and blurriness. To identify such elements, we apply k-means clustering on the strict upper triangular elements $\mathbf{V}_{i,j}$ $(i < j)$ of the variance matrix $\mathbf{V}$ to assign the elements into k clusters $C = \{c_1, c_2, \ldots, c_k\}$ with respect to the value. Next, we split the k clusters into two groups, $G_{low} = \{c_1, \ldots, c_m\}$ with low variance value and $G_{high} = \{c_{m+1}, \ldots, c_k\}$ with high variance value. The

hyper-parameters k and m are empirically set to 3 and 1, respectively. More details can be found in the supplementary Section A.2. We assume that G_{high} contains the domain-specific style and G_{low} contains domain-invariant content.

Finally, we propose an instance *selective* whitening (ISW) loss that selectively suppresses only to the style-encoded covariances. Let the mask matrix $\mathbf{M}$ in Eq. (11) change to $\tilde{\mathbf{M}} \in \mathbb{R}^{C \times C}$ for the ISW loss as

$$\tilde{\mathbf{M}}_{i,j} = \begin{cases} 1, & \text{if } \mathbf{V}_{i,j} \in G_{high} \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

The ISW loss is defined as

$$\mathcal{L}_{\text{ISW}} = \mathbb{E}[\|\mathbf{\Sigma_s} \odot \tilde{\mathbf{M}}\|_1]. \tag{17}$$

The networks continue training for the remaining epochs incorporating the proposed ISW loss.

4.4. Network architecture with proposed ISW loss

IBN-Net [42] has explored a number of ResNet [17]-based architectures to combine instance normalization with batch normalization and proposed several IBN blocks based on a residual block (Fig. 5(a)). Among the proposed blocks, IBN-b, which adds an instance normalization layer right after the addition operation of a residual block (Fig. 5(b)),
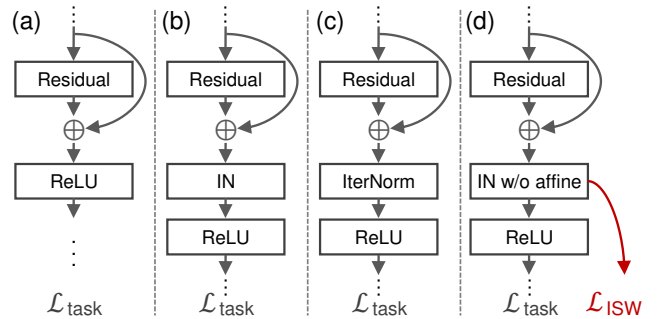

Figure 5. **Architecture comparison with other methods**: (a) the original residual block [17]; (b) IBN-b [42] combining instance normalization with batch normalization; (c) IterNorm [21] employing Newton's iterations for efficient whitening; (d) Our proposed ISW loss applied to instance normalization.

shows the best generalization performance on semantic segmentation tasks. After all, they add three instance normalization layers after the first three convolution groups (*i.e.,* conv1, conv2_x, and conv3_x). We follow this architectural approach as our baseline. As shown in Fig. 5(d), we simply add our proposed ISW loss to the instance normalization layer. Our loss in total is described as

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda(\frac{1}{L}\sum_{i}^{L} \mathcal{L}_{\text{ISW}}^{i}), \qquad (18)$$

where λ denotes the weight of our ISW loss and is empirically set to 0.6, $\mathcal{L}_{\text{task}}$ is the task loss (*e.g.,* a per-pixel cross-entropy loss for semantic segmentation), i indicates the layer index, and L is the number of layers to which the ISW loss is applied. The hyper-parameter λ is analyzed in the supplementary Section A.2. L is set to three by following IBN-Net. An affine transformation is not used since the subsequent convolution operation after a whitening transformation can do the equivalent job, and empirically, we found no performance gain by explicitly adding the affine transformation.

5. Experiments

This section describes the experimental setup and presents evaluation results to assess the effectiveness of our proposed methods on semantic segmentation with comparison to other methods. Furthermore, we provide an in-depth analysis of our results including the covariance matrices.

5.1. Experimental Setup

We train our model on several datasets (*e.g.,* Cityscapes) and show its performance on other datasets (*e.g.,* BDD-100K, Mapillary, GTAV, and SYNTHIA) to measure the generalization capability on unseen domains. For fair comparisons with other normalization techniques, we re-implement IBN-Net [42] and IterNorm [21] on our baseline models and compare them with our methods. As described in Section 4.4, our proposed loss can easily be added to existing models, so we apply our methods to various backbone networks such as ResNet [17], ShuffleNetV2 [36] and MobileNetV2 [52] and show wide applicability of the proposed methods. For all the quantitative experiments, mean Intersection over Union (mIoU) is used to measure the segmentation performance.

