Domain Generalization via Balancing Training Difficulty and Model Capability

Xueying Jiang, Jiaxing Huang, Sheng Jin, Shijian Lu*
S-lab, Nanyang Technological University
xueying003@e.ntu.edu.sg
{Jiaxing.Huang, Sheng.Jin, Shijian.Lu}@ntu.edu.sg

Abstract

Domain generalization (DG) aims to learn domain-generalizable models from one or multiple source domains that can perform well in unseen target domains. Despite its recent progress, most existing work suffers from the misalignment between the difficulty level of training samples and the capability of contemporarily trained models, leading to over-fitting or under-fitting in the trained generalization model. We design MoDify, a Momentum Difficulty framework that tackles the misalignment by balancing the seesaw between the model's capability and the samples' difficulties along the training process. MoDify consists of two novel designs that collaborate to fight against the misalignment while learning domain-generalizable models. The first is MoDify-based Data Augmentation which exploits an RGB Shuffle technique to generate difficulty-aware training samples on the fly. The second is MoDify-based Network Optimization which dynamically schedules the training samples for balanced and smooth learning with appropriate difficulty. Without bells and whistles, a simple implementation of MoDify achieves superior performance across multiple benchmarks. In addition, MoDify can complement existing methods as a plug-in, and it is generic and can work for different visual recognition tasks.

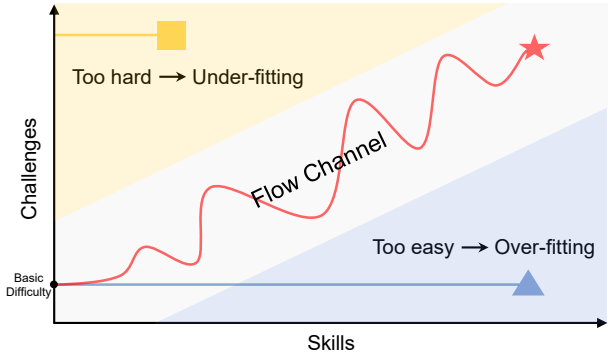

Figure 1. Illustration of the proposed MoDify framework. Training domain-generalizable models often suffer from clear under-fitting (or over-fitting) if keep feeding over-difficult (or over-easy) training samples, especially at the early (or later) training stage, both leading to degraded generalization of the trained models (as illustrated in yellow/blue lines). Inspired by the Flow Theory [16] that a learner usually has better learning outcome when the learner's skill and the task difficulty are well aligned (i.e., lying within the *Flow Channel*), the proposed MoDify schedules the training samples adaptively according to the alignment between the sample difficulty and the capability of contemporarily trained models (as illustrated in red line).

1. Introduction

Deep neural networks (DNNs) [23, 30, 48] have achieved significant progress in recent years with numerous network architectures and learning algorithms designed for various discriminative tasks. In the area of computer vision, DNNs have achieved great success in various visual recognition tasks such as image segmentation [7, 58, 52], object detection [45, 5], etc. However, deep network training often suffers from a misfitting problem, being either over-fitting or under-fitting due to the misalignment between the capacity of networks under training and the complexity of training data. While applying a misfit deep network model to the data from a different domain, the misfitting problem can be greatly enlarged due to the distribution bias and distribution shift across domains.

Domain generalization aims to mitigate the misfitting problem by learning a domain generalizable model that can work well in new domains. It has been widely studied via different augmentation strategies, e.g., domain randomization [41, 56, 26, 51], feature augmentation [31, 40, 11], and data augmentation [61, 39], targeting to obtain generalization capability by *seeing* more training data with various diverse characteristics. However, the aforementioned methods mostly neglect the misalignment between the difficulty level of training samples and the capability of the contem-

*Corresponding author.

porary models along the training process, leading to misfit deep network models and degraded performance.

The Flow Theory [16] has been widely studied in the field of learning and education, which suggests that a learner has optimal learning outcomes when the capability of the learner is well aligned with the difficulty level of learning tasks throughout the learning process. Inspired by this theory, we design **MoDify**, a **Mo**mentum **Dif**ficult**y** framework that aims to tackle the misfitting problem in deep network training. The idea is to dynamically gauge the difficulty level of training samples along the training process, and feed training samples whose difficulty level is well aligned with the capability of the contemporary deep network model under training. This directly leads to a balanced learning process between the difficulty level of training samples and the model capability as illustrated in Fig. 1, which helps mitigate the misfitting problem effectively.

MoDify consists of two novel designs for balanced and smooth learning. The first is MoDify-based **D**ata **A**ugmentation (**MoDify-DA**) that produces augmented training samples with relevant difficulty levels on the fly. The second is MoDify-based **N**etwork **O**ptimization (**MoDify-NO**) that achieves progressive network training by considering the difficulty level of training samples. The two designs work in a collaborative manner to maintain the difficulty-capability balance, which coordinate the augmentation and network training smoothly according to the model's capability. Moreover, we employ an efficient yet effective RGB Shuffle technique that enables online sample augmentation by shuffling the color channels while preserving spatial structures efficiently. RGB Shuffle improves the generalization of the trained model effectively. MoDify has three desirable features: 1) it is generic and performs well across different visual recognition tasks such as image semantic segmentation and object detection; 2) it is an online technique with negligible computational cost; 3) it is complementary with existing DG methods and can be incorporated with consistent performance boosts.

In summary, the contributions of this work are threefold. *First*, we propose MoDify, a novel momentum difficulty framework that effectively addresses the network misfitting problem by maintaining the balance between the difficulty level of training samples and the capability of the contemporary models along the training process. *Second*, we design MoDify-DA and MoDify-NO, the former generates difficulty-aware augmentation samples on the fly while the latter coordinates for a smooth learning process by dropping over-simple samples and postponing over-difficult samples to a later training phase. *Third*, extensive experiments show that a simple implementation of MoDify achieves superior performance consistently across multiple benchmarks and visual recognition tasks.

