Diffusion-based Image Translation with Label Guidance for Domain Adaptive Semantic Segmentation

Duo Peng\(^1\)  Ping Hu\(^2\)  Qiuhong Ke\(^3\)  Jun Liu\(^1,\ast\)

\(^1\)Singapore University of Technology and Design  \(^2\)Boston University  \(^3\)Monash University

duo_peng@mymail.sutd.edu.sg, pinghu@bu.edu, qiuhong.ke@monash.edu, jun_liu@sutd.edu.sg

Abstract

Translating images from a source domain to a target domain for learning target models is one of the most common strategies in domain adaptive semantic segmentation (DASS). However, existing methods still struggle to preserve semantically-consistent local details between the original and translated images. In this work, we present an innovative approach that addresses this challenge by using source-domain labels as explicit guidance during image translation. Concretely, we formulate cross-domain image translation as a denoising diffusion process and utilize a novel Semantic Gradient Guidance (SGG) method to constrain the translation process, conditioning it on the pixel-wise source labels. Additionally, a Progressive Translation Learning (PTL) strategy is devised to enable the SGG method to work reliably across domains with large gaps. Extensive experiments demonstrate the superiority of our approach over state-of-the-art methods.

1. Introduction

Semantic segmentation is a fundamental computer vision task that aims to assign a semantic category to each pixel in an image. In the past decade, fully supervised methods [47, 59, 29, 84] have achieved remarkable progress in semantic segmentation. However, the progress is at the cost of a large number of dense pixel-level annotations, which are significantly labor-intensive and time-consuming to obtain [3, 47]. Facing this problem, one alternative is to leverage synthetic datasets, e.g., GTA5 [57] or SYNTHIA [60], where the simulated images and corresponding pixel-level labels can be generated by the computer engine freely. Although the annotation cost is much cheaper than manual labeling, segmentation models trained with such synthetic datasets often do not work well on real images due to the large domain discrepancy.

*Corresponding Author

The task of domain adaptive semantic segmentation (DASS) aims to address the domain discrepancy problem by transferring knowledge learned from a source domain to a target one with labels provided for the source domain.
only. It is a very challenging problem as there can be a large discrepancy (e.g., between synthetic and real domains), resulting in great difficulties in knowledge transfer. In DASS, one type of methods [24, 7, 61, 5, 83, 12, 48, 68, 49, 66, 72] focuses on aligning the data distributions from different domains in the feature space, which obtains promising performance, while another type of approaches [86, 87, 40, 81, 38, 42] aims to generate pseudo labels for unlabeled target images, also making great achievements. Besides above approaches, image translation [85] is also an effective and important way to handle the DASS problem. It focuses on translating the labeled source images into the target domain, then using the translated images and source labels to train a target-domain segmentation model. In the past few years, lots of DASS studies [54, 23, 6, 44, 16] have been conducted based on image translation. This is because image translation allows for easy visual inspection of the adaptation method’s results, and the translated images can be saved as a dataset for the future training of any model, which is convenient. Existing image translation methods mainly adopt generative adversarial networks (GANs) [19] to translate images across a large domain gap. However, due to the intractable adversarial training, GAN models can be notoriously hard to train and prone to show unstable performance [62, 55]. Furthermore, many studies [51, 11, 55] have also shown that GAN-based methods struggle with preserving local details, leading to semantic inconsistencies between the translated images and corresponding labels, as shown in the second row of Fig. 1 (b). Training the segmentation model on such mismatched image and label pairs will cause sub-optimal domain adaptation performance. Facing these inherent limitations, the development of GAN-based image translation methods is increasingly encountering a bottleneck. Yet, there is currently little research exploring alternative approaches to improve image translation for DASS beyond GANs.

Denoising Diffusion Probabilistic Models (DDPMs), also known as diffusion models, have recently emerged as a promising alternative to GANs for various tasks, such as image generation [11, 69], image restoration [73, 79], and image editing [1, 52], etc. Diffusion models have been shown to have a more stable training process [11, 55, 22]. Inspired by its strong capability, we propose to exploit the diffusion model for image translation in DASS. Specifically, based on the diffusion technique, we propose a label-guided image translation framework to preserve local details. We observe that existing image translation methods [23, 54, 6, 44, 41] mainly focus on training a translation model with image data alone, while neglecting the utilization of source-domain labels during translation, as shown in Fig. 1 (a). Since source labels can explicitly indicate the semantic category of each pixel, introducing them to guide the translation process should improve the capability of the translation model to preserve details. A straightforward idea of incorporating the source labels is to directly train a conditional image translation model, where the translated results are semantically conditioned on the pixel-wise source labels. However, it is non-trivial since the source labels and target images are not paired, which cannot support the standard conditional training. Recent advances [11, 55, 58] have shown that a pre-trained unconditional diffusion model can become conditional with the help of the gradient guidance [11]. The gradient guidance method can guide the pre-trained unconditional diffusion model to provide desired results by directly affecting the inference process (without any training). In light of this, we propose to first train an unconditional diffusion-based image translation model, and then apply gradient guidance to the image translation process, making it conditioned on source labels. However, we still face two challenges: (1) Traditional gradient guidance methods generally focus on guiding the diffusion model based on image-level labels (i.e., classification labels), while the DASS problem requires the image translation to be conditioned on pixel-level labels (i.e., segmentation labels). (2) The gradient guidance methods typically work within a single domain, whereas the DASS task requires it to guide the image translation across different domains. To address the first challenge, we propose a novel Semantic Gradient Guidance (SGG), which can precisely guide the image translation based on fine-grained segmentation labels. To tackle the second challenge, we carefully design a Progressive Translation Learning strategy (PTL), which is proposed to drive the SGG method to reliably work across a large domain gap. With the help of SGG and PTL, our framework effectively handles the image translation for DASS in a fine-grained manner, which is shown in the third row of Fig. 1 (b).

