InstaFormer: Instance-Aware Image-to-Image Translation with Transformer

Soohyun Kim   Jongbeom Baek   Jihye Park   Gyeongnyeon Kim   Seungryong Kim*

Korea University, Seoul, Korea

{shkim1211,baem0911,ghp1112,kkn9975,seungryong_kim}@korea.ac.kr

Abstract

We present a novel Transformer-based network architecture for instance-aware image-to-image translation, dubbed InstaFormer, to effectively integrate global- and instance-level information. By considering extracted content features from an image as tokens, our networks discover global consensus of content features by considering context information through a self-attention module in Transformers. By augmenting such tokens with an instance-level feature extracted from the content feature with respect to bounding box information, our framework is capable of learning an interaction between object instances and the global image, thus boosting the instance-awareness. We replace layer normalization (LayerNorm) in standard Transformers with adaptive instance normalization (AdaIN) to enable a multi-modal translation with style codes. In addition, to improve the instance-awareness and translation quality at object regions, we present an instance-level content contrastive loss defined between input and translated image. We conduct experiments to demonstrate the effectiveness of our InstaFormer over the latest methods and provide extensive ablation studies.

1. Introduction

For a decade, image-to-image translation (I2I), aiming at translating an image in one domain (i.e., source) to another domain (i.e., target), has been popularly studied, to the point of being deployed in numerous applications, such as style transfer [14, 21], super-resolution [11, 30], inpainting [24, 44], or colorization [62, 63].

In particular, most recent works have focused on designing better disentangled representation to learn a multi-modal translation from unpaired training data [22, 33, 43]. While they have demonstrated promising results, most of these methods only consider the translation on an *whole* image, and do not account for the fact that an image often contain many object instances of various sizes, thus showing
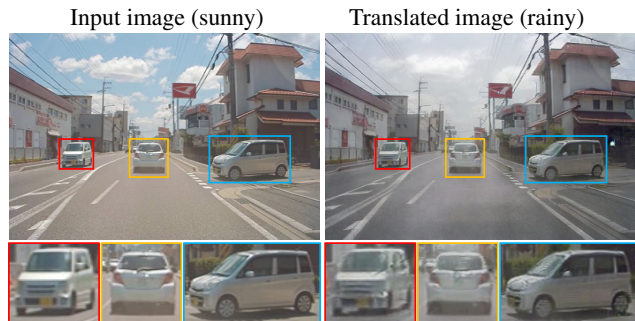
_____
*Corresponding author

Figure 1. **Results of InstaFormer for instance-aware image-to-image translation.** Our InstaFormer effectively considers global- and instance-level information with Transformers, which enables high quality instance-level translation.

the limited performance at content-rich scene translation, e.g., driving scene, which is critical for some downstream tasks, such as domain adaptive object detection [3], that require well-translated object instances.

To address the aforementioned issues, some methods [3, 27, 49] seek to explicitly consider an object instance in an image within deep convolutional neural networks (CNNs). This trend was initiated by instance-aware I2I (INIT) [49], which treats the object instance and global image separately. Following this [49], some variants were proposed, e.g., jointly learning translation networks and object detection networks, called detection-based unsupervised I2I (DUNIT) [3], or using an external memory module, called memory-guided unsupervised I2I (MGUIT) [27]. While these methods improve an instance-awareness to some extent, they inherit limitation of CNN-based architectures [3, 27, 49], e.g., local receptive fields or limited encoding of relationships or interactions between pixels or patches within an image which are critical in differentiating an object instance from an whole image and boosting its translation.

To tackle these limitations, for the first time, we present to utilize Transformer [53] architecture within I2I networks that effectively integrates global- and instance-level information present in an image, dubbed InstaFormer. We follow common disentangled representation approaches [22, 33] to extract both content and style vectors. By considering extracted content features from an image as *tokens*,

our Transformer-based aggregator mixes them to discover global consensus by considering global context information through a self-attention module, thus boosting the instance-awareness during translation. In addition, by augmenting such tokens by an instance-level feature extracted from the global content feature with respect to bounding box information, our framework is able to learn an interaction between not only object instance and global image, but also different instances, followed by a position embedding technique to consider both global- and instance-level patches at once, which helps the networks to better focus on the object instance regions. We also replace layer normalization (LayerNorm) [1] in Transformers with adaptive instance normalization (AdaIN) [21] to facilitate a multi-modal translation with extracted or random style vectors. Since aggregating raw content and style vectors directly with Transformers requires extremely large computation [12, 29], we further propose to apply a convolutional patch embedding and deconvolutional module at the beginning and end of our Transformer-based aggregator. In addition, to improve the instance-awareness and quality of translation images at object regions, we present an instance-level content contrastive loss defined between input and translated images.

In experiments, we demonstrate our framework on several benchmarks [8, 15, 49] that contain content-rich scenes. Experimental results on various benchmarks prove the effectiveness of the proposed model over the latest methods for instance-aware I2I. We also provide an ablation study to validate and analyze components in our model.

2. Related Work

Image-to-Image Translation. While early efforts for I2I are based on supervised learning [26], most recent state-of-the-arts focus on unpaired settings [2, 13, 37, 61, 65, 67]. CycleGAN [68] attempts this by proposing a cycle-consistency loss which has been one of standard losses for unpaired I2I. Inspired by CycleGAN, numerous methods utilize cycle-consistency [7, 20, 22, 31, 33, 59], and they can be largely divided into uni-modal models [36, 59, 61] and multi-modal models [7, 22, 33] methods. Specifically, MUNIT [22] assumes that an image representation can be disentangled into a domain-specific style and a domain-invariant content representation and uses these disentangled latent features with cycle-consistency to generate the translations. However, the content in translated image can be easily distorted, and cycle mapping requires multiple generators and discriminators. To address these, CUT [43]and F-LSeSim [65] propose novel losses inspired by infoNCE [42] to directly compute distance between input and translated images in an one-side framework without cycle-consistency. However, they still have shown limited performance to encode an object-awareness at the translated image.