2. Related Work

Domain generalization (DG) aims to generalize the model learned on one or multiple domains to unseen target domains which have been explored in various computer vision tasks, such as object detection [39, 26, 33], semantic segmentation [39, 56, 41, 26, 31, 61]. Most existing DG methods can be broadly categorized into single-source DG [26, 11, 39, 56, 41, 61, 51, 40, 31, 62] and multi-source DG [60, 36, 35, 28, 2, 32, 20, 19, 18, 11, 61, 43, 59], both targeting to learn domain-invariant feature representations from various aspects, including *domain alignment* [60, 36, 35, 28], *meta-learning* [2, 32, 20, 19, 18] and *augmentation strategies* [26, 11, 39, 56, 41, 61, 51, 40, 31]. Our work belongs to single-source DG, aiming to address a more challenging issue when only one single source domain is available during training.

Single-source DG usually works by domain randomization that augments data [61, 62, 43, 59] or domains [41, 56, 26, 51]. Most existing methods aim to enhance the variation of synthetic images in a source domain by adversarial data augmentation [43, 59] or designing customized modules [61, 62]. However, they largely neglect the misalignment between the difficulty level of training samples and the capability of contemporary models during training, leading to degraded generalization performance. In this work, we design an effective and efficient strategy to address the misfitting problem.

Flow Theory [15, 14, 16], which is a well-established theory in psychology, suggests that optimal learning could be achieved when the skills level of the learner is aligned with the difficulty level of the tasks during learning. It has been extensively studied in the field of education [13, 24] and has more recently been applied to game designs [6, 29]. We introduce the Flow Theory into computer vision research for tackling the domain generalization challenge. The idea is that domain generalization often suffers from unbalance between the difficulty level of training samples and the capability of contemporarily trained models. Flow theory can fit in perfectly by scheduling the training samples according to their difficulties while training domain generalizable networks.

Curriculum Learning [3] is a widely studied learning strategy, which involves starting with easier training samples [50, 9, 22] or sub-tasks [1, 42, 37, 34] and gradually increasing the difficulty level. It has attracted increasing attention recently with the advance of deep learning, and it has been studied in various visual recognition tasks such as domain adaption [57, 10]. For instance, [57] divides the semantic segmentation task into sub-tasks and learns from easy to difficult ones, aiming to decrease the learning difficulty in the early training period. [10] ranks the training samples according to pseudo-label correctness probabilities and learns from them sequentially during training.

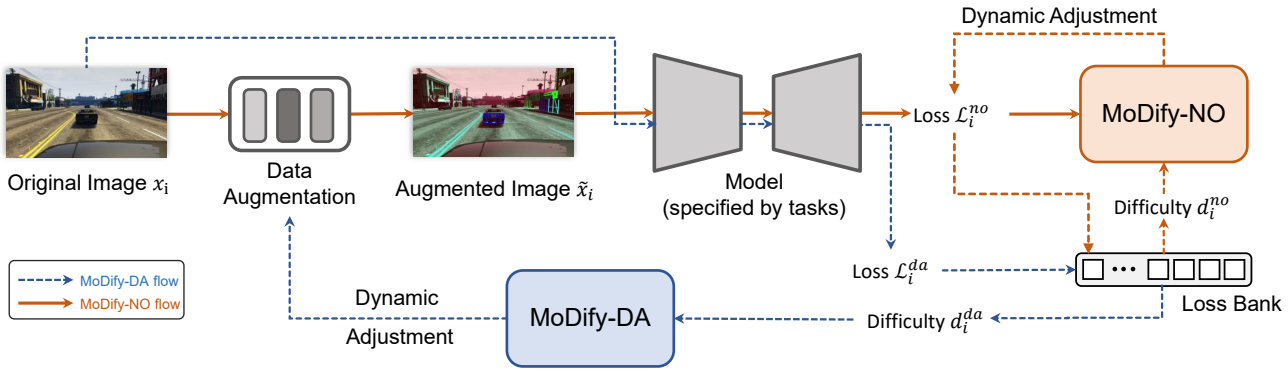

Figure 2. Overall architecture of the proposed Momentum Difficulty (MoDify). In the MoDify-DA flow (highlighted by blue arrows), the network takes the original image x_i as input and generates its loss $\mathcal{L}_i^{da}$, and applies the loss to compute the difficulty level d_i^{da} with the Loss Bank. MoDify-DA dynamically adjusts the strength of data augmentation according to the d_i^{da}. In the MoDify-NO flow (highlighted by red arrows), the network takes the augmented image $\widetilde{x}_i$ as input (with a difficulty level of d_i^{da}). Then the difficulty degree d_i^{no} of the augmented image $\widetilde{x}_i$ is calculated in the same way. MoDify-NO decides whether postpone, drop, or learn from this sample based on the d_i^{no} from the Loss Bank. Noted the sample is fed for training only if its difficulty level is aligned with the model's capability. Additionally, MoDify-DA introduces little computational overhead without involving any back propagation.

However, most existing curriculum learning methods require the pre-defined difficulty levels of the samples in training. We instead dynamically augment and feed difficulty-aware training samples according to the capability of contemporarily trained network models along the training process.

3. Method

This section presents the proposed **Mo**mentum **Dif**ficulty (**MoDify**) framework. First, the problem definition and overview are presented in Sec. 3.1 and Sec. 3.2, respectively. The detailed designs of MoDify are then introduced, including Loss Bank-based Difficulty Assessment in Sec. 3.3 and Difficulty-Aware Training Strategy in Sec. 3.4. Finally, loss functions are presented in Sec. 3.5.

3.1. Problem Definition

Domain generalization (DG) aims to learn a generalizable model (trained in source domain $\mathcal{S}$) that can work in various unseen target domains $\mathcal{T} = \{\mathcal{T}_1, ..., \mathcal{T}_K\}$. During training, only the dataset $D^s = \{(x^s, y^s)\}$ of the source domain is available.