In summary, our contributions are three-fold. (i) We propose a novel diffusion-based image translation framework with the guidance of pixel-wise labels, which achieves image translation with fine-grained semantic preservation. To the best of our knowledge, this is the first study to exploit the diffusion model for DASS. (ii) We devise a Semantic Gradient Guidance (SGG) scheme accompanied with a Progressive Translation Learning (PTL) strategy to guide the translation process across a large domain gap in DASS. (iii) Our method achieves the new state-of-the-art performance on two widely used DASS settings. Remarkably, our method outperforms existing GAN-based image translation methods significantly on various backbones and settings.

2. Related Work

Domain adaptive semantic segmentation (DASS) is a task that aims to improve the semantic segmentation model’s adaptation performance in new target scenarios without access to the target labels. Current DASS meth-
ods [24, 61, 40, 8, 23] have pursued three major directions to bridge the domain gap. The first direction aims to align data distributions in the latent feature space. With the advent of adversarial learning [14], many methods [24, 7, 56, 61, 5, 37, 83, 12, 48, 68, 49, 66, 72] adopt the adversarial learning to align feature distributions by fooling a domain discriminator. However, the model based on adversarial learning can be hard to train [82, 62, 53], as it requires carefully balancing the capacity between the feature encoder (i.e., generator) and the domain discriminator, which is intractable. The second direction considers exploring the network’s potential of self-training on the target domain [81, 50, 40, 38, 71, 42, 86, 87]. Self-training methods typically select the results with high confidence scores as pseudo labels to fine-tune the segmentation model. These methods usually introduce inevitable label noise [44, 30, 76], as we cannot fully guarantee that the selected pseudo labels are correct especially when the domain gap is large. The third direction is inspired by the image translation technique [85], aiming to transfer the style of source labeled images to that of the target domain [8, 54, 23, 6, 44, 16]. Inspired by the advances of the generative adversarial networks (GANs), they mostly adopt GANs as image translation models. While remarkable progress has been achieved, many studies [51, 11, 55] have indicated that GAN-based image translation methods tend to show limited capacity for preserving semantic details. Unlike existing methods that handle image translation based on the input source image, our method enables the model to translate images further conditioned on the source labels, which enables fine-grained semantic preservation.

Denoising Diffusion Probabilistic Models (DDPMs), also namely diffusion models, are a class of generative models inspired by the nonequilibrium thermodynamics. Diffusion models [64, 22] generally learn a denoising model to gradually denoise from an original common distribution, e.g., Gaussian noise, to a specific data distribution. It is first proposed by Sohl-Dickstein et al. [64], and has recently attracted much attention in a wide range of research fields including computer vision [11, 69, 13, 1, 52, 9, 9, 17, 4, 20], nature language processing [2, 18, 43], audio processing [28, 28, 34, 67, 74] and multi-modality [33, 45, 27, 70]. Based on the diffusion technique, Dhariwal et al. proposed a gradient guidance method [11] to enable pre-trained unconditional diffusion models to generate images based on a specified class label. Although effective, the gradient guidance method generally works with image-level labels and is limited to a single domain, making it challenging to be adopted for DASS. To address these issues, in this paper, we propose a novel Semantic Gradient Guidance (SGG) method and a Progressive Translation Learning (PTL) strategy, enabling the utilization of pixel-level labels for label-guided cross-domain image translation.

3. Revisiting Diffusion Models

As our method is proposed based on the Denoising Diffusion Probabilistic Model (DDPM) [64, 22], here, we briefly revisit DDPM. DDPM is a generative model that converts data from noise to clean images by gradually denoising. A standard DDPM generally contains two processes: a diffusion process and a reverse process. The diffusion process contains no learnable parameters and it just used to generate the training data for diffusion model’s learning. Specifically, in the diffusion process, a slight amount of Gaussian noise is repeatedly added to the clean image $x_0$, making it gradually corrupted into a noisy image $x_T$ after $T$-step noise-adding operations. A single step of diffusion process (from $x_{t-1}$ to $x_t$) is represented as follows:

$$q(x_t|x_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}x_{t-1}, \beta_t I),$$

where $\beta_t$ is the predefined variance schedule. Given a clean image $x_0$, we can obtain $x_T$ by repeating the above diffusion step from $t = 1$ to $t = T$. Particularly, an arbitrary image $x_t$ in the diffusion process can also be directly calculated as:

$$x_t = \sqrt{\alpha_t}x_0 + \sqrt{1-\alpha_t}\epsilon,$$

where $\alpha_t = 1-\beta_t$, $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ and $\epsilon \sim \mathcal{N}(0, I)$. Based on this equation, we can generate images from $x_0$ to $x_T$ for training the diffusion model.

