Generalized Semantic Segmentation by Self-Supervised Source Domain Projection and Multi-Level Contrastive Learning

Liwei Yang[1*], **Xiang Gu**[1*], **Jian Sun**[1,2,3†]

[1] School of Mathematics and Statistics, Xi'an Jiaotong University, China
[2] Pazhou Laboratory (Huangpu), China
[3] Peng Cheng Laboratory, China
{yangliwei, xianggu}@stu.xjtu.edu.cn, jiansun@xjtu.edu.cn

Abstract

Deep networks trained on the source domain show degraded performance when tested on unseen target domain data. To enhance the model's generalization ability, most existing domain generalization methods learn domain invariant features by suppressing domain sensitive features. Different from them, we propose a Domain Projection and Contrastive Learning (DPCL) approach for generalized semantic segmentation, which includes two modules: Self-supervised Source Domain Projection (SSDP) and Multi-Level Contrastive Learning (MLCL). SSDP aims to reduce domain gap by projecting data to the source domain, while MLCL is a learning scheme to learn discriminative and generalizable features on the projected data. During test time, we first project the target data by SSDP to mitigate domain shift, then generate the segmentation results by the learned segmentation network based on MLCL. At test time, we can update the projected data by minimizing our proposed pixel-to-pixel contrastive loss to obtain better results. Extensive experiments for semantic segmentation demonstrate the favorable generalization capability of our method on benchmark datasets.

Introduction

Deep learning (Long, Shelhamer, and Darrell 2015; Chen et al. 2017; Zheng et al. 2021) has achieved breakthroughs in semantic segmentation, benefiting from the large-scale densely-annotated training images. Nonetheless, obtaining the labeled image data for segmentation is time consuming in real life. For instance, labeling a single image with resolution of 2048×1024 in Cityscapes (Cordts et al. 2016) costs 1.5 hours, and even 3.3 hours for adverse weather conditions (Sakaridis, Dai, and Van Gool 2021). An alternative solution is training with synthetic data (Richter et al. 2016; Ros et al. 2016). However, CNNs are sensitive to domain shift and generalize poorly from synthetic to real data.

To deal with this challenge, Domain Adaptation (DA) methods (Zou et al. 2018; Hoffman et al. 2018; Yang and Soatto 2020; Kundu et al. 2022) align the distributions of source and target domains. However, DA assumes that target data is available in the training process which is hard to fulfill in real-life scenarios. Therefore, Domain Generalization (DG) has been widely studied to overcome this limitation. DG aims to learn a model on source domain data which is generalized well on the unseen target domain. The essence of DG is to learn domain-agnostic features (Li et al. 2018a; Dou et al. 2019; Chen et al. 2022a).

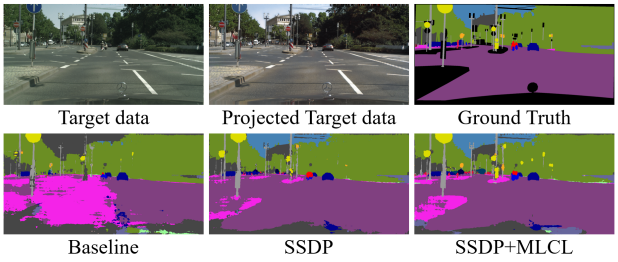

Figure 1: Comparison of results for baseline (DeepLabV3+ with backbone ResNet50), our method using Self-supervised Source Domain Projection (SSDP), and using both SSDP and Multi-Level Contrastive Learning (MLCL).

This work considers Domain Generalization Semantic Segmentation (DGSS), in which we can only use source domain data in training. Existing DGSS methods are mainly divided into three categories. (1) The normalization and whitening-based methods utilize different normalization techniques such as instance normalization or whitening to standardize the feature distribution among different samples (Pan et al. 2018; Choi et al. 2021; Xu et al. 2022; Peng et al. 2022). (2) Generalizable feature learning methods aim to learn domain-agnostic representation (Chen et al. 2021; Kim et al. 2022). (3) Domain randomization-based methods learn synthetic to real generalization by increasing the variety of training data. (Yue et al. 2019; Peng et al. 2021; Lee et al. 2022). However, domain randomization methods require unlabeled auxiliary datasets for generalization.

In this paper, we propose Self-supervised Source Domain Projection (SSDP) and Multi-Level Contrastive Learning (MLCL) schemes for domain generalization semantic segmentation. Specifically, we first design SSDP, aiming to learn a projection to map the unseen target domain data to the source domain by projecting augmented data to its original data in the training phase. Secondly, for augmented data projected to the source domain, we further propose

*These authors contributed equally.
†Corresponding author.

MLCL to learn a better generalizable segmentation network by contrasting the features with the guidance of labels at the pixel, instance and class levels. At test time, given an unlabeled target domain image, we first project it onto the source domain by SSDP, then segment it by the learned semantic segmentation model. Extensive experiments show that our SSDP and MLCL schemes improve the generalization performance of our segmentation model. Figure 1 illustrates an example of segmentation results by the baseline method and its improved versions respectively using SSDP and SSDP+MLCL. Our code is available at https://github.com/liweiyangv/DPCL.

The main contributions can be summarized as follows.

- We propose a Self-supervised Source Domain Projection (SSDP) approach for projecting data onto the source domain, to mitigate domain shift in the test phase.
- We propose a Multi-Level Contrastive Learning (MLCL) scheme, which considers the relationship among pixels features, instance prototypes and class prototypes. In particularly, we propose to deal with pixel-to-pixel contrastive learning as a Transition Probability Matrix matching problem.
- We apply our method to urban-scene segmentation task. Extensive experiments show the effectiveness of our DPCL for domain generalization.