Instance-Aware Image-to-Image Translation. Some methods attempted to address the aforementioned issues [3, 27, 41, 49]. INIT [49] attempted to translate the whole image and object instances independently. DUNIT [3] proposed to further train detection module and adopted instance-consistency loss for object-awareness. MGUIT [27] utilizes bounding box to read and write class-wise memory module, and has access to class-aware features on memory at test-time. The aforementioned methods inherit limitation of CNN-based architecture, e.g., local receptive fields or limited encoding of relationships or interactions within an image [22, 33, 68].

Vision Transformers and Image Generation Recently, Vision Transformers (ViT) have shown to attain highly competitive performance for a wide range of vision applications, such as image classification [10, 12, 52, 55], object detection [5, 9, 69], and semantic segmentation [58, 66]. Inspired by ViT [12], some improvements are made to improve the computational complexity [29, 38, 54, 55]. For example, Swin Transformer [38] proposes relative position biases, and restricts self-attentioncomputation within shifted windows. MLP-Mixer [50] suggests to replace self-attention with an MLP, achieving memory efficiency and competitive performance [35, 39, 51]. In this paper, we introduce ViT-based aggregator to further enhance to learn instance-awareness by aggregating information from local region as well as global image.

On the other hands, there exist several efforts to adapt Vision Transformers to image generation tasks [6, 23, 28, 34, 64]. As seminal work, TransGAN [28] first presents a GAN structure using pure Transformer, but has only validated on low-resolution images. [64] has achieved success on generating high-resolution images. [23] leverages Transformers to build the bipartite structure to allow long-range interactions. To our best knowledge, our work is the first attempt to adopt Transformers in instance-aware image translation.

3. Methodology

3.1. Overview

Our approach aims to learn a multi-modal mapping between two domains $\mathcal{X} \subset \mathbb{R}^{H \times W \times 3}$ and $\mathcal{Y} \subset \mathbb{R}^{H \times W \times 3}$ without paired training data, but with a dataset of unpaired instances $X = \{x \in \mathcal{X}\}$ and $Y = \{y \in \mathcal{Y}\}$. Especially, we wish to model such a mapping function to have an ability that jointly accounts for whole image and object instances. Unlike conventional I2I methods [3, 16, 22, 27, 33, 49] that were formulated in a two-sided framework to exploit a cycle-consistency constraint, which often generates some distortions on the translated images and requires auxiliary networks for inverse mapping [68], we formulate our approach in an one-sided framework [43].

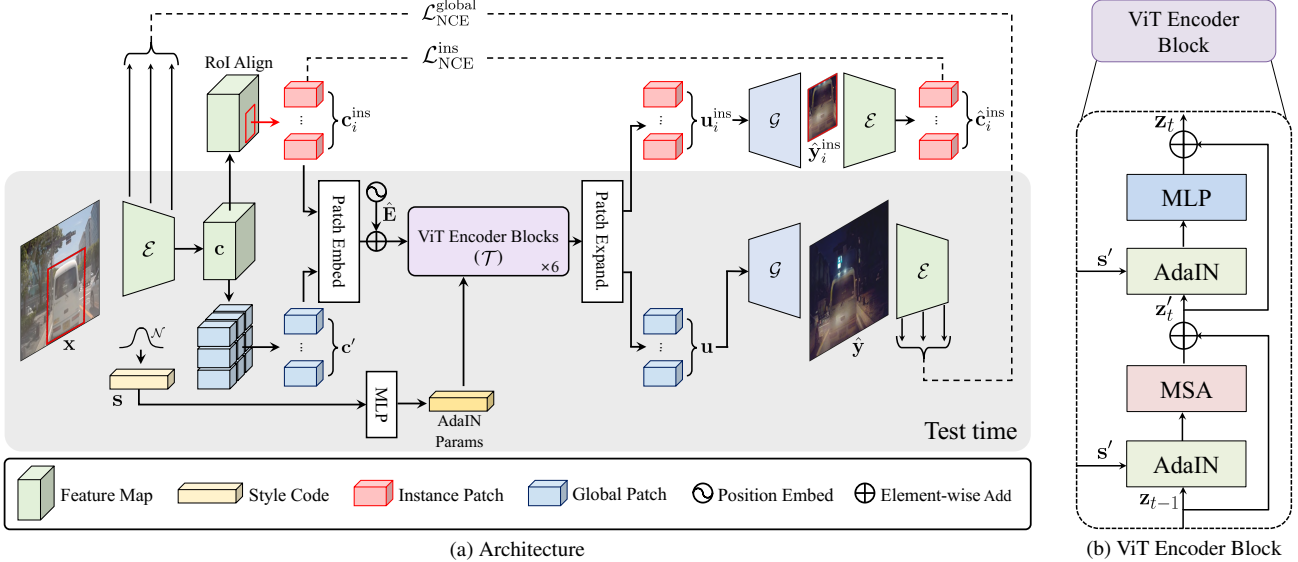In specific, as illustrated in Fig. 2, our framework,

Figure 2. **Network configuration:** (a) overall architecture for image-to-image translation, (b) ViT encoder block in details. Our networks consist of content encoder, Transformer encoder, and generator. The gray background represents the test phase, where we have no access on object instance bounding box (Best viewed in color).