The reverse process is opposite to the diffusion process, which aims to train the diffusion model to gradually convert data from the noise $x_T$ to the clean image $x_0$. One step of the reverse process (from $x_t$ to $x_{t-1}$) can be formulated as:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t, t), \Sigma_\theta(x_t, t)).$$

Given the noisy image $x_T$, we can denoise it to the clean image $x_0$ by repeating the above equation from $t = T$ to $t = 1$. In Eq. 3, the covariance $\Sigma_\theta$ is generally fixed as a predefined value, while the mean $\mu_\theta$ is estimated by the diffusion model. Specifically, the diffusion model estimates $\mu_\theta$ via a parameterized noise estimator $\epsilon_\theta(x_t, t)$ which is usually a U-Net [59]. Given $x_t$ as the input, the diffusion model will output the value of $\mu_\theta$ as follows:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{1-\beta_t}} \left( x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right),$$

For training the diffusion model, the images generated in the diffusion process are utilized to train the noise estimator.
4. Method

We address the problem of domain adaptive semantic segmentation (DASS), where we are provided the source images $X^S$ with corresponding source labels $Y^S$ and the target images $X^T$ (without target labels). Our goal is to train a label-guided image translation model to translate source images into target-like ones. Then we can use the translated data to train a target-domain segmentation model. To achieve this goal, we first train a basic image translation model based on the diffusion technique, which is the baseline of our approach (Sec. 4.1). Then, we impose our SGG method to the baseline model, making it conditioned on the pixel-wise source labels (Sec. 4.2). We further propose a PTL training strategy to enable our SGG to work across a large domain gap (Sec. 4.3). Finally, we detail the training and inference procedures (Sec. 4.4).

4.1. Diffusion-based Image Translation Baseline

We start with the baseline of our framework, which is a basic image translation model based on diffusion models. Before translation, we use the (unlabeled) target domain images $X^T$ to pre-train a standard diffusion model via Eq. 5. As the diffusion model is essentially a denoising model, we use the noise as an intermediary to link the two large-gap domains. As shown in Fig. 2, we first leverage the diffusion process to add $n$-step noise onto the source image $x_0$ ($x_0 \in X^S$), making it corrupted into $x_n$, and then use the pre-trained diffusion model to reverse (denoise) it for $n$ steps, obtaining $x_n^0$. Since the diffusion model is pre-trained on the target domain, the reverse (denoising) process will convert the noise into target-domain content, making the denoised image $x_n^0$ shown an appearance closer to the target-domain images than the original source-domain image $x_0$. Note that the diffusion model solely trained on the target domain is able to denoise the noisy image from a different source domain towards the target domain, because the diffusion model’s training objective is derived from the variational bound on the negative log-likelihood $\mathbb{E}[-\log p_d(X^T)]$, indicating it learns to generate target-domain data from various distributions. This property of diffusion model has been verified in [65] and has been widely used in many studies [52, 15, 9].

4.2. Semantic Gradient Guidance (SGG)

The baseline model presented above handles image translation in an unconditional manner, where denoising from much noise however will freely generate new content that may not be strictly consistent with the original image. Inspired by the gradient guidance [11], we here propose a novel SGG method to enable the unconditional baseline to be conditioned on source labels.

Our SGG consists of two guidance modules, namely the Local Class-regional Guidance (LCG) and the Global Scene-harmonious Guidance (GSG). They are proposed to affect the reverse process of the baseline model, aiming to guide the denoising process to generate content semantically consistent with source labels. Specifically, the LCG is proposed to preserve local details, while the GSG is designed to enhance the global harmony of the denoised images. As shown in Fig. 3 (a), we alternately apply the LCG and GSG modules at each reverse step, aiming to provide both local-precision and global-harmony guidance.

(1) Local Class-regional Guidance (LCG)

The key idea of LCG is to use a pixel classifier to calculate loss to measure if the generated (denoised) pixel has the same semantic as the source semantic label, and then utilize the loss gradient as the guidance signal to guide the pixel to generate same-semantic content. Assuming that we already have a pre-trained segmentation model $g$ (i.e., the pixel classifier) that can perform pixel classification well over domains (note that we will address this assumption in Sec. 4.3), based on $g$, LCG can carry out guidance as follows.

Here, we use $x_k^0$ to denote the input of the LCG module, and $x_{k-1}^0$ as its output, where $k \in \{n, n-2, n-4, \ldots\}$. As shown in Fig. 3 (b), given a source label map $y \in Y^S$, we can leverage $y$ to obtain a binary mask $m_c$ that indicates the region of the $c$-th class in the source image ($c \in \mathcal{C}$ and $\mathcal{C}$ denotes all classes in the source dataset). Given the input $x_k^0$, we can multiply $x_k^0$ with $m_c$ to obtain the pixels inside the $c$-th class region, which can be fed into the pre-trained segmentation model $g$ to predict the label probabilities of the pixels. The output can be used to calculate the cross-entropy loss with the source label, aiming to use the loss gradient to guide the generation of pixels in the $c$-th class region. As shown in Fig. 3 (b), we can use classifier $g$ to
compute the cross-entropy loss $L_{\text{local}}$ for pixels in the local $c$-th class region, which is formulated as:

$$L_{\text{local}}(x_k^\phi, y, c) = L_{\text{ce}} \left( g(x_k^\phi \odot m_c), y \odot m_c \right),$$

where $L_{\text{ce}}$ is the cross-entropy loss, $\odot$ denotes the element-wise multiplication and $y \odot m_c$ is the source label of the class $c$.