Related Works

Domain Generalization

Domain generalization attempts to improve model generalization ability on the unseen target domain. As for classification task, domain generalization is mainly based on domain alignment of source domains to learn domain-invariant features (Li et al. 2018b; Matsuura and Harada 2020), meta-learning to learn generalizable features (Dou et al. 2019; Chen et al. 2022a), or data augmentation to expand source data to improve generalization capabilities (Li et al. 2021a).

As for semantic segmentation, the existing domain generalization methods can be classified into three categories: 1) Normalization and whitening based-methods utilize Instance Normalization (IN) or Instance Whitening (IW) to standardize global features by erasing the style-specific information and prevent model overfitting on the source domain (Pan et al. 2018; Choi et al. 2021; Xu et al. 2022; Peng et al. 2022). For instance, ISW (Choi et al. 2021) utilizes IW to disentangle features into domain-invariant and domain-specific parts, and normalize domain-specific features. DIRL (Xu et al. 2022) proposes a sensitivity-aware prior module to guide the feature recalibration and feature whitening, and learns style insensitive features. (Peng et al. 2022) designs a semantic normalization and whitening scheme to align category-level features from different data. 2) Generalizable feature learning-based methods focus on learning domain-invariant features, such as utilizing attention mechanism (Chen et al. 2021) or meta-learning framework (Kim et al. 2022). 3) Randomization-based methods synthesize images with different styles to expand source domain (Yue et al. 2019; Peng et al. 2021; Lee et al. 2022). DPRC (Yue et al. 2019) randomizes the synthetic images with the styles of real images and learns a generalizable model. WildNet (Lee et al. 2022) leverages various contents and styles from the wild to learn generalized features. Different from domain randomization methods, which utilize auxiliary dataset to expand source domain data, we adopt a self-supervised scheme to train a source domain projection network, which projects data with different distributions onto the source domain. Based on the projected data, we further propose a multi-level contrastive learning strategy to learn discriminative features.

Test-Time Adaptation

Test-time adaptation aims to improve the performance of source trained model against domain shift with a test-time adaptation strategy. Existing methods are mainly designed for classification task and can be categorized in two ways. (1) Update model's parameters at test time by utilizing self-supervised loss. Tent (Wang et al. 2021a) adopts entropy minimization to fine-tune BN layers in the test phase. Adacontrast (Chen et al. 2022b) conducts test-time contrastive learning and learns a target memory queue to denoise pseudo label. (2) Learn the model to adapt to test data without using extra loss at test time. For example, (Xiao et al. 2022) learns to adapt the model's parameters based on only one test data using the meta-learning framework. Different from the above adaptation schemes, we update the target data by projecting it onto the source domain by our SSDP network, then we iterate the projected data by our pixel-to-pixel contrastive loss, while fixing the parameters of learned models.

Contrastive Learning

Contrastive learning has shown compelling performance in representation learning (Wu et al. 2018; Chen et al. 2020a,b,c). Supervised contrastive learning (Khosla et al. 2020) pulls the sample pairs in the same class closer and pushing away the negative pairs which have different labels to learn discriminative features. (Wang et al. 2021b; Huang et al. 2022) utilizes supervised contrastive learning scheme in semantic segmentation to constrain pixel-level features. Except for pixel-wise contrastive learning, recent works also utilize other contrastive learning for segmentation, such as prototype-wise (Hu, Cui, and Wang 2021; Kwon et al. 2021) or distribution-wise (Li et al. 2021b). Besides traditional InfoNCE loss, (Hendrycks et al. 2020; Englesson and Azizpour 2021) propose to minimize Jensen-Shannon (JS) divergence among the predictive distributions of samples with different augmentation strategies to learn a robust model. Different from recent work, we define multi-level contrastive learning for pixel features, instance prototypes and class prototypes. Specifically, we reformulate pixel-to-pixel contrastive learning based on transition probability matrix, which shows a better generalization ability in the experiments.

Method

In this paper, we focus on a Single-source Domain Generalization (SDG) setting. We denote our source domain as

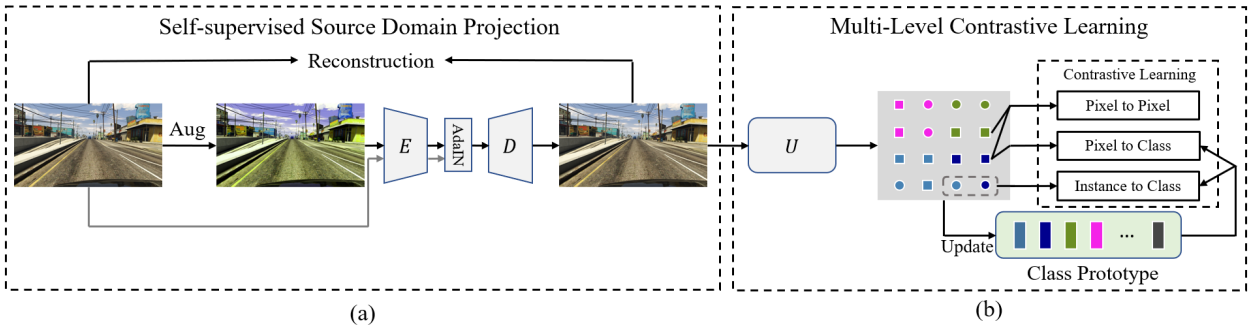

Figure 2: Overview of our proposed DPCL. (a) Self-supervised Source Domain Projection (SSDP) sub-network aims to project data onto the source domain. E, D are the encoder and decoder of SSDP. (b) Multi-Level Contrastive Learning (MLCL) based on projected data attempts to learn discriminative features. U is the feature extractor of segmentation network.