3.2. Momentum Difficulty (MoDify)

The proposed MoDify aims to address the imbalance issue between the difficulty level of augmented training samples and the capability of contemporarily trained models while training domain generalizable networks. It tackles this challenge by augmenting and scheduling difficulty-aware training samples dynamically according to the capability of contemporarily trained network models.

Overview. Fig. 2 illustrates the overall architecture of the proposed MoDify, which includes a loss bank and a difficulty-aware training framework comprising two specific strategies (i.e. MoDify-DA and MoDify-NO).

In the MoDify-DA flow, the original image x_i is fed into the visual task network to obtain its loss $\mathcal{L}_i^{da}$. Then, the loss bank takes $\mathcal{L}_i^{da}$ as input and outputs the difficulty degree d_i^{da} of x_i, where the loss bank is updated in a momentum-based manner. Finally, the augmented samples $\widetilde{x}_i$ are generated based on the difficulty degree d_i^{da}. In the MoDify-NO flow, the difficulty degree d_i^{no} of the augmented image $\widetilde{x}_i$ is obtained using the same approach as in the MoDify-DA flow. The network is updated only when the value of d_i^{no} falls within a moderate range, allowing it to prioritize learning samples with appropriate difficulty.

This dual flow mechanism ensures that the model learns samples whose difficulty is aligned with the model's capability, thus making the training process more efficient and smooth by rejecting undesirably samples.

3.3. Loss Bank-based Difficulty Assessment

The Loss Bank is a crucial component of MoDify that establishes a consistent measure of training samples' difficulty by maintaining a list containing loss values for each sample processed by the visual task network. The size of the Loss Bank is based on the total number of samples in the training dataset, rather than the mini-batch size, providing a global-scale measurement of the training samples' difficulty.

The overall process of Loss Bank updating is shown in Algorithm 1. Specifically, the values of Loss Bank $B = \{V_i \mid i \in \{1, ..., N\}\}$ are defined on-the-fly by a set of data

Algorithm 1 Loss Bank during training

```
Initialization:
B = {V_i = α | i ∈ {1, ..., N}}, epoch j
for j = 1 to j = M do
    for i = 1 to i = N do
        Update V_i by Eqn. 1 and L_i^da
        Update d_i^da by Eqn. 2 and L_i^da
        Update d_i^no by Eqn. 2 and L_i^no
    end for
end for
```

samples. N equals the number of samples in the training dataset. We adopt a momentum manner to update the Loss Bank during training:

$$V_i = \lambda V_i' + (1-\lambda)\mathcal{L}_i^{da}, \tag{1}$$

where V_i' and V_i denote the i-th sample's value of the last epoch and the current epoch separately, and λ is the momentum coefficient.

Difficulty Degree. The proposed loss bank is utilized to assess the difficulty level of the samples, which provides a global and dynamic perspective on the samples' loss values. For each sample x_i, we first fed it into the visual task network and output its loss $\mathcal{L}_i$, where the loss has a different formulation according to the tasks. Then we use the relative rank of the loss $\mathcal{L}_i$ in the loss bank as the difficulty degree, which is formulated as below:

$$d_i = \frac{\sum_{k=1}^{N} I(\mathcal{L}_i < V_k)}{N}, \tag{2}$$

where $V_k \in B$ represents the loss value of x_k in Loss Bank and $I(x)$ is an indicator function.

Remark 1. *MoDify is an efficient training framework using the lightweight and simple Loss Bank. For instance, in comparison to DG methods using image translation GANs [44], which typically utilizes 9 convolutional layers with about 11,000,000 parameters, MoDify is much more efficient, which only utilizes a fixed-length list with approximately N parameters. Here N equals to the size of samples contained in the training dataset.*

3.4. Difficulty-Aware Training Strategy

This subsection introduces the designed difficulty-aware training strategies of MoDify, including MoDify-DA and MoDify-NO. Besides, the data augmentation strategy used in MoDify-DA is also introduced.

MoDify-DA. We propose the MoDify-DA strategy that dynamically adjusts the strength of data augmentation. For each input original image x_i, MoDify-DA calculates its augmentation degree based on the sample's difficulty degree d_i^{da} using Eqn. 2. We utilize $1 - d_i^{da}$ as the augmentation degree of x_i and use it as the probability to augment the input image so that samples with higher difficulty levels remain unchanged and at the same time simpler samples are augmented.

A simple yet effective data augmentation method is utilized to improve the domain invariance in this section. In Domain Generalization (DG) tasks, learning domain-invariant features is crucial for better generalization performance, especially as the source and target domains frequently differ in style and color but share spatial layout similarities. Leveraging spatial information such as edges and shapes can be beneficial. For instance, while the color of a simulated car and a real car may differ, their shape is often similar. Motivated by this observation, we select an appropriate data augmentation method called RGB Shuffle. This method randomly permutes the R, G, and B channels of a training image, effectively altering its style while preserving its structural information.

Compared with offline data augmentation methods [25, 17], MoDify-DA is designed to perform strategy online. In contrast to online data augmentation techniques like [53], which involves multi-round perturbation, and [49], requiring an extra model for parameter selection of data augmentation, MoDify-DA requires just one additional round for deciding the degree of data augmentation, eliminating the need for an auxiliary parameter optimization model. Therefore, MoDify-DA exhibits more efficiency that brings only little computational cost.

MoDify-NO. We propose MoDify-NO strategy that enables the network focus on samples with a moderate difficulty degree. For each input image $\widetilde{x}_i$, MoDify-NO decides whether or not to learn from this sample based on its difficulty level d_i^{no} using Eqn. 2. To achieve this, we dynamically adjust the weight w_i used for the sample's loss function $w_i\mathcal{L}_i$, which is formulated as:

$$w_i = \begin{cases} 1.0, & d_i^{no} \in (T_{easy}, T_{hard}) \\ 0.0, & others \end{cases}, \tag{3}$$

where T_{hard} and T_{easy} represent the thresholds for filtering out samples that are either too easy or too difficult. We set $T_{hard} = 0.95$ and $T_{easy} = 0.05$ in experiments.