In the standard reverse process, given the noisy image $x_k^\phi$, the diffusion model denoises it by estimating the mean value $\mu_\theta$ (Eq. 4) and then using it to obtain the denoised image $x_{k-1}^\phi$ via sampling (Eq. 3). Therefore, the mean $\mu_\theta$ will determine the content of the denoised image. Based on this, we guide the reverse process by using the computed loss gradient $\nabla x_k^\phi L_{\text{local}}$ to adjust $\mu_\theta$, as shown in Fig. 3 (b). As the loss gradient is computed based on the source label, it can enable the generated content to correspond to the label class. We follow the previous gradient guidance method [11] to formulate the adjustment operation as:

$$\mu_\theta(x_k^\phi, k, c) = \mu_\theta(x_k^\phi, k) + \lambda \Sigma_\theta \nabla x_k^\phi L_{\text{local}}(x_k^\phi, y, c),$$

where $\mu_\theta$ is the original output of the pre-trained diffusion model, $\Sigma_\theta$ is a fixed value already defined in Eq. 3 and $\lambda$ is a hyper-parameter which controls the guidance scale.

Next, we adopt the adjusted mean value $\mu_\theta$ to sample the denoised image:

$$x_{k-1,c}^\phi \sim \mathcal{N}(\mu_\theta(x_k^\phi, k, c), \Sigma_\theta).$$

With the above process, the guided denoised image $x_{k-1,c}^\phi$ can preserve the semantics for pixels in the $c$-th class region. We parallelly perform the above operations for different $c$, and combine different guided regions into a complete image $x_{k-1}^\phi$:

$$x_{k-1}^\phi = \sum_c m_c \odot x_{k-1,c}^\phi,$$

where $x_{k-1}^\phi$ is the output of the LCG module.

(2) Global Scene-harmonious Guidance (GSG)

The LCG is performed locally, achieving a per-category guidance for image translation. As a complement, we also propose a global guidance module to enhance the global harmony of the translated image, which further boosts the translation quality.

As shown in Fig. 3 (c), our proposed GSG also follows the similar operation flow as LCG, i.e., “computing loss gradient $\rightarrow$ adjusting mean value $\rightarrow$ sampling new image”. Given $x_k^\phi (k \in \{n-1, n-3, n-5, \ldots\})$ as the input of the GSG module, the loss function is computed based on the whole image to achieve the global guidance:

$$L_{\text{global}}(x_k^\phi, y) = L_{\text{ce}}(g(x_k^\phi), y).$$

In SGG, we apply the LCG or GSG at each reverse (denoising) step of the baseline model. By doing so, we can gradually apply the guidance in a step-wise manner, which is suitable for the step-wise translation process of the baseline model. To achieve both local-precise and global-harmony guidance, we incorporate an equal number of LCG and GSG modules into our approach. Specifically, since the baseline model has $n$ reverse steps, we use $\frac{n}{2}$ LCG modules.
and \( \frac{n}{2} \) GSG modules. As for the arrangement of LCG and GSG modules, we design several plausible options: (1) first using \( \frac{1}{2} \) LCG modules and then \( \frac{1}{2} \) GSG modules, (2) first using \( \frac{1}{2} \) GSG modules and then \( \frac{1}{2} \) LCG modules, (3) alternating between LCG and GSG, and (4) randomly mixing the two types of modules. We observe that all options perform at the same level. Since the alternating option shows a slightly better performance than others, we adopt the alternating scheme for SGG.

4.3. Progressive Translation Learning (PTL)

The proposed SGG method is based on the assumption that we already have a segmentation model trained over domains. That is, if we want to guide the translation towards the target domain, we need to have a segmentation model that is pre-trained on the target domain. However, in DASS, we only have labeled images in the source domain for training a source-domain segmentation network. To enable the SGG to reliably work for translation towards the target domain with the source-domain segmentation model, we propose a Progressive Translation Learning (PTL) strategy.

As mentioned before, our baseline model can directly translate images by first diffusing for \( n \) steps and then reversing for \( n \) steps (see Fig. 2). We formulate this process as: \( \mathcal{X}^n = \phi^n(\mathcal{X}^S) \), where \( \mathcal{X}^S \) denotes all source-domain images, \( \phi^n \) denotes the baseline model that handles \( n \)-step diffusion and reverse process, \( \mathcal{X}^n \) denotes the translated output images which is related to the parameter \( n \). As the diffusion model can denoise the image towards the target domain, the parameter \( n \) essentially controls the amount of target-domain information added to the source image. Inspired by [15], we can flexibly change the domain of translated images by varying the value of \( n \). When \( n \) is small, we can obtain translated images \( \mathcal{X}^n \) with minor differences to source-domain images \( \mathcal{X}^S \). Conversely, when \( n \) is large enough, the translated images \( \mathcal{X}^n \) can completely approach the target domain. By walking through \( n \) from 1 to \( N \), our baseline model can generate a set of image domains \( \{ \mathcal{X}^n | n = 1, 2, ..., N \} \), where \( \mathcal{X}^N \) approximately follows the target-domain distribution. Since each step of diffusion/reverse operation only adds/removes a very small amount of noise, the translated images from neighboring domains, i.e., \( \mathcal{X}^n \) and \( \mathcal{X}^{n+1} \), show very slight differences.