dubbed InstaFormer, consists of content encoder $\mathcal{E}$ and generator $\mathcal{G}$, similar to [43,65], and additional encoder $\mathcal{T}$ with Transformers [53] to improve the instance-awareness by considering global consensus between whole image and object instances. To translate an image $\mathbf{x}$ in domain $\mathcal{X}$ to domain $\mathcal{Y}$, our framework first extracts a content feature map $\mathbf{c} = \mathcal{E}(\mathbf{x}) \in \mathbb{R}^{h \times w \times l_c}$ from $\mathbf{x}$, with height h, width w, and l_c channels, and randomly draws a style latent code $\mathbf{s} \in \mathbb{R}^{1 \times 1 \times l_s}$ from the prior distribution $q(\mathbf{s}) \sim \mathcal{N}(0, \mathbf{I})$ to achieve a multi-modal translation. Instead of directly feeding $\mathbf{c}$ and $\mathbf{s}$ to the generator $\mathcal{G}$, as done in the literature [22,33], we aggregate information in the content $\mathbf{c}$ to discover global consensus between the global image and object instances in a manner that we first extract an object instance content vector $\mathbf{c}_i^{\text{ins}}$ for i-th object bounding box with parameters $B_i = [x_i, y_i, h_i, w_i]$, where (x_i, y_i) represent a center point, and h_i and w_i represent height and width of the box and $i \in 1, ..., N$ where N is the number of instance, and then mix $\{\mathbf{c}, \{\mathbf{c}_i^{\text{ins}}\}_i, \mathbf{s}\}$ through the proposed Transformer module $\mathcal{T}$ to extract global embedding $\mathbf{u}$ and instance embedding $\mathbf{u}_i^{\text{ins}}$, which are independently used to generate global-level translated image $\hat{\mathbf{y}} = \mathcal{G}(\mathbf{u}) \in \mathbb{R}^{h \times w \times 3}$ and instance-level translated images $\hat{\mathbf{y}}_i^{\text{ins}} = \mathcal{G}(\mathbf{u}_i^{\text{ins}}) \in \mathbb{R}^{h_i \times w_i \times 3}$. In our framework, during training, we have access to the ground-truth object bounding boxes, while we do not access them at test-time.

To train our networks, we first use an adversarial loss defined between a translated image $\hat{\mathbf{y}}$ and a real image $\mathbf{y}$ from $\mathcal{Y}$ with discriminators, and a global content contrastive loss defined between $\mathbf{x}$ and $\hat{\mathbf{y}}$ to preserve the global content. To improve the disentanglement ability for content and style,

following [22], we also use both image reconstruction loss and style reconstruction loss by leveraging an additional style encoder for $\mathcal{Y}$. To improve the instance-awareness and the quality of translation images at object instance regions, we newly present an instance-level content contrastive loss between $\mathbf{x}$ and $\hat{\mathbf{y}}$.

3.2. Content and Style Mixing with Transformers

Most existing I2I methods [3,22,27,33,43,49] attempted to aggregate a content feature map with deep CNNs with residual connections, which are often called residual blocks, often inserted between encoder and generator networks. They are thus limited in the sense that they inherit limitation of CNN-based architecture, e.g., local receptive fields or limited encoding of relationships or interactions between pixels and patches within an image [22,33,49]. In instance-aware I2I task, enlarging the receptive fields and encoding an interaction between objects and global image may be of prime importance. For instance, if an image contains a *car* object on the *road*, using the context information of not only global background, e.g., road, but also other instances, e.g., other cars or person, would definitely help to translate the image more focusing on the instance, but existing CNN-based methods [22,33,43] would limitedly handle this.

To overcome this, we present to utilize Transformer architecture [53] to enlarge the receptive fields and encode the interaction between features for instance-aware I2I. To this end, extracted content vector $\mathbf{c} \in \mathbb{R}^{h \times w \times l_c}$ from $\mathbf{x}$ can be flattened as a sequence $\mathbf{c}' = \text{Reshpae}(\mathbf{c})$ with the number of tokens hw and channel l_c, which can be directly used as input for Transformers. However, this requires extremely

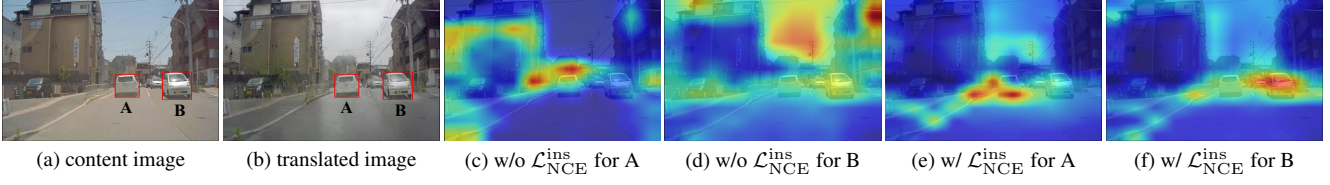| (a) content image | (b) translated image | (c) w/o $\mathcal{L}_{\text{NCE}}^{\text{ins}}$ for A | (d) w/o $\mathcal{L}_{\text{NCE}}^{\text{ins}}$ for B | (e) w/ $\mathcal{L}_{\text{NCE}}^{\text{ins}}$ for A | (f) w/ $\mathcal{L}_{\text{NCE}}^{\text{ins}}$ for B |

Figure 3. **Visualization of learned self-attention.** For (a) content image containing instances A and B, our networks generate (b) translated image, considering attention maps (c,d) without $\mathcal{L}_{\text{NCE}}^{\text{ins}}$ and (e,f) with $\mathcal{L}_{\text{NCE}}^{\text{ins}}$ for instance A, B, respectively.

high computational complexity due to the huge number of tokens hw, e.g., full HD translation.

Patch Embedding and Expanding. To address this issue, inspired by a patch embedding in ViT [12], we first apply sequential convolutional blocks to reduce the spatial resolutions. Instead of applying a single convolution in ViT [12] for extracting non-overlapping patches, we use sequential overlapped convolutional blocks to improve the stability of training while reducing the number of parameters involved [57]. We define this process as follows:

$$\mathbf{p} = \text{Conv}(\mathbf{c}) \in \mathbb{R}^{(h/k) \times (w/k) \times l_c'}, \tag{1}$$

where $k \times k$ is the stride size of convolutions, and l_c' is a projected channel size. After feed-forwarding Transformer blocks such that $\mathbf{z} = \mathcal{T}(\mathbf{p}) \in \mathbb{R}^{(h/k) \times (w/k) \times l_c'}$, downsampled feature map $\mathbf{z}$ should be upsampled again with additional deconvolutional blocks, which are symmetric architectures to the convolutions, defined as follows:

$$\mathbf{u} = \text{DeConv}(\mathbf{z}) \in \mathbb{R}^{h \times w \times l_c}. \tag{2}$$

In addition, for a multi-modal translation, we leverage a style code vector $\mathbf{s} \in \mathbb{R}^{1 \times 1 \times l_s}$, and thus this should be considered during mixing with Transformers [53]. Conventional methods [21, 22, 33] attempted to mix content and style vectors using either concatenation [33] or AdaIN [21]. In our framework, by slightly changing the normalization module in Transformers, we are capable of simultaneously mixing content and style vectors such that $\mathcal{T}(\mathbf{p}, \mathbf{s})$.

