# Learning Intra-Domain Style-Invariant Representation for Unsupervised Domain Adaptation of Semantic Segmentation

Zongyao Li[a,*], Ren Togo[b], Takahiro Ogawa[b], Miki Haseyama[b]

[a]*Graduate School of Information Science and Technology, Hokkaido University, N-14, W-9, Kita-ku, Sapporo 060-0814, Japan*
[b]*Faculty of Information Science and Technology, Hokkaido University, N-14, W-9, Kita-ku, Sapporo 060-0814, Japan*

**Abstract**

In this paper, we aim to tackle the problem of unsupervised domain adaptation (UDA) of semantic segmentation and improve the UDA performance with a novel conception of learning intra-domain style-invariant representation. Previous UDA methods focused on reducing the inter-domain inconsistency between the source domain and the target domain. However, due to the different data distributions of the two domains, reducing the inter-domain inconsistency cannot ensure the generalization ability of the trained model in the target domain. Therefore, to improve the UDA performance, we take into consideration the intra-domain diversity of the target domain for the first time in studies on UDA and aim to train the model to generalize well to the diverse intra-domain styles. To achieve this, we propose a self-ensembling method to learn the intra-domain style-invariant representation and we introduce a semantic-aware multimodal image-to-image translation model to obtain images with diversified intra-domain styles. Our method achieves state-of-the-art performance on two synthetic-to-real adaptation benchmarks, and we demonstrate the effectiveness of our method by conducting extensive experiments.

*Keywords:* Style-invariant representation; Self-ensembling; Domain adaptation

*Corresponding author
*Email addresses:* li@lmd.ist.hokudai.ac.jp (Zongyao Li), togo@lmd.ist.hokudai.ac.jp (Ren Togo), ogawa@lmd.ist.hokudai.ac.jp (Takahiro Ogawa), miki@ist.hokudai.ac.jp (Miki Haseyama)

## 1. Introduction

Unsupervised domain adaptation (UDA) aims to transfer knowledge from a domain that is rich in ground truth labels to an unlabeled domain. UDA is especially promising for tasks that have a shortage of ground truth labels such as semantic segmentation. In recent years, synthetic data (e.g., GTA5 [1] and SYNTHIA [2]) have drawn researchers' interest as an appropriate candidate for the source domain in UDA of semantic segmentation. Labels of synthetic data can be produced automatically, and thus leveraging those synthetic data may considerably alleviate the burden of human annotation.

Unlike semi-supervised learning (SSL) in which labeled data and unlabeled data typically have the same distributions, distributions of the two domains in UDA are quite different and the images have major visual differences. Therefore, aligning the feature distributions of the two domains is considered the key to transferring the knowledge. Researchers have tried to achieve this by using various approaches such as modifying the images to make the two domains visually similar [3, 4, 5] and using adversarial learning to make the domain of the features or segmentation outputs indistinguishable [6, 7, 8]. Despite significant achievements, a problem that has not attracted sufficient attention is that alignment of the feature distributions cannot ensure generalization ability of the trained model in the target domain. Due to the different intrinsic data distributions and some nontransferable features, the two domains cannot be completely aligned, and a model trained with supervision signals from only the source domain may therefore not generalize well in the target domain. Although pseudo labels can provide supervision signals in the target domain [9, 10, 11], the final performance still depends on the model that generates the pseudo labels.

In this study, to tackle the problem of generalization in the target domain, we focused on learning intra-domain style-invariant representation for UDA of semantic segmentation. The underlying concept is that if the learned representation is invariant to the varied characteristics (e.g., brightness, saturation and texture, which are referred to as intra-domain styles in this paper) of the target domain, the segmentation model may perform well on the unknown samples in the target domain. This concept is some-

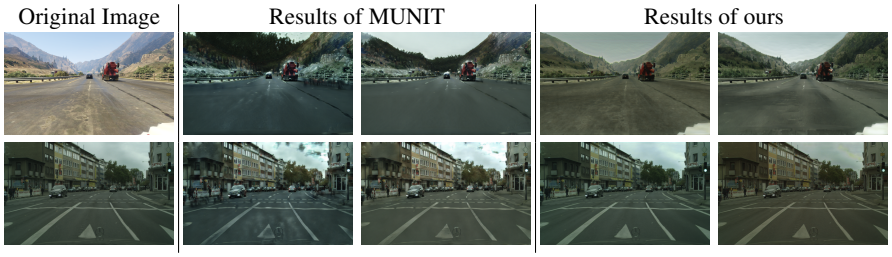


Figure 1: Inconsistent semantic contents (sky region) can be observed in the results of MUNIT, whereas the semantic-aware MUNIT proposed in our method produced diversified results with consistent contents. 1st row: GTA5-to-Cityscapes translation. 2nd row: diversification of the target domain image.

what similar to data augmentation, which is considered to be usually helpful for the enhancement of convolutional neural networks' (CNNs') generalization ability in supervised learning. However, in our study, the style of an image cannot be modified appropriately by using usual augmentation techniques. In addition, more importantly, the style-invariant representation is learned via not only supervised learning with labeled source domain samples but also unsupervised learning with unlabeled target domain samples. Therefore, we propose a self-ensembling method to integrate the supervised and unsupervised learning of intra-domain style-invariant representation and, additionally, construct a multimodal unpaired image-to-image (I2I) translation model to obtain images with diverse intra-domain styles.

The idea of self-ensembling originated from studies of SSL [12, 13]. SSL was used for UDA of semantic segmentation in a previous work [14] but only as a usual SSL technology that does not consider the generalization problem and the intra-domain styles. In this study, we used a self-ensembling architecture [13] that consists of a student model trained with style-diversified images and a teacher model updated as the exponential moving average of the student model. By training with images with diversified intra-domain styles, the learning of the intra-domain style-invariant representation integrates into a supervised loss of the source domain and a teacher-student consistency loss of the target domain. Moreover, pseudo labels are subsequently involved in the training to further improve the UDA performance.

As mentioned above, images with diversified intra-domain styles are indispensable for the realization of our conception. In our method, we translate the source domain

images to different target domain styles and meanwhile diversify the styles of the target domain images. Such a task can be accomplished by an existing multimodal unpaired I2I translation method [15] named multimodal unsupervised image- to-image translation (MUNIT). However, we found that the existing method cannot meet an essential requirement in our study, which is the consistency of semantic contents in the translation results. An example is shown in Fig. 1. The semantic contents in the sky region are inconsistent in the translation results of MUNIT. To overcome this problem, we adapt the MUNIT architecture to content-consistent translation by introducing pixel-level semantic information as additional guidance for the translation. As shown in Fig. 1, the consistency of semantic contents in our translation results is enhanced compared to that in the results of MUNIT, and learning the style-invariant representation therefore becomes realizable.

In this paper, we make the following contributions.

- We propose the conception of learning intra-domain style-invariant representation for UDA of semantic segmentation, which can make the trained model generalize better to the diverse intra-domain styles of the target domain.
- We propose a self-ensembling method for learning the intra-domain style-invariant representation and construct a semantic-aware version of MUNIT for style diversification.
- We achieved state-of-the-art UDA performance on GTA5-to-Cityscapes and SYNTHIA-to-Cityscapes benchmarks and we conducted extensive experiments for further analyses.

## 2. Related Work

### *2.1. UDA of Semantic Segmentation*

UDA of semantic segmentation is considered a challenging task due to the complexity of transferring pixel-level semantic knowledge. There are generally three main components in the technologies for UDA of semantic segmentation: I2I translation, adversarial learning, and semi-supervised learning.

*I2I translation* technologies can modify some characteristics (e.g., color and texture) that are collectively called "styles" of an image, typically for reducing the visual domain gap. Cycle-consistent generative adversarial network (CycleGAN) [16]-based models were the most frequently used models in previous works [3, 17]. The generative adversarial network (GAN) [18]-based unpaired I2I translation method is used to transfer the visual style of the source domain images to that of the target domain. The same goal was achieved in some works [19, 20] by using style transfer technology [21, 22]. In addition to them, [14] trains a one-sided I2I translation model with the GAN framework and the adaptive instance normalization (AdaIN) layer [23]. [24] reduces the visual domain gap by swapping the low-frequency spectrum of the Fourier Transform of the images instead of training a neural network.