Specifically, we start with the source-domain segmentation model \( g_0 \), which is well-trained on source images \( \mathcal{X}^S \) with source labels \( \mathcal{Y}^S \). As the domain gap between two adjacent domains is very small, the segmentation network trained on source domain \( \mathcal{X}^S \) can also work well for the adjacent domain \( \mathcal{X}^1 \). Based on this property, we can use \( g_0 \) to carry out SGG guidance for image translation of the next domain \( \mathcal{X}^1 \). We formulate this operation in a general representation, i.e., given \( g_n \), which is well-trained on \( \mathcal{X}^n \), we can guide the translation of \( \mathcal{X}^{n+1} \) as follows:

\[
\mathcal{X}^{n+1} = SGG(\phi^{n+1}(\mathcal{X}^S), g_n),
\]

where \( SGG(\phi^{n+1}, g_n) \) denotes the SGG-guided baseline model with \( (n + 1) \)-step diffusion and reverse process, which is equipped with \( g_n \).

Under the \( g_0 \)-based SGG guidance, the translated images \( \mathcal{X}^1 \) have fine-grained semantic consistency with source labels \( \mathcal{Y}^S \). Then, we can use the guided images \( \mathcal{X}^1 \) to fine-tune the segmentation model \( g_0 \), making it further fit with this domain. This operation is also formulated in a general representation, i.e., given the guided images \( \mathcal{X}^{n+1} \), we can fine-tune the segmentation model \( g_n \) as follows:

\[
L_{ft} = L_{ce}(g_n(\mathcal{X}^{n+1}), \mathcal{Y}^S) + L_{ce}(g_n(\mathcal{X}^S), \mathcal{Y}^S).
\]

We name the fine-tuned segmentation model as \( g_{n+1} \). The former loss component aims to use the guided images \( \mathcal{X}^{n+1} \) to adapt the segmentation model to the new domain. The latter component is to train the segmentation model on the source domain images \( \mathcal{X}^S \). We empirically observe combining both that augments the model’s learning scope leads to slightly better results than using the former loss only.

After fine-tuning the segmentation model \( g_0 \) on \( \mathcal{X}^1 \), we can perform SGG-guided image translation again using the fine-tuned model (namely \( g_1 \)), obtaining guided images \( \mathcal{X}^{n+1} \). Then we can use \( \mathcal{X}^{n+1} \) to further fine-tune the segmentation model \( g_1 \) for adapting to a new domain. In this manner, we can perform Eq. 11 and Eq. 12 iteratively with \( n \) increasing from 1 to \( N \), finally obtaining the translated target-domain images \( \mathcal{X}^N \) and target-fine-tuned segmentation model \( g_N \). We use the segmentation model \( g_N \) for test-time inference. Throughout the whole process of PTL, the segmentation model always performs SGG guidance at the adjacent domain with little domain discrepancy, which is the key to enabling the SGG to work across a large domain gap.

4.4. Training and Inference

Training. First, we train a standard diffusion model with the unlabeled target-domain images \( \mathcal{X}^T \) via Eq. 5. After that, we keep the pre-trained diffusion model frozen and use it to denoise the source-diffused image, which forms our basic image translation model, namely baseline model \( \phi^n \) (Sec. 4.4). Our baseline model \( \phi^n \) can translate source images \( \mathcal{X}^S \) into the images \( \mathcal{X}^n \). The model \( \phi^n \) is flexible with a variable parameter \( n \), which can control the domain of the translated images \( \mathcal{X}^n \). When \( n = 1 \), the translated images are extremely close to the source domain. When \( n = N \), the translated images approximately approach the target domain. We use the source-domain labeled data \( \{ \mathcal{X}^S, \mathcal{Y}^S \} \) to train a segmentation model \( g_0 \). Then, we set \( n = 1 \) for the baseline model \( \phi^n \) and use it to translate source images \( \mathcal{X}^S \) into the adjacent-domain images \( \mathcal{X}^1 \). During the
translation process, we use the segmentation model $g_0$ to carry out SGG guidance (Eq.11), obtaining the guided images $X^1$. Then, we use the guided images and source labels $\{X^1,Y^{S}\}$ to fine-tune the segmentation model $g_1$ (Eq. 12), obtaining fine-tuned segmentation model $g_1$. We set $n = 2$ for the baseline model $\phi^n$, and use $g_1$ to carry out SGG guidance, obtaining the guided images $X^2$. Next, we can use $\{X^2,Y^{S}\}$ to execute the fine-tuning again. In this way, we alternate the SGG guidance and the fine-tuning of segmentation model with $n$ growing from 1 to $N$ (i.e., PTL strategy). Finally, we obtain the translated target-domain images $X^N$ and the target-fine-tuned segmentation model $g_N$. Inference. After training, we directly use the segmentation model $g_N$ for inference.

5. Experiment

5.1. Implementation Details

Datasets. As a common practice in DASS, we evaluate our framework on two standard benchmarks (GTA5 [57] \rightarrow \text{Cityscapes} [10], and SYNTHIA [60] \rightarrow \text{Cityscapes} [10]). GTA5 and SYNTHIA provide 24,996 and 9400 labeled images, respectively. Cityscapes consists of 2975 and 500 images for training and validation respectively. Following the standard protocol [24], we report the mean intersection over Union (mIoU) on 19 classes for GTA5 \rightarrow \text{Cityscapes} and 16 classes for Synthia \rightarrow \text{Cityscapes}.