Any forms of Transformers [12, 38, 50, 55, 60] can be considered as a candidate in our framework, and in experiments, ViT-like [12] architecture is considered for $\mathcal{T}$. In the following, we explain the details of Transformer modules.

Transformer Aggregator. In order to utilize Transformer to process content patch embeddings $\mathbf{p}$, our work is built upon the ViT encoder, which is composed of an multi-head self-attention (MSA) layer and a feed-forward MLP with GELU [53], where normalization layers are applied before both parts. Especially, for I2I, we adopt AdaIN instead of LayerNorm [1] to control the style of the output with the affine parameters from style vector $\mathbf{s}$ and to enable multimodal outputs. In specific, content patch embedding $\mathbf{p}$ is first reshaped, and position embedding is achieved such that

$$\mathbf{z}_0 = \text{Reshape}(\mathbf{p}) + \mathbf{E} \in \mathbb{R}^{(h/k \cdot w/k) \times l_c'}, \tag{3}$$

where $\mathbf{E}$ represents a position embedding [53], which will be discussed in the following.

These embedded tokens $\mathbf{z}_0$ are further processed by the sequential Transformer encoder blocks as follows:

$$\begin{aligned} \mathbf{z}_t' &= \text{MSA}\left(\text{AdaIN}\left(\mathbf{z}_{t-1}, \mathbf{s}'\right)\right) + \mathbf{z}_{t-1}, \\ \mathbf{z}_t &= \text{MLP}\left(\text{AdaIN}\left(\mathbf{z}_t', \mathbf{s}'\right)\right) + \mathbf{z}_t', \end{aligned} \tag{4}$$

where $\mathbf{z}_t'$ and $\mathbf{z}_t$ denote the output of MSA and MLP modules for t-th block respectively and $t \in 1, ..., T$, respectively, $\mathbf{s}'$ indicates AdaIN parameters extracted from S. After L Transformer modules, followed by reshaping to original resolution, we finally achieve the output of Transformer block $\mathcal{T}$ such that $\mathbf{z}_T = \mathcal{T}(\mathbf{p}, \mathbf{s}')$. As exemplified in Fig. 3, our learned self-attention well considers the interaction between object instances and global image.

3.3. Instance-Aware Content and Style Mixing

So far we discussed a method for content and style mixing with Transformers [12]. This framework can improve the translation quality especially at instance regions to some extent, but the nature of irregular shape of object instances may hinder the performance boosting of our framework. In particular, global-level aggregation itself is limited to capture details of a tiny object and it is not always guaranteed that an object is located in a single regular patch. To overcome this, we present a novel technique to aggregate *instance*-level content features and *global*-level content features simultaneously, which enables the model to pay more attention to the relationships between global scenes and object instances.

In specific, given ground-truth bounding boxes with parameters B_i, we extract instance-level content feature maps through through ROI Align [18] module defined as follows:

$$\mathbf{c}_i^{\text{ins}} = \text{RoIAlign}(\mathbf{c}; B_i) \in \mathbb{R}^{k \times k \times l_c}, \tag{5}$$

where $k \times k$ is a fixed spatial resolution. This can be further processed with the convolutional blocks as proposed above such that

$$\mathbf{p}_i^{\text{ins}} = \text{Conv}(\mathbf{c}_i^{\text{ins}}) \in \mathbb{R}^{1 \times 1 \times l_c'}. \tag{6}$$

In our framework, by concatenating $\mathbf{p}$ and $\mathbf{p}_i^{\text{ins}}$, we build a new input for Transformer $\hat{\mathbf{z}}_0$ such that

$$\hat{\mathbf{z}}_0 = \text{Reshape}(\text{Cat}(\mathbf{p}, \{\mathbf{p}_i^{\text{ins}}\}_i)) + \hat{\mathbf{E}} \in \mathbb{R}^{(h/k \cdot w/k + N) \times l_c'}, \tag{7}$$
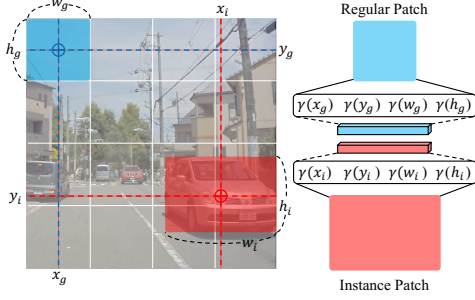
Figure 4. **Illustration of building position embedding for regular patches and instance-level patches.**

where $\mathrm{Cat}(\cdot, \cdot)$ denotes a concatenation operator and $\hat{\mathbf{E}}$ is a corresponding positional embedding. Transformer blocks are then used to process $\hat{\mathbf{z}}_0$ similarly to above to achieve $\hat{\mathbf{z}}_T$, which is decomposed into $\mathbf{z}_T$ and $\mathbf{z}_{T,i}^{\mathrm{ins}}$.

3.4. Instance-Aware Position Embedding

Since Transformer [53] block itself does not contain positional information, we add positional embedding $\mathbf{E}$ as described above. To this end, our framework basically utilizes existing technique [12], but the main difference is that our proposed strategy enables simultaneously considering regularly-partitioned patches $\mathbf{p}$ and instance patches $\mathbf{p}_i^{\mathrm{ins}}$ in terms of their spatial relationships.