$\mathcal{S}$ and unseen target domain as $\mathcal{T}$. Notably, $\mathcal{T}$ has different distribution with the source domain. $\mathcal{S}$ can be represented as $\{(x_i, y_i)\}_{i=1}^{n}$, where (x_i, y_i) denote the i-th image and its pixel-wise class label, n is the number of samples in $\mathcal{S}$. SDG aims to train the segmentation model on $\mathcal{S}$ and generalize it to the unseen target domain $\mathcal{T}$.

The proposed method DPCL mainly has two components. As shown in Fig. 2, we first utilize a Self-supervised Source Domain Projection (SSDP) block to project data from other distributions to the source domain. Then, we propose a Multi-Level Contrastive Learning (MLCL) scheme to learn discriminative features based on projected data. Next, we will explain our formulation and each module in detail.

Self-Supervised Source Domain Projection

The SSDP aims to project data onto the source domain to mitigate domain shift at test time. Since target data is not available in training, we can not directly obtain a style transfer network from target to source like domain adaptation methods (Hoffman et al. 2018). In this paper, we adopt a data augmentation strategy to generate data with different distributions from the source domain, and project augmented data to its corresponding original data in the source domain.

We denote our SSDP as a mapping $F : \mathcal{T} \rightarrow \mathcal{S}$. Given a target data x, it aims to make $F(x)$ close to the source domain $\mathcal{S}$. Since target domain data is unavailable in training, we use data augmentation over source domain data to simulate domain shift in the training phase. We project the augmented data to original data to learn our SSDP. Specifically, our design of SSDP is shown in Fig. 3. We denote $x_a = A(x)$ as the augmented data, where A is an augmentation function. We input both original data x and augmented data x_a into encoder E of SSDP and get feature f_x and f_{x_a}. As for f_{x_a}, we adopt instance normalization to get normalized feature $\hat{f}_{x_a}$ to eliminate its style information. Meanwhile, we calculate channel-wise standard deviation and mean of feature f_x which contain style information of x as affine parameters to transform normalized feature $\hat{f}_{x_a}$. We assume the transformed feature $\tilde{f}_{x_a}$ contains content information of x_a and style information of x. Then we

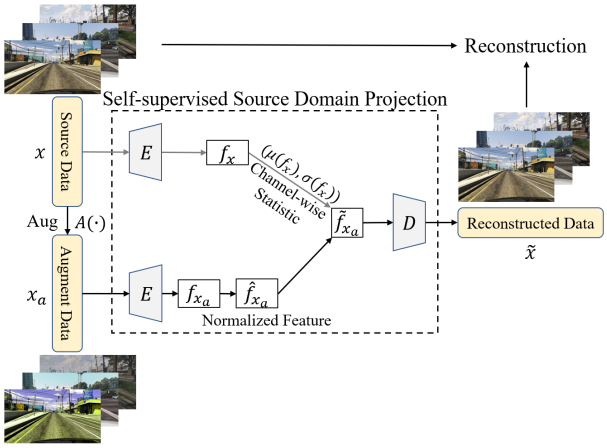

Figure 3: Illustration of Self-supervised Source Domain Projection sub-network. f_x, f_{x_a} are the features of source data x and augment data x_a, $\hat{f}_{x_a}$ is the feature after instance normalization of f_{x_a}. $\mu(f_x), \sigma(f_x)$ are the channel wise mean and standard deviation of f_x. $\tilde{f}_{x_a}$ is the renormalized feature of $\hat{f}_{x_a}$. $\tilde{x}$ is the reconstructed original image.

input $\tilde{f}_{x_a}$ into decoder D to get reconstructed original data $\tilde{x}$. Since we only utilize data augmentation to create sample pair x and x_a, our scheme of training SSDP can be regarded as a self-supervised way.

Formally, we use a standard instance normalization to get the normalized feature $\hat{f}_{x_a}$ by

$$\hat{f}_{x_a} = \frac{f_{x_a} - \mu(f_{x_a})}{\sigma(f_{x_a})} \tag{1}$$

where $\mu(f_{x_a}), \sigma(f_{x_a})$ are the channel-wise mean and standard deviation of feature f_{x_a}, then we use the same statistics of f_x to transform normalized feature $\hat{f}_{x_a}$ by

$$\tilde{f}_{x_a} = \sigma(f_x)\hat{f}_{x_a} + \mu(f_x) \tag{2}$$

Then we input the transformed feature $\tilde{f}_{x_a}$ into decoder D and get reconstructed data $\tilde{x}$. In the experiment,we use L_1

loss for enforcing image reconstruction:

$$\mathcal{L}_{recon} = ||x - \tilde{x}||_1 \tag{3}$$

Different from the traditional autoencoder, the input of our SSDP is the augmented data and original data, the output is the original data in the source domain. We adopt AdaIN (Karras, Laine, and Aila 2019) in the feature space to make SSDP project augmented data to the original data.

In the test phase, we do not have the paired source data x to provide source style information for each target data x_t. We use mean and standard deviation cluster center of source data features to alter $\sigma(f_x)$ and $\mu(f_x)$ in Eq. (2). Specifically, we cluster the mean and standard deviation of training data features into q centers after training over the source domain. We denote mean cluster centers as $\mu_{\mathcal{S}} = \{\mu_1, \mu_2, ..., \mu_q\}$, standard deviation centers as $\sigma_{\mathcal{S}} = \{\sigma_1, \sigma_2, ..., \sigma_q\}$. Given a target data x_t, we use L_2 distance to find the closest center $\hat{\mu}$ of $\mu(f_{x_t})$ in $\mu_{\mathcal{S}}$, *i.e.*,

$$\hat{\mu} = \arg\min_{\tilde{\mu}} ||\tilde{\mu} - \mu(f_{x_t})||_2, \tilde{\mu} \in \mu_{\mathcal{S}} \tag{4}$$

We can get the closest standard deviation center $\hat{\sigma}$ in the same way. Then we use $\hat{\mu}, \hat{\sigma}$ to transform the normalized feature $\hat{f}_{x_t}$ by using Eq. (2), and get the projected data by sending the renormalized feature $\tilde{f}_{x_t}$ into decoder D.