Network Architecture. To make a fair and comprehensive comparison, we test three typical types of networks as the segmentation model $g$: (1) DeepLab-V2 [3] architecture with ResNet-101 [21] backbone; (2) FCN-8s [47] architecture with VGG16 [63] backbone; (3) DAFormer [25] architecture with SegFormer [75] backbone. All of them are initialized with the network pre-trained on ImageNet [36]. For the diffusion model, we follow the previous work [22] to adopt a U-Net [59] architecture with Wide-ResNet [80] backbone. The diffusion model is trained from scratch without loading any pre-trained parameters.

Parameter Setting. Following DDPM [22], we train the diffusion model with a batch size of 4, using the Adam [35] as the optimizer with learning rate as $2 \times 10^{-5}$ and momentum as 0.9 and 0.99. Specifically, for the hyper-parameters of diffusion model, we still follow [22] to set $\beta_t$ from $\beta_1 = 10^{-4}$ to $\beta_T = 0.02$ linearly and $T = 1000$. The covariance $\Sigma_0$ is fixed and defined as $\Sigma_0 = \beta_1 I$. We train the segmentation model $g$ by using the SGD optimizer [32] with a batch size of 2, a learning rate of $2.5 \times 10^{-4}$ and a momentum of 0.9. In SGG, we set $\lambda = 80$. In PTL, we set $N = 600$. We pre-train the segmentation model on the source domain for 20,000 iterations and fine-tune it on other intermediate domains for 300 iterations each. Since the discrepancy between adjacent domains is minor, fine-tuning for 300 iterations is enough to effectively adapt to a new domain. As the SGG guidance needs the segmentation model $g$ to classify for noisy and masked images, during the training of the segmentation model, we also generate noisy and masked images for data augmentation to train a robust segmentation model.

5.2. Comparative Studies

We compared the results of our method with the state-of-the-art methods on two DASS tasks, i.e., “GTA5 \rightarrow \text{Cityscapes}” and “SYNTHIA \rightarrow \text{Cityscapes}”, using three different backbone networks: ResNet-101 [21], VGG16 [63] and SegFormer [75]. To ensure reliability, we run our model five times and report the average accuracy. The comparative results are reported in Tab. 1. Overall, our method consistently gains the best performance across all settings and backbones. Remarkably, when compared to existing image translation methods (marked with *), our method brings 3.2% ~ 20.1% improvement on all backbones and settings, which demonstrates the superiority of our diffusion-based approach.

5.3. Ablation Studies

Effect of Baseline model. In Sec. 4.1, we propose a Diffusion-based Image Translation Baseline, i.e, the baseline model $\phi^n$. To study the effectiveness of the baseline model $\phi^n$, we set $n = N$ to directly translate source images

<table>
<thead>
<tr>
<th>Method</th>
<th>Venue</th>
<th>GTA5 \rightarrow Cityscapes</th>
<th>SYN \rightarrow Cityscapes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\text{Res}$</td>
<td>$\text{Vgg}$</td>
</tr>
<tr>
<td>Source only</td>
<td>-</td>
<td>37.3</td>
<td>27.1</td>
</tr>
<tr>
<td>CyCADA [23]</td>
<td>ICLM’18</td>
<td>42.7</td>
<td>35.4</td>
</tr>
<tr>
<td>IIT [54]</td>
<td>CVPR’18</td>
<td>35.7</td>
<td>-</td>
</tr>
<tr>
<td>GdCoSe [6]</td>
<td>CVPR’19</td>
<td>38.1</td>
<td>-</td>
</tr>
<tr>
<td>BDL [44]</td>
<td>CVPR’19</td>
<td>48.5</td>
<td>41.3</td>
</tr>
<tr>
<td>CLAN [49]</td>
<td>CVPR’19</td>
<td>43.2</td>
<td>36.6</td>
</tr>
<tr>
<td>Intra [56]</td>
<td>CVPR’20</td>
<td>45.3</td>
<td>34.1</td>
</tr>
<tr>
<td>SUIT [41]</td>
<td>PR’20</td>
<td>45.3</td>
<td>40.6</td>
</tr>
<tr>
<td>LDR [76]</td>
<td>ECCV’20</td>
<td>45.9</td>
<td>43.6</td>
</tr>
<tr>
<td>CCM [60]</td>
<td>ECCV’20</td>
<td>49.9</td>
<td>-</td>
</tr>
<tr>
<td>ProDA [81]</td>
<td>CVPR’21</td>
<td>57.5</td>
<td>-</td>
</tr>
<tr>
<td>CTF [50]</td>
<td>CVPR’21</td>
<td>56.1</td>
<td>-</td>
</tr>
<tr>
<td>CADA [77]</td>
<td>WACV’21</td>
<td>49.2</td>
<td>44.9</td>
</tr>
<tr>
<td>UPLR [71]</td>
<td>ICCV’21</td>
<td>52.6</td>
<td>-</td>
</tr>
<tr>
<td>DPL [8]</td>
<td>ICCV’21</td>
<td>53.3</td>
<td>46.5</td>
</tr>
<tr>
<td>BAPA [46]</td>
<td>ICCV’21</td>
<td>57.4</td>
<td>-</td>
</tr>
<tr>
<td>CIR [16]</td>
<td>AAA’21</td>
<td>49.1</td>
<td>-</td>
</tr>
<tr>
<td>CRAM [52]</td>
<td>TIT’22</td>
<td>48.6</td>
<td>41.8</td>
</tr>
<tr>
<td>ADAS [39]</td>
<td>CVPR’22</td>
<td>45.8</td>
<td>-</td>
</tr>
<tr>
<td>CPSL [42]</td>
<td>CVPR’22</td>
<td>55.7</td>
<td>-</td>
</tr>
<tr>
<td>FDA [78]</td>
<td>CVPR’22</td>
<td>50.5</td>
<td>42.2</td>
</tr>
<tr>
<td>DAP [30]</td>
<td>CVPR’22</td>
<td>59.8</td>
<td>-</td>
</tr>
<tr>
<td>DaFormer [25]</td>
<td>CVPR’22</td>
<td>68.3</td>
<td>-</td>
</tr>
<tr>
<td>HRDA [26]</td>
<td>ECCV’22</td>
<td>-</td>
<td>73.8</td>
</tr>
<tr>
<td>ProCA [31]</td>
<td>ECCV’22</td>
<td>56.3</td>
<td>-</td>
</tr>
<tr>
<td>Bi-CL [38]</td>
<td>ECCV’22</td>
<td>57.1</td>
<td>-</td>
</tr>
<tr>
<td>Deco-Net [37]</td>
<td>ECCV’22</td>
<td>59.1</td>
<td>-</td>
</tr>
<tr>
<td>Ours</td>
<td>ICCV’23</td>
<td>61.0</td>
<td>48.1</td>
</tr>
</tbody>
</table>