The deep networks are often biased towards learning lower frequency functions [45], so we use high frequency functions to alleviate such bias. We denote $\gamma(\cdot)$ as a sinusoidal mapping into $\mathbb{R}^{2K}$ such that $\gamma(a) = (\sin(2^0\pi a), \cos(2^0\pi a), ..., \sin(2^{K-1}\pi a), \cos(2^{K-1}\pi a))$ for a scalar a.

In specific, as a global feature map is divided into regular girds, each regular patch can be represented to have center coordinates (x_g, y_g) with patch width w_g and height h_g of regular size for g-th patch $\mathbf{p}(g)$. After embedding for each information through $\gamma(\cdot)$ and concatenating along the channel axis, it is further added to the patch embedded tokens.

$$\mathbf{E} = \mathrm{Cat}(\gamma(x_g), \gamma(y_g), \gamma(w_g), \gamma(h_g)) \quad (8)$$

Unlike regular patches, which have the same size of width and height for each, instance patches contain positional information of corresponding bounding boxes, which contain the centerpoint coordinates (x_i, y_i) and width and height (w_i, h_i). Instance-wise $\mathbf{E}$ is denoted as:

$$\mathbf{E}^{\mathrm{ins}} = \mathrm{Cat}(\gamma(x_i), \gamma(y_i), \gamma(w_i), \gamma(h_i)). \quad (9)$$

Then $\hat{\mathbf{E}} = \mathrm{Cat}(\mathbf{E}, \mathbf{E}^{\mathrm{ins}})$. Fig. 4 illustrates the difference in how regular patch and instance patch are handled.

3.5. Loss Functions

Adversarial Loss. Adversarial loss aims to minimize the distribution discrepancy between two different features [17,
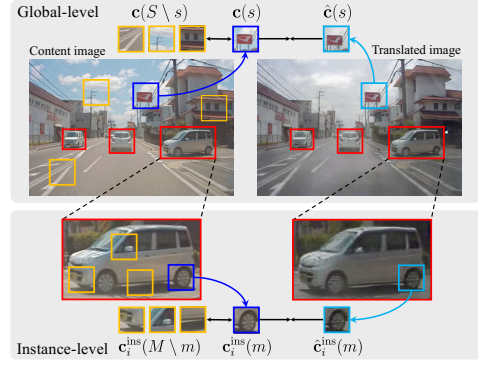

Figure 5. **Illustration of global content loss and instance-level content loss.** Blue box indicates a positive sample, while yellow box means a negative sample (Best viewed in color).

40]. We adopt this to learn the translated image $\hat{\mathbf{y}}$ to be similar to an image $\mathbf{y}$ from $\mathcal{Y}$ defined such that

$$\mathcal{L}_{\mathrm{GAN}} = \mathbb{E}_{\mathbf{x} \sim \mathcal{X}}[\log(1 - \mathcal{D}(\hat{\mathbf{y}}))] + \mathbb{E}_{\mathbf{y} \sim \mathcal{Y}}[\log \mathcal{D}(\mathbf{y})], \quad (10)$$

where $\mathcal{D}(\cdot)$ is the discriminator.

Global Content Loss. To define the content loss between $\mathbf{x}$ and $\hat{\mathbf{y}}$, we exploit infoNCE loss [42], defined as

$$\ell(\hat{\mathbf{v}}, \mathbf{v}^+, \mathbf{v}^-) =$$
$$-\log\left[\frac{\exp(\hat{\mathbf{v}} \cdot \mathbf{v}^+/\tau)}{\exp(\hat{\mathbf{v}} \cdot \mathbf{v}^+/\tau) + \sum_{n=1}^{N}\exp(\hat{\mathbf{v}} \cdot \mathbf{v}_n^-/\tau)}\right], \quad (11)$$

where τ is the temperature parameter, and $\mathbf{v}^+$ and $\mathbf{v}^-$ represent positive and negative for $\hat{\mathbf{v}}$.

We set *pseudo* positive samples between input image $\mathbf{x}$ and translated image $\hat{\mathbf{y}}$. For the content feature from translated image $\hat{\mathbf{c}}(s) = \mathcal{E}(\hat{\mathbf{y}})$, we set positive patches $\mathbf{c}(s)$, and negative patches $\mathbf{c}(S \setminus s)$ from $\mathbf{x}$, where $S \setminus s$ represents indexes except for s, following [43,65]. Global content loss function is then defined as

$$\mathcal{L}_{\mathrm{NCE}}^{\mathrm{global}} = \mathbb{E}_{\mathbf{x} \sim \mathcal{X}} \sum_l \sum_s \ell(\hat{\mathbf{c}}_l(s), \mathbf{c}_l(s), \mathbf{c}_l(S \setminus s)), \quad (12)$$

where $\mathbf{c}_l$ is feature at l-th level, $s \in \{1, 2, ..., S_l\}$ and S_l is the number of patches in each l-th layer.

Instance-level Content Loss. To improve the instance-awareness and the quality of translation images at object regions, we newly present an instance-level content contrastive loss. Our instance-level content loss is then defined such that

$$\mathcal{L}_{\mathrm{NCE}}^{\mathrm{ins}} = \mathbb{E}_{\mathbf{x} \sim \mathcal{X}} \sum_i \sum_m \ell(\hat{\mathbf{c}}_i^{\mathrm{ins}}(m), \mathbf{c}_i^{\mathrm{ins}}(m), \mathbf{c}_i^{\mathrm{ins}}(M \setminus m)),$$
$$(13)$$

where $m \in \{1, 2, ..., M_i\}$ and M_i is the number of patches at each instance. Fig. 5 illustrates how our suggested content losses work, with the procedure to define positive and negative samples.

