TSIT: A Simple and Versatile Framework for Image-to-Image Translation

Liming Jiang[1], Changxu Zhang[2], Mingyang Huang[3], Chunxiao Liu[3], Jianping Shi[3], and Chen Change Loy[1]✉

[1] Nanyang Technological University
[2] University of California, Berkeley
[3] SenseTime Research

liming002@ntu.edu.sg zhangcx@berkeley.edu
{huangmingyang,liuchunxiao,shijianping}@sensetime.com ccloy@ntu.edu.sg

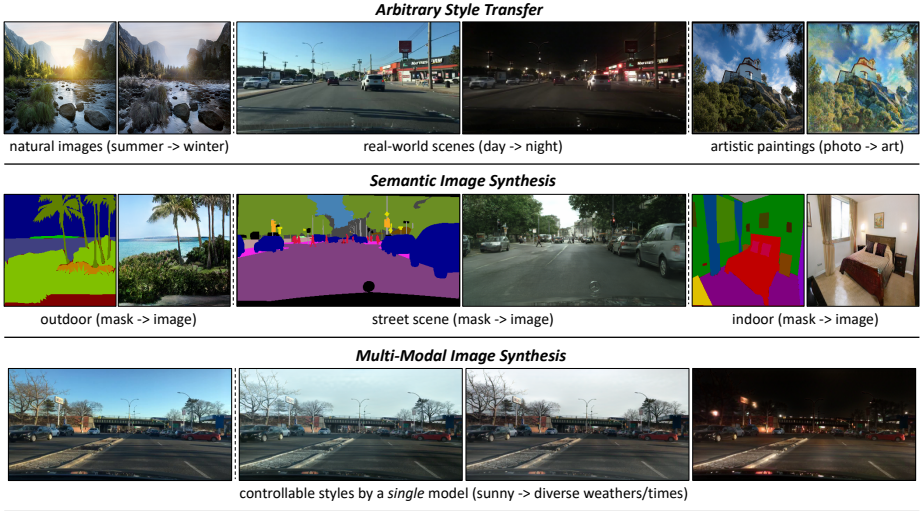

Fig. 1. Our framework is simple and versatile for various image-to-image translation tasks. For unsupervised arbitrary style transfer, diverse scenarios (*e.g.*, natural images, real-world scenes, artistic paintings) can be handled. For supervised semantic image synthesis, our method is robust to different scenes (*e.g.*, outdoor, street scene, indoor). Multi-modal image synthesis is feasible by a *single* model with controllable styles.

Abstract. We introduce a simple and versatile framework for image-to-image translation. We unearth the importance of normalization layers, and provide a carefully designed two-stream generative model with newly proposed feature transformations in a coarse-to-fine fashion. This allows multi-scale semantic structure information and style representation to be effectively captured and fused by the network, permitting our method to scale to various tasks in both unsupervised and supervised settings. No additional constraints (*e.g.*, cycle consistency) are needed, contributing to a very clean and simple method. Multi-modal image synthesis with arbitrary style control is made possible. A systematic study compares the proposed method with several state-of-the-art task-specific baselines, verifying its effectiveness in both perceptual quality and quantitative evaluations. GitHub: https://github.com/EndlessSora/TSIT.

1 Introduction

Image-to-image translation [16] aims at translating one image representation to another. Recent advances [10, 31, 22, 23, 33], especially Generative Adversarial Networks (GANs) [10], have made remarkable success in various image-to-image translation tasks. Previous studies usually present specialized solutions for a specific form of application, ranging from arbitrary style transfer [54, 45, 13, 28, 14, 25, 50] in the unsupervised setting, to semantic image synthesis [16, 4, 35, 42, 34, 29] in the supervised setting.

In this study, we are interested in devising a general and unified framework that is applicable to different image-to-image translation tasks without degradation in synthesis quality. This is *non-trivial* given the different natures of different tasks. For instance, in certain conditional image synthesis tasks (*e.g.*, arbitrary style transfer), paired data are usually not available. Under this unsupervised setting, translation task demands additional constraints on cycle consistency [54, 45, 20, 28], semantic features [40], pixel gradients [1], or pixel values [37]. In semantic image synthesis (*i.e.*, translation from segmentation labels to images), training pairs are available. This task is more data-dependent and typically needs losses to minimize per-pixel distance between the generated sample and ground truth. In addition, specialized structures [4, 42, 34, 29] are required to maintain spatial coherence and resolution. Due to the different needs, existing methods exploit their own specially designed components. It is difficult to cross-use these components or integrate them into a unified framework.

To address the aforementioned challenges, we propose a Two-Stream Image-to-image Translation (TSIT) framework, which is *versatile* for various image-to-image translation tasks (see Fig. 1). The framework is simple as it is based purely on feature transformation. Unlike previous approaches [34, 13] that only consider either semantic structure or style representation, we factorize *both* the structure and style in multi-scale *feature levels* via a symmetrical *two-stream* network. The two streams jointly influence the new image generation in a coarse-to-fine manner via a consistent feature transformation scheme. Specifically, the content spatial structure is preserved by an element-wise feature adaptive denormalization (FADE) from the content stream, while the style information is exerted by feature adaptive instance normalization (FAdaIN) from the style stream. Standard loss functions such as adversarial loss and perceptual loss are used, without additional constraints like cycle consistency. The pipeline is applicable to both unsupervised and supervised settings, easing the preparation of data.

The **contributions** of our work are summarized as follows. We propose TSIT, a simple and versatile framework, which is effective for various image-to-image translation tasks. Despite the succinct design, our network is readily adaptable to various tasks and achieves compelling results. The good performance is achieved by 1) *multi-scale* feature normalization (FADE and FAdaIN) scheme that captures *coarse-to-fine* structure and style information, and 2) a *two-stream* network design that integrates *both* content and style effectively, reducing artifacts and making multi-modal image synthesis possible (see Fig. 1). In comparison to several state-of-the-art task-specific baselines [14, 50, 4, 35, 42, 34,

29], our method achieves comparable or even better results in both perceptual quality and quantitative evaluations.

2 Related Work

Image-to-image translation. Existing methods can be classified into two categories: unsupervised and supervised. With only unpaired data, unsupervised image-to-image translation problem is inherently ill-posed. Additional constraints are needed on *e.g.*, cycle consistency [54, 45, 20, 28], semantic features [40], pixel gradients [1], or pixel values [37]. In contrast, supervised methods, such as `pix2pix` [16], are more data-dependent, requiring well-annotated paired training samples. Subsequent approaches [4, 35, 42, 34, 29] extend the supervised problem for generating high-resolution images or keeping effective semantic meaning.

Limited by learning only one-to-one mapping between two domains, some of the GAN-based methods [54, 45, 20, 28] suffer from generating images with low diversity. Recent studies explore more deeply into both multi-domain translation [6, 27] and multi-modal translation [14, 25, 49], significantly increasing generation diversity. MUNIT [14] is a representative method that disentangles the domain-invariant content and the domain-specific style representation, enriching the synthesized images. Multi-mapping translation is defined in a very recent work, DMIT [50], which is designed to capture the multi-modal image nature in each domain.

Existing image-to-image translation methods lack the scalability to adapt to different tasks under diverse difficult settings. Different demands of unsupervised and supervised settings oblige previous methods to exploit customized modules. Cross-using these components will be suboptimal due to either degradation in quality or introduction of additional constraints. It is non-trivial to integrate them into a single framework and improve robustness. In this study, we design a two-stream network with newly proposed feature transformations inspired by [34] and [13]. Our method is succinct yet able to link various tasks.

Arbitrary style transfer. Style transfer is closely relevant to image-to-image translation in the unsupervised setting. Style transfer aims at retaining the content structure of an image, while manipulating its style representation adopted from other images. Classical methods [9, 18, 3, 8] gradually improve this task from optimization-based to real-time, allowing multiple style transfer during inference. Huang *et al.* introduce AdaIN [13], an effective normalization strategy for arbitrary style transfer. Several studies [46, 52, 44, 5, 24, 30, 39] improve stylization via wavelet transforms [46], graph cuts [52], or iterative error-correction [39]. Besides, most collection-guided [14] style transfer methods are GAN-based [54, 45, 28, 14, 25, 50], showing impressive results.

Previous works usually consider either content or style information. In contrast, our framework succeeds in seeking a balance between content and style, and adaptively fuses them well. The proposed method achieves user-controllable multi-modal style manipulation by only a *single* model. Compared to customized

style transfer methods, our approach achieves better synthesis quality in many scenarios including natural images, real-world scenes, and artistic paintings.