5.1.1 Implementation details

We adopt DeepLabV3+ [5] for a semantic segmentation architecture, and SGD optimizer with an initial learning rate of 1e-2 and momentum of 0.9 is used. Besides, we follow the polynomial learning rate scheduling [32] with the power of 0.9. We train all the models for 40K iteration, except for multi-source models, which are trained for 110K iterations. To prevent the model from overfitting, color and positional augmentations such as color jittering, Gaussian blur,

random cropping, random horizontal flipping, and random scaling with the range of [0.5, 2.0] are conducted. For the photometric transformation in ISW, we apply color jittering and Gaussian blur. Also, as suggested by IBN-Net, we add three instance normalization layers after the first three convolution groups and apply our proposed loss. Further details are provided in the supplementary Section A.3.

5.1.2 Datasets

To verify the generalization capability of our methods, we conduct the experiments on five different datasets.

Real-world datasets Cityscapes [9] is a large-scale dataset containing high-resolution (*e.g.,* 2048×1024) urban scene images collected from 50 different cities in primarily Germany. It provides 3,450 finely-annotated images and 20,000 coarsely-annotated images. We use only a finely-annotated set for training and validation. BDD-100K [60] is another real-world dataset that contains diverse urban driving scene images with the resolution of 1280×720. The images are collected from various locations in the US. For a semantic segmentation task, 7,000 training and 1,000 validation images are provided. The last real-world dataset we use is Mapillary [40], a diverse street-view dataset consisting of 25,000 high-resolution images with a minimum resolution of 1920×1080 collected from all around the world.

Synthetic datasets GTAV [45] is a large-scale dataset containing 24,966 driving-scene images generated from Grand Theft Auto V game engine. It has 12,403, 6,382, and 6,181 images of size 1914×1052 for a train, a validation, and a test set, respectively. It has 19 object categories compatible with Cityscapes. Also, we use SYNTHIA [48], composed of photo-realistic synthetic images containing 9,400 samples with a resolution of 960×720.

5.2. Quantitative Evaluation

This subsection provides ablation studies, the comparisons of our results against other normalization methods, the evaluation on multiple source domains, and the analysis of computational cost. Since the experiments follow domain generalization settings, the model cannot access any datasets other than the source data.

5.2.1 Effectiveness of instance selective whitening loss

To verify the effectiveness of our methods, we conduct comparisons with other normalization methods and ablation studies on instance whitening (IW), instance-relaxed whitening (IRW), and instance selective whitening (ISW). **Note that all the experiments in this subsection are performed three times and averaged for fair comparisons**.

Table 1 shows the generalization performance of the models trained on GTAV dataset. ISW outperforms other methods on all datasets except the source dataset (*i.e.,* GTAV). Especially, ISW shows a significant improvement on real-world datasets (i.e., Cityscapes, BDD-100K, and

Models (GTAV)	C	B	M	S || G
Baseline	28.95	25.14	28.18	26.23 || 73.45
†SW [43]	29.91	27.48	29.71	27.61 || **73.50**
†IBN-Net [42]	33.85	32.30	37.75	27.90 || 72.90
†IterNorm [21]	31.81	32.70	33.88	27.07 || 73.19
Ours (IW)	33.21	32.67	37.35	27.57 || 72.06
Ours (IRW)	33.57	33.18	38.42	27.29 || 71.96
Ours (ISW)	**36.58**	**35.20**	**40.33**	**28.30** || 72.10

Table 1. Comparison of mIoU(%). Compared models are trained on GTAV train set, and validated on Cityscapes (C), BDD-100K (B), Mapillary (M), SYNTHIA (S) and GTAV (G) validation sets. ResNet-50 with an output stride of 16 is used. † denotes our own re-implemented models. SW denotes Switchable Whitening [43].

Mapillary). Table 2 shows the generalization performance of those models trained on Cityscapes dataset. Although IterNorm outperforms our models on GTAV, the performance gap is minimal. ISW outperforms other normalization and baseline models on BDD-100K, Mapillary, and SYNTHIA datasets.

Baseline, Switchable Whitening (SW), and IBN-Net, which are less generalizable than our method, tend to overfit the source domain, suffering from performance degradation on the target domain due to the large domain shift. Our method may sacrifice the performance on the source domains (*i.e.,* training and evaluating on the same dataset) as shown in the last column in Table 1 and 2. However, our models shows good generalizability, which is critical when deployed in the wild, where large domain-shift is expected.