Table 1. Performance comparison in terms of mIoU (%). Three backbone networks, ResNet-101 (Res), VGG16 (Vgg) and SegFormer (Seg), are used in our study. The results of the task “GTA5 \rightarrow \text{Cityscapes}” (“SYNTHIA \rightarrow \text{Cityscapes}”) are averaged over 19 (16) classes. * denotes previous image translation methods.
into the target domain. After translation, we use the translated images and source images, along with the source labels, to train a segmentation model for inference. As shown in Tab. 2 (a), our baseline model achieves 48.3% and 45.1% mIoU on two DASS tasks, respectively. This performance is comparable to the results of previous image translation methods (marked with * in Tab. 1), which demonstrates the effectiveness of our baseline model.

**Effect of SGG.** To demonstrate the effectiveness of SGG, during the PTL training, we remove the SGG guidance and directly use the unguided images of each intermediate domain to fine-tune the segmentation model. By comparing the results in Tab. 2 (b) and (f), we can see that when adopting SGG, our framework shows a significant improvement (more than 6%) on both tasks, clearly demonstrating the effectiveness of SGG in preserving semantics.

**Effect of PTL.** The key design of PTL is to generate intermediate domains to decompose the large gap into small ones. To investigate the effectiveness of PTL, we construct an ablation experiment by removing the generation of intermediate domains, i.e., we set $n = N$ for the baseline model $\phi^N$ and use it to translate source images into the target domain, i.e., $X^N = \phi^N(X^S)$. During the translation, we directly use the source-trained segmentation model to execute SGG guidance to guide the translation process. Finally, we use the guided target-domain images to fine-tune the segmentation model. From Tab. 2 (c) and (f), we can see that our framework without PTL strategy consistently incurs an obvious performance drop on both DASS tasks, which demonstrates the effectiveness of PTL strategy. We show in Fig. 4 some translated image examples and their corresponding intermediate domains. We can see that the image domain is gradually transferred, and all images contain semantics that are well preserved in a fine-grained manner, which demonstrates that our PTL can enable the SGG to reliably work on each intermediate domain.

**Effect of LCG and GSG.** In order to evaluate the impact of the LCG and GSG modules, we conduct ablation experiments by removing each module respectively. As shown in Tab. 2 (d) and (e), we find that the model’s performance without LCG (or GSG) degrades, which demonstrates their usefulness. Furthermore, to comprehensively demonstrate the effect of LCG and GSG modules, we provide some qualitative ablation results in Supplementary.

**Effect of the second loss component of fine-tuning loss in PTL.** As mentioned in Sec. 4.3, in our PTL strategy, we propose a fine-tuning loss $L_{ft}$ to progressively fine-tune the segmentation model towards the target domain, which is formulated as follows:

$$L_{ft} = \frac{1}{\lambda} \left( L_{src}(g_n(X^{n+1}), Y^S) + L_{ce}(g_n(X^S), Y^S) \right). \quad (13)$$

The fine-tuning loss has two components. The former loss component $L_{ada}$ aims to adapt the segmentation model to a new domain, which is closer to the target domain. The later component $L_{src}$ is to train the segmentation model on the source domain. As shown in Tab. 3, compared to using the former loss $L_{ada}$ only, the combination of both components can augment the model’s learning scope, leading to better results (i.e., a gain of 0.4% and 0.3%).