Multi-Level Contrastive Learning

Based on the projected data by SSDP, we further propose a multi-level contrastive learning scheme for learning discriminative features. Using traditional cross-entropy as task loss only penalizes pixel-wise predictions independently but ignores semantic relationships among pixels. To investigate the semantics at different levels and their relations, we propose multi-level contrastive learning for learning model of semantic segmentation in the feature space. Our segmentation model consists of feature extractor U and classifier H.

Different from image classification, semantic segmentation aims to predict class label for each pixel, and there may exist more than one instance in an image to be segmented. We consider the semantic class relationship among multi-level features, including pixel, instance and class levels to learn discriminative and generalizable features. Specifically, we adopt prototype for instance-level and class-level feature representations by average pooling features in each connected region or total area of each class in each image according to the ground truth segmentation mask.

Construction of Class Prototype. Taking class-level prototype as example, we calculate prototype by average pooling the features in each class region:

$$p^k = \frac{\sum_{i=1}^{H'W'} y_{z_i}^k z_i}{\sum_{i=1}^{H'W'} y_{z_i}^k} \tag{5}$$

where H', W' respectively denotes height and width of feature map. y_{z_i} is one-hot label for pixel feature z_i, *i.e.*, $y_{z_i}^k = 1$, when z_i belongs to class k. To obtain the class prototype in the whole training dataset, we update class prototypes using moving average strategy by

$$\hat{P}^k = \gamma P^k + (1 - \gamma) p^k$$

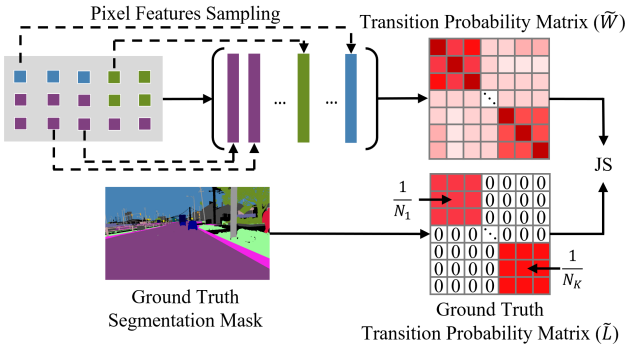

Figure 4: Illustration of pixel-to-pixel contrastive learning.

where $\hat{P}^k$, P^k are the updated and historical class prototype for class k, p^k is k-th class prototype calculated in the current training batch, γ is momentum set as 0.999.

We next present our multi-level contrastive learning loss considering pixel-to-pixel, pixel-to-class and instance-to-class feature relations in the feature maps. In the following paragraphs, the features and prototypes are l_2 normalized. The "pixel" in this work represents the pixel in the feature maps instead of the original image grid.

Pixel-to-Pixel Contrastive Learning. This learning loss is to constrain the pixels in feature maps having the same class label should be closer and different class labels should be distant in the feature space. To realize this goal, we propose a novel pixel-to-pixel contrastive loss. As shown in Fig. 4, for the k-th class, we first sample N_k features to avoid memory explosion when using all pixels in the feature maps. We denote N as the number of features sampled in a batch, *i.e.*, $N = \sum_{k=1}^{K} N_k$, K is class number. We next calculate the similarity matrix $W \in \mathbb{R}^{N \times N}$ over the sampled pixel-wise features, in which $W_{ij} = \exp(z_i \cdot z_j / \tau)$, "$\cdot$" denotes inner product, τ is temperature. We can also get the ground truth label matrix L, which implies the semantic relationship among sampled pixels, *i.e.*, $L_{ij} = 1$ if $y_{z_i} = y_{z_j}$ else $L_{ij} = 0$. Then, we can calculate the transition probability matrix $\widetilde{W}$ and $\widetilde{L}$ by normalizing each row of W and L:

$$\widetilde{W} = D_W^{-1} W, \widetilde{L} = D_L^{-1} L \tag{6}$$

where $D_W = \text{diag}(W\vec{1}), D_L = \text{diag}(L\vec{1})$. We define pixel-to-pixel contrastive loss by calculating distribution distance in each row between $\widetilde{W}$ and $\widetilde{L}$:

$$\mathcal{L}_{pp} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{M}(\tilde{w}_i, \tilde{l}_i) \tag{7}$$

where $\tilde{w}_i$ and $\tilde{l}_i$ are the i-th row in matrix $\widetilde{W}$ and $\widetilde{L}$. Both $\tilde{w}_i$ and $\tilde{l}_i$ in Eq. (7) are in probability simplex, and $\mathcal{M}(\cdot)$ is the distribution distance metric. In this paper, we adopt JS divergence as metric $\mathcal{M}(\cdot)$, which is a symmetric divergence. Note that our pixel-to-pixel semantic similarity loss is different from the supervised contrastive learning loss (Khosla et al. 2020) in two aspects. Firstly, our loss considers the feature with itself as positive pair, positioning along the diagonal of the row normalized matrix $\widetilde{W}$. Secondly, we use

a symmetric JS divergence as our distribution metric. In the experiment, we will show that our proposed pixel-to-pixel contrastive loss produces better generalization performance than the standard supervised contrastive loss.