Table 3 explains the wide applicability of our work. The first group is reported by adopting ShuffleNetV2, and the second group is using MobileNetV2 as backbone networks. In both cases, our model with ISW outperforms the baseline and IBN-Net on real-world datasets. To further validate the capability of our method, we present the comparison with baselines trained on multiple synthetic domains, GTAV, and SYNTHIA. For the training, we aggregate the training domains without any joint training methodologies. Learning domain-invariant features across multiple datasets is essen-

Models (Cityscapes)	B	M	G	S || C
Baseline	44.96	51.68	42.55	23.29 || **77.51**
†SW [43]	48.49	55.82	44.87	26.10 || 77.30
†IBN-Net [42]	48.56	57.04	45.06	26.14 || 76.55
†IterNorm [21]	49.23	56.26	**45.73**	25.98 || 76.02
Ours (IW)	48.19	58.90	45.21	25.81 || 76.06
Ours (IRW)	48.67	**59.20**	45.64	26.05 || 76.13
Ours (ISW)	**50.73**	58.64	45.00	**26.20** || 76.41

Table 2. Comparison of mIoU(%). The models are trained on Cityscapes train set. ResNet-50 with an output stride of 16 is used. † denotes re-implemented models.

Models (GTAV)	C	B	M	S || G
Baseline	25.56	22.17	28.60	23.33 || **66.47**
†IBN-Net [42]	27.10	31.82	34.89	**25.56** || 65.44
Ours (ISW)	**30.98**	**32.06**	**35.31**	24.31 || 64.99
Baseline	25.92	25.73	26.45	24.03 || **68.12**
†IBN-Net [42]	30.14	27.66	27.07	**24.98** || 67.66
Ours (ISW)	**30.86**	**30.05**	**30.67**	24.43 || 67.48

Table 3. Comparison of mIoU(%). The models are trained on GTAV train set. The backbone networks of the first group are ShuffleNetV2 [36] and the second group is MobileNetV2 [52].

tial to optimize the model on different distributions of multiple datasets. Table 4 shows our model trained on multiple datasets performs better than other models due to its generalization ability by extracting domain-invariant features during training.

Models (G + S)	C	B	M || G	S
Baseline	35.46	25.09	31.94 || 68.48	67.99
IBN-Net	35.55	32.18	38.09 || **69.72**	66.90
Ours	**37.69**	**34.09**	**38.49** || 68.26	**68.77**

Table 4. Comparison of mIoU(%). The models are trained on multiple synthetic domains. The backbone is ResNet-50 with an output stride of 16. † denotes re-implemented models.

5.2.2 Comparison with other DG and DA methods

This subsection compares our method with two existing DG methods on semantic segmentation task, based on the results reported in the papers [42, 61]. DRPC [61] proposes a domain randomization method, which maps the synthetic images to multiple auxiliary real domains using image-to-image translation with the style of real images (*e.g.,* ImageNet). As shown in Table 5, our model gains the largest performance increase on average, compared to other methods such as IBN-Net [42] and DRPC [61]. Our method shows a large amount of performance improvement on BDD-100K and Mapillary datasets that involve significantly more diverse driving scenes than Cityscapes.

In addition, we compare the result of our method with those reported from several domain adaptation methods. See the supplementary Section A.1.

Models (GTAV)	C		B		M	
Baseline IBN-Net [42]	22.20 29.60	7.40 ↑	N/A		N/A	
Baseline DRPC [61]	32.45 **37.42**	4.97↑	26.73 32.14	5.41↑	25.66 34.12	8.46↑
Baseline Ours (ISW)	28.95 36.58	**7.63↑**	25.14 **35.20**	**10.06↑**	28.18 **40.33**	**12.15↑**

Table 5. mIoU(%) comparison with IBN-Net and DRPC trained on GTAV train set. The backbone is ResNet-50. Note that IBN-Net does not report the performance on BDD-100K and Mapillary.

Models	# of Params	GFLOPS	Inference Time (ms)
Baseline	45.082M	554.31	10.48
[†]IBN-Net [42]	45.083M	554.31	10.51
[†]IterNorm [21]	45.081M	554.31	40.31
Ours	45.081M	554.31	10.43

Table 6. Comparison of computational cost. Tested with the image size of 2048×1024 on NVIDIA A100 GPU. The inference time is averaged over 500 trials. [†] denotes re-implemented models.