*Adversarial learning* is used to align the two domains in the feature space. The adversarial loss derived from the domain discriminator can be imposed on either the intermediate features [3] or final outputs [6] of the segmentation network. There are also some more sophisticated methods based on adversarial learning. [25] explores local region-level consistency across domains with adversarial learning and further integrates local region-level adversarial learning with global image-level adversarial learning. [26] improves the feature alignment by separating the features into different semantic groups and performing class-wise adversarial learning. [27] weights the adversarial loss based on the alignment score produced by two classifiers.

*Semi-supervised learning* is similar to UDA in terms of exploiting unlabeled data, and thus some SSL technologies are useful for UDA of semantic segmentation. Pseudo label is a widely used strategy that is powerful even in the simplest manner [17]. It can be further improved by elaborate designs such as weighting the pseudo label with estimated reliability [9]. Other useful SSL technologies include Entropy Minimization [28] used in [29, 30] and Self-Ensembling [13] used in [14]. Moreover, the scale-invariance constraint in [31] can be interpreted as an SSL technology that makes use of the information of semantic structures in the target domain.

### 2.2. Unpaired I2I translation

Unpaired I2I translation learns a mapping between two domains without paired data. Due to the absence of direct supervision, most methods are constructed on the basis of the GAN framework in which the distribution of each domain is learned by a discriminator. Cycle-consistency [16] is also a crucial ingredient for mitigating the ill-posedness of the unpaired translation. Some methods [32, 15] learn disentangled latent representation and use normalization layers (e.g., conditional instance normalization (CIN) [33] and AdaIN [23]) with trainable parameters to realize the multimodal translation. In this study, we further introduced a semantic-aware architecture to enhance the consistency of semantic contents in the multimodal results.

### 2.3. Self-Ensembling

Self-ensembling minimizes the discrepancy between the predictions of the classification network and its ensemble. The ensemble can be an exponential moving average (EMA) of the network [13, 34] or the prediction [12]. Perturbations to the inputs are important for the self-ensembling. Instead of random perturbations such as Gaussian noise, [35, 36] propose perturbations based on adversarial training to improve the generalization performance. We consider the perturbation from a totally different perspective, i.e., diversification of the intra-domain style rather than addition of trivial noise.

## 3. Proposed Method

First, we provide the problem setting. Given a source domain dataset $\mathcal{S}$ comprised of image-label pairs $\{x^s, y^s\} \in \mathcal{S}$ and a target domain dataset $\mathcal{T}$ comprised of unlabeled images $\{x^t\} \in \mathcal{T}$, we aim to transfer the semantic knowledge from $\mathcal{S}$ to $\mathcal{T}$ at the pixel level. Our method for learning intra-domain style-invariant representation is presented in Section 3.1. The I2I translation model that generates the style-diversified images used in the method presented in Section 3.1 is described in Section 3.2. All of the implementation details are provided in Section 3.3.

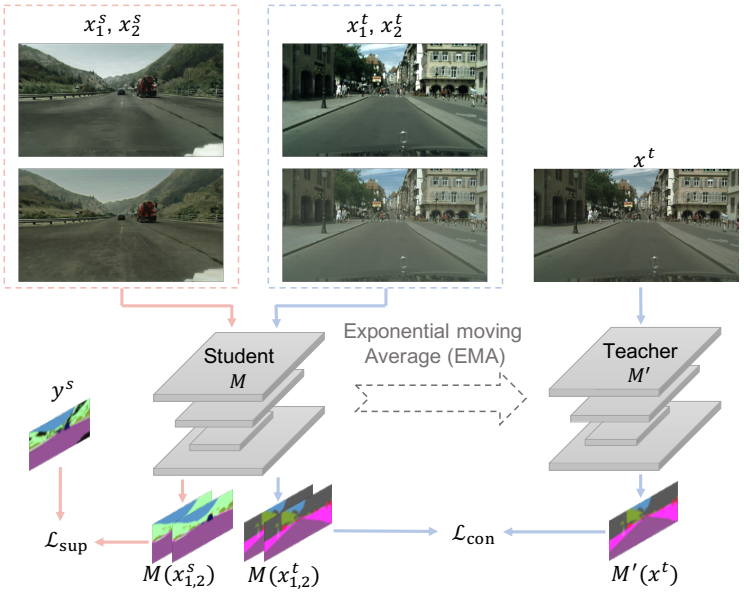


Figure 2: Proposed self-ensembling method for learning intra-domain style-invariant representation. In our method, we first feed a source domain image $x^s$ and a target domain image $x^t$ into the multimodal I2I translation model (not shown in this figure) and obtain the images $x_1^s$, $x_2^s$ and $x_1^t$, $x_2^t$ with diversified intra-domain styles. To train the student model $M$, $x_1^s$ and $x_2^s$ are fed into $M$ along with the source domain label $y^s$ to calculate the supervised learning loss $\mathcal{L}_{\text{sup}}$. Next, the unsupervised consistency loss $\mathcal{L}_{\text{con}}$ is calculated with the outputs of feeding $x_1^t$, $x_2^t$ into the student model $M$ and $x^t$ into the teacher model $M'$. After updating $M$ at each training step, $M'$ is updated as the exponential moving average of $M$.

### *3.1. UDA by Learning Intra-Domain Style-Invariant Representation*

#### *3.1.1. Overview of the self-ensembling architecture*

In our method, we employ a self-ensembling architecture to learn the intra-domain style-invariant representation for UDA. The self-ensembling architecture consists of a student model $M$ and a teacher model $M'$ with the same structures. The student model is trained with the labeled data from $\mathcal{S}$ and with the unlabeled data from $\mathcal{T}$ simultaneously. Meanwhile, the teacher model is updated as the EMA of the student model as the following equation:

$$\theta'_k = \alpha\theta'_{k-1} + (1-\alpha)\theta_k, \tag{1}$$

where $\alpha$ is the EMA weight parameter, $\theta'_k$ is the weight of $M'$ at training step $k$, $\theta'_{k-1}$ at step $k-1$, and $\theta_k$ is the weights of $M$ at training step $k$. The student model and teacher

model are updated alternately. Fig. 2 illustrates the training of the self-ensembling architecture. The training consists of three components: supervised learning with $\mathcal{S}$, unsupervised learning with $\mathcal{T}$, and pseudo-label learning with $\mathcal{T}$, which is not included in Fig. 2.

*3.1.2. Supervised learning with the source domain*

Given an image-label pair $\{x^s, y^s\}$ from the source domain $\mathcal{S}$, we use a multimodal I2I translation model to translate $x^s$ into the target domain and sample two translation results $x_1^s$ and $x_2^s$ with different intra-domain styles. Then, with the images $x_1^s$, $x_2^s$ and label $y^s$, we calculate the cross-entropy loss:

$$\mathcal{L}_{\text{sup}} = \mathbb{E}_{x^s,y^s}[-\frac{1}{2HW}\sum_{k}\sum_{h,w,c} y^{s(h,w,c)}\log M(x_k^s)^{(h,w,c)}], \tag{2}$$

where $M(x_k^s)$ is the probability predicted by the model $M$ for the image $x_k^s$, $(h, w, c)$ means the element of the channel $c$ at the spatial position $(h, w)$, and $H$ and $W$ denote the height and width of the image, respectively.

Since the domain translation can hardly be perfect, the student model learns the semantic knowledge partly in the target domain with the above supervised loss. On the other hand, the intra-domain style-invariant representation is also motivated by the supervised learning. Due to the presence of the ground truth label, there is no need to enforce an additional constraint and the supervised loss implicitly encourages consistent predictions of $x_1^s$ and $x_2^s$.

*3.1.3. Unsupervised learning with the target domain*

Given a target domain image $x^t$, we again use the multimodal I2I translation model to obtain two style-diversified copies $x_1^t$ and $x_2^t$ of $x^t$. But this time, the translation does not change the image domain and only diversifies the intra-domain style of $x^t$. Due to the absence of the ground truth label, the prediction of the teacher model $M'$ is used to calculate an unsupervised consistency loss:

$$\mathcal{L}_{\text{con}} = \mathbb{E}_{x^t}[\frac{1}{2}\sum_{k}||M(x_k^t) - M'(x^t)||_2]. \tag{3}$$

This consistency loss is twofold: one is the consistency between $M(x_k^t)$ and $M'(x^t)$, and the other is the consistency between $M(x_1^t)$ and $M(x_2^t)$. The former is a consistency term of SSL, which forces the student model to make predictions that are consistent with those by the teacher model. The teacher model aggregates information of the consecutive student models and consequently tends to be more accurate than the student model. Therefore, the prediction of the teacher model can be used as targets for training the student model. Moreover, considering that the teacher model can be regarded as an aggregate of the student models, the consistency constraint encourages the student model to be smooth in the vicinity of $x^t$ and yields a movement of the decision boundary towards low-density regions so that the model becomes more reliable for the target domain according to the "smoothness assumption" of SSL. However, different from the traditional consistency term in the SSL methods [12, 13], we sample from the vicinity of $x^t$ by changing its intra-domain style instead of adding trivial noise.