Semantic image synthesis. We define semantic image synthesis as in [34], aiming at synthesizing a photorealistic image from a semantic segmentation mask. Semantic image synthesis is a special form of supervised image-to-image translation. The domain gap of this task is large. Therefore, keeping effective semantic information to enhance fidelity without losing diversity is challenging.

`Pix2pix` [16] first adopts conditional GAN [31] in the semantic image synthesis task. Pix2pixHD [42] contains a multi-scale generator and multi-scale discriminators to generate high-resolution images. SPADE [34] takes a noise map as input, and resizes the semantic label map for modulating the activations in normalization layers by a learned affine transformation. CC-FPSE [29] employs a weight prediction network for generator. A semantics-embedding discriminator is used to enhance fine details and semantic alignments between the generated samples and the input semantic layouts. In addition to these GAN-based methods, CRN [4] applies a cascaded refinement network with regression loss as the supervision. SIMS [35] is a semi-parametric method, retrieving fragments from a memory bank and refining the canvas by a refinement network.

Different from prior works, we design a symmetrical two-stream framework. The network learns feature-level semantic structure information and style representation instead of directly resizing the input mask like SPADE [34]. Coarse-to-fine feature representations are learned by neural networks, adaptively keeping high fidelity without diminishing diversity.

3 Methodology

We consider three key requirements in formulating a robust and scalable method to link various tasks: 1) *Both* semantic structure information and style representation should be considered and fused adaptively. 2) The content and style information should be learned by networks in *feature level* instead of in image level to fit the nature of diverse semantic tasks. 3) The network structure and loss functions should be *simple* for easy training without additional constraints.

Based on the aforementioned considerations, we design a Two-Stream Image-to-image Translation (TSIT) framework (see Fig. 2). We will detail our method in this section, including the network structure (Sec. 3.1), the feature transformation scheme (Sec. 3.2), and the objective functions (Sec. 3.3).

3.1 Network Structure

As illustrated in Fig. 2, TSIT consists of four components: content stream, style stream, generator, and discriminators (omitted in Fig. 2). The first three main components are fully convolutional and symmetrically designed. The details of the submodules, including content/style residual block, FADE residual block,

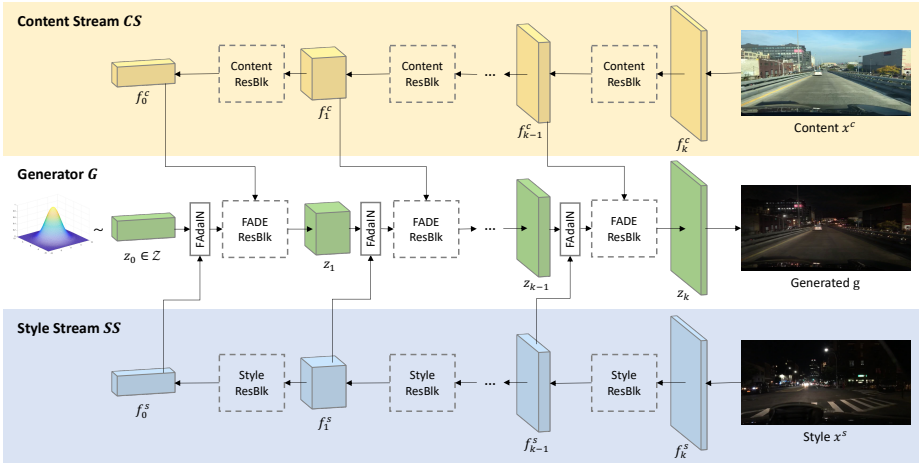

Fig. 2. The proposed Two-Stream Image-to-image Translation (TSIT) framework. The multi-scale patch-based discriminators are omitted. A Gaussian noise map is taken as the latent input for the generator. The feature representations of the content and style images are extracted by the corresponding streams for multi-scale feature transformations. The symmetrical networks fuses semantic structure and style representation in an end-to-end training. Submodules of our network are shown in Fig. 3.

FADE module in the FADE residual block, are as shown in Fig. 3. We will discuss them separately in this section.

Content/style stream. Unlike the traditional conditional GAN [31], we place the two-stream networks, *i.e.*, content stream and style stream, on each side of the generator (see Fig. 2). These two streams are symmetrical with the same network structure, aiming at extracting corresponding feature representations in different levels. We construct content/style stream based on standard residual blocks [11]. We call them content/style residual blocks. As shown in Fig 3 (a), each block has three convolutional layers, one of which is designed for the learned skip connection. The activation function is Leaky ReLU. The function of content/style stream is to extract features and feed them to the corresponding feature transformation layers in the generator. Multi-scale content/style representation in *feature levels* can be learned by content/style stream, adaptively fitting different feature transformations.

Generator. The generator has a completely inverse structure *w.r.t.* the content/style stream. This is intentionally designed to consistently match the level of semantic abstraction at different feature scales. A noise map is sampled from a Gaussian distribution as the latent input, and the feature maps from corresponding layers in content/style stream are taken as multi-scale feature inputs. The proposed feature transformations are implemented by a FADE residual block (Fig. 3 (b)) and a FAdaIN module. In the FADE residual block, we use an inverse architecture *w.r.t.* the content/style residual block and replace the batch

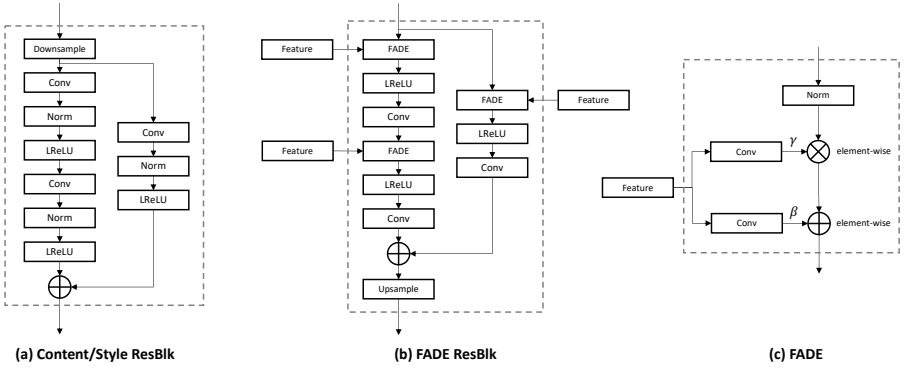

Fig. 3. Submodules of our framework. (a) is a content/style residual block in the symmetrical content/style streams. (b) is a FADE residual block in the generator. (c) is a FADE module in the FADE residual block. It performs *element-wise* denormalization by modulating the normalized activation using a learned affine transformation defined by the modulation parameters γ and β.

normalization [15] layer with the FADE module (Fig. 3 (c)). The FADE module performs *element-wise* denormalization by modulating the normalized activation using a learned affine transformation defined by the modulation parameters γ and β. The FAdaIN module is used to exert style information through feature adaptive instance normalization. More discussions are given in Sec. 3.2.

The entire image generation process is performed in a coarse-to-fine manner. In particular, multi-scale content/style features are injected to refine the generated image constantly from high-level latent code to low-level image representation. Semantic structure and style information are learnable and effectively fused in an end-to-end training.

Discriminators. We exploit the standard multi-scale patch-based discriminators (omitted in Fig. 2) in [42, 34]. Three regular discriminators with an identical architecture are included to discriminate images at different scales. Despite the same structure, patch-based training allows the discriminator operating at the coarsest scale to have the largest receptive field, capturing global information of the image. Whereas the one operating at the finest scale has the smallest receptive field, making the generator produce better details. Multi-scale patch-based discriminators further improve the robustness of our method for image-to-image translation tasks in different resolutions. Besides, the discriminators also serve as feature extractors for the generator to optimize the feature matching loss.

3.2 Feature Transformation

We propose a new feature transformation scheme, considering *both* semantic structure information and style representation, and fusing them adaptively. Let x^c be the content image and x^s be the style image. CS, SS, G, D denote content

stream, style stream, generator, and discriminators, respectively. Sampled from a Gaussian distribution, $z_0 \in \mathbb{Z}$ is a noise map as the latent input for the generator (Fig. 2). Let $z_i \in \{z_0, z_1, z_2, ..., z_k\}$ be the feature map after i-th residual block in the generator, with k denoting the the total number of residual blocks (*i.e.*, the upsampling times in the generator). Let $f_i^c \in \{f_0^c, f_1^c, f_2^c, ..., f_k^c\}$ represent the corresponding feature representations extracted by the content stream (Fig. 2), $f_i^s \in \{f_0^s, f_1^s, f_2^s, ..., f_k^s\}$ with the similar meaning in the style stream.