Pixel-to-Class Contrastive Learning. This loss is to enforce that pixel-level features in the feature maps should be closer to their own class centers, represented by class prototypes. We introduce our pixel-to-class similarity loss as

$$\mathcal{L}_{pc} = \frac{1}{H'W'} \sum_{i=1}^{H'W'} -y_{z_i}^T \log \frac{\exp(z_i \cdot P^k/\tau)}{\sum_{P^a \in \mathcal{P}} \exp(z_i \cdot P^a/\tau)} \quad (8)$$

where $\mathcal{P}$ is the set of class prototypes. Specifically, we use class prototype P before updating in the current batch to calculate pixel-to-class contrastive loss. In fact, pixel-to-class contrastive loss is a standard classification loss, to ensure each pixel can be classified by the class prototype classifier.

Instance-to-Class Contrastive Learning. In addition to the above losses, we also constrain that the class prototype can correctly classify each instance prototype, which is computed by average pooling features in each connected region of each class. We can use contrastive loss like Eq. (8), however, toughly pulling all different instance prototypes of a class closer to the class prototype may lose the diversity of instance-level feature of the class. We adopt the margin triplet loss (Schroff, Kalenichenko, and Philbin 2015) as our instance-to-class contrastive learning loss:

$$\mathcal{L}_{ic} = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{M_k} \sum_{m,n} \max\{d(p_m^k, P^k) + \xi - d(p_n^{\setminus k}, P^k), 0\} \quad (9)$$

where p_m^k is m-th instance prototype in k-th class, $p_n^{\setminus k}$ is the n-th instance prototype in all classes except k, M_k is the total number of instance pairs for class k, ξ is the margin, d is L_2 distance. We can use Eq. (5) to get each instance prototype by substituting the class mask with instance binary mask. Each binary mask is accessed by extracting the connected region in each class mask as (Wang et al. 2020).

Multi-Level Contrastive Loss. Totally, our multi-level contrastive loss is defined as

$$\mathcal{L}_{mlcl} = \lambda \mathcal{L}_{pp} + \mathcal{L}_{pc} + \mathcal{L}_{ic} \quad (10)$$

where we only have one hyper-parameter λ in the loss to balance the contribution of pixel-to-pixel contrastive loss.

Training Methods

The training phase of our approach consists of two stages. First, we use Eq. (3) to pre-train our SSDP network by reconstructing the original data from augmented data. In the second stage, we freeze the parameters of SSDP and only use it for data projection. Based on the projected data, except for task loss and multi-level contrastive loss, we utilize a divergence loss to make class prototypes apart from each other after each update, which is denoted as

$$\mathcal{L}_{div} = \frac{1}{K(K-1)} \sum_{j=1}^{K} \sum_{i \neq j}^{K} max\{\hat{P}^i (\hat{P}^j)^T, 0\} \quad (11)$$

Finally, we use the following total loss to train our segmentation model based on projected data in the second stage

$$\mathcal{L}_{total} = \mathcal{L}_{task} + \mathcal{L}_{mlcl} + \mathcal{L}_{div} \quad (12)$$

we use a per-pixel cross-entropy loss for segmentation task loss $\mathcal{L}_{task}$. To avoid feature mode collapse by using $\mathcal{L}_{mlcl}$ at the beginning, we warm up our segmentation model only using $\mathcal{L}_{task}$ for ten epochs and then use $\mathcal{L}_{total}$ to train.

Testing Process

In testing, we first project target data by our SSDP to mitigate domain shift. Then we send the projected data into segmentation model to generate its prediction for segmentation. Our class prototypes obtained in the training phase can also be regarded as a classifier. So we average the softmax probabilities predicted by classifier H and class prototypes (based on features in the second last layer) to make a more reliable prediction. Except for standard test process, we also propose a test-time adaptation scheme by minimizing our proposed pixel-to-pixel contrastive loss. Different from existing test-time adaptation methods (Wang et al. 2021a; Chen et al. 2022b), which commonly update model parameters in the test time process, we fix all the network parameters, and only optimize the input image of segmentation network, taking the projected data by SSDP as initialization. Specifically, given a projected target domain data by SSDP, we first compute its pseudo label by averaging the predictions from classifier H and class prototypes, then we randomly sample one thousand pixel of each class from this image without replacement to construct our pixel-to-pixel contrastive loss in Eq. (7), we iterate the projected data once by gradient descent to minimize the loss, and get the refined prediction of segmentation mask.

Experiment

In this section, we will evaluate our method on different domain generalization benchmarks.

Experimental Setups

Synthetic Datasets. GTAV (G) (Richter et al. 2016) is a synthetic dataset, which contains 24966 images with resolution of 1914×1052 along with their pixel-wise semantic labels, and it has 12,403, 6,382, and 6,181 images for training, validation, and test, respectively. SYNTHIA (S) (Ros et al. 2016) is an another synthetic dataset. The subset SYNTHIA-RANDCITYSCAPES is used in our experiments which contains 9400 images with resolution of 1280×760.

Real-World Datasets. Cityscapes (C) (Cordts et al. 2016) is a high resolution dataset (2048×1024) of 5000 vehicle-captured urban street images taken from 50 cities primarily in Germany. BDD (B) (Yu et al. 2020) is another real-world dataset that contains diverse urban driving scene images in resolution of 1280×720. The last real-world dataset we use is Mapillary (M) (Neuhold et al. 2017), which consists of 25,000 high-resolution images with a minimum resolution of 1920×1080 collected from all around the world.