	CycleGAN [68]		UNIT [36]		MUNIT [22]		DRIT [33]		INIT [49]		DUNIT [3]		MGUIT [27]		InstaFormer	
	CIS	IS	CIS	IS	CIS	IS	CIS	IS	CIS	IS	CIS	IS	CIS	IS	CIS	IS
sunny→night	0.014	1.026	0.082	1.030	1.159	1.278	1.058	1.224	1.060	1.118	1.166	1.259	<u>1.176</u>	<u>1.271</u>	**1.200**	**1.404**
night→sunny	0.012	1.023	0.027	1.024	1.036	1.051	1.024	1.099	1.045	1.080	<u>1.083</u>	1.108	**1.115**	**1.130**	1.115	<u>1.127</u>
sunny→rainy	0.011	1.073	0.097	1.075	1.012	1.146	1.007	1.207	1.036	1.152	1.029	<u>1.225</u>	<u>1.092</u>	1.213	**1.158**	**1.394**
sunny→cloudy	0.014	1.097	0.081	1.134	1.008	1.095	1.025	1.104	1.040	1.142	1.033	1.149	<u>1.052</u>	<u>1.218</u>	**1.130**	**1.257**
cloudy→sunny	0.090	1.033	0.219	1.046	1.026	1.321	1.046	1.249	1.016	1.460	1.077	1.472	<u>1.136</u>	<u>1.489</u>	**1.141**	**1.585**
Average	0.025	1.057	0.087	1.055	1.032	1.166	1.031	1.164	1.043	1.179	1.079	1.223	<u>1.112</u>	<u>1.254</u>	**1.149**	**1.353**

Table 1. **Quantitative evaluation on INIT dataset [49].** For evaluation, we perform bidirectional translation for each domain pair. We measure CIS [22] and IS [48](higher is better). Our results shows the best results in terms of CIS and IS.

Image Reconstruction Loss. We additionally make use of image reconstruction loss to help disentanglement between content and style. For regularization, we use a reconstruction loss to ensure that our $\mathcal{G}$ can reconstruct an image for domain $\mathcal{Y}$. To be specific, $\mathbf{y}$ is fed into $\mathcal{E}$ and style encoder $\mathcal{S}$ to obtain a content feature map $\mathbf{c}^{\mathcal{Y}} = \mathcal{E}(\mathbf{y})$ and a style code $\mathbf{s}^{\mathcal{Y}} = \mathcal{S}(\mathbf{y})$. We then compare the reconstructed image $\mathcal{G}(\mathcal{T}(\mathbf{c}^{\mathcal{Y}}, \mathbf{s}^{\mathcal{Y}}))$ for domain $\mathcal{Y}$ with $\mathbf{y}$ as follows:

$$\mathcal{L}_{\text{recon}}^{\text{img}} = \mathbb{E}_{\mathbf{y} \sim \mathcal{Y}}[\|\mathcal{G}(\mathcal{T}(\mathbf{c}^{\mathcal{Y}}, \mathbf{s}^{\mathcal{Y}})) - \mathbf{y}\|_1]. \quad (14)$$

Style Reconstruction Loss. In order to better learn disentangled representation, we compute L1 loss between style code from the translated image and randomly generated style code in order to enable mapping generated style features to Gaussian distribution such that

$$\mathcal{L}_{\text{recon}}^{\text{style}} = \mathbb{E}_{\mathbf{x} \sim \mathcal{X}, \mathbf{y} \sim \mathcal{Y}}[\|\mathcal{S}(\hat{\mathbf{y}}) - \mathbf{s}\|_1]. \quad (15)$$

Total Loss. The total loss function is as follows:

$$\min_{\mathcal{E}, \mathcal{G}, \mathcal{S}} \max_{\mathcal{D}} \mathcal{L}(\mathcal{E}, \mathcal{G}, \mathcal{D}) = \mathcal{L}_{\text{GAN}} + \lambda^{\text{glob}} \mathcal{L}_{\text{NCE}}^{\text{global}} + \lambda^{\text{ins}} \mathcal{L}_{\text{NCE}}^{\text{ins}}$$
$$+ \lambda^{\text{style}} \mathcal{L}_{\text{recon}}^{\text{style}} + \lambda^{\text{img}} \mathcal{L}_{\text{recon}}^{\text{img}}, \quad (16)$$

where λ^{glob}, λ^{ins}, λ^{style}, and λ^{img} are weights that control the importance of each loss.

4. Experiments

4.1. Implementation Details

We first summarize implementation details in our framework. We conduct experiments using a single 24GB RTX 3090 GPU. Training datasets are resized to the size of 352×352. We employ an Adam optimizer for 200 epochs using a step decay learning rate scheduler. A batch size of 8, an initial learning rate of 2e-4. The number of NCE layers L is 3. For the loss weights, we set as $\lambda^{\text{glob}} = 1$, $\lambda^{\text{ins}} = 1$, $\lambda^{\text{style}} = 10$, and $\lambda^{\text{img}} = 5$. As described above, we implement our framework with the most representative vision Transformer-based, i.e., ViT [12], but we will show our framework works with MLP-Mixer [50] in the following. We will make our code publicly available.

4.2. Experimental Setup

We conduct experiments on two standard datasets for instance-aware I2I, INIT dataset [49] and KITTI-Cityscapes dataset [8, 15]. INIT dataset [49] provides street scene images including 4 domain categories (sunny, night, rainy, cloudy) with object bounding box annotations for car, person, and traffic sign. We conduct translation experiments for sunny→night, night→sunny, sunny→rainy, sunny→cloudy, and cloudy→sunny. KITTI object detection benchmark [15] and Cityscapes [8] dataset are used to evaluate domain adaptation for object detection on KITTI→Cityscapes. KITTI contains 7,481 images for training and 7,518 images for testing with the bounding box annotations for 6 object classes. Cityscapes dataset consists of 5,000 images with pixel-level annotations for 30 classes.

In this section, we compared our InstaFormer with recent state-of-the-art instance-aware I2I methods: INIT [49], DUNIT [3], MGUIT [27], and several unsupervised image-to-image translation methods: CycleGAN [68], UNIT [36], CUT [43], MUNIT [22], and DRIT [33].