Feature adaptive denormalization (FADE). Our method is inspired by spatially adaptive denormalization (SPADE) [34]. Different from SPADE that resizes a semantic mask as its input, we generalize the input to multi-scale *feature representation* f_i^c of the content image x^c. In this way, we fully exploit semantic information captured by the content stream CS.

Formally, we define N as the batch size, L_i as the number of feature map channels in each layer. H_i and W_i are height and width, respectively. We first apply batch normalization [15] to normalize the generator feature map z_i in a channel-wise manner. Then, we modulate the normalized feature by using the learned parameters scale γ_i and bias β_i. The denormalized activation ($n \in N$, $l \in L_i$, $h \in H_i$, $w \in W_i$) is:

$$\gamma_i^{l,h,w} \cdot \frac{z_i^{n,l,h,w} - \mu_i^l}{\sigma_i^l} + \beta_i^{l,h,w}, \tag{1}$$

where μ_i^l and σ_i^l are the mean and standard deviation, respectively, of the generator feature map z_i before the batch normalization [15] in channel l:

$$\mu_i^l = \frac{1}{NH_iW_i} \sum_{n,h,w} z_i^{n,l,h,w}, \tag{2}$$

$$\sigma_i^l = \sqrt{\frac{1}{NH_iW_i} \sum_{n,h,w} \left(z_i^{n,l,h,w}\right)^2 - \left(\mu_i^l\right)^2}. \tag{3}$$

The denormalization operation is *element-wise*, and the parameters $\gamma_i^{l,h,w}$ and $\beta_i^{l,h,w}$ are learned by one-layer convolutions from f_i^c in the FADE module (see Fig. 3 (c)). Compared to previous conditional normalization methods [8, 13, 34], FADE experiences more perceptible influence from coarse-to-fine feature representations, thus it can better preserve semantic structure information.

Feature adaptive instance normalization (FAdaIN). To better fuse style representation, we introduce another feature transformation, named feature adaptive instance normalization (FAdaIN). This method is inspired by adaptive instance normalization (AdaIN) [13], with a generalization to enable the style stream SS to learn multi-scale *feature-level* style representation f_i^s of the style image x^s more effectively.

We use the same notation z_i to represent the feature map after i-th FADE residual block in the generator. FAdaIN adaptively computes the affine param-

eters from the corresponding style feature f_i^s with the same scale from SS:

$$\mathrm{FAdaIN}\left(z_i, f_i^s\right) = \sigma\left(f_i^s\right)\left(\frac{z_i - \mu\left(z_i\right)}{\sigma\left(z_i\right)}\right) + \mu\left(f_i^s\right), \tag{4}$$

where $\mu\left(z_i\right)$ and $\sigma\left(z_i\right)$ are the mean and standard deviation, respectively, of z_i.

Exploiting FAdaIN, coarse-to-fine style features at different layers can be fused adaptively with the corresponding semantic structure features learned by FADE, allowing our framework to be trained end-to-end and versatile to different tasks. Furthermore, owing to the effectiveness of FAdaIN in capturing multi-scale style feature representations, multi-modal image synthesis is made possible with arbitrary style control.

3.3 Objective

We use standard losses in our objective function. Following [34, 29], we adopt a hinge loss term [26, 32, 51] as our adversarial loss. For the generator, we apply hinge-based adversarial loss, perceptual loss [18], and feature matching loss [42]. For the multi-scale discriminators, only hinge-based adversarial loss is used to distinguish whether the image is real or fake. The generator and discriminator are trained alternately to play a min-max game. The generator loss $\mathcal{L}_G$ and the discriminator loss $\mathcal{L}_D$ can be written as:

$$\mathcal{L}_G = -\mathbb{E}\left[D\left(g\right)\right] + \lambda_P \mathcal{L}_P\left(g, x^c\right) + \lambda_{FM} \mathcal{L}_{FM}\left(g, x^s\right), \tag{5}$$

$$\mathcal{L}_D = -\mathbb{E}\left[\min\left(-1 + D\left(x^s\right), 0\right)\right] - \mathbb{E}\left[\min\left(-1 - D\left(g\right), 0\right)\right], \tag{6}$$

where $g = G\left(z_0, x^c, x^s\right)$ denotes the generated image, z_0, x^c, x^s denote the input noise map in latent space, the content image, and the style image, respectively. $\mathcal{L}_P$ is the perceptual loss [18] that minimizes the difference between the feature representations extracted by VGG-19 [18] network. $\mathcal{L}_{FM}$ is the feature matching loss [42] that matches the intermediate features at different layers of multi-scale discriminators. λ_P and λ_{FM} are the corresponding weights.

The simple objective functions make our framework stable and easy to train. Thanks to the two-stream network, the typical KL loss [22, 50, 34, 29] for multi-modal image synthesis becomes optional. Despite the simplicity, TSIT is a highly versatile tool, readily adaptable to various image-to-image translation tasks.

4 Experiments

4.1 Settings

Implementation details. We use Adam [21] optimizer and set $\beta_1 = 0$, $\beta_2 = 0.9$. Two time-scale update rule [12] is applied, where the learning rates for the generator (including two streams) and the discriminators are 0.0001 and

0.0004, respectively. We exploit Spectral Norm [32] for all layers in our network. We adopt SyncBN and IN [41] for the generator and the multi-scale discriminators, respectively. For the perceptual loss [18], we use the feature maps of `relu1_1`, `relu2_1`, `relu3_1`, `relu4_1`, `relu5_1` layers from a pretrained VGG-19 [38] model, with the weights [1/32, 1/16, 1/8, 1/4, 1]. For the feature matching loss [42], we select features of three layers from the discriminator at each scale. All the experiments are conducted on NVIDIA Tesla V100 GPUs. Please refer to our *Appendix* for additional implementation details.

Applications. The proposed framework is versatile for various image-to-image translation tasks. We consider three representative applications of conditional image synthesis: arbitrary style transfer (unsupervised), semantic image synthesis (supervised), and multi-modal image synthesis (enriching generation diversity). Please refer to our *Appendix* for details of our application exploration.

Datasets. For arbitrary style transfer, we consider diverse scenarios. We use Yosemite summer → winter dataset (natural images) provided by [54]. We classify BDD100K [48] (real-world scenes) into different times and perform day → night translation. Besides, we use Photo → art dataset (artistic paintings) in [54]. For semantic image synthesis, we select several challenging datasets (*i.e.*, Cityscapes [7] and ADE20K [53]). For multi-modal image synthesis, we further classify BDD100K [48] into different time and weather conditions, and perform controllable time and weather translation. The details of the datasets can be found in the *Appendix*.

Evaluation metrics. Besides comparing perceptual quality, we employ the standard evaluation protocol in prior works [14, 2, 19, 34, 29] for quantitative evaluation. For arbitrary style transfer, we apply Fréchet Inception Distance (FID, evaluating similarity of distribution between the generated images and the real images, lower is better) [12] and Inception Score (IS, considering clarity and diversity, higher is better) [36]. For semantic image synthesis, we strictly follow [34, 29], adopting FID [12] and segmentation accuracy (mean Intersection-over-Union (mIoU) and pixel accuracy (accu)). The segmentation models are: DRN-D-105 [47] for Cityscapes [7], and UperNet101 [43] for ADE20K [53].

Baselines. We compare our method with several state-of-the-art task-specific baselines. For a fair comparison, we mainly employ GAN-based methods. In the unsupervised setting, MUNIT [14] and DMIT [50] are included, with the strong ability to capture the multi-modal nature of images while keeping quality. In the supervised setting, we compare against CRN [4], SIMS [35], pix2pixHD [42], SPADE [34], and CC-FPSE [29].