The latter consistency is similar to that between $x_1^s$ and $x_2^s$ in the supervised learning with the source domain. Although there are no ground truth supervision signals in the consistency loss, the predictions of $x_1^t$ and $x_2^t$ are both encouraged to be consistent with the teacher model's prediction $M'(x^t)$ so that the intra-domain style-invariant representation learning is enforced.

#### *3.1.4. Pseudo-label learning with the target domain*

Like many previous methods, we use pseudo labels of the target domain images to further improve the UDA performance. Inspired by [9] that estimates the prediction uncertainty to rectify the learning with pseudo labels, we adopt a similar rectification strategy for the pseudo-label learning. Different from the uncertainty estimation by using two classifiers in [9], we take advantage of the style-diversified images $x_1^t$ and $x_2^t$ to estimate the prediction uncertainty. The pseudo-label loss is defined as follows:

$$\mathcal{L}_{\text{psl}} = \text{Exp}(-\text{KLD}(x_1^t, x_2^t))\mathcal{L}'_{\text{sup}} + \text{KLD}(x_1^t, x_2^t), \tag{4}$$

where $\text{KLD}(x_1^t, x_2^t)$ is the Kullback–Leibler (KL) divergence between $M(x_1^t)$ and $M(x_2^t)$, and $\mathcal{L}'_{\text{sup}}$ is the cross-entropy loss using pseudo labels with the same form

as $\mathcal{L}_{\mathrm{sup}}$.

In Eq. (4), $\mathrm{KLD}(x_1^t, x_2^t)$ measures the inconsistency between the predictions of model $M$ for the style-diversified copies $x_1^t$ and $x_2^t$. If $M$ produces predictions with a large divergence for a pixel of $x_1^t$ and $x_2^t$, the predictions are considered ambiguous and the model is considered to be unreliable for the pixel in light of the ambiguous predictions. The pseudo labels of such pixels are also more noisy, and hence we assign the pixels small weights in the form of $\mathrm{Exp}(-\mathrm{KLD}(x_1^t, x_2^t))$ for the cross-entropy loss $\mathcal{L}'_{\mathrm{sup}}$ using pseudo labels. Meanwhile, to compensate for the training for the pixels with small weights, we train the student model $M$ to minimize also the KL divergence term $\mathrm{KLD}(x_1^t, x_2^t)$ to make the predictions for $x_1^t$ and $x_2^t$ consistent. As a result, model $M$ is trained under less effects of the unreliable pseudo labels compared to the normal pseudo-label learning without the rectification, and the minimization of both $\mathcal{L}'_{\mathrm{sup}}$ and $\mathrm{KLD}(x_1^t, x_2^t)$ leads to the learning of the intra-domain style-invariant representation.

#### *3.1.5. Training procedure*

The training of the self-ensembling architecture is multi-step. The model is first trained without pseudo-label learning as the following loss function:

$$\mathcal{L}_{\mathrm{init}} = \mathcal{L}_{\mathrm{sup}} + \omega\lambda_{\mathrm{con}}\mathcal{L}_{\mathrm{con}}, \tag{5}$$

where $\lambda_{\mathrm{con}}$ is the loss weight of $\mathcal{L}_{\mathrm{con}}$, and $\omega$ is a weight rising from zero to one at the beginning of the training. Then pseudo labels are produced with the trained model, and pseudo-label learning is involved in the training with the following loss function:

$$\mathcal{L}_{\mathrm{final}} = \mathcal{L}_{\mathrm{sup}} + \mathcal{L}_{\mathrm{psl}} + \omega\lambda_{\mathrm{con}}\mathcal{L}_{\mathrm{con}}. \tag{6}$$

The production of pseudo labels and training with $\mathcal{L}_{\mathrm{final}}$ are performed twice iteratively in our method.

### *3.2. Multimodal Unpaired I2I Translation*

To obtain style-diversified images for the intra-domain style-invariant representation learning described in Section 3.1, we construct an unpaired multimodal I2I trans-

lation model based on the MUNIT [15] architecture. Despite the success in domain transfer, some semantic contents are inconsistent in the translation results of MUNIT as shown in Fig. 1. Such inconsistency can disturb the learning since the style-invariant representation should be learned for the same semantic contents. Therefore, we adapt the architecture of MUNIT to introduce pixel-level semantic information into the translation. The semantic information can serve as additional guidance for the translation to preserve the original contents better. We describe the MUNIT architecture and the modification in our semantic-aware MUNIT as follows.

*3.2.1. MUNIT architecture*

MUNIT learns disentangled representations to create many-to-many mappings across the two domains. It assumes that an image $x$ can be decomposed into and generated from a content latent code $f_{\text{cont}}$ and a style latent code $f_{\text{sty}}$. The content space is shared by both domains, and the style space is specific to each domain. For each domain, two encoders $E_{\text{cont}}$ and $E_{\text{sty}}$ are trained to extract the content code $f_{\text{cont}} = E_{\text{cont}}(x)$ and style code $f_{\text{sty}} = E_{\text{sty}}(x)$, respectively, and a decoder $G$ is trained to generate the translated image $x' = G(f_{\text{cont}}, f'_{\text{sty}})$, where $f'_{\text{sty}}$ is sampled from the normal distribution $\mathcal{N}(0, I)$. The encoders and decoders are denoted by $\{E^s_{\text{cont}}, E^s_{\text{sty}}, G^s\}$ for the source domain and by $\{E^t_{\text{cont}}, E^t_{\text{sty}}, G^t\}$ for the target domain.

The loss function of MUNIT is comprised of an adversarial loss and several reconstruction losses. Take the $\mathcal{S}$-to-$\mathcal{T}$ translation as an example. Given a source domain image $x^s$, the image should be reconstructed by $G^s$ from its latent codes $f^s_{\text{cont}} = E^s_{\text{cont}}(x^s)$ and $f^s_{\text{sty}} = E^s_{\text{sty}}(x^s)$, which is formulated by the following loss function:

$$\mathcal{L}^x_{\text{recon}} = \mathbb{E}_{x^s}[||x^s - G^s(f^s_{\text{cont}}, f^s_{\text{sty}})||_1]. \tag{7}$$

In addition, after translating $x^s$ to the target domain as $x^s_{s2t} = G^t(f^s_{\text{cont}}, f'_{\text{sty}})$, the latent codes should be reconstructed by encoding $x^s_{s2t}$ with the encoders of the target domain as follows:

$$\mathcal{L}^{\text{cont}}_{\text{recon}} = \mathbb{E}_{x^s, x^s_{s2t}}[||f^s_{\text{cont}} - E^t_{\text{cont}}(x^s_{s2t})||_2], \tag{8}$$

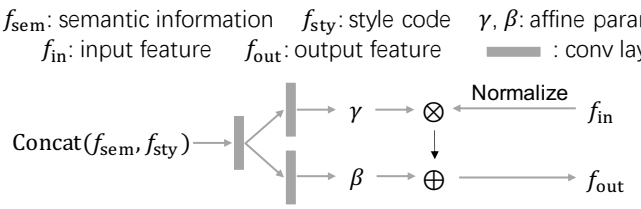


Figure 3: Structure of the spatially-adaptive normalization layer. $\mathrm{Concat}(f_{\mathrm{sem}}, f_{\mathrm{sty}})$ has been resized to the spatial size of $f_{\mathrm{in}}$.

$$\mathcal{L}_{\mathrm{recon}}^{\mathrm{sty}} = \mathbb{E}_{x^s, x_{s2t}^s}[||f'_{\mathrm{sty}} - E_{\mathrm{sty}}^t(x_{s2t}^s)||_2]. \tag{9}$$

To make $x_{s2t}^s$ realistic, the following adversarial loss derived from a domain-specific discriminator $D^t$ is imposed:

$$\mathcal{L}_{\mathrm{adv}} = \mathbb{E}_{x_{s2t}^s, x^t}[\log(1 - D^t(x_{s2t}^s)) + \log D^t(x^t)]. \tag{10}$$

The final objective function for the $\mathcal{S}$-to-$\mathcal{T}$ translation is defined as follows:

$$\begin{aligned} \min_{E,G} \max_{D} \mathcal{L}(E, G, D) = \mathcal{L}_{\mathrm{adv}} + \lambda_x \mathcal{L}_{\mathrm{recon}}^x \\ + \lambda_c \mathcal{L}_{\mathrm{recon}}^{\mathrm{cont}} + \lambda_s \mathcal{L}_{\mathrm{recon}}^{\mathrm{sty}}, \end{aligned} \tag{11}$$

where $\lambda_x$, $\lambda_c$ and $\lambda_s$ are hyper-parameters. The learning of the opposite $\mathcal{T}$-to-$\mathcal{S}$ translation is performed simultaneously in the same manner.