5.2.3 Computational cost analysis

To ensure our method requires no additional computational cost, we report the number of parameters, GFLOPS, and inference time. As seen in Fig. 5, all the models in Table 6 share the same network architecture, but with different normalization methods. As shown in Table 6, our approach performs a whitening transformation without additional computational cost.

5.3. Qualitative Analysis

Comparison of covariance matrices To show how the covariance matrix is selectively whitened, we visualize the covariance matrix of intermediate feature maps from IBN-Net [42] and our model with ISW. As shown in Fig. 6, the first pair of covariance matrices are from the first convolution layer and the others are from the second convolution layer. Note that the style information mainly exists in the early layers of the network as pointed out in IBN-Net. Moreover, the style information is encoded as a form of the features covariance as revealed in previous studies [13, 14]. Hence, the covariance matrices are sparser at the second pair, compared to the first ones. By comparing the covariance maps from IBN-Net and ISW, we can find the ones from ours are whitened but a small number of covariance elements remain large, showing our ISW selectively eliminates the covariance.

Reconstructing images with whitened features For in-depth analysis, we reconstruct input images from the whitened feature maps of our ISW model. For the experiment, we adopt U-Net [46] as reconstruction networks. To newly train a decoder, we append the decoder to the backbone of a pre-trained baseline and train the decoder. We then replace the backbone network with the pre-trained ISW
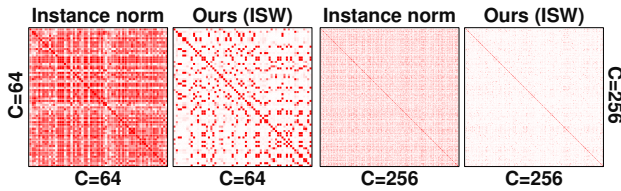

Figure 6. Visualization of covariance matrices extracted from IBN-Net and our model. The first and the second pairs are extracted from the first and the second convolution layers, respectively.

Figure 7. Reconstructed images from ISW-whitened feature maps using U-Net; the first row: a baseline backbone, the second row: an ISW model backbone. The image contents are properly maintained while the style such as illumination and colors vanish.

model. As seen in Fig. 7, generated images preserve the relevant content information for segmentation while the style information such as illumination and colors is suppressed. These examples support the validity of our approach that selectively suppresses the style information.

6. Discussions

In this section, we discuss potential issues and improvements of our approach for further research.

Affine parameters. Most of the normalization layers contain affine parameters to recover the original distribution and enhance the representation of a network. We attempted to deploy this by adding affine parameters or a 1×1 convolution layer after the normalization layer incorporating our proposed whitening loss. Despite our effort, this approach did not improve our method. We conjecture it is because affine parameters or a 1×1 convolution layer do not have sufficient complexity in recovering the original distribution.

Photometric transformation. Our method adopted photometric transformation to separate the style and content information, where we found that applying color transform and Gaussian blur does not harm the content information. We expect our approach can be further improved by exploring various photometric augmentation techniques.

7. Conclusions

This paper proposed a novel instance selective whitening (ISW) loss, which facilitates disentangling the covariances of the intermediate features into the style- and content-related ones and suppressing only the former to learn the domain-invariant feature representation. We focused on solving the domain generalization problem in urban-scene segmentation, which has practical impact when deployed in the wild but has not been studied much. In this regard, we strive to promote the importance of the domain generalization and inspire new research paths in this area.

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)*, 2017. 2

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

[3] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2007. 1

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 2

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

[6] Yuhua Chen, Wen Li, and Luc Van Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[7] Wonwoong Cho, Sungha Choi, David Keetae Park, Inkyu Shin, and Jaegul Choo. Image-to-image translation via group-wise deep whitening-and-coloring transformation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3

[8] Sungha Choi, Joanne T Kim, and Jaegul Choo. Cars can't fly up in the sky: Improving urban-scene segmentation via height-driven attention networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[9] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 6

[10] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2

[11] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning (ICML)*, 2015. 1

[12] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 2016. 1

[13] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. 2, 3, 8

[14] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 3, 8

[15] Muhammad Ghifary, W. Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *International Conference on Computer Vision (ICCV)*, 2015. 2

[16] Rui Gong, Wen Li, Yuhua Chen, and Luc Van Gool. Dlow: Domain flow for adaptation and generalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5, 6

[18] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International Conference on Machine Learning (ICML)*, 2018. 1, 3