Implementation Details. We use ResNet50 (He et al. 2016), ShuffleNetV2 (Ma et al. 2018) and MobileNetV2 (Sandler et al. 2018) as our segmentation backbones for the task GTAV to Cityscapes, BDD and Mapillary and the task Cityscapes to BDD, SYNTHIA and GTAV. We take SGD

Backbone	Method	Train on GTAV (G)				Train on Cityscapes (C)			
		C	B	M	Mean	B	S	G	Mean
ResNet50	Baseline	28.95	25.12	28.18	27.42	44.91	23.29	42.55	36.92
	SW	29.91	27.48	29.71	29.03	48.49	26.10	44.87	39.82
	IBN-Net	33.85	32.30	37.75	34.63	48.56	26.14	45.06	39.92
	DPRC	37.42	32.14	34.12	34.56	49.86	26.58	45.62	40.69
	GTR	37.53	33.75	34.52	35.27	50.75	26.47	45.79	41.00
	IRW	33.57	33.18	38.42	35.06	48.67	26.05	45.64	40.12
	ISW	36.58	35.20	40.33	37.37	50.73	26.20	45.00	40.64
	SANSAW	39.75	37.34	41.86	39.65	**52.95**	**28.32**	**47.28**	**42.85**
	DIRL	41.04	39.15	41.60	40.60	51.80	26.50	46.52	41.60
	DPCL	**44.87**	**40.21**	**46.74**	**43.94**	52.29	26.60	46.00	41.63
	DPCL+TTA (C)	**46.34**	40.67	48.28	45.10	52.23	26.68	46.26	41.72
	DPCL+TTA (C+E)	46.02	**41.14**	**48.79**	**45.32**	**53.30**	26.91	47.25	42.49
ShuffleNetV2	Baseline	25.56	22.17	28.60	25.44	38.09	21.25	36.45	31.93
	IBN-Net	27.10	31.82	34.89	31.27	41.89	22.99	40.91	35.26
	ISW	30.98	32.06	35.31	32.78	41.94	22.82	40.17	34.98
	DIRL	31.88	32.57	36.12	33.52	42.55	**23.74**	41.23	35.84
	DPCL	**36.66**	**34.35**	**39.92**	**36.98**	**43.96**	23.24	**41.93**	**36.38**
	DPCL+TTA (C)	**39.12**	**35.86**	**42.19**	**39.06**	44.18	23.60	42.23	36.67
	DPCL+TTA (C+E)	37.94	35.40	41.15	38.16	**44.53**	**23.95**	**43.49**	**37.32**
MobileNetV2	Baseline	25.92	25.73	26.45	26.03	40.13	21.64	37.32	33.03
	IBN-Net	30.14	27.66	27.07	28.29	44.97	23.23	41.13	36.44
	ISW	30.86	30.05	30.67	30.53	45.17	22.91	41.17	36.42
	DIRL	34.67	32.78	34.31	33.92	**47.55**	23.29	41.43	37.42
	DPCL	**37.57**	**35.45**	**40.30**	**37.77**	46.23	**24.68**	**44.17**	**38.36**
	DPCL+TTA (C)	**41.16**	36.59	**42.94**	**40.23**	46.37	24.76	44.32	38.48
	DPCL+TTA (C+E)	39.13	**36.86**	41.83	39.27	46.76	**25.17**	**45.49**	**39.14**

Table 1: Results for the task G to C, B and M and the task C to B, S and G in mIoU. The best and second best results of methods without TTA are bolded and underlined respectively. The best TTA methods are also bolded.

optimizer with an initial learning rate of 1e-3, and train segmentation model for 40k iterations with batch size of 8, momentum of 0.9 and weight decay of 5e-4. We adopt the polynomial learning rate scheduling (Liu, Rabinovich, and Berg 2015) with the power of 0.9. We use color-jittering and Gaussian noise as image augmentation. We also use random cropping, random horizontal flipping, and random scaling to avoid the model over-fitting. As for our SSDP subnet, we adopt the same architecture with generator in CycleGAN (Zhu et al. 2017) and train it with Adam optimizer. We utilize the same image data augmentation with our segmentation network. In the multi-level contrastive learning, we sample thirty pixel features in each class from a batch of images, half of which are with incorrect prediction by segmentation classifier and half of which are with correct prediction according to their labels. We respectively use $q = 10$ and $q = 5$ for the task trained on GTAV and Cityscapes. The other parameters are set as $\xi = 0.5, \tau = 0.1, \lambda = 5$.

Compared Methods. Our baseline model is DeepLabV3+ trained by cross-entropy loss in source domain for segmentation. We compare with the DG methods: SW (Pan et al. 2019), IBN-Net (Pan et al. 2018), DPRC (Yue et al. 2019), GTR (Peng et al. 2021), IRW, ISW (Choi et al. 2021), DIRL (Xu et al. 2022) and SANSAW (Peng et al. 2022).

Comparison with State-of-the-Art Methods

As for the synthetic to real generalization, we follow DIRL (Xu et al. 2022) to evaluate the generalization performance from GTAV to Cityscapes, BDD and Mapillary. As shown in Table 1, our method outperforms the other methods clearly and consistently across three different network backbones, especially for the task from GTAV to Cityscapes and Mapillary. When using ResNet50, our method improves performance from 41.04 to 44.87 on Cityscapes dataset and from 41.60 to 46.74 on Mapillary dataset compared with DIRL. Except for standard test process, we also show our method performance with test-time adaptation as discussed in the subsection of Testing process, which is denoted as DPCL+TTA (C). As for the backbone of ShuffleNetV2 and MobileNetV2, our method respectively improves the mIoU by 2.08 and 2.46 using test-time adaptation. Except for contrastive loss, we additionally try TTA by minimizing sum of entropy and our pixel-to-pixel contrastive loss (normalized by number of selected pixels), dubbed DPCL+TTA (C+E), and it further improves performance with ResNet50. We also visualize the qualitative comparisons with other methods shown in Fig. 5 to show superiority of our methods DPCL.