### *3.2.2. Our semantic-aware MUNIT*

MUNIT sometimes fails to preserve the original semantic contents because the translation network can hardly understand the semantic meanings of the image contents. Therefore, to solve this problem, we introduce pixel-level semantic information of the image into the translation. Since the ground truth labels are accessible only in the source domain, we cannot use the labels as semantic information directly. We found that an appropriate substitute for the labels is the prediction by a pre-trained segmentation network that can make meaningful predictions for images of both domains. Accordingly, we pre-train a segmentation network with a simple UDA method [6] and

introduce the network prediction as the semantic information. Some information may be misleading due to inaccurate predictions, while, on the other hand, the predicted probability distribution can be richer with latent information than one-hot ground truth labels.

MUNIT uses the AdaIN [23] layer, which is a normalization layer with trainable parameters, to render the image with the style code in the decoder network. The style code and semantic information in our method are both supposed to guide the translation, and thus it is reasonable to introduce them together via the normalization layer. However, the AdaIN layer takes only one-dimensional style codes and cannot deal with pixel-level inputs. To tackle this problem, we replace the AdaIN layer with a spatially-adaptive instance normalization layer inspired by [37]. As shown in Fig. 3, the concatenation of the style code and semantic information is processed by convolutional layers and output as pixel-level affine parameters for the normalization. We use this spatially-adaptive instance normalization layer as all normalization layers in the decoders. By introducing the semantic information via the normalization layers, the translation network becomes aware of the semantic meanings and thus can translate the image more appropriately.

The training of our semantic-aware MUNIT is the same as that of MUNIT. Besides the loss items described in Section 3.2.1, we also use the cycle-consistency loss and perceptual loss as in MUNIT but replace the VGG network with a segmentation network pre-trained by a simple UDA method [6]. The pre-trained segmentation network is also used for acquiring semantic information.

During the training of the self-ensembling model, style-diversified images $x_k^s$, $x_k^t$ (k=1, 2) are dynamically generated by the target domain decoder $G^t$. Specifically, $x_k^s$ are target-domain-like images generated as follows:

$$x_k^s = G^t(E^s_{\text{cont}}(x^s), f_k, M_{\text{pre}}(x^s)), \tag{12}$$

where $f_k$ are style codes randomly sampled from the normal distribution $\mathcal{N}(0, I)$, and $M_{\text{pre}}$ is the pre-trained segmentation model for extracting semantic information. $x_k^t$

**Algorithm 1:** Training procedure of the semantic-aware MUNIT.

**Input:** Labeled source domain dataset $\mathcal{S}$, unlabeled target domain dataset $\mathcal{T}$, segmentation model $M_{\text{pre}}$ pre-trained with $\mathcal{S}$ and $\mathcal{T}$.

**Output:** Optimal translation model composed of encoders $E^s_{\text{cont}}$, $E^s_{\text{sty}}$, $E^t_{\text{cont}}$, $E^t_{\text{sty}}$, and decoders $G^s$, $G^t$.

1 Initialize the encoders, the decoders, and discriminators $D^s$, $D^t$;
2 **for** *each iteration with $x^s$ randomly drawn from $\mathcal{S}$ and $x^t$ from $\mathcal{T}$* **do**
3 Encode $x^s$ with $E^s_{\text{cont}}$ and $E^s_{\text{sty}}$ as content code $f^s_{\text{cont}} = E^s_{\text{cont}}(x^s)$ and style code $f^s_{\text{sty}} = E^s_{\text{sty}}(x^s)$;
4 Reconstruct $x^s$ from $f^s_{\text{cont}}$ and $f^s_{\text{sty}}$ with $G^s$ as $G^s(f^s_{\text{cont}}, f^s_{\text{sty}}, M_{\text{pre}}(x^s))$ and calculate the loss of Eq. (7);
5 Sample style code $f'_{\text{sty}}$ from $\mathcal{N}(0, I)$ and translate $x^s$ to the target domain as $x^s_{s2t} = G^t(f^s_{\text{cont}}, f'_{\text{sty}}, M_{\text{pre}}(x^s))$;
6 Reconstruct $f^s_{\text{cont}}$ by encoding $x^s_{s2t}$ with $E^t_{\text{cont}}$ as $E^t_{\text{cont}}(x^s_{s2t})$ and calculate the loss of Eq. (8);
7 Reconstruct $f'_{\text{sty}}$ by encoding $x^s_{s2t}$ with $E^t_{\text{sty}}$ as $E^t_{\text{sty}}(x^s_{s2t})$ and calculate the loss of Eq. (9);
8 Input $x^s_{s2t}$ and $x^t$ to $D^t$ and calculate the loss of Eq.(10);
9 Repeat the procedure of lines 3-8 from encoding $x^t$ (changing the encoders, decoders, and discriminator accordingly);
10 Update the encoders and decoders to minimize all the above losses;
11 Update $D^t$ to maximize the loss of Eq. (10) with $x^t$ and $x^s_{s2t}$;
12 Update $D^s$ similarly with $x^s$ and $x^t_{t2s}$;
13 **end for**

are generated similarly without changing the domain as follows:

$$x^t_k = G^t(E^t_{\text{cont}}(x^t), f_k, M_{\text{pre}}(x^t)). \tag{13}$$

### *3.3. Implementation Details*

Algorithms 1 and 2 detail the training procedure of the semantic-aware MUNIT and the self-ensembling method, respectively. The two algorithms are implemented in sequence to implement the whole proposed method. The network structures, training parameters, and some other details in the implementation are as follows.

**Algorithm 2:** Training procedure of the self-ensembling method.

**Input:** Labeled source domain dataset $\mathcal{S}$, unlabeled target domain dataset $\mathcal{T}$, segmentation model $M_{\rm pre}$ pre-trained with $\mathcal{S}$ and $\mathcal{T}$, encoders $E^s_{\rm cont}$, $E^t_{\rm cont}$ and decoders $G^s$, $G^t$ trained with Algorithm 1.

**Output:** Optimal segmentation model $M$ in the target domain.

1 **for** *k=0,1,2...(until 2 in our implementation)* **do**
2  **if** $k \neq 0$ **then**
3  Produce pseudo labels for each $x^t$ of $\mathcal{T}$ with model $M$;
4  **end if**
5  Initialize student model $M$ and make a copy of $M$ as teacher model $M'$;
6  **for** *each iteration with* $x^s$ *randomly drawn from* $\mathcal{S}$ *and* $x^t$ *from* $\mathcal{T}$ **do**
7  Translate $x^s$ as $x^s_k = G^t(E^s_{\rm cont}(x^s), f_k, M_{\rm pre}(x^s))$ $(k = 1, 2)$ to the target domain with style codes $f_1$, $f_2$ sampled from $\mathcal{N}(0, I)$;
8  Generate copies $x^t_k = G^t(E^t_{\rm cont}(x^t), f_k, M_{\rm pre}(x^t))$ $(k = 1, 2)$ for $x^t$ with style codes $f_1$, $f_2$ sampled from $\mathcal{N}(0, I)$;
9  Input $x^s_1$, $x^s_2$ to $M$ and calculate the cross-entropy loss of Eq. (2);
10  Input $x^t_1$, $x^t_2$ to $M$ and $x^t$ to $M'$, and calculate the unsupervised consistency loss of Eq. (3);
11  **if** $k \neq 0$ **then**
12  Calculate the pseudo-label loss of Eq. (4) with the pseudo labels newly produced at line 3;
13  **end if**
14  Update $M$ to minimize all the above losses;
15  Update $M'$ as Eq. (1);
16  **end for**
17 **end for**

#### *3.3.1. Semantic-aware MUNIT*

The network structures of the encoders and decoders in our semantic-aware MUNIT are the same as those of MUNIT [15] except for the normalization layers in the decoders. The spatially-adaptive instance normalization layers used in the decoders consist of three convolutional layers with 128 filters of 3×3. The discriminator consists of six convolutional layers with {64, 128, 256, 512, 512, 1} filters. The first five layers have filters of 4×4 and a stride of 2, and the last layer has a 1×1 filter and a stride of 1. We adopt the Adam optimizer with the same parameters as those in MUNIT. The hyper-parameters $\lambda_x$, $\lambda_c$, $\lambda_s$ and the weights of the cycle-consistency loss and perceptual loss are set to 10, 1, 1, 10, 0.1, respectively. All of the input images are resized to have a long side of 1,024 pixels and the original ratio is kept unchanged.