Remark 2. *The MoDify framework dynamically adjusts the data augmentation degree of training samples in line with the model's capability. Model capability is gauged by the loss of each iteration: $M_c = 1.0 - \frac{\mathcal{L}_i - \mathcal{L}_{min}}{\mathcal{L}_{max} - \mathcal{L}_{min}}$, where $\mathcal{L}_{max}$ and $\mathcal{L}_{min}$ denote the max and min losses in training. Fig. 3 provides an in-depth explanation. During training, it is noticeable that points with a specific color ranging from red to blue are distributed from the left-bottom corner to the top-right corner, which matches the distribution of the flow channel in Fig. 1. This phenomenon indicates that with the proposed MoDify strategy, the difficulty level of augmented samples increases along with the improvement of the model's capability.*

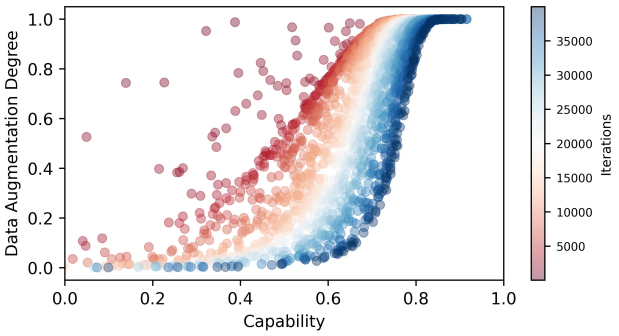

Figure 3. Visualization of the model's capability versus the augmentation degree for new training samples (indicating the difficult level of augmented training samples) along the training iterations. Colors indicate different training iterations, ranging from red to blue as the number of iterations increases. The illustration shows that a low (or high) data augmentation degree is automatically adopted to generate training samples of low (or high) difficult levels at the early (or late) training stage when the capability of contemporarilytrained models is low (or high).

MoDify achieves a balance between the model's capability and training samples' difficulty in an online manner, using only an additional forward pass. This approach improves the model's generalization performance by alleviating over-fitting and under-fitting issues.

3.5. Loss Functions

There are two tasks chosen as instantiation to evaluate the effectiveness of the proposed method, including semantic segmentation and object detection. For the semantic segmentation task, the loss function used to supervise between the predicted segmentation result and the ground truth is the Cross Entropy loss [54]. For the object detection task, two losses are adopted, including the *bounding box loss* and the *classification loss*. Specifically, the *bounding box loss* $\mathcal{L}_{\text{bbox}}$ is the smooth L1 loss [21], and the *classification loss* $\mathcal{L}_{\text{cls}}$ is the Cross Entropy loss [54].

4. Experiments

This section presents experiments including datasets, metrics, and implementation details, domain generalization evaluations for semantic segmentation and object detection tasks, ablation studies, and discussions respectively. More details are described in the ensuing subsections.

4.1. Datasets and Metrics

Datasets. We evaluate MoDify over multiple datasets across different visual DG tasks on semantic segmentation and object detection, which involve two synthetic source datasets including GTAV [46] and SYNTHIA [47] and three real target datasets including Cityscapes [12], BDD100K [55], and Mapillary [38]). GTAV is a large-scale dataset containing 24,966 high-resolution synthetic images with a size of 1914×1052, which shares 19 classes with Cityscapes, BDD100K, and Mapillary. SYNTHIA consists of photo-realistic synthetic images containing 9,400 samples with a resolution of 960×720, which shares 16 classes with the three target datasets. Cityscapes, BDD100K, and Mapillary consist of 2975, 7000, and 18000 real-world training images and 500, 1000, and 2000 validation images respectively.

DG for Semantic Segmentation. We study two synthetic-to-real semantic segmentation tasks, including GTAV $\rightarrow$ {Cityscapes, BDD100K, Mapillary} and SYNTHIA $\rightarrow$ {Cityscapes, BDD100K, Mapillary}.

DG for Object Detection. We evaluate our methods on several DG scenarios for object detection: SYNTHIA $\rightarrow$ { Cityscapes, BDD100K, Mapillary}.

Metrics. The evaluation metric is the mean Intersection-over-Union (mIoU) for the semantic segmentation task and is the mean Average Precision (mAP) with an IoU threshold equals to 0.5 for the object detection task.

4.2. Implementation Details

Semantic Segmentation. We employ DeepLab-V2 [7] as the segmentation model. Two backbones are used for experiments, including ResNet-50 and ResNet-101 [23]. We use SGD [4] with momentum 0.9 as the optimizer. The weight decay is set to $5e^{-4}$ and the learning rate is $2.5e^{-4}$, which is decayed by the polynomial policy [7].

Object Detection. Faster R-CNN [45] is adopted as the detection model. ResNet-101 is used as the backbone. SGD [4] with momentum 0.9 and weight decay $1e^{-4}$ is adopted. The initial learning rate is set to $2e^{-2}$, which is decayed to $2e^{-3}$ and $2e^{-4}$ at the 16 and 22 epochs, respectively.

4.3. Domain Generalizable Semantic Segmentation

We compare MoDify against state-of-the-art DG-based semantic segmentation methods, including IBN-NET [39], DRPC [56], GLTR [41], FSDR [26], WildNet [31], and SHADE [61], on Cityscapes, BDD100K and Mapillary validation sets. The results are reported in Tab. 1 and Tab. 2 using GTAV and SYNTHIA as source domains respectively. Moreover, we compare the methods using two backbones for a fair comparison. The performance is analyzed in detail as follows:

GTAV $\rightarrow$ {Cityscapes, BDD100K, Mapillary}. As shown in Tab. 1, the setting GTAV $\rightarrow$ {Cityscapes, BDD100K, Mapillary} is used for comparison. MoDify achieves the best performance with 46.8% and 44.2% mean mIoU on both backbones. Specifically, when using ResNet-101 as the backbone, MoDify significantly outperforms existing best methods by 2.1%, 0.5%, and 2.0% mIoU on