4.2 Results

Arbitrary style transfer. The results of *Yosemite summer → winter season transfer* are shown in Fig. 4. Baselines [14, 50] tend to impose the color of the style image (winter) to the whole content image (summer). Besides, MUNIT sometimes introduces unnecessary artistic effects, and DMIT generates some

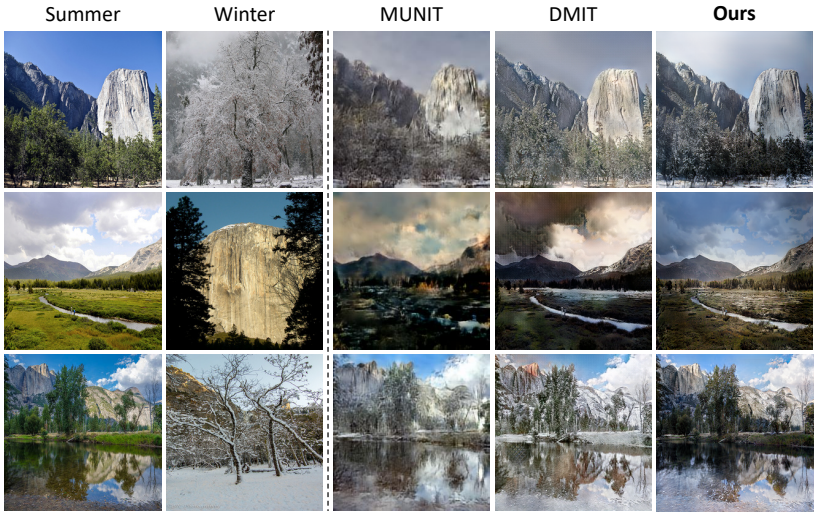

Fig. 4. Yosemite summer → winter season transfer results compared to baselines.

Fig. 5. BDD100K day → night time translation results compared to baselines.

grid-like artifacts. In comparison, our generated results are clearer and more semantics-aware spatially. The results of *BDD100K day → night time translation* are shown in Fig. 5. Some objects (*e.g.*, road sign, car) generated by MUNIT are too dark, and the whole image tends to have some unnatural colors. DMIT introduces obvious artifacts to the car or sky. In contrast, our method produces more photorealistic samples in this task. In *photo → art style transfer*, we choose some hard cases to make a clear comparison (see Fig. 6) due to the very strong ability of all the methods in this task. Our method can transfer the styles well while effectively keeping the content structure. MUNIT tends to impose a homogeneous color to the image. Although DMIT achieves slightly better stylization than our method in certain cases (in Row 3 of Fig. 6), it also brings some grid-like distortions.

The quantitative evaluation results are shown in Table 1. Our approach achieves better performance than baselines [14, 50] in all the tasks. We also note

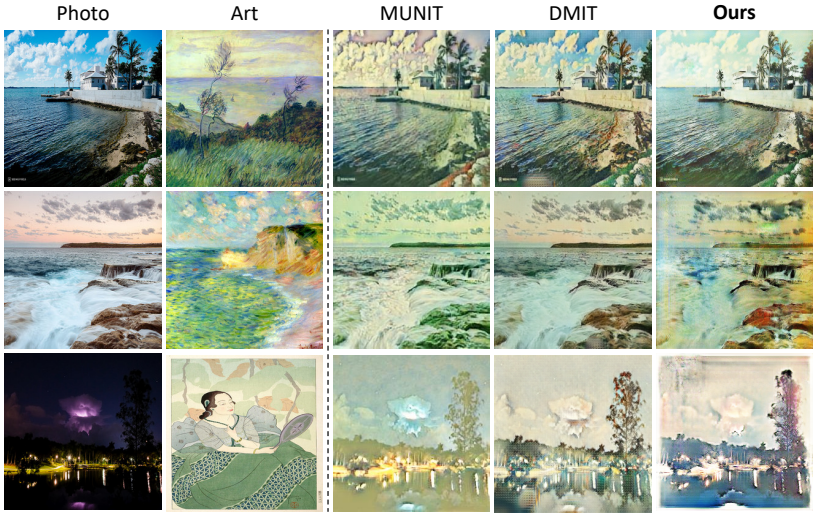

Fig. 6. Photo → art style transfer results compared to baselines.

Table 1. The FID and IS scores of our method compared to state-of-the-art methods in arbitrary style transfer tasks. A lower FID and a higher IS indicate better performance.

	summer → winter		day → night		photo → art	
Methods	FID ↓	IS ↑	FID ↓	IS ↑	FID ↓	IS ↑
MUNIT [14]	118.225	2.537	110.011	2.185	167.314	3.961
DMIT [50]	87.969	2.884	83.898	2.156	166.933	3.871
Ours	**80.138**	**2.996**	**79.697**	**2.203**	**165.561**	**4.020**

that the gap is relatively small in photo → art style transfer, in line with the close qualitative performance in this task (see Fig. 6).

Semantic image synthesis. We choose two state-of-the-art baselines, SPADE [34] and CC-FPSE [29], to show some qualitative comparison results of semantic image synthesis (Fig. 7). Our method demonstrates better perceptual quality than these task-specific baselines. In street scene (Column 1), our method generates better details on key objects (car, pedestrian). In road scene (Column 2), SPADE generates atypical colors on the roads, while CC-FPSE produces unnatural edges on the cars, hardly fitting the background (road). For outdoor natural images (Column 3), all the methods share a similar generation quality. Our method is slightly better due to less distortions on the grass. In indoor scene (Column 4 and 5), SPADE and CC-FPSE produce obvious artifacts in some cases (Column 5). In contrast, our method is more robust to diverse scenarios.

The quantitative evaluation results are shown in Table 2 (the values used for comparison are taken from [34, 29]). The proposed method achieves comparable performance with the very strong specialized methods [4, 35, 42, 34, 29] for semantic image synthesis. Note that SIMS [35] yields the best FID score but poor segmentation performance on Cityscapes, because it stitches image patches from a memory bank of training set while not keeping the exactly consistent position

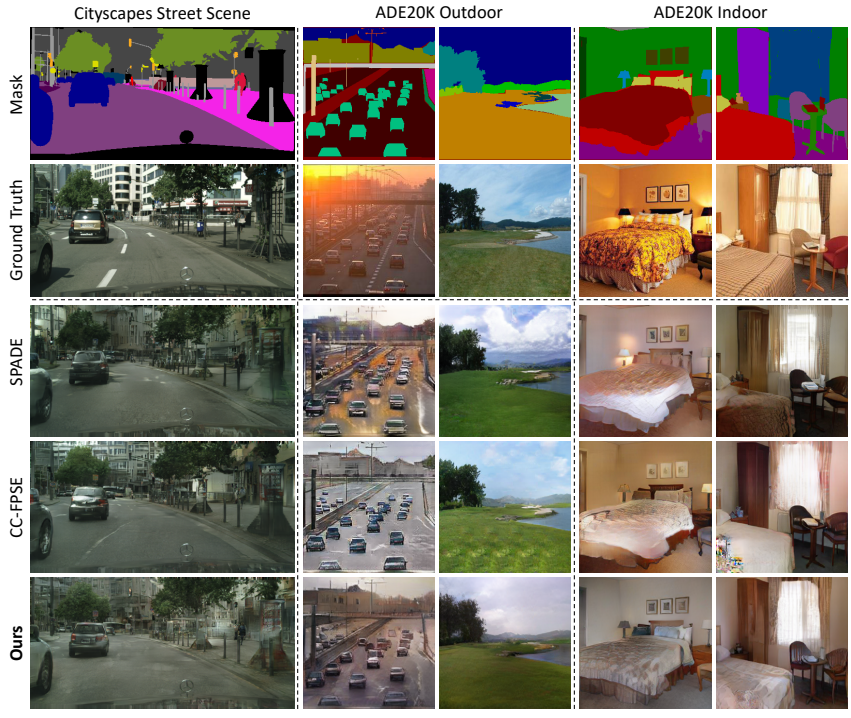

Fig. 7. Semantic image synthesis results compared to baselines.

Table 2. The mIoU, pixel accuracy (accu) and FID scores of our method compared to state-of-the-art methods in semantic image synthesis tasks. A higher mIoU, a higher pixel accuracy (accu) and a lower FID indicate better performance.

	Cityscapes			ADE20K		
Methods	mIoU ↑	accu ↑	FID ↓	mIoU ↑	accu ↑	FID ↓
CRN [4]	52.4	77.1	104.7	22.4	68.8	73.3
SIMS [35]	47.2	75.5	**49.7**	N/A	N/A	N/A
pix2pixHD [42]	58.3	81.4	95.0	20.3	69.2	81.8
SPADE [34]	62.3	81.9	71.8	38.5	79.9	33.9
CC-FPSE [29]	65.5	82.3	54.3	**43.7**	**82.9**	31.7
Ours	**65.9**	**82.7***	59.2	38.6	80.8	**31.6**

in the synthesized image. Our approach achieves state-of-the-art segmentation performance on Cityscapes and the best FID score on ADE20K, suggesting its robustness to fit the nature of different image-to-image translation tasks.