*3.3.2. Self-ensembling*

We used two structures, Deeplab V2 [38] with ResNet101 [39] and FCN-8s [40] with VGG16 [41], for the segmentation network. We use the optimizer parameters provided by [17]. The batch size is set to 1 for both structures. The EMA parameter $\alpha$ and weight parameter $\lambda_{\mathrm{con}}$ are set to 0.99 and 1, respectively. The ramp-up parameter $\omega$ increases as $\omega = \mathrm{Exp}(-5(1-k)^2)$, where $k$ increases linearly from zero to one during the first 20,000 training iterations. The pseudo labels are selected with a probability threshold of 0.9 to be used in the pseudo-label learning. Only when the ratio of the selected labels is less than 50%, the threshold is ignored to ensure that at least half of the pseudo labels are used. In addition, color jitter transformation was found to be effective as a supplementary intra-domain style augmenter in our experiments. Therefore, the color jitter transformation is further imposed on the generated images.

## 4. Experiments

We conducted experiments on two benchmarks GTA5-to-Cityscapes and SYNTHIA-to-Cityscapes, both of which are synthetic-to-real adaptations. Datasets are first described in Section 4.1. The main results for the benchmarks are presented and compared to results of state-of-the-art methods in Section 4.2. Finally, results of extensive supplementary experiments to further analyze and validate the effectiveness of our method are shown in Section 4.3.

*4.1. Datasets*

The Cityscapes dataset [42] is a real-world dataset consisting of urban scene images with resolutions of 2,048×1,024. It contains a training set of 2,975 images and a validation set of 500 images. The validation set was used as the test data in the experiments. The images were resized to a resolution of 1,024×512 pixels to be fed into the segmentation network.

The GTA5 dataset [1] consists of 24,966 synthesized urban scene images with resolutions of 1,914×1,052. The images are rendered from the GTA5 video game. Nineteen common categories are shared by GTA5 and Cityscapes. In the self-ensembling training, the images were resized to a resolution of 1,280×720 pixels.

Table 1: Results of mean intersection over union (IoU) and per-category IoUs for GTA5-to-Cityscapes benchmark.

| Method | Structure | road | side. | buil. | wall | fence | pole | tlig. | tsign | vege. | terr. | sky | pers. | rider | car | truck | bus | train | moto. | bicy. | mIoU |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BDL [17] | ResNet101 | 91.0 | 44.7 | 84.2 | 34.6 | 27.6 | 30.2 | 36.0 | 36.0 | 85.0 | **43.6** | 83.0 | 58.6 | 31.6 | 83.3 | 35.3 | 49.7 | 3.3 | 28.8 | 35.6 | 48.5 |
| CAG-UDA [43] | | 90.4 | 51.6 | 83.8 | 34.2 | 27.8 | **38.4** | 25.3 | 48.4 | 85.4 | 38.2 | 78.1 | 58.6 | 34.6 | 84.7 | 21.9 | 42.7 | **41.1** | 29.3 | 37.2 | 50.2 |
| RectPLL [9] | | 90.4 | 31.2 | 85.1 | 36.9 | 25.6 | 37.5 | **48.8** | **48.5** | 85.3 | 34.8 | 81.1 | 64.4 | 36.8 | **86.3** | 34.9 | 52.2 | 1.7 | 29.0 | 44.6 | 50.3 |
| FDA [24] | | 92.5 | 53.3 | 82.4 | 26.5 | 27.6 | 36.4 | 40.6 | 38.9 | 82.3 | 39.8 | 78.0 | 62.6 | 34.4 | 84.9 | 34.1 | **53.1** | 16.9 | 27.7 | 46.4 | 50.5 |
| SIM [44] | | 90.6 | 44.7 | 84.8 | 34.3 | 28.7 | 31.6 | 35.0 | 37.6 | 84.7 | 43.3 | 85.3 | 57.0 | 31.5 | 83.8 | 42.6 | 48.5 | 1.9 | 30.4 | 39.0 | 49.2 |
| PIT [45] | | 87.5 | 43.4 | 78.8 | 31.2 | 30.2 | 36.3 | 39.9 | 42.0 | 79.2 | 37.1 | 79.3 | **65.4** | **37.5** | 83.2 | **46.0** | 45.6 | 25.7 | 23.5 | 49.9 | 50.6 |
| LTIR [46] | | 92.9 | **55.0** | 85.3 | 34.2 | 31.1 | 34.9 | 40.7 | 34.0 | 85.2 | 40.1 | **87.1** | 61.0 | 31.1 | 82.5 | 32.3 | 42.9 | 0.3 | 36.4 | 46.1 | 50.2 |
| PCEDA [47] | | 91.0 | 49.2 | 85.6 | 37.2 | 29.7 | 33.7 | 38.1 | 39.2 | 85.4 | 35.4 | 85.1 | 61.1 | 32.8 | 84.1 | 45.6 | 46.9 | 0.0 | 34.2 | 44.5 | 50.5 |
| **Ours** | | **93.0** | 54.0 | **86.6** | **42.6** | **34.7** | 35.9 | 40.8 | 43.3 | **86.0** | 43.2 | 85.4 | 61.5 | 34.4 | 83.7 | 29.2 | 50.1 | 4.0 | **36.5** | **50.9** | **52.4** |
| BDL [17] | VGG16 | 89.2 | 40.9 | 81.2 | 29.1 | 19.2 | 14.2 | 29.0 | 19.6 | 83.7 | 35.9 | 80.7 | **54.7** | 23.3 | 82.7 | 25.8 | 28.0 | 2.3 | **25.7** | 19.9 | 41.3 |
| FDA [24] | | 86.1 | 35.1 | 80.6 | 30.8 | 20.4 | 27.5 | 30.0 | 26.0 | 82.1 | 30.3 | 73.6 | 52.5 | 21.7 | 81.7 | 24.0 | 30.5 | 29.9 | 14.6 | 24.0 | 42.2 |
| SIM [44] | | 88.1 | 35.8 | 83.1 | 25.8 | 23.9 | 29.2 | 28.8 | 28.6 | 83.0 | 36.7 | 82.3 | 53.7 | 22.8 | 82.3 | 26.4 | 38.6 | 0.0 | 19.6 | 17.1 | 42.4 |
| PIT [45] | | 86.2 | 35.0 | 82.1 | 31.1 | 22.1 | 23.2 | 29.4 | 28.5 | 79.3 | 31.8 | 81.9 | 52.1 | 23.2 | 80.4 | **29.5** | 26.9 | **30.7** | 20.5 | 1.2 | 41.8 |
| LTIR [46] | | 92.5 | **54.5** | **83.9** | 34.5 | 25.5 | **31.0** | 30.4 | 18.0 | **84.1** | 39.6 | 83.9 | 53.6 | 19.3 | 81.7 | 21.1 | 13.6 | 17.7 | 12.3 | 6.5 | 42.3 |
| PCEDA [47] | | 90.2 | 44.7 | 82.0 | 28.4 | **28.4** | 24.4 | 33.7 | 35.6 | 83.7 | **40.5** | 75.1 | 54.4 | **28.2** | 80.3 | 23.8 | **39.4** | 0.0 | 22.8 | 30.8 | 44.6 |
| **Ours** | | **93.7** | 53.6 | 83.5 | **35.1** | 21.1 | 28.6 | **36.2** | **42.0** | 82.2 | 32.4 | **86.5** | 47.3 | 19.4 | **83.8** | 26.0 | 30.7 | 30.2 | 13.1 | **32.1** | **46.2** |

Table 2: Results of mean IoU and per-category IoUs for SYNTHIA-to-Cityscapes benchmark. "mIoU" is the mean IoU over all of the 16 categories, and "mIoU*" is that over 13 categories excluding 3 categories marked by "*". Results of "-" were not reported in the papers.