We further compare our methods with other methods from Cityscapes to BDD, SYNTHIA and GTAV, shown in Table 1. Our method achieves the best performance with backbone ShuffleNetV2 and MobileNetV2, achieves the second

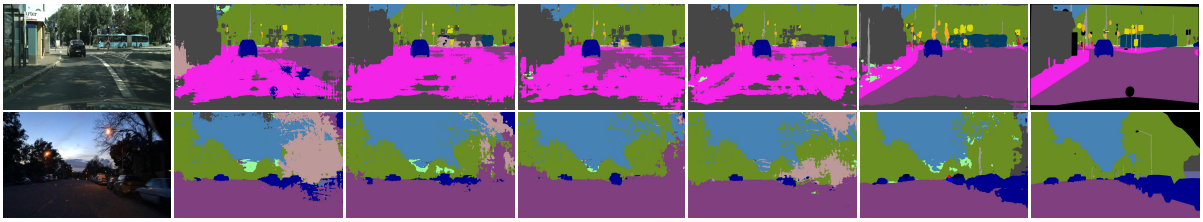

Figure 5: Visual comparison of different domain generalization semantic segmentation methods using ResNet50, trained on GTAV (G) and tested on unseen target domains of Cityscapes (C) (Cordts et al. 2016) and BDD (B) (Yu et al. 2020).

SSDP	$\mathcal{L}_{mlcl}$	$\mathcal{L}_{div}$	C	B	M	Mean
			28.95	25.12	28.18	27.42
✓			40.13	39.47	43.13	40.91
✓	✓		43.68	39.89	45.05	42.87
✓	✓	✓	44.87	40.21	46.74	43.94

Table 2: Ablation study for domain generalization task G to C, B and M with ResNet50 in mIoU, SSDP denotes our Self-supervised Source Domain Projection network, $\mathcal{L}_{mlcl}$ denotes Multi-Level Contrastive Learning loss and $\mathcal{L}_{div}$ denotes class prototype divergence loss.

Method	C	B	M	Mean
SSDP (w/ AE)	36.41	34.47	36.61	35.83
SSDP (w/o AdaIN)	43.13	39.77	45.10	42.67
SSDP	44.87	40.21	46.74	43.94

Table 3: Results of different designs of Source Domain Projection network for task G to C, B and M using ResNet50.

best performance with backbone ResNet50 among the compared methods. Our method DPCL+TTA (C+E) further improves the performance for three backbones.

Ablation Study

We examine each component of our method DPCL to check how they contribute in the domain generalization on the task GTAV to Cityscapes, BDD and Mapillary. As show in Table 2, the baseline method shows lowest performance on three unseen target domains. Our method improves baseline in average accuracy from 27.42 to 40.91 by using our SSDP. This shows that our SSDP for projecting data can mitigate domain shift in the test phase. Based on the projected data, we add our multi-level contrastive learning module, further improving the performance. Finally, we add diversity constraint $\mathcal{L}_{div}$ to our class prototypes and produce the best performance, especially in the task GTAV to Mapillary.

Comparison of Different Designs of SSDP. We compare different designs of SSDP shown in Table 3. In the first row, we use a standard Auto-Encoder in SSDP network, denoted as SSDP (w/ AE), which aims to reconstruct original input image and obtains 35.83 mean mIoU. In the second row, we input augmented data into SSDP and directly reconstruct original data without AdaIN technique in the feature space named SSDP (w/o AdaIN). It improves the average performance from 35.83 to 42.67, which is superior than DIRL. The last row is the SSDP that we adopt, which reconstructs the original data from augmented data with AdaIN technique (Karras, Laine, and Aila 2019) in the feature space and shows effectiveness in average performance.

Method	Scl-CE	Scl-JS	Ours-CE	Ours-JS
Mean mIoU	43.08	43.65	43.46	43.94

Table 4: Results of different choices of pixel-to-pixel contrastive loss for task G to C, B and M using ResNet50. Mean mIoU is obtained over the three target dataset.

Comparison of Different Choices of Pixel-to-Pixel Contrastive Learning. In this paragraph, we compare our pixel-to-pixel contrastive loss with supervised contrastive loss (Khosla et al. 2020) under the same hyper-parameter setting. The method Scl-CE is the standard supervised contrastive loss used in (Khosla et al. 2020). Compared with ours, Scl-CE discards the diagonal values of W and L and uses cross-entropy loss (see appendix). Scl-JS masks out the diagonal vector of matrix W and L, but uses JS divergence as metric $\mathcal{M}(\cdot)$. Ours-CE and Ours-JS are respectively our loss using cross-entropy and JS divergence as metric $\mathcal{M}(\cdot)$. Table 4 shows that Ours-JS achieves consistently better performance than the other variants of losses.

Due to the space limit, more visualization results and empirical analysis, *e.g.*, sensitivity to hyper-parameters, ablation for multi-level contrastive learning loss, data augmentation, choices of $\mathcal{L}_{ic}$, etc., are in the appendix (Yang, Gu, and Sun 2023).

Conclusion

In this paper, we propose a novel domain generalization semantic segmentation method DPCL, consisting of modules of Self-supervised Source Domain Projection (SSDP) and Multi-Level Contrastive Learning (MLCL). Comprehensive experiments demonstrate the effectiveness of SSDP and MLCL in domain generalization semantic segmentation. In the future, we plan to further improve the learning schemes on the segmentation model, and try transformer-based backbones in our framework.

Acknowledgements

This work was supported by National Key R&D Program 2021YFA1003002, NSFC (12125104, U20B2075, 11971373, 61721002, U1811461), and the Fundamental Research Funds for the Central Universities.

References

Chen, C.; Li, J.; Han, X.; Liu, X.; and Yu, Y. 2022a. Compound Domain Generalization via Meta-Knowledge Encoding. In *CVPR*.