* The value differs from the earlier version of this paper [17]. The official code of DRN [47] does not provide the implementation of the "accu" metric. The new accu value 82.7% (still the best among the compared methods) is obtained by including 255-labeled pixels, consistent with [34, 29]. The previously reported accu 94.4% omits 255-labeled pixels, which may be more reasonable due to its consistency with the training of the segmentation model and the calculation of mIoU.

Fig. 8. BDD100K multi-modal image synthesis results for different time and weather translation by a *single* model.

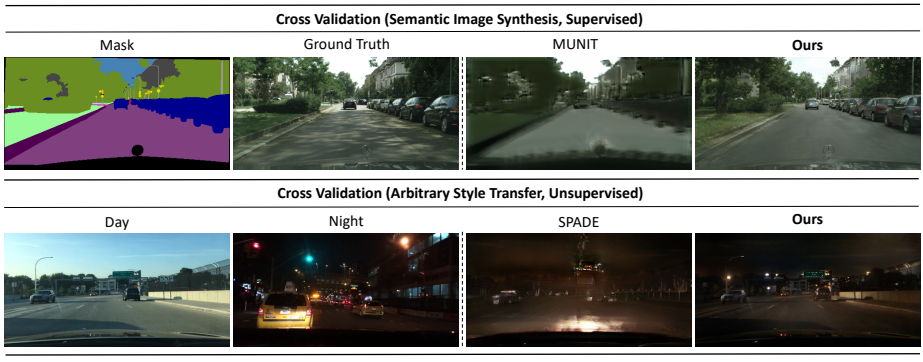

Fig. 9. Cross validation of ineffectiveness of task-specific methods in inverse settings.

Multi-modal image synthesis. We perform multi-modal image synthesis for time and weather image-to-image translation (see Fig. 8) on BDD100K [48]. Training only a *single* model, we translate the images of weather *sunny* to different times and weathers (*i.e.*, *night*, *snowy*, *cloudy*, *rainy*). Our method effectively adapts to different style control and keeps photorealistic generation quality. Although the weather *snowy* is not very obvious in BDD100K [48], our approach successfully introduces some snow-like effects on trees and grounds (Column 2).

Cross validation. We also conduct experiments to evaluate the performance of existing specialized methods in inverse settings (*i.e.*, using unsupervised methods to do semantic image synthesis / using supervised methods to perform arbitrary style transfer). We selected two representative methods, MUNIT [14] and SPADE [34]. Without modifying the architecture, we tuned the loss weights and tried to get the best generation results. To ensure a fair comparison, we also tried to

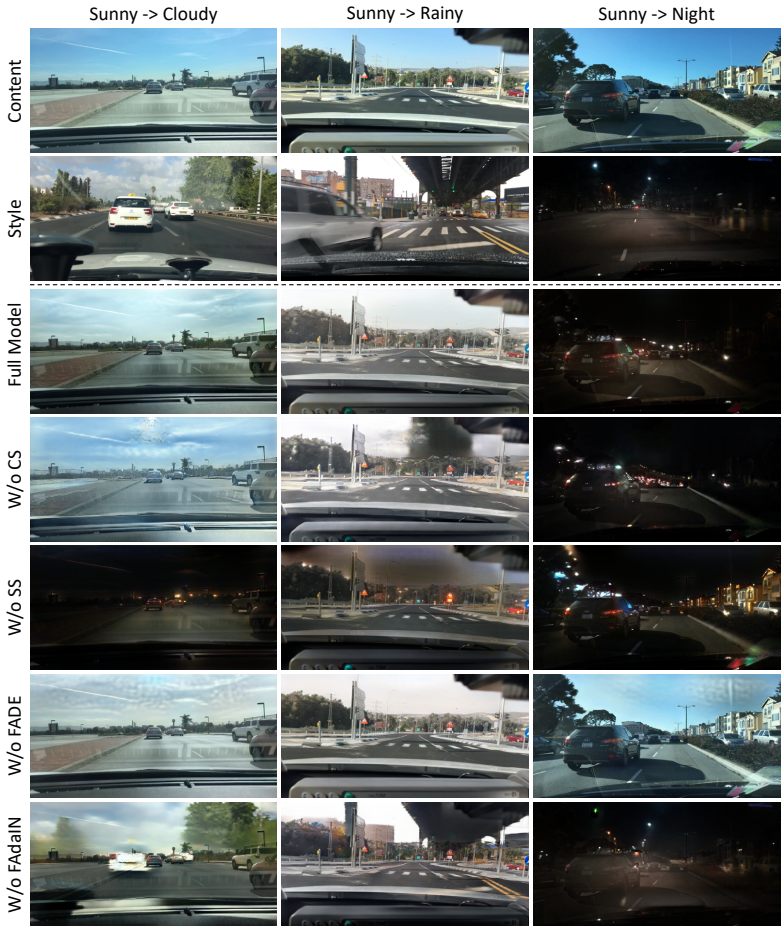

Fig. 10. Ablation studies of key modules (*i.e.*, content stream (CS), style stream(SS)) and feature transformations in multi-modal image synthesis task.

compute perceptual loss with the content (day) image for SPADE to match the setting of TSIT. Representative results of cross validation are shown in Fig. 9. The proposed method shows much better results than baseline methods. MUNIT fails to adapt to semantic image synthesis. SPADE loses details of key objects and introduces very strong artifacts despite translating the color correctly.

Ablation studies. We present ablation studies of key modules (*i.e.*, content stream (CS), style stream(SS)) and the proposed feature transformations (see Fig. 10. More ablation study results can be found in the *Appendix*). We perform multi-modal image synthesis to show the effectiveness of different components. Our full model generates high-quality results (Row 3). When we directly inject the resized content image without CS, the semantic structure information be-

comes weak, leading to several artifacts in the sky (Row 4). Without SS, the model cannot perform multi-modal image synthesis at all (Row 5). The style representation is dominated by the night style. When we concatenate the feature maps of CS with the ones of the generator instead of using FADE, the concatenation introduces too much content information, leading to several failure cases (*e.g.*, *sunny* $\rightarrow$ *night* in Row 6). If we discard FAdaIN by concatenating the feature maps of SS with the ones of the generator, the style becomes too strong, causing serious style regionalization problem (Row 7).

5 Conclusion

We propose TSIT, a simple and versatile framework for image-to-image translation. The proposed symmetrical two-stream network allows the image generation to be effectively conditioned on the multi-scale feature-level semantic structure information and style representation via feature transformations. A systematic study verifies the effectiveness of our method in diverse tasks compared to state-of-the-art task-specific baselines. We believe that designing a unified and versatile framework for more tasks is an important direction in the image generation area. Incorporating unconditional image synthesis tasks and introducing more variability into the two streams/latent space can be interesting future works.

Acknowledgements. This work is supported by the SenseTime-NTU Collaboration Project, Singapore MOE AcRF Tier 1 (2018-T1-002-056), and NTU NAP.

References

1. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised pixel-level domain adaptation with generative adversarial networks. In: CVPR (2017)
2. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: ICLR (2018)
3. Chen, D., Yuan, L., Liao, J., Yu, N., Hua, G.: StyleBank: An explicit representation for neural image style transfer. In: CVPR (2017)
4. Chen, Q., Koltun, V.: Photographic image synthesis with cascaded refinement networks. In: ICCV (2017)
5. Chiu, T.Y.: Understanding generalized whitening and coloring transform for universal style transfer. In: ICCV (2019)
6. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In: CVPR (2018)
7. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
8. Dumoulin, V., Shlens, J., Kudlur, M.: A learned representation for artistic style. arXiv preprint **arXiv:1610.07629** (2016)
9. Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. arXiv preprint **arXiv:1508.06576** (2015)

10. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS (2014)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
12. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: NeurIPS (2017)
13. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: ICCV (2017)
14. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: ECCV (2018)
15. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML (2015)
16. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR (2017)
17. Jiang, L., Zhang, C., Huang, M., Liu, C., Shi, J., Loy, C.C.: TSIT: A simple and versatile framework for image-to-image translation. arXiv preprint **arXiv:2007.12072v1** (2020)
18. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: ECCV (2016)
19. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: CVPR (2019)
20. Kim, T., Cha, M., Kim, H., Lee, J.K., Kim, J.: Learning to discover cross-domain relations with generative adversarial networks. In: ICML (2017)
21. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint **arXiv:1412.6980** (2014)
22. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint **arXiv:1312.6114** (2013)
23. Kingma, D.P., Mohamed, S., Rezende, D.J., Welling, M.: Semi-supervised learning with deep generative models. In: NeurIPS (2014)
24. Kotovenko, D., Sanakoyeu, A., Lang, S., Ommer, B.: Content and style disentanglement for artistic style transfer. In: ICCV (2019)
25. Lee, H.Y., Tseng, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Diverse image-to-image translation via disentangled representations. In: ECCV (2018)
26. Lim, J.H., Ye, J.C.: Geometric GAN. arXiv preprint **arXiv:1705.02894** (2017)
27. Liu, A.H., Liu, Y.C., Yeh, Y.Y., Wang, Y.C.F.: A unified feature disentangler for multi-domain image translation and manipulation. In: NeurIPS (2018)
28. Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. In: NeurIPS (2017)
29. Liu, X., Yin, G., Shao, J., Wang, X., et al.: Learning to predict layout-to-image conditional convolutions for semantic image synthesis. In: NeurIPS (2019)
30. Lu, M., Zhao, H., Yao, A., Chen, Y., Xu, F., Zhang, L.: A closed-form solution to universal style transfer. In: ICCV (2019)
31. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint **arXiv:1411.1784** (2014)
32. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. arXiv preprint **arXiv:1802.05957** (2018)
33. Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al.: Conditional image generation with PixelCNN decoders. In: NeurIPS (2016)
34. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: CVPR (2019)

35. Qi, X., Chen, Q., Jia, J., Koltun, V.: Semi-parametric image synthesis. In: CVPR (2018)
36. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training GANs. In: NeurIPS (2016)
37. Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R.: Learning from simulated and unsupervised images through adversarial training. In: CVPR (2017)
38. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint **arXiv:1409.1556** (2014)
39. Song, C., Wu, Z., Zhou, Y., Gong, M., Huang, H.: ETNet: Error transition network for arbitrary style transfer. In: NeurIPS (2019)
40. Taigman, Y., Polyak, A., Wolf, L.: Unsupervised cross-domain image generation. arXiv preprint **arXiv:1611.02200** (2016)
41. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint **arXiv:1607.08022** (2016)
42. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional GANs. In: CVPR (2018)
43. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: ECCV (2018)
44. Yang, S., Wang, Z., Wang, Z., Xu, N., Liu, J., Guo, Z.: Controllable artistic text style transfer via shape-matching GAN. In: ICCV (2019)
45. Yi, Z., Zhang, H., Tan, P., Gong, M.: DualGAN: Unsupervised dual learning for image-to-image translation. In: ICCV (2017)
46. Yoo, J., Uh, Y., Chun, S., Kang, B., Ha, J.W.: Photorealistic style transfer via wavelet transforms. In: ICCV (2019)
47. Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks. In: CVPR (2017)
48. Yu, F., Xian, W., Chen, Y., Liu, F., Liao, M., Madhavan, V., Darrell, T.: BDD100K: A diverse driving video database with scalable annotation tooling. arXiv preprint **arXiv:1805.04687** (2018)
49. Yu, X., Cai, X., Ying, Z., Li, T., Li, G.: SingleGAN: Image-to-image translation by a single-generator network using multiple generative adversarial learning. In: ACCV (2018)
50. Yu, X., Chen, Y., Liu, S., Li, T., Li, G.: Multi-mapping image-to-image translation via learning disentanglement. In: NeurIPS (2019)
51. Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-attention generative adversarial networks. arXiv preprint **arXiv:1805.08318** (2018)
52. Zhang, Y., Fang, C., Wang, Y., Wang, Z., Lin, Z., Fu, Y., Yang, J.: Multimodal style transfer via graph cuts. In: ICCV (2019)
53. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ADE20K dataset. In: CVPR (2017)
54. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV (2017)

Appendix

This appendix provides the supplementary information that is not elaborated in the main paper: Sec. A details the different applications we have explored. Sec. B describes details of the datasets used in our experiments. Sec. C provides additional implementation details. Sec. D presents several supplementary ablation studies. Sec. E shows more examples of the images generated by our method.

A Application Exploration

We have introduced a Two-Stream Image-to-image Translation (TSIT) framework in the main paper. The proposed framework is simple and versatile for various image-to-image translation tasks under both unsupervised and supervised settings. We have considered three important and representative applications of conditional image synthesis: arbitrary style transfer (unsupervised), semantic image synthesis (supervised), and multi-modal image synthesis (enriching generation diversity). We employ a two-stream network, namely "content" stream and "style" stream, on these applications.

For the unsupervised arbitrary style transfer application, we feed the content image to the content stream and the style image to the style stream, and let the networks learn different levels of *feature representations* of the content and style. The proposed feature transformations, FADE and FAdaIN, adaptively fuse content and style feature maps, respectively, at different scales in the generator. In contrast to prior works, our method is more adaptable to style transfer tasks in diverse scenarios (*e.g.*, natural images, real-world scenes, and artistic paintings).

We further expand the application of our method to cater to semantic image synthesis under the supervised setting. The definition of "content" and "style" can be more general: all the images that provide semantic structure information can be content images, and all the images representing the global style distribution can be considered as style images. Therefore, when we inject semantic segmentation masks to the content stream and the corresponding real images to the style stream, semantic image synthesis task in the supervised setting can be handled. Despite a rather large domain gap in this task, our framework yields comparable or even better results over the state-of-the-art task-specific methods, suggesting the high adaptability of our approach.

It is noteworthy to highlight that the newly proposed feature transformations and the symmetrical two-stream network can effectively disentangle the semantic structure and style information. Thanks to the clean disentanglement, the high-level multi-modal nature of the images can be captured by our framework, contributing to high-fidelity multi-modal image synthesis.

B Dataset Details

In this section, we discuss the detailed information of all the datasets we explored, including the source, preprocessing, number of images, resolution, *etc.*

For arbitrary style transfer under the unsupervised setting, paired data are not needed. We perform style transfer tasks in diverse scenarios (*e.g.*, natural images, real-world scenes, and artistic paintings).

- **Yosemite summer → winter.** We use this unpaired dataset provided by [54], containing rich natural images collected via Flickr API. We perform season transfer using this dataset, with 1,231 summer images and 962 winter images for training. The resolution is 256×256.
- **BDD100K day → night.** We conduct time translation on BDD100K [48] dataset, which is captured at diverse locations in the United States. All the images are in real-world scenes, mostly street/road scenes. We classify the dataset into different times. The training set contains 12,454 daytime images and 22,884 nighttime images. The original images are scaled to 512×256.
- **Photo → art.** We utilize the art dataset collected in [54]. The art images of this dataset were downloaded from Wikiart.org. The dataset consists of photographs and diverse artistic paintings (Monet: 1,074; Cézanne: 584; van Gogh: 401; Ukiyo-e: 1,433). To test the robustness of the models for arbitrary style transfer, we combine all the artistic styles, yielding 6,853 photos and 3,492 paintings for training. All the images are uniformly resized to 256×256.

For semantic image synthesis under the supervised setting, we follow [29, 34] and select several challenging datasets.

- **Cityscapes.** Cityscapes [7] dataset contains street scene images mostly collected in Germany, with 2,975 images for training and 500 images for evaluation. The dataset provides instance-wise, dense pixel annotations of 30 classes. All the image sizes are adjusted to 512×256.
- **ADE20K.** We use ADE20K [53] dataset consisting of challenging in-the-wild images with fine annotations of 150 semantic classes. The sizes of training and validation sets are 20,210 and 2,000, respectively. All the images are scaled to 256×256.

For multi-modal image synthesis, we use BDD100K [48] dataset, details of which have been described earlier.

- **BDD100K sunny → different time/weather conditions**. We further classify the images in BDD100K [48] dataset into different time and weather conditions, constituting a training set of 10,000 sunny images and 10,000 images of other time and weather conditions (night: 2,500; cloudy: 2,500; rainy: 2,500; snowy: 2,500). The resolution is 512×256.

C Additional Implementation Details

We provide more implementation details in this section, including the network architecture specifics, detailed feature shapes, hyperparameters, *etc.*

Network architecture specifics. Our framework consists of four components: content stream, style stream, generator, and discriminators. The first three components maintain a symmetrical structure, using fully convolutional networks.

The number of residual blocks k (*i.e.*, downsampling/upsampling times) in the content/style stream and the generator equals to 7. Let inc, $outc$, kn, s, p denote the input channel, the output channel, the kernel size, the stride, and the zero-padding amount, respectively.