| Method | Structure | road | side. | buil. | wall* | fence* | pole* | tlig. | tsign | vege. | sky | pers. | rider | car | bus | moto. | bicy. | mIoU | mIoU* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BDL [17] | ResNet101 | 86.0 | 46.7 | 80.3 | - | - | - | 14.1 | 11.6 | 79.2 | 81.3 | 54.1 | 27.9 | 73.7 | **42.2** | 25.7 | 45.3 | - | 51.4 |
| CAG-UDA [43] | | 84.7 | 40.8 | 81.7 | 7.8 | 0.0 | 35.1 | 13.3 | 22.7 | **84.5** | 77.6 | **64.2** | 27.8 | 80.9 | 19.7 | 22.7 | 48.3 | 44.5 | 52.6 |
| RectPLL [9] | | 87.6 | 41.9 | **83.1** | **14.7** | **1.7** | **36.2** | **31.3** | 19.9 | 81.6 | 80.6 | 63.0 | 21.8 | **86.2** | 40.7 | 23.6 | **53.1** | 47.9 | 54.9 |
| FDA [24] | | 79.3 | 35.0 | 73.2 | - | - | - | 19.9 | 24.0 | 61.7 | 82.6 | 61.4 | 31.1 | 83.9 | 40.8 | **38.4** | 51.1 | - | 52.5 |
| SIM [44] | | 83.0 | 44.0 | 80.3 | - | - | - | 17.1 | 15.8 | 80.5 | 81.8 | 59.9 | **33.1** | 70.2 | 37.3 | 28.5 | 45.8 | - | 52.1 |
| PIT [45] | | 83.1 | 27.6 | 81.5 | 8.9 | 0.3 | 21.8 | 26.4 | **33.8** | 76.4 | 78.8 | **64.2** | 27.6 | 79.6 | 31.2 | 31.0 | 31.3 | 44.0 | 51.8 |
| LTIR [46] | | **92.6** | 53.2 | 79.2 | - | - | - | 1.6 | 7.5 | 78.6 | **84.4** | 52.6 | 20.0 | 82.1 | 34.8 | 14.6 | 39.4 | - | 49.3 |
| PCEDA [47] | | 85.9 | 44.6 | 80.8 | 9.0 | 0.8 | 32.1 | 24.8 | 23.1 | 79.5 | 83.1 | 57.2 | 29.3 | 73.5 | 34.8 | 32.4 | 48.2 | 46.2 | 53.6 |
| **Ours** | | 91.9 | **54.6** | 81.3 | 7.2 | 1.1 | 33.8 | 29.6 | 30.0 | 78.5 | 80.0 | 61.6 | 28.9 | 82.4 | 32.8 | 37.3 | 52.6 | **49.0** | **57.1** |
| BDL [17] | VGG16 | 72.0 | 30.3 | 74.5 | 0.1 | 0.3 | 24.6 | 10.2 | 25.2 | 80.5 | 80.0 | 54.7 | 23.2 | 72.7 | 24.0 | 7.5 | 44.9 | 39.0 | 46.1 |
| FDA [24] | | 84.2 | 35.1 | 78.0 | 6.1 | 0.4 | 27.0 | 8.5 | 22.1 | 77.2 | 79.6 | 55.5 | 19.9 | 74.8 | **24.9** | 14.3 | 40.7 | 40.5 | 47.3 |
| PIT [45] | | 81.7 | 26.9 | 78.4 | **6.3** | 0.2 | 19.8 | 13.4 | 17.4 | 76.7 | 74.1 | 47.5 | 22.4 | **76.0** | 21.7 | **19.6** | 27.7 | 38.1 | 44.9 |
| LTIR [46] | | **89.8** | **48.6** | **78.9** | - | - | - | 0.0 | 4.7 | **80.6** | 81.7 | 36.2 | 13.0 | 74.4 | 22.5 | 6.5 | 32.8 | - | 43.8 |
| PCEDA [47] | | 79.7 | 35.2 | 78.7 | 1.4 | **0.6** | 23.1 | 10.0 | **28.9** | 79.6 | 81.2 | 51.2 | **25.1** | 72.2 | 24.1 | 16.7 | **50.4** | 41.1 | 48.7 |
| **Ours** | | 84.6 | 40.3 | 74.5 | 0.5 | 0.1 | **27.7** | **25.4** | 25.1 | 78.0 | **81.8** | **58.0** | 19.4 | 70.5 | 24.3 | 17.7 | 41.5 | **41.8** | **49.3** |

The SYNTHIA dataset [2] is a synthetic dataset consisting of photo-realistic images of driving scenarios rendered from a virtual city. We used the SYNTHIA-RAND-CITYSCAPES subset that contains 9,400 images with resolutions of 1,280×760. It shares 16 common categories with Cityscapes.

### *4.2. Main Results and Comparison with Results of State-of-the-art Methods*

The results obtained by of our method and eight recent state-of-the-art methods are shown in Table 1 and Table 2. Here we summarize the common ideas in the previous methods. Pseudo labels were used in most methods except PIT and PCEDA. BDL [17],

FDA [24], LTIR [46] and PCEDA [47] perform I2I translation to reduce the visual domain gap, and FDA uniquely uses a translation approach based on Fourier Transform. BDL, RectPLL [9], SIM [44] and LTIR share a similar component of the output-space adversarial learning. PIT [45] is a distinctive method that explores the domain-invariant interactive relation between the image-level information and pixel-level information.

Table 1 shows the results for the GTA5-to-Cityscapes benchmark. As shown in Table 1, our method achieved the best performance with both base structures. Our mean IoUs were 1.8 and 1.6 higher than the second best ones, respectively. Moreover, our method performed best in 7 categories in both structure settings, which is superior to all of the most competitive methods. For the more challenging benchmark SYNTHIA-to-Cityscapes shown in Table 2, mean IoUs were calculated over 13 categories and 16 categories respectively following some previous works. In the ResNet101 setting, our method outperformed the second best one by 2.2 and 1.1 over 13 categories and 16 categories, respectively. In the VGG16 setting, our method showed slight superiority over the second best one by 0.6 and 0.7 over 13 categories and 16 categories, respectively. In conclusion, our method achieved state-of-the-art results for the two benchmarks.

From Table 1 and Table 2, better adaptation performance was achieved with GTA5 than with SYNTHIA as the source domain, which we attribute to the following factors: dataset size and domain gap. The GTA5 dataset has a larger size (24,966 training images) than that of the SYNTHIA dataset (9,400 training images). And more importantly, the domain gap between GTA5 and Cityscapes is less than that between SYNTHIA and Cityscapes. For example, images of GTA5 and Cityscapes are all obtained with an inside car view, while SYNTHIA is composed of images with different views and angles such as a bird's-eye view. Such differences make the adaptation from SYNTHIA harder than that from GTA5 and consequently lead to inferior performance.

### *4.3. Supplementary Results and Analyses*

#### *4.3.1. Ablation study*

The results of an ablation study for analyzing the contribution of each component in our method are shown in Table 3. First, training with only the original source domain images in which no adaptation was performed was set as a baseline with mean IoUs of

Table 3: Ablation study for the components in our method with the ResNet101 structure. $\mathcal{L}_{\mathrm{sup}}$, $\mathcal{L}_{\mathrm{con}}$ and $\mathcal{L}_{\mathrm{psl}}$ are defined in Section 3.1.

| Method | Components | | | Mean IoU | |
|---|---|---|---|---|---|
| | $\mathcal{L}_{\mathrm{sup}}$ | $\mathcal{L}_{\mathrm{con}}$ | $\mathcal{L}_{\mathrm{psl}}$ | GTA5 | SYNTHIA |
| Source only (non-adaptation) | | | | 35.1 | 33.8 |
| Source only (style-diversified) | ✓ | | | 46.9 | 40.5 |
| Self-ensembling | ✓ | ✓ | | 48.1 | 42.1 |
| Self-ensembling + pseudo-label learning | ✓ | ✓ | ✓ | 52.4 | 49.0 |

Table 4: Comparison of style diversification measures with the ResNet101 structure. The results were obtained using the self-ensembling architecture without pseudo-label learning. "PM" denotes the measure used in the proposed method.

| Style diversification measure | Mean IoU | |
|---|---|---|
| | GTA5 | SYNTHIA |
| Color jitter | 43.5 | 40.1 |
| CycleGAN + color jitter | 46.5 | 39.5 |
| MUNIT | 43.9 | 40.6 |
| MUNIT + color jitter | 43.2 | 40.7 |
| Semantic-aware MUNIT | 47.8 | 41.4 |
| Semantic-aware MUNIT + color jitter (PM) | **48.1** | **42.1** |

35.1 and 33.8 for GTA5-to-Cityscapes and SYNTHIA-to-Cityscapes, respectively. The use of style-diversified source domain images without the self-ensembling architecture greatly improved the mean IoUs to 46.9 and 40.5 owing to the domain transfer and intra-domain style diversification. By using the target domain images together with the self-ensembling architecture, the mean IoUs were further improved to 48.1 and 42.1. Finally, the enhancement with pseudo-label learning raised the mean IoUs to 52.4 and 49.0.