4.3. Experimental Results

Qualitative Evaluation. We first conduct qualitative comparisons of our method to CycleGAN [68], UNIT [36], MUNIT [22], DRIT [33], and MGUIT [27] on sunny→night, night→sunny, sunny→cloudy, and sunny→rainy tasks in INIT dataset [49]. As shown in Fig. 6, our model generates higher quality translated results, particularly at object instance regions. Especially, as exemplified in the highlighted regions in Fig. 7, our model is good at capturing local regions within multiple instances thanks to Transformer-based architecture that simultaneously consider object instances and global image, and proposed instance-level contrastive learning. Note that our attention map visualization also proves this, well illustrated in Fig. 3. Note that MGUIT [27] has access on their trained memory module during test-time, which is additional burden.

Quantitative Evaluation. Following the common practice [3, 27, 49], we evaluate our InstaFormer with inception score (IS) [48] and conditional inception score (CIS) [22]. Since the metrics above are related to diversity of translated images, we also evaluate our methods with fréchet incep-

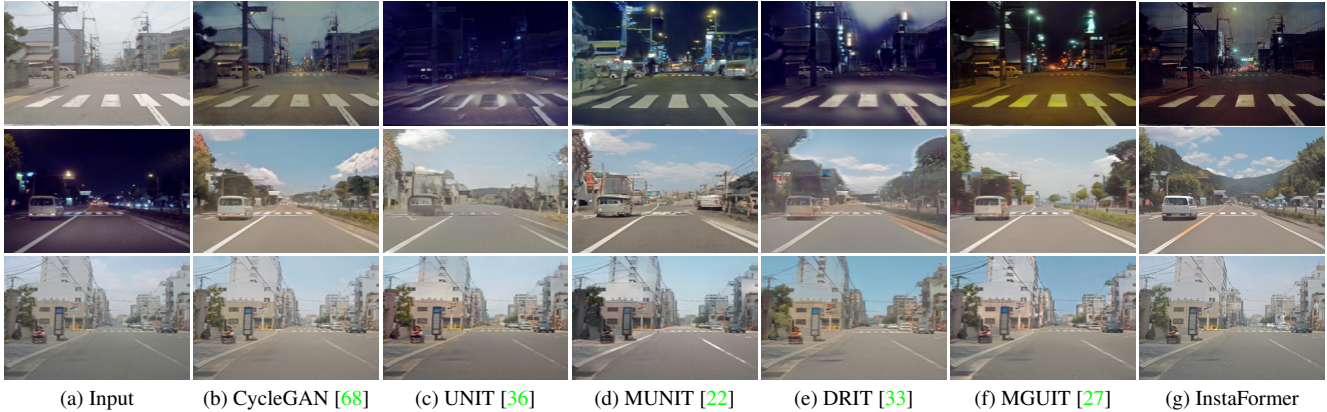| (a) Input | (b) CycleGAN [68] | (c) UNIT [36] | (d) MUNIT [22] | (e) DRIT [33] | (f) MGUIT [27] | (g) InstaFormer |

Figure 6. **Qualitative comparison on INIT dataset [49]:** (top to bottom) sunny→night, night→sunny, and cloudy→sunny results. Among the methods, ours preserves object details well and show realistic results.
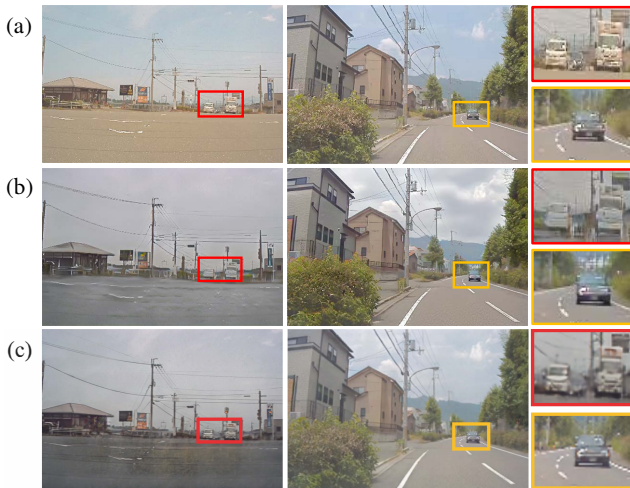

Figure 7. **Visual comparison with MGUIT [27]:** (a) input, (b) MGUIT [27], and (c) InstaFormer. We show the results for sunny→rainy (left) and sunny→cloudy (right).
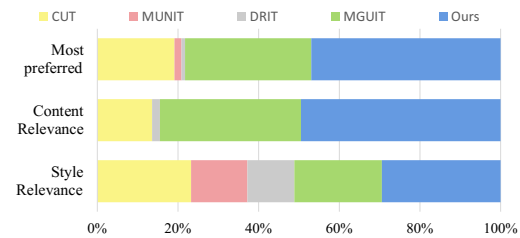

Figure 8. **User study results on INIT dataset [49].** Our method is most preferred for overall quality, semantic consistency and style relevance, compared to CUT [43], MUNIT [22], DRIT [33], and MGUIT [27].

tion score (FID) [19] and structural similarity index measure (SSIM) [56] in terms of quality of translated images. Note that we evaluate the results under the same settings for all the methods. We adopt FID to measure the distance between distributions of real images and synthesized images in a deep feature domain. In addition, since SSIM index is an error measurement which is computed between the original content images and synthesized images, we apply to measure instance-wise structural consistency. It should be noted that for image translation tasks, there often exists some discrepancy between quantitative evaluations and human perceptions [4], thus the user study in the following would be a better precise metric.

As shown in Table 1, our InstaFormer outperforms the current state-of-the-art methods in terms of diversity (CIS, IS). Furthermore, in terms of global distribution, or instance-level similarity as shown in Table 2, FID and SSIM

score show our InstaFormer tends to outperform prior methods in almost all the comparisons. In particular, results on SSIM demonstrate that our network is faithfully designed to encode an instance-awareness. Our method improves the FID score by a large margin compared to previous leading methods MGUIT [27] on INIT dataset [49].