[19] Xiaowei Hu, Chi-Wing Fu, Lei Zhu, and Pheng-Ann Heng. Depth-attentional features for single-image rain removal. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[20] Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. Decorrelated batch normalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[21] Lei Huang, Yi Zhou, Fan Zhu, Li Liu, and Ling Shao. Iterative normalization: Beyond standardization towards efficient whitening. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3, 5, 6, 7, 8

[22] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *European Conference on Computer Vision (ECCV)*, 2018. 3

[23] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 2015. 2

[24] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *European Conference on Computer Vision (ECCV)*, 2018. 3

[25] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. *arXiv preprint arXiv:1710.03463*, 2017. 2

[26] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M Hospedales. Episodic training for domain generalization. In *International Conference on Computer Vision (ICCV)*, 2019. 2

[27] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[28] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature

transforms. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2, 3

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

[30] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[31] Yiying Li, Yongxin Yang, Wei Zhou, and Timothy M Hospedales. Feature-critic networks for heterogeneous domain generalization. *arXiv preprint arXiv:1901.11448*, 2019. 2

[32] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Parsenet: Looking wider to see better. In *CoRR*, 2015. 6

[33] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[34] Ping Luo. Learning deep architectures via generalized whitened neural networks. In *International Conference on Machine Learning (ICML)*, 2017. 2, 3

[35] Liqian Ma, Xu Jia, Stamatios Georgoulis, Tinne Tuytelaars, and Luc Van Gool. Exemplar guided unsupervised image-to-image translation with semantic consistency. *arXiv preprint arXiv:1805.11145*, 2018. 3

[36] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *European Conference on Computer Vision (ECCV)*, 2018. 6, 7

[37] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *International Conference on Computer Vision (ICCV)*, 2017. 2

[38] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning (ICML)*, 2013. 2

[39] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[40] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *International Conference on Computer Vision (ICCV)*, 2017. 6

[41] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 3

[42] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *European Conference on Computer Vision (ECCV)*, 2018. 2, 3, 5, 6, 7, 8

[43] Xingang Pan, Xiaohang Zhan, Jianping Shi, Xiaoou Tang, and Ping Luo. Switchable whitening for deep representation learning. In *International Conference on Computer Vision (ICCV)*, 2019. 2, 3, 7

[44] Mohammad Mahfujur Rahman, Clinton Fookes, Mahsa Baktashmotlagh, and Sridha Sridharan. Correlation-aware adversarial domain adaptation and generalization. *Pattern Recognition*, 2020. 2

[45] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision (ECCV)*, 2016. 3, 6

[46] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention (MICCAI)*, 2015. 8

[47] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[49] Subhankar Roy, Aliaksandr Siarohin, Enver Sangineto, Samuel Rota Bulo, Nicu Sebe, and Elisa Ricci. Unsupervised domain adaptation using feature-whitening and consensus loss. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3

[50] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[51] Fatemeh Sadat Saleh, Mohammad Sadegh Aliakbarian, Mathieu Salzmann, Lars Petersson, and Jose M Alvarez. Effective use of synthetic data for urban scene semantic segmentation. In *European Conference on Computer Vision (ECCV)*, 2018. 3

[52] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 6, 7

[53] Seonguk Seo, Yumin Suh, Dongwan Kim, Jongwoo Han, and Bohyung Han. Learning to optimize domain specific normalization for domain generalization. *arXiv preprint arXiv:1907.04275*, 2019. 2

[54] Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening and coloring batch transform for gans. *arXiv preprint arXiv:1806.00420*, 2018. 2, 3

[55] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision (ECCV)*, 2016. 2, 3

[56] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker.

Learning to adapt structured output space for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[57] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. <mark>Instance normalization: The missing ingredient for fast stylization.</mark> *arXiv preprint arXiv:1607.08022*, 2016. 2, 4

[58] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 3

[59] Neha S Wadia, Daniel Duckworth, Samuel S Schoenholz, Ethan Dyer, and Jascha Sohl-Dickstein. Whitening and second order optimization both destroy information about the dataset, and can make generalization impossible. *arXiv preprint arXiv:2008.07545*, 2020. 3

[60] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 6

[61] 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 *International Conference on Computer Vision (ICCV)*, 2019. 2, 3, 7

[62] Yang Zhang, Philip David, and Boqing Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *International Conference on Computer Vision (ICCV)*, 2017. 3

[63] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[64] Yi Zhu, Karan Sapra, Fitsum A Reda, Kevin J Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[65] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *European Conference on Computer Vision (ECCV)*, 2018. 1, 3