In the content/style stream, we use a series of content/style residual blocks with the nearest neighbor downsampling. The scale factor of downsampling is 2. By default, we use instance normalization [41] for the content/style residual blocks, and the negative slope of Leaky ReLU is 0.2. Thus, the structure of Content/Style ResBlk($inc, outc$) is: Downsample(2)$-$Conv($inc, inc, kn3\times 3, s1, p1$)$-$IN $-$ LReLU(0.2)$-$Conv($inc, outc, kn3 \times 3, s1, p1$)$-$IN $-$ LReLU(0.2) with the learned skip connection Conv($inc, outc, kn1\times1, s1, p0$)$-$IN $-$ LReLU(0.2).

In the generator, we construct several FADE residual blocks with the nearest neighbor upsampling. The scale factor of upsampling is 2. FAdaIN layers are applied before each FADE residual block. The FADE residual block contains a FADE submodule, which performs *element-wise* denormalization using a learned affine transformation defined by the modulation parameters γ and β. Let $normc$, $featc$ indicate the normalized channel and the injected feature channel, respectively. Then, the convolutional layers in FADE($normc, featc$) can be represented as: Conv($featc, normc, kn3 \times 3, s1, p1$). By default, we adopt SyncBN for the generator, and the negative slope of Leaky ReLU is 0.2. The structure of FADE ResBlk($inc, outc$) is: FADE(inc, inc)$-$LReLU(0.2)$-$Conv($inc, inc, kn3 \times 3, s1, p1$)$-$FADE($inc, inc$)$-$LReLU(0.2)$-$Conv($inc, outc, kn3 \times 3, s1, p1$)$-$Upsample(2) with the learned skip connection FADE(inc, inc)$-$LReLU(0.2)$-$Conv($inc, outc, kn1 \times 1, s1, p0$).

As mentioned in the main paper, we exploit the same multi-scale patch-based discriminators as [42, 34]. The detailed network architectures and the layers used for feature matching loss [42] are also identical.

Feature shapes. In the content/style stream, we put an input layer at the entrance. The feature channel is adjusted to 64 after the input layer, while the resolution remains unchanged. Then, the feature channels after each of the $k(7)$ residual blocks are: 128, 256, 512, 1024, 1024, 1024, 1024. Since the scale factor of downsampling is 2 (as described in the network architecture specifics above), the resolution of the features is halved after each residual block. The generator feature shapes are strictly corresponding and opposite to that of content/style stream. The discriminator feature shapes are identical to that in [42, 34], where the resolution is halved on every step of the pyramid.

Additional training details. For perceptual loss, we use the feature reconstruction loss that requires a content target [18].

In the arbitrary style transfer and multi-modal image synthesis tasks, the content target is the content image. The loss weights are $\lambda_P = 1, \lambda_{FM} = 1$, and the batch size is 1. We train our models for 200 epochs on Yosemite summer $\rightarrow$ winter, 10 epochs on BDD100K day $\rightarrow$ night, 40 epochs on Photo $\rightarrow$ art, and 20 epochs on BDD100K sunny $\rightarrow$ different time/weather conditions. The models are trained on 1 NVIDIA Tesla V100 GPU, with around 10 GB memory consumption. For multi-modal image synthesis, similar to [16], at the inference

Table 3. The quantitative evaluation on ablation studies of the key modules (*i.e.*, content stream (CS), style stream (SS)) and the feature transformations in multi-modal image synthesis task. A lower FID and a higher IS indicate better performance.

	multi-modal image synthesis				
Metrics	full model	w/o CS	w/o SS	w/o FADE	w/o FAdaIN
FID ↓	**85.876**	89.429	86.263	86.463	89.795
IS ↑	**2.934**	2.851	2.734	2.881	2.890

phase we run the generator network in exactly the same manner as during the training phase. For the cross validation of SPADE [34], the hyperparameters obtaining the best generation results are $\lambda_P = 10, \lambda_{FM} = 10$.

In the semantic image synthesis task, the content target is the ground truth real image. The corresponding loss weights are $\lambda_P = 20, \lambda_{FM} = 10$, and the batch size is 16. We perform 200 epochs of training on Cityscapes and ADE20K. The models are trained on 2 NVIDIA Tesla V100 GPUs, each with about 32 GB memory consumption. We also find that in semantic image synthesis, weakening/removing the style stream can sometimes contribute to a performance boost. Besides, exploiting variational auto-encoders [22] can help in certain cases. For the cross validation of MUNIT [14], since the loss functions are very different from ours, we use its default hyperparameters in unsupervised image-to-image translation.

D Supplementary Ablation Studies

We ablate the key modules (*i.e.*, content stream (CS), style stream(SS)) and the proposed feature transformations in the main paper. We perform multi-modal image synthesis to clearly show the effectiveness of different components. Due to the space constraints, we only provide qualitative evaluation results. In this section, we will first show the quantitative evaluation results of key component ablation studies in the main paper. Then, we will dig deeper and present more supplementary ablation study results.

Quantitative evaluation of key component ablation studies. We conduct quantitative evaluation on ablation studies of the key components in multi-modal image synthesis task. As shown in Table 3, using the full model we introduced, the lowest FID score and highest IS score have been achieved. This means the generated images by our full model are the most photorealistic, clearest, and of the highest diversity. Without any key module of TSIT, the quantitative performance will drop. This verifies the necessity of these components for our method.

Feature channel ablation studies. We also study how the number of feature channels in the two streams (*i.e.*, content stream (CS) and style stream (SS)) affects the image synthesis results. We conduct quantitative evaluation of feature channel ablation studies, covering all of the discussed tasks. Note that

Table 4. The quantitative evaluation on ablation studies of CS/SS feature channels for unsupervised arbitrary style transfer (day → night). A lower FID and a higher IS indicate better performance.

	arbitrary style transfer (day → night)		
Metrics	full model	channels ÷ 2	channels ÷ 4
FID ↓	**79.697**	82.357	95.199
IS ↑	**2.203**	2.142	2.101

Table 5. The quantitative evaluation on ablation studies of CS/SS feature channels for supervised semantic image synthesis (Cityscapes). A higher mIoU, a higher pixel accuracy (accu) and a lower FID indicate better performance.

	semantic image synthesis (Cityscapes)		
Metrics	full model	channels ÷ 2	channels ÷ 4
mIoU ↑	**65.9**	61.0	56.6
accu ↑	**82.7**	82.1	81.5
FID ↓	**59.2**	71.8	74.4

Table 6. The quantitative evaluation on ablation studies of CS/SS feature channels for multi-modal image synthesis. A lower FID and a higher IS indicate better performance.

	multi-modal image synthesis		
Metrics	full model	channels ÷ 2	channels ÷ 4
FID ↓	**85.876**	93.258	97.297
IS ↑	**2.934**	2.851	2.813

we should change the channels in CS/SS at the same time to maintain a symmetrical structure. As presented in Table 4, Table 5 and Table 6, in different tasks under either unsupervised or supervised setting, the best performance is achieved by the full model of TSIT. As we reduce the channel numbers in the two-stream network, the image synthesis quality gradually degrade. For more channels, memory consumption will increase exponentially.

Feature-level/Image-level injection ablation studies. To verify the importance of the feature-level injection, We further conduct feature-level/image-level injection ablation studies. TSIT performs feature-level injections from the content/style stream to the generator to adapt to diverse tasks. In comparison, the direct injection of resized images (*i.e.*, the direct application of AdaIN in arbitrary style transfer, and SPADE in semantic image synthesis) can be regarded as the image-level injections. We provide quantitative evaluation results under this setting. As shown in Table 7 and Table 8, compared to our feature-level injection scheme, the image-level injection leads to a performance drop. This suggests the significance of feature-level injection in TSIT.

Table 7. The quantitative evaluation on ablation studies of feature-level (FAdaIN)/ image-level (AdaIN) injection for unsupervised arbitrary style transfer (day → night). A lower FID and a higher IS indicate better performance.

	arbitrary style transfer (day → night)	
Metrics	feature-level	image-level
FID ↓	**79.697**	80.618
IS ↑	**2.203**	2.182

Table 8. The quantitative evaluation on ablation studies of feature-level (FADE)/ image-level (SPADE) injection for supervised semantic image synthesis (Cityscapes). A higher mIoU, a higher pixel accuracy (accu) and a lower FID indicate better performance.

	semantic image synthesis (Cityscapes)	
Metrics	feature-level	image-level
mIoU ↑	**65.9**	59.7
accu ↑	**82.7**	81.7
FID ↓	**59.2**	60.1

E More Examples of Generated Images

We show more examples of generated results by our method in Fig. 11 and Fig. 12. Several generated images of arbitrary style transfer, covering diverse scenarios, are presented in Fig. 11. We also show more synthesized exmaples of semantic image synthesis in Fig. 12. These examples feature both outdoor and indoor scenes, generated from the corresponding semantic segmentation label maps. All the images synthesized by our proposed method are very photorealistic.