**Parameter Analysis.** For parameters $\beta$, $T$ and $\Sigma_\theta$, we follow the previous work DDPM [22] to set their values. In this paper, we only need to study the impact of $N$ (the num-

---

**Table 2. Ablation studies of different components on the task “GTA5 \(\rightarrow\) Cityscapes” (G\(\rightarrow\)C) and “SYNTHIA \(\rightarrow\) Cityscapes” (S\(\rightarrow\)C) with backbone Res-101.**

<table>
<thead>
<tr>
<th>Model</th>
<th>G(\rightarrow)C</th>
<th>S(\rightarrow)C</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Baseline Model</td>
<td>48.3</td>
<td>45.1</td>
</tr>
<tr>
<td>(b) Ours w/o SGG</td>
<td>55.3</td>
<td>32.7*</td>
</tr>
<tr>
<td>(c) Ours w/o PTL</td>
<td>58.3</td>
<td>56.5</td>
</tr>
<tr>
<td>(d) Ours w/o LCG</td>
<td>59.8</td>
<td>59.1</td>
</tr>
<tr>
<td>(e) Ours w/o GSG</td>
<td>60.5</td>
<td>59.4</td>
</tr>
<tr>
<td>(f) Ours (full)</td>
<td>61.9</td>
<td>61.0</td>
</tr>
</tbody>
</table>

---

**Figure 4. The translated images of our approach (including images from intermediate domains).** $n$ denotes the serial number of intermediate domains. We can see that the semantic knowledge is transferred smoothly and precisely.

**Figure 5. Parameter analysis on $N$ and $\lambda$.**
number of domains in PTL) and \( \lambda \) (the guidance scale parameter in SGG). As for \( N \), we desire the number of domains as large as possible, as we want to smoothly bridge the domain gap. On the other hand, we need to avoid inducing too much training time, i.e., the number of domains should be limited. We present the ablation results of \( N \) in Fig. 5 (blue curve). We find the accuracy begins to level off when \( N > 600 \). We thus set the optimal value of \( N \) to 600 with the consideration of saving training time. As for \( \lambda \), we study its impact with different values and show the results in Fig. 5 (red curve). We can see that the best option of \( \lambda \) is \( \lambda = 80 \).

### Training & Inference Time

We compare our method with other methods in terms of training and inference time on the task “GTA5→Cityscapes”.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Training Time</th>
<th>Inference Time</th>
<th>Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BDL [44]</td>
<td>1.8 days</td>
<td>58.40 ms</td>
<td>48.5</td>
</tr>
<tr>
<td>ProCA [31]</td>
<td>1.9 days</td>
<td>57.48 ms</td>
<td>56.3</td>
</tr>
<tr>
<td>Bi-CL [38]</td>
<td>1.7 days</td>
<td>57.91 ms</td>
<td>57.1</td>
</tr>
<tr>
<td>Deco-Net [37]</td>
<td>1.9 days</td>
<td>59.67 ms</td>
<td>59.1</td>
</tr>
<tr>
<td>Ours</td>
<td>2.3 days</td>
<td>57.68 ms</td>
<td>61.9</td>
</tr>
</tbody>
</table>

### Ablation Study on the Loss Component \( L_{src} \)

<table>
<thead>
<tr>
<th>Model</th>
<th>( G \rightarrow C )</th>
<th>( S \rightarrow C )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) ( L_{src} )</td>
<td>61.5</td>
<td>60.7</td>
</tr>
<tr>
<td>(b) ( L_{src} + L_{rec} )</td>
<td>61.9</td>
<td>61.0</td>
</tr>
</tbody>
</table>

### Ablation Studies on Different Arrangement Options of LCG and GSG Modules in SGG

<table>
<thead>
<tr>
<th>Model</th>
<th>( G \rightarrow C )</th>
<th>( S \rightarrow C )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) LCG/GSG</td>
<td>61.4</td>
<td>60.5</td>
</tr>
<tr>
<td>(b) GSG/LCG</td>
<td>61.3</td>
<td>60.5</td>
</tr>
<tr>
<td>(c) Alternate</td>
<td>61.9</td>
<td>61.0</td>
</tr>
<tr>
<td>(d) RandMix</td>
<td>61.6</td>
<td>60.7</td>
</tr>
</tbody>
</table>

Table 3 shows the ablation results, where “LCG/GSG” denotes first using \( \frac{1}{2} \) LCG modules and then \( \frac{1}{2} \) GSG modules, “GSG/LCG” represents first using \( \frac{1}{2} \) GSG modules and then \( \frac{1}{2} \) LSG modules, “Alternate” refers to alternating between LCG and GSG, and “RandMix” represents randomly arranging the two types of modules. We can see that the alternating option performs best, outperforming others by 0.3%~0.6%. Therefore, we adopt the alternating option to arrange LCG and GSG modules in the SGG scheme.

### Qualitative Segmentation Results

Fig. 6 shows some qualitative segmentation results. The “Source Only” results were obtained by directly applying the segmentation model trained on the source domain to the target domain. We can see that even only using our baseline model, the segmentation results are improved obviously (e.g., the road and car). After adding the SGG scheme to our baseline, the boundary of small objects (e.g., the person) becomes more precise, which demonstrates the SGG’s effectiveness in preserving details. When further using the PTL strategy, our approach can provide more accurate segmentation results, indicating that the PTL can help SGG to preserve details better.

### 6. Conclusion

In this paper, we have proposed a novel label-guided image translation framework to tackle DASS. Our approach leverages the diffusion model and incorporates a Semantic Gradient Guidance (SGG) scheme to guide the image translation based on source labels. We have also proposed a Progressive Translation Learning (PTL) strategy to facilitate our SGG in working across a large domain gap. Extensive experiments have shown the effectiveness of our method.

### Acknowledgement

This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-PhD-2022-01-027).
References


of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 9924–9935. 2022. 7


[50] Haoyu Ma, Xiangru Lin, Zifeng Wu, and Yizhou Yu. Coarse-to-fine domain adaptive semantic segmentation with photo-