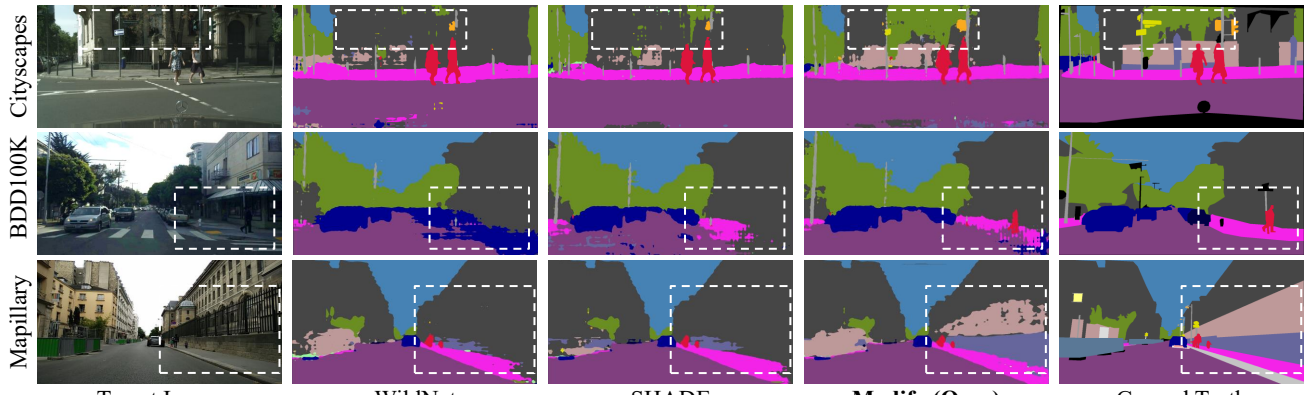

Figure 4. Qualitative illustration of domain generalizable semantic segmentation for GTAV to Cityscapes (Row 1), BDD (Row 2), and Mapillary (Row 3). White boxes highlight regions with clear differences across the compared methods. Compared with other methods, MoDify predicts better building shapes in Row 1, better sidewalk in Row 2, and more accurate fence structures in Row 3.

Net	Method	Cityscapes	BDD100K	Mapillary	Mean
ResNet-101	IBN-Net [39]	37.4	34.2	36.8	36.1
	DRPC [56]	42.5	38.7	38.1	39.8
	GLTR [41]	43.7	39.6	39.1	40.8
	FSDR [26]	44.8	41.2	43.4	43.1
	WildNet [31]	45.8	41.7	47.1	44.9
	SHADE [61]	46.7	43.7	45.5	45.3
	MoDify (Ours)	**48.8**	**44.2**	**47.5**	**46.8**
ResNet-50	SW [40]	29.9	27.5	29.7	29.0
	IterNorm [27]	31.8	32.7	33.9	32.8
	ASG [8]	31.9	N/A	N/A	N/A
	IBN-Net [39]	33.9	32.3	37.8	34.6
	DRPC [56]	37.4	32.1	34.1	34.6
	ISW [11]	36.6	35.2	40.3	37.4
	GLTR [41]	38.6	N/A	N/A	N/A
	SiamDoGe [51]	43.0	37.5	40.6	40.4
	SHADE [61]	44.7	39.3	43.3	42.4
	WildNet [31]	44.6	38.4	46.1	43.0
	MoDify (Ours)	**45.7**	**40.1**	**46.2**	**44.0**

Table 1. Benchmarking Domain generalization over semantic segmentation task GTAV → {Cityscapes, BDD100K, Mapillary} in mIoU. Best in **bold**, second underlined.

Net	Method	Cityscapes	BDD100K	Mapillary	Mean
R101	IBN-Net [39]	37.5	33.0	33.7	34.7
	DRPC [56]	37.6	34.3	34.1	35.4
	GLTR [41]	39.7	35.3	36.4	37.1
	FSDR [26]	40.8	37.4	39.6	39.3
	MoDify (Ours)	**43.4**	**39.5**	**42.3**	**41.7**
R50	DRPC [56]	35.7	31.5	32.7	33.3
	MoDify (Ours)	**38.9**	**33.7**	**36.2**	**36.3**

Table 2. Benchmarking domain generalization over semantic segmentation task SYNTHIA → {Cityscapes, BDD100K, Mapillary} in mIoU. R50 and R101 represent ResNet-50 and ResNet-101, respectively. Best in **bold**, second underlined.

Method	Cityscapes	BDD100K	Mapillary	Mean
Faster R-CNN [45]	24.3	20.1	20.8	21.7
IBN-Net[39]	30.1	23.1	22.3	25.1
FSDR[26]	33.5	25.2	24.9	27.8
MoDify (Ours)	**37.0**	**26.1**	**26.9**	**30.0**

Table 3. Benchmarking domain generalization over object detection task SYNTHIA → {Cityscapes, BDD100K, Mapillary} in mIoU. Faster-RCNN with ResNet-101 is the base framework for all methods. Best in **bold**, second underlined.

the Cityscapes, BDD100K, and Mapillary datasets, respectively. Moreover, MoDify achieves better performance with ResNet-50, surpassing the second-best methods with 1.8%, 1.7%, and 0.1% mIoU on three datasets, respectively.

SYNTHIA → {Cityscapes, BDD100K, Mapillary}. Tab. 2 shows the results under the setting SYNTHIA → {Cityscapes, BDD100K, Mapillary}. We can see that MoDify achieves the best performance on ResNet-50 and ResNet-101 backbones. Specifically, MoDify (ResNet-50) outperforms previous methods with 3.2%, 2.2%, and 3.5% mIoU on the three datasets, respectively. Moreover, MoDify (ResNet-101) improves 2.6%, 2.1%, and 2.7% mIoU as compared with the second-best performance.

As the two tables show, MoDify outperforms all state-of-the-art DG methods clearly and consistently across both tasks and two network backbones. The superior segmentation performance is largely attributed to the proposed MoDify which balances the model's capability and the samples' difficulties along the training process, mitigating the misalignment issue effectively. Moreover, qualitative illustrations in Fig. 4 demonstrate the effectiveness of the proposed MoDify which produces better semantic segmentation consistently across different target datasets and domains.