#### *4.3.2. Measures of style diversification*

The intra-domain style diversification plays a key role in our method. Therefore, we compared several different style diversification measures and the results are shown in Table 4. Color jitter simply modifies the brightness, contrast, saturation and hue of an image without transferring its domain. It contributed to the intra-domain style-invariant representation learning, whereas by using solely the color jitter, the UDA performance was not satisfactory. By using CycleGAN [16], the one-to-one translation model, to translate the source domain images to the target domain before imposing

Table 5: Study on the influence of the number of sampled styles for one image with the ResNet101 structure.

| Number of sampled styles | Mean IoU | |
|---|---|---|
| | GTA5 | SYNTHIA |
| 1 | 42.4 | 38.6 |
| 2 | **48.1** | **42.1** |
| 4 | 48.0 | 41.9 |
| 8 | 48.1 | 42.0 |

the color jitter, the performance was improved for GTA-to-Cityscapes but not for the other benchmark. MUNIT, the base model of our I2I translation model, had slightly better performance than the color jitter. In addition, imposing the color jitter following the translation with MUNIT did not achieve further improvement. Our I2I translation model, the semantic-aware version of MUNIT, outperformed its original version and also all of the other above-mentioned measures. Moreover, slight improvement was achieved by combining it with the color jitter, which was adopted in the proposed method.

*4.3.3. Number of sampled styles*

It may be conjectured that sampling more than two style-diversified copies for each image could further improve the performance. To verify this, we did a study on the influence of the number of sampled intra-domain styles for one image. However, as shown in Table 5, our method failed unexpectedly to gain further improvement by increasing the number of sampled styles. According to the results, sampling two styles is sufficient for the style-invariant representation learning. On the other hand, the results also suggest that the improvement by sampling two styles was due to the proposed intra-domain style-invariant representation learning, not to the double copies in a mini-batch, because the performance should have been further improved by sampling four and eight styles if the number of copies in a mini-batch is the reason.

For the reason why the performance was not in proportion to the number of sampled styles, our explanation is as follows. First, the Cityscapes target domain has a relatively consistent global style since the images were all collected in the same country and in similar weather and illumination conditions. Moreover, the camera used for collecting the images was also the same one and positioned with the same angle. Consequently,

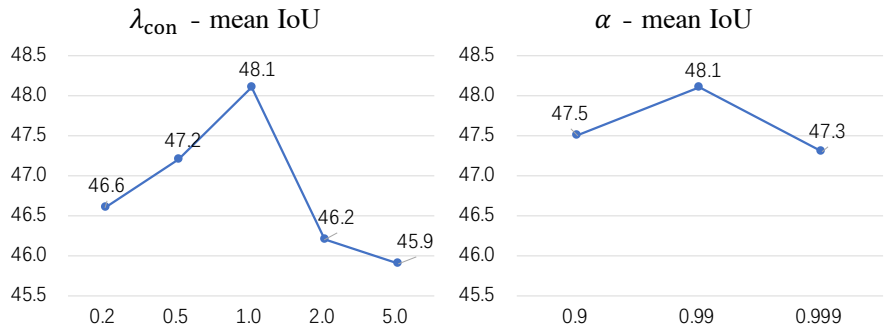


Figure 4: Influence of hyper-parameters $\lambda_{\text{con}}$ and $\alpha$ on mean IoU. The experiments were conducted without pseudo-label learning on GTA5-to-Cityscapes with the ResNet101 structure.

the slight variance of the intra-domain styles in the target domain made it redundant to sample more styles. However, for applying our method to a target domain with highly diverse intra-domain styles, the influence of the number of sampled styles should be studied again. Another possible reason is that the style sampling for each image is random and independent so that the insufficiency of the sampled styles for one image can be remedied to some extent by the style sampling for the other images. In other words, the diverse intra-domain styles may not necessarily be covered by the style sampling for each single image. Since we trained the model for a large number of iterations, a wide range of styles were sampled with different images to learn the style-invariant representation, and the need of sampling a large number of styles for one image was thus reduced.

#### *4.3.4. Hyper-parameter analyses*

Results of analyses for hyper-parameters $\lambda_{\text{con}}$ and $\alpha$ are shown in Fig. 4. $\lambda_{\text{con}}$ is the weight parameter for the consistency loss in the self-ensembling architecture, and $\alpha$ is the EMA updating parameter for the teacher model. Our method achieved the best performance at $\lambda_{\text{con}} = 1.0$ and $\alpha = 0.99$ without pseudo-label learning. As shown in Fig. 4, a small $\lambda_{\text{con}} = 0.2$ and a large $\lambda_{\text{con}} \geq 2.0$ clearly reduced the performance, and $\alpha$ had only a slight effect on the performance.

Figure 5: Examples of our I2I translation results for GTA5-to-Cityscapes. The 1st row: original GTA5 images. The 2nd to 5th rows: style-diversified translation results to Cityscapes.

#### *4.3.5. Examples of our I2I translation results*

Examples of the two domain translations and the target domain diversification are shown in Fig. 5, Fig. 6, and Fig. 7, respectively. In the translation results for GTA5-to-Cityscapes of Fig. 5, diversified intra-domain styles appeared in the surface texture of roads, buildings and terrains. In the translation results for SYNTHIA-to-Cityscapes of Fig. 6, color temperature appeared diverse as well as the texture of roads and vegetation. In the diversification results for Cityscapes of Fig. 7, image contrast and light intensity showed diversified characteristics. The above-mentioned varied characteristics are abstracted as the intra-domain styles and underlie the proposed method.

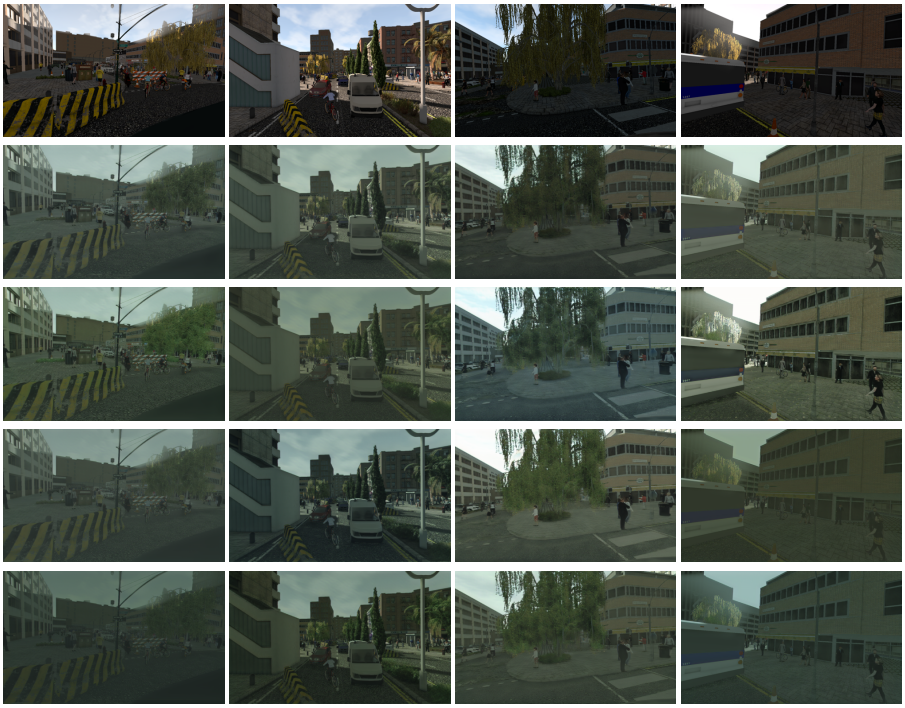

Figure 6: Examples of our I2I translation results for SYNTHIA-to-Cityscapes. The 1st row: original SYNTHIA images. The 2nd to 5th rows: style-diversified translation results to Cityscapes.