Chen, D.; Wang, D.; Darrell, T.; and Ebrahimi, S. 2022b. Contrastive Test-Time Adaptation. In *CVPR*.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans PAMI*, 40(4): 834–848.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020a. A simple framework for contrastive learning of visual representations. In *ICML*.

Chen, T.; Kornblith, S.; Swersky, K.; Norouzi, M.; and Hinton, G. E. 2020b. Big self-supervised models are strong semi-supervised learners. In *NeurIPS*.

Chen, W.; Yu, Z.; Mello, S. D.; Liu, S.; Alvarez, J. M.; Wang, Z.; and Anandkumar, A. 2021. Contrastive Syn-to-Real Generalization. In *ICLR*.

Chen, X.; Fan, H.; Girshick, R.; and He, K. 2020c. Improved baselines with momentum contrastive learning. arXiv:2003.04297.

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

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *CVPR*.

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

Englesson, E.; and Azizpour, H. 2021. Generalized jensen-shannon divergence loss for learning with noisy labels. In *NeurIPS*.

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

Hendrycks, D.; Mu, N.; Cubuk, E. D.; Zoph, B.; Gilmer, J.; and Lakshminarayanan, B. 2020. Augmix: A simple data processing method to improve robustness and uncertainty. In *ICLR*.

Hoffman, J.; Tzeng, E.; Park, T.; Zhu, J.-Y.; Isola, P.; Saenko, K.; Efros, A.; and Darrell, T. 2018. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*.

Hu, H.; Cui, J.; and Wang, L. 2021. Region-aware contrastive learning for semantic segmentation. In *ICCV*.

Huang, J.; Guan, D.; Xiao, A.; Lu, S.; and Shao, L. 2022. Category contrast for unsupervised domain adaptation in visual tasks. In *CVPR*.

Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *CVPR*.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. In *NeurIPS*.

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

Kundu, J. N.; Kulkarni, A.; Bhambri, S.; Jampani, V.; and Radhakrishnan, V. B. 2022. Amplitude Spectrum Transformation for Open Compound Domain Adaptive Semantic Segmentation. In *AAAI*.

Kwon, H.; Jeong, S.; Kim, S.; and Sohn, K. 2021. Dual Prototypical Contrastive Learning for Few-shot Semantic Segmentation. arXiv:2111.04982.

Lee, S.; Seong, H.; Lee, S.; and Kim, E. 2022. WildNet: Learning Domain Generalized Semantic Segmentation from the Wild. In *CVPR*.

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

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

Li, L.; Gao, K.; Cao, J.; Huang, Z.; Weng, Y.; Mi, X.; Yu, Z.; Li, X.; and Xia, B. 2021a. Progressive domain expansion network for single domain generalization. In *CVPR*.

Li, S.; Xie, B.; Zang, B.; Liu, C. H.; Cheng, X.; Yang, R.; and Wang, G. 2021b. Semantic distribution-aware contrastive adaptation for semantic segmentation. arXiv:2105.05013.

Liu, W.; Rabinovich, A.; and Berg, A. C. 2015. Parsenet: Looking wider to see better. arXiv:1506.04579.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*.

Ma, N.; Zhang, X.; Zheng, H.-T.; and Sun, J. 2018. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *ECCV*.

Matsuura, T.; and Harada, T. 2020. Domain generalization using a mixture of multiple latent domains. In *AAAI*.

Neuhold, G.; Ollmann, T.; Rota Bulo, S.; and Kontschieder, P. 2017. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*.

Pan, X.; Luo, P.; Shi, J.; and Tang, X. 2018. Two at once: Enhancing learning and generalization capacities via ibn-net. In *ECCV*.

Pan, X.; Zhan, X.; Shi, J.; Tang, X.; and Luo, P. 2019. Switchable whitening for deep representation learning. In *ICCV*.

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

Peng, D.; Lei, Y.; Liu, L.; Zhang, P.; and Liu, J. 2021. Global and local texture randomization for synthetic-to-real semantic segmentation. *IEEE Trans IP*, 30: 6594–6608.

Richter, S. R.; Vineet, V.; Roth, S.; and Koltun, V. 2016. Playing for data: Ground truth from computer games. In *ECCV*.

Ros, G.; Sellart, L.; Materzynska, J.; Vazquez, D.; and Lopez, A. M. 2016. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*.

Sakaridis, C.; Dai, D.; and Van Gool, L. 2021. ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *ICCV*.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In *CVPR*.

Wang, D.; Shelhamer, E.; Liu, S.; Olshausen, B.; and Darrell, T. 2021a. Tent: Fully Test-Time Adaptation by Entropy Minimization. In *ICLR*.

Wang, W.; Zhou, T.; Yu, F.; Dai, J.; Konukoglu, E.; and Van Gool, L. 2021b. Exploring cross-image pixel contrast for semantic segmentation. In *ICCV*.

Wang, Z.; Yu, M.; Wei, Y.; Feris, R.; Xiong, J.; Hwu, W.-m.; Huang, T. S.; and Shi, H. 2020. Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation. In *CVPR*.

Wu, Z.; Xiong, Y.; Yu, S. X.; and Lin, D. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*.

Xiao, Z.; Zhen, X.; Shao, L.; and Snoek, C. G. M. 2022. Learning to Generalize across Domains on Single Test Samples. In *ICLR*.

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

Yang, L.; Gu, X.; and Sun, J. 2023. Generalized Semantic Segmentation by Self-Supervised Source Domain Projection and Multi-Level Contrastive Learning. arXiv:2303.01906.

Yang, Y.; and Soatto, S. 2020. Fda: Fourier domain adaptation for semantic segmentation. In *CVPR*.

Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; and Darrell, T. 2020. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*.

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

Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P. H.; et al. 2021. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*.

Zou, Y.; Yu, Z.; Kumar, B.; and Wang, J. 2018. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*.