4.4. Domain Generalizable Object Detection

Apart from semantic segmentation, the object detection task is also used to evaluate the effectiveness of the proposed method over the DG-based object detection tasks SYNTHIA → {Cityscapes, BDD100K, Mapillary}. State-of-the-art methods IBN-NET [39] and FSDR [26] are used to compare with MoDify. As shown in Tab. 3, on all

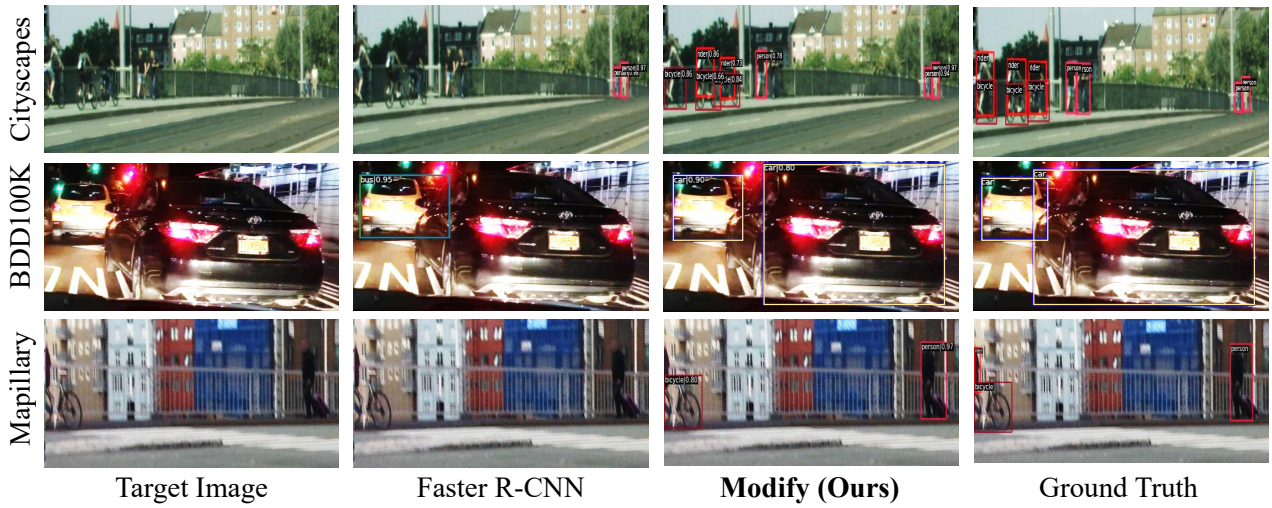

Figure 5. Qualitative illustration of domain generalizable object detection for GTAV to Cityscapes (1st row), BDD (2nd row), and Mapillary (3rd row). The images are cropped and zoomed in for better visualization. In the first row, MoDify performs the best for finding the persons on the sidewalk. In the second row, MoDify predicts both objects accurately with correct categories. In the last row, two objects including a person and a bicycle are predicted correctly by MoDify while Faster R-CNN misses both objects.

Index	RGB Shuffle	MoDify-DA	MoDify-NO	mIoU
1				36.6
2	✓			43.8
3	✓	✓		46.3
4			✓	43.3
5	✓		✓	46.5
6	✓	✓	✓	**48.8**

Table 4. Ablation study of the proposed components in MoDify over the domain generalizable semantic segmentation task GTAV → Cityscapes, using ResNet-101 as the backbone.

three datasets, MoDify beats the second-best performance by 3.5%, 0.9% and 2.0% mAP improvements, respectively. The qualitative comparison results are shown in Fig. 5, showing cases of all three datasets. Compared with the results of other methods, the detection results of MoDify are more precise with fewer false positive predictions.

4.5. Ablation Study

We examine different MoDify designs to find out how they contribute to network generalization in semantic segmentation. Specifically, we trained six models over the UDG task GTAV → Cityscapes, and Tab. 4 presents the corresponding experimental results.

We can observe that the baseline trained with the GTA data only does not perform well due to domain bias. MoDify-DA and MoDify-NO outperform the Baseline by a significant margin, which demonstrates the importance of balancing the seesaw between the model's capability and the samples' difficulty level throughout the training process to achieve domain-generalizable models. When combined with RGB Shuffle, MoDify-NO achieves slightly better results than MoDify-DA, which can be largely attributed to its direct connection to network updates during training. Furthermore, MoDify consistently achieves the best performance. This indicates that MoDify-DA and MoDify-NO are complementary, where the two designs work collaboratively to generate difficulty-aware augmentation samples and coordinate the augmentation and network training smoothly. Besides, the Color shuffle augmentation strategy improves the Baseline by a large margin, demonstrating the effectiveness of this simple yet effective technique.

4.6. Discussions

This section covers three main parts, including analysis on the losses of different methods during training, the compatibility of our method with existing approaches, as well as its computational cost.

Fig. 6 shows the loss curves of the methods using strong data augmentation, no data augmentation, and MoDify during training, as well as the mIoU performance in the target domain. As shown in Fig 6, we can observe that the proposed MoDify balances the model's fitting to the source domain data between the Strong DA-based method and the No DA-based method during training. The method with no data augmentation has the lowest loss during training, but the worst performance due to over-fitting on the source domain (as illustrated in yellow line). Strong data augmentation leads to high loss and sub-optimal performance due to under-fitting on the source domain (as illustrated in red line). Our method achieves the best performance with moderate loss, indicating that MoDify alleviates the misfitting

	Cityscapes		BDD100K		Mapillary	
	Base	+Ours	Base	+Ours	Base	+Ours
ISW [11]	36.6	**40.8**	35.2	**38.0**	40.3	**43.7**
SHADE [61]	44.7	**46.0**	39.3	**40.5**	43.3	**44.2**

Table 5. MoDify's training strategies are complementary to existing domain generalization methods. For the task GTAV $\rightarrow$ {Cityscapes, BDD100K, Mapillary} (using ResNet-50 as the backbone), including MoDify (Ours) consistently boosts the performance of domain generalization.