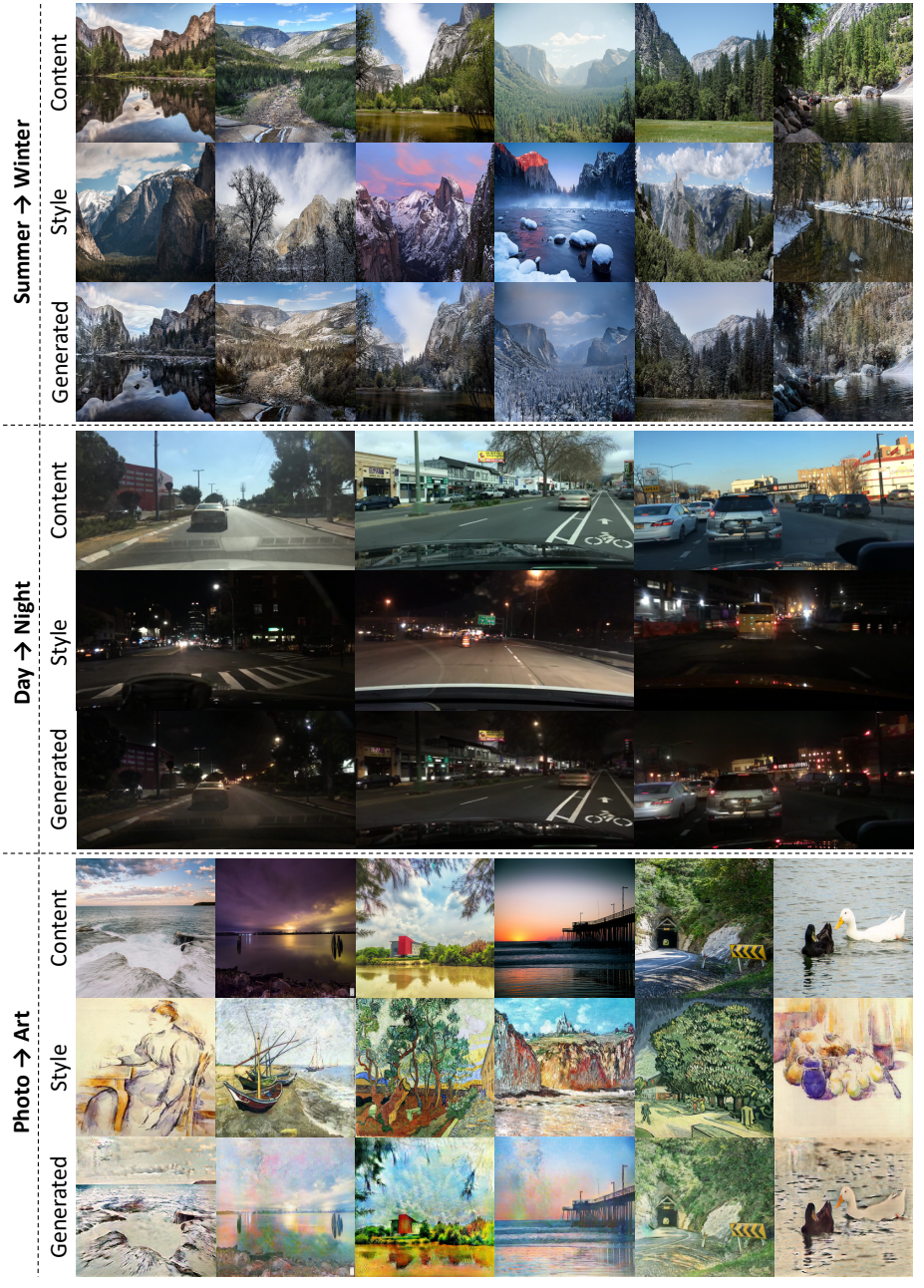

Fig. 11. More examples of images generated by our method in the arbitrary style transfer task (unsupervised). Rows 1-3 show Yosemite summer → winter season transfer results. Rows 4-6 are BDD100K day → night translation results. Rows 7-9 present photo → art style transfer results.

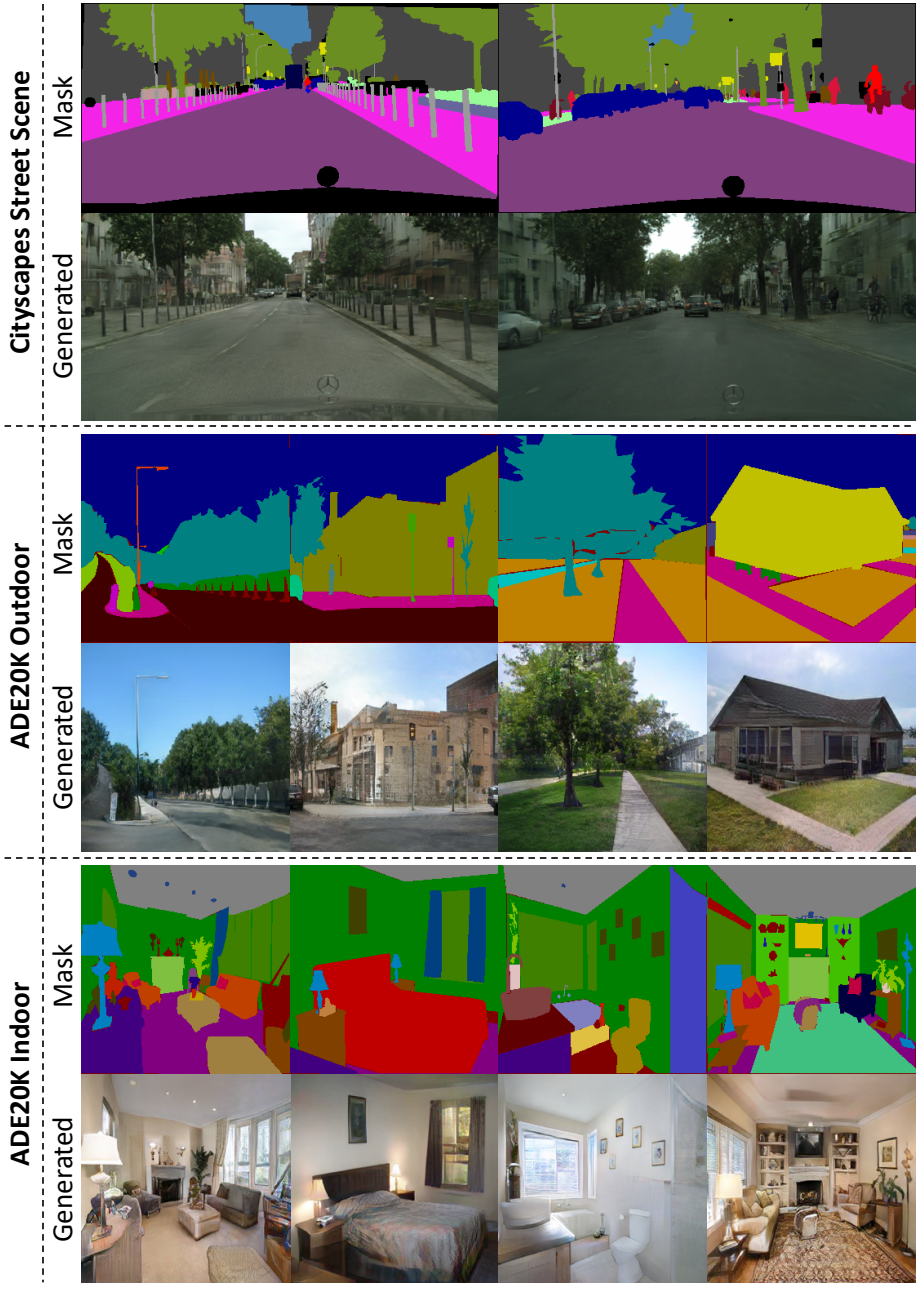

Fig. 12. More examples of images generated by our method in the semantic image synthesis task (supervised). Row 1 and 2 show generated results on Cityscapes dataset. Row 3 and 4 are outdoor synthesized results on ADE20K dataset. Row 5 and 6 present indoor synthesized results on ADE20K dataset.