## 5. Conclusion

In this paper, we have proposed a novel concept of learning intra-domain style-invariant representation for UDA of semantic segmentation, and we constructed a method based on the proposed concept. Learning representation invariant to the diversified intra-domain styles contributes to the generalization in the target domain. To realize this, we first trained a semantic-aware multimodal I2I translation model to obtain images with diversified intra-domain styles and consistent semantic contents. Then we used the generated images to train the segmentation model with the self-ensembling architecture. By further employing pseudo-label learning, our method achieved state-of-the-art performance for two benchmarks. Moreover, we demonstrated the effectiveness of our method by conducting elaborate experiments and analyses.

In future researches, we think that the clearest direction for improving our work is

Figure 7: Examples of our I2I translation results for diversifying Cityscapes. The 1st row: original Cityscapes images. The 2nd to 5th rows: style-diversified translation results.

the I2I translation for diversifying the intra-domain styles. Specifically, although our semantic-aware MUNIT succeeded in diversifying the intra-domain styles of the images with preservation of most contents, some objects that are hard to recognize for the pre-trained segmentation model were still prone to be translated inappropriately, possibly resulting in misleading the training of the UDA model. Therefore, how to mitigate the influence of the misleading semantic information is a key to improve the image translation and accordingly improve the final UDA performance. Moreover, it may limit the training of the semantic-aware MUNIT that the pre-trained segmentation model is not updated during the training. Intuitively, more accurate semantic information may improve the I2I translation, and hence we can consider an architecture that combines the training of the semantic-aware MUNIT and the self-ensembling, for example, in a collaborative manner. In conclusion, since the style-diversified images are essential for learning the intra-domain style-invariant representation, we believe that it

is the most significant to improve the I2I translation in terms of the style-diversification and the content preservation.

**Acknowledgements**

This study was partly supported by JSPS KAKENHI Grant Number JP17H01744. This study was conducted on the Data Science Computing System of Education and Research Center for Mathematical and Data Science, Hokkaido University.

**References**

[1] S. R. Richter, V. Vineet, S. Roth, V. Koltun, Playing for data: Ground truth from computer games, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 102–118.

[2] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, A. M. Lopez, The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3234–3243.

[3] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, T. Darrell, Cycada: Cycle-consistent adversarial domain adaptation, in: Proceedings of the International Conference on Machine Learning, PMLR, 2018, pp. 1989–1998.

[4] R. Li, W. Cao, Q. Jiao, S. Wu, H.-S. Wong, Simplified unsupervised image translation for semantic segmentation adaptation, Pattern Recognition 105 (2020) 107343.

[5] A. Dundar, M.-Y. Liu, Z. Yu, T.-C. Wang, J. Zedlewski, J. Kautz, Domain stylization: A fast covariance matching framework towards domain adaptation, IEEE Transactions on Pattern Analysis and Machine Intelligence 43 (7) (2021) 2360–2372.

[6] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, M. Chandraker, Learning to adapt structured output space for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7472–7481.

[7] Z. Zheng, Y. Yang, Unsupervised scene adaptation with memory regularization in vivo, in: Proceedings of the International Conference on International Joint Conferences on Artificial Intelligence, 2021, pp. 1076–1082.

[8] W. Chen, H. Hu, Generative attention adversarial classification network for unsupervised domain adaptation, Pattern Recognition 107 (2020) 107440.

[9] Z. Zheng, Y. Yang, Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation, International Journal of Computer Vision 129 (4) (2021) 1106–1120.

[10] Y. Chen, C. Yang, Y. Zhang, Y. Li, Deep conditional adaptation networks and label correlation transfer for unsupervised domain adaptation, Pattern Recognition 98 (2020) 107072.

[11] J. Liang, R. He, Z. Sun, T. Tan, Exploring uncertainty in pseudo-label guided unsupervised domain adaptation, Pattern Recognition 96 (2019) 106996.

[12] S. Laine, T. Aila, Temporal ensembling for semi-supervised learning, arXiv preprint arXiv:1610.02242.

[13] A. Tarvainen, H. Valpola, Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, in: Advances in Neural Information Processing Systems, 2017, pp. 1195–1204.

[14] J. Choi, T. Kim, C. Kim, Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 6830–6840.

[15] X. Huang, M.-Y. Liu, S. Belongie, J. Kautz, Multimodal unsupervised image-to-image translation, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 172–189.

[16] J.-Y. Zhu, T. Park, P. Isola, A. A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2223–2232.

[17] Y. Li, L. Yuan, N. Vasconcelos, Bidirectional learning for domain adaptation of semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 6936–6945.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.

[19] Y. Zhang, Z. Qiu, T. Yao, D. Liu, T. Mei, Fully convolutional adaptation networks for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6810–6818.

[20] Z. Wu, X. Han, Y.-L. Lin, M. Gokhan Uzunbas, T. Goldstein, S. Nam Lim, L. S. Davis, Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 518–534.

[21] L. A. Gatys, A. S. Ecker, M. Bethge, Image style transfer using convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2414–2423.

[22] D. Ulyanov, V. Lebedev, A. Vedaldi, V. S. Lempitsky, Texture networks: Feed-forward synthesis of textures and stylized images., in: Proceedings of International Conference on Machine Learning, Vol. 1, 2016, p. 4.

[23] X. Huang, S. Belongie, Arbitrary style transfer in real-time with adaptive instance normalization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1501–1510.

[24] Y. Yang, S. Soatto, Fda: Fourier domain adaptation for semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 4085–4095.

[25] J. Huang, D. Guan, A. Xiao, S. Lu, Multi-level adversarial network for domain adaptive semantic segmentation, Pattern Recognition 123 (2022) 108384.

[26] L. Du, J. Tan, H. Yang, J. Feng, X. Xue, Q. Zheng, X. Ye, X. Zhang, Ssf-dan: Separated semantic feature based domain adaptation network for semantic segmentation, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 982–991.

[27] Y. Luo, P. Liu, L. Zheng, T. Guan, J. Yu, Y. Yang, Category-level adversarial adaptation for semantic segmentation using purified features, IEEE Transactions on Pattern Analysis and Machine Intelligence.

[28] Y. Grandvalet, Y. Bengio, Semi-supervised learning by entropy minimization, in: Advances in Neural Information Processing Systems, 2005, pp. 529–536.

[29] T.-H. Vu, H. Jain, M. Bucher, M. Cord, P. Pérez, Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2517–2526.

[30] M. Chen, H. Xue, D. Cai, Domain adaptation for semantic segmentation with maximum squares loss, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 2090–2099.

[31] D. Guan, J. Huang, S. Lu, A. Xiao, Scale variance minimization for unsupervised domain adaptation in image segmentation, Pattern Recognition 112 (2021) 107764.

[32] A. Almahairi, S. Rajeshwar, A. Sordoni, P. Bachman, A. Courville, Augmented cyclegan: Learning many-to-many mappings from unpaired data, in: Proceedings of International Conference on Machine Learning, PMLR, 2018, pp. 195–204.

[33] V. Dumoulin, J. Shlens, M. Kudlur, A learned representation for artistic style, arXiv preprint arXiv:1610.07629.

[34] Y. Luo, J. Zhu, M. Li, Y. Ren, B. Zhang, Smooth neighbors on teacher graphs for semi-supervised learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8896–8905.

[35] S. Park, J. Park, S.-J. Shin, I.-C. Moon, Adversarial dropout for supervised and semi-supervised learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018.

[36] T. Miyato, S.-i. Maeda, M. Koyama, S. Ishii, Virtual adversarial training: a regularization method for supervised and semi-supervised learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 41 (8) (2018) 1979–1993.

[37] T. Park, M.-Y. Liu, T.-C. Wang, J.-Y. Zhu, Semantic image synthesis with spatially-adaptive normalization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2337–2346.

[38] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (4) (2017) 834–848.

[39] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[40] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.

[41] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.

[42] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3213–3223.

[43] Q. Zhang, J. Zhang, W. Liu, D. Tao, Category anchor-guided unsupervised domain adaptation for semantic segmentation, in: Advances in Neural Information Processing Systems, 2019, pp. 435–445.

[44] Z. Wang, M. Yu, Y. Wei, R. Feris, J. Xiong, W.-m. Hwu, T. S. Huang, H. Shi, Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12635–12644.

[45] F. Lv, T. Liang, X. Chen, G. Lin, Cross-domain semantic segmentation via domain-invariant interactive relation transfer, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 4334–4343.

[46] M. Kim, H. Byun, Learning texture invariant representation for domain adaptation of semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12975–12984.

[47] Y. Yang, D. Lao, G. Sundaramoorthi, S. Soatto, Phase consistent ecological domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9011–9020.