Models	Params	FLOPs	Inference Time
ISW [11]	45.1M	556.22G	13.6ms
WildNet [31]	45.1M	556.19G	21.3ms
SHADE [61]	45.1M	556.19G	14.5ms
Ours	45.1M	556.19G	13.7ms

Table 6. Comparison of computational cost. The tests are carried out using the image size of 1024 $\times$ 2048 on NVIDIA V100 GPU.

issue during training (as illustrated in green line).

MoDify is complementary to existing domain generalization networks, which can be easily incorporated into them with consistent performance boosts but little extra parameters and computation. We evaluated this feature by incorporating MoDify's training strategies into two competitive domain generalization networks including ISW [11] and SHADE [61] as shown in Tab. 5. During training, we balance the difficulty level of the augmented samples and the capability of networks. As Tab. 5 shows, the incorporation of MoDify improves the semantic segmentation of state-of-the-art networks consistently. As the incorporation of MoDify just includes a training-free loss bank without changing network structures, the inference has few extra parameters and computation once the model is trained.

To validate the computational efficiency of our proposed method, we conducted a detailed analysis of its parameters, floating-point operations per second (FLOPs), and inference time. The results are presented in Tab. 6. It can be observed that our approach incurs only a minor additional computational cost during testing. Additionally, we calculated that the training time required by our proposed method is only 1.4 times that of the baseline method's training time. These findings confirm the practicality and efficiency of our proposed method.

5. Methodology Limitation

MoDify enhances the model's generalization ability by adjusting the strength of image-level data augmentation during the training process, which leads to better performance of visual task models in several challenging scenarios. However, there are certain region-level difficult samples where the performance of the model is relatively poor. To illustrate this, some failure cases are visualized in the supplementary material. Future work could explore a fine-grained region adaptive strategy, applying data augmentation with appropriate levels to different image regions, which is a more targeted approach.

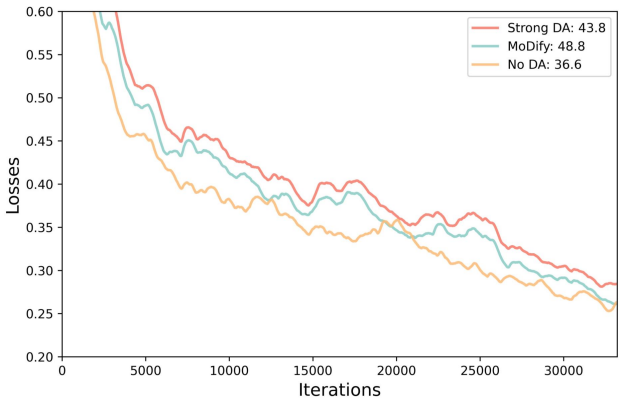

Figure 6. Visualization of the losses during the training process for strong data augmentation (Strong DA), MoDify, and no data augmentation (No DA), respectively. No data augmentation results in low loss but poor performance due to over-fitting. Strong data augmentation leads to high loss and sub-optimal performance due to under-fitting. Our method achieves the best performance with moderate loss, indicating that MoDify alleviates the misfitting issue effectively. Results are obtained on the semantic segmentation task from GTAV [46] to Cityscapes [12] with ResNet-101.

6. Conclusion

This paper presents the Momentum Difficulty (MoDify) technique that tackles domain generalization challenges by mitigating the misalignment between the overall difficulty degree of training samples and the capability of the contemporary deep network model along with the training process. Specifically, we designed MoDify-based Data Augmentation (MoDify-DA) and MoDify-based Network Optimization (MoDify-NO), which coordinate the augmentation and the network training smoothly. The proposed MoDify has three valuable features: 1) it is generic to various visual recognition tasks with consistently superior performance; 2) it is an online yet lightweight technique in various downstream; 3) it complements with existing domain generalization methods with consistent performance boosts.

Acknowledgement

This study is supported under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s).

References

[1] Shuang Ao, Tianyi Zhou, Guodong Long, Qinghua Lu, Liming Zhu, and Jing Jiang. CO-PILOT: COllaborative Planning and reInforcement Learning On sub-Task curriculum. *Advances in Neural Information Processing Systems*, pages 10444–10456, 2021.

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

[3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *International Conference on Machine Learning*, pages 41–48, 2009.

[4] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of Computational Statistics*, pages 177–186. Springer, 2010.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the IEEE/CVF European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[6] Jenova Chen. Flow in games (and everything else). *Communications of the ACM*, (4):31–34, 2007.

[7] 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*, (4):834–848, 2017.

[8] Wuyang Chen, Zhiding Yu, Zhangyang Wang, and Animashree Anandkumar. Automated synthetic-to-real generalization. In *International Conference on Machine Learning*, pages 1746–1756. PMLR, 2020.

[9] Xinlei Chen and Abhinav Gupta. Webly supervised learning of convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1431–1439, 2015.

[10] Jaehoon Choi, Minki Jeong, Taekyung Kim, and Changick Kim. Pseudo-labeling curriculum for unsupervised domain adaptation. *British Machine Vision Conference*, 2019.

[11] 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*, pages 11580–11590, 2021.

[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*, pages 3213–3223, 2016.

[13] Mihaly Csikszentmihalyi. Flow and education. *NAMTA journal*, (2):2–35, 1997.

[14] Mihaly Csikszentmihalyi. Flow and the psychology of discovery and invention. *HarperPerennial, New York*, 1997.

[15] Mihaly Csikszentmihalyi. *Flow: The psychology of happiness*. Random House, 2013.

[16] Mihaly Csikszentmihalyi and Mihaly Csikzentmihaly. *Flow: The psychology of optimal experience*. Harper & Row New York, 1990.

[17] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019.

[18] 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*, 2019.