User Study. We also conducted a user study on 110 participants to evaluate the quality of synthesized images in the experiments with the following questions: "Which do you think has better image quality in overall/ similar content to content image / represent style similar to target domain?" on INIT dataset, summarized in Fig. 8. Our method ranks the first in every case, especially on content relevance and overall preference. Note that no standard evaluation metric has been emerged yet, human evaluation has an effect as evaluation metrics in image translation tasks.

4.4. Ablation Study

In order to validate the effectiveness of each component in our method, we conduct a comprehensive ablation study. In particular, we analyze the effectiveness of instance-level loss ($\mathcal{L}_{\mathrm{NCE}}^{\mathrm{ins}}$), Transformer encoder ($\mathcal{T}$), and AdaIN, shown in Fig. 9. It should be noted that CUT [43] can be regarded as the setting without $\mathcal{L}_{\mathrm{NCE}}^{\mathrm{ins}}$, $\mathcal{T}$, AdaIN from InstaFormer. Without $\mathcal{L}_{\mathrm{NCE}}^{\mathrm{ins}}$, our self-attention module has limited capa-

| (a) Content image | (b) InstaFormer | (c) MLP-Mixer [50] | (d) w/o $\mathcal{L}_{NCE}^{ins}$ | (e) w/o $\mathcal{L}_{NCE}^{ins}, \mathcal{T}$ | (f) CUT [43] | (g) w/o AdaIN |

Figure 9. **Ablation study on different settings:** instance-level loss ($\mathcal{L}_{NCE}^{ins}$), Transformer encoder ($\mathcal{T}$), normalization, and another backbone (MLP-Mixer). Note that CUT equals to the setting w/o $\mathcal{L}_{NCE}^{ins}$, $\mathcal{T}$, and AdaIN.

Methods	sunny→night		night→sunny		Average	
	FID↓	SSIM↑	FID↓	SSIM↑	FID↓	SSIM↑
CUT [43]	**75.28**	0.698	<u>80.72</u>	0.634	**78.00**	0.666
MUNIT [22]	100.32	0.703	98.04	0.631	99.18	0.680
DRIT [33]	<u>79.59</u>	0.312	99.33	0.266	89.46	0.289
MGUIT [27]	98.03	<u>0.836</u>	82.17	**0.848**	90.10	<u>0.842</u>
InstaFormer	84.72	**0.872**	**71.65**	<u>0.818</u>	<u>79.05</u>	**0.845**

Table 2. **Quantitative evaluation with FID [19] metric for data distribution and SSIM [56] index measured at each instance.**
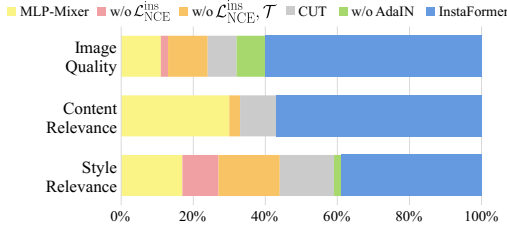

Figure 10. **User study results on ablation study.**

Method	Pers	Car	Truc.	Bic	mAP
DT [25]	28.5	40.7	25.9	29.7	31.2
DAF [22]	39.2	40.2	25.7	48.9	38.5
DARL [32]	46.4	58.7	27.0	49.1	45.3
DAOD [47]	47.3	59.1	28.3	49.6	46.1
DUNIT [3]	<u>60.7</u>	65.1	32.7	<u>57.7</u>	54.1
MGUIT [27]	58.3	<u>68.2</u>	<u>33.4</u>	**58.4**	<u>54.6</u>
InstaFormer	**61.8**	**69.5**	**35.3**	55.3	**55.5**

Table 3. **Results for domain adaptive detection.** We compare the per-class Average Precision for KITTI → CityScape.

4.5. Domain Adaptive Object Detection

Additionally, we evaluate our method on the task of unsupervised domain adaptation for object detection. We follow the experimental setup in DUNIT [3]. We used Faster-RCNN [46] as baseline detector. In Table 3, we report the per-class average precisions (AP) for the KITTI→Cityscapes case [8, 15]. Compared to DUNIT [3] and MGUIT [27], our model shows impressive results. It should be noted that we do not access any information about bounding box information on test-time, while DUNIT contains object detection network and MGUIT has access to trained external memory by reading class-aware features. In particular, our model significantly outperforms other methods in almost all classes, which indicates that our suggested instance loss has strength on instance-awareness.

5. Conclusion

In this paper, we have presented Transformer-based networks, dubbed InstaFormer, for instance-aware image-to-image translation, which enables boosting the translation of object instances as well as global image. By simultaneously considering instance-level features and global-level features with Transformers, we learned an interaction between not only object instance and global image, but also different instances. To improve the instance-awareness during translation, we proposed an instance-level content contrastive loss. Experiments on various datasets with evaluation metrics have shown that our framework outperforms the existing solutions for instance-aware I2I.

bility to focus on objects, thus generating images containing blurred objects, as evaluated in Fig. 3 as well. To validate the effect of $\mathcal{T}$ in our model, we conduct ablation experiments by replacing it with Resblocks (without $\mathcal{L}_{NCE}^{ins}$, $\mathcal{T}$). Without Transformers, it fails to capture global relationship between features. It is obvious that CUT [43] shows limited results containing artifacts, while InstaFormer dramatically improves object-awareness and quality of the generated image thanks to our architecture. Since AdaIN helps to understand global style by leveraging affine parameters, the result without AdaIN, which is replaced with LayerNorm, shows limited preservation on style with a single-modal output. We also validate our ablation study results on human evaluation. 110 participants are asked to consider three aspects: overall quality, semantic consistency and style consistency, summarized in Fig. 10, where we also validate the superiority of each proposed component.

In addition, we conduct experiments using MLP-Mixer [50]-based aggregator that replaces $\mathcal{T}$ consisted of ViT [12] blocks to justify robustness of our framework. Fig. 9(c) shows result examples by MLP-Mixer [50]-based aggregator. Although