[19] Yingjun Du, Jun Xu, Huan Xiong, Qiang Qiu, Xiantong Zhen, Cees GM Snoek, and Ling Shao. Learning to learn with variational information bottleneck for domain generalization. In *Proceedings of the IEEE/CVF European Conference on Computer Vision*, pages 200–216. Springer, 2020.

[20] Yingjun Du, Xiantong Zhen, Ling Shao, and Cees GM Snoek. Metanorm: Learning to normalize few-shot batches across domains. In *International Conference on Learning Representations*, 2020.

[21] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1440–1448, 2015.

[22] Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R Scott, and Dinglong Huang. Curriculumnet: Weakly supervised learning from large-scale web images. In *Proceedings of the IEEE/CVF European Conference on Computer Vision*, pages 135–150, 2018.

[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*, pages 770–778, 2016.

[24] Jean Heutte, Fabien Fenouillet, Charles Martin-Krumm, Ilona Boniwell, and Mihaly Csikszentmihalyi. Proposal for a conceptual evolution of the flow in education (eduflow) model. In *European Conference on Positive Psychology*, 2016.

[25] Daniel Ho, Eric Liang, Xi Chen, Ion Stoica, and Pieter Abbeel. Population based augmentation: Efficient learning of augmentation policy schedules. In *International Conference on Machine Learning*, pages 2731–2741. PMLR, 2019.

[26] 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*, pages 6891–6902, 2021.

[27] Lei Huang, Yi Zhou, Fan Zhu, Li Liu, and Ling Shao. Iterative normalization: Beyond standardization towards efficient whitening. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4874–4883, 2019.

[28] 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*, pages 8484–8493, 2020.

[29] Kristian Kiili, Sara De Freitas, Sylvester Arnab, and Timo Lainema. The design principles for flow experience in educational games. *Procedia Computer Science*, pages 78–91, 2012.

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, (6):84–90, 2017.

[31] 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*, pages 9936–9946, 2022.

[32] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M Hospedales. Episodic training for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1446–1455, 2019.

[33] Chuang Lin, Zehuan Yuan, Sicheng Zhao, Peize Sun, Changhu Wang, and Jianfei Cai. Domain-invariant disentangled network for generalizable object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8771–8780, 2021.

[34] Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher-student curriculum learning. *IEEE Transactions on Neural Networks and Learning Systems*, (9):3732–3740, 2019.

[35] Toshihiko Matsuura and Tatsuya Harada. Domain generalization using a mixture of multiple latent domains. In *Proceedings of the AAAI Conference on Artificial Intelligence*, number 07, pages 11749–11756, 2020.

[36] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5715–5725, 2017.

[37] Sanmit Narvekar, Jivko Sinapov, Matteo Leonetti, and Peter Stone. Source task creation for curriculum learning. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 566–574, 2016.

[38] 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*, pages 4990–4999, 2017.

[39] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *Proceedings of the IEEE/CVF European Conference on Computer Vision*, pages 464–479, 2018.

[40] 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*, pages 1863–1871, 2019.

[41] 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*, pages 6594–6608, 2021.

[42] Anastasia Pentina, Viktoriia Sharmanska, and Christoph H Lampert. Curriculum learning of multiple tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5492–5500, 2015.

[43] 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*, pages 12556–12565, 2020.

[44] Mohammad Mahfujur Rahman, Clinton Fookes, Mahsa Baktashmotlagh, and Sridha Sridharan. Multi-component image translation for deep domain generalization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 579–588. IEEE, 2019.

[45] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 2015.

[46] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Proceedings of the IEEE/CVF European Conference on Computer Vision*, pages 102–118. Springer, 2016.

[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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3234–3243, 2016.

[48] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2015.

[49] Zhiqiang Tang, Yunhe Gao, Leonid Karlinsky, Prasanna Sattigeri, Rogerio Feris, and Dimitris Metaxas. Onlineaugment: Online data augmentation with less domain knowledge. In *Proceedings of the IEEE/CVF European Conference on Computer Vision*, pages 313–329. Springer, 2020.

[50] Yunchao Wei, Xiaodan Liang, Yunpeng Chen, Xiaohui Shen, Ming-Ming Cheng, Jiashi Feng, Yao Zhao, and Shuicheng Yan. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (11):2314–2320, 2016.

[51] Zhenyao Wu, Xinyi Wu, Xiaoping Zhang, Lili Ju, and Song Wang. SiamDoGe: Domain generalizable semantic segmentation using siamese network. In *Proceedings of the IEEE/CVF European Conference on Computer Vision*, pages 603–620. Springer, 2022.

[52] 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*, pages 12077–12090, 2021.

[53] Xiaogang Xu and Hengshuang Zhao. Universal adaptive data augmentation. *International Joint Conference on Artificial Intelligence*, 2023.

[54] Ma Yi-de, Liu Qing, and Qian Zhi-Bai. Automated image segmentation using improved pcnn model based on cross-entropy. In *Proceedings of International Symposium on In-*

telligent Multimedia, Video and Speech Processing, pages 743–746. IEEE, 2004.

[55] 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*, pages 2636–2645, 2020.

[56] 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*, pages 2100–2110, 2019.

[57] Yang Zhang, Philip David, Hassan Foroosh, and Boqing Gong. A curriculum domain adaptation approach to the semantic segmentation of urban scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (8):1823–1841, 2019.

[58] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2881–2890, 2017.

[59] Long Zhao, Ting Liu, Xi Peng, and Dimitris Metaxas. Maximum-entropy adversarial data augmentation for improved generalization and robustness. *Advances in Neural Information Processing Systems*, pages 14435–14447, 2020.

[60] Shanshan Zhao, Mingming Gong, Tongliang Liu, Huan Fu, and Dacheng Tao. Domain generalization via entropy regularization. *Advances in Neural Information Processing Systems*, pages 16096–16107, 2020.

[61] 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 IEEE/CVF European Conference on Computer Vision*, pages 535–552. Springer, 2022.

[62] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *Proceedings of the IEEE/CVF European Conference on Computer Vision*, pages 561–578. Springer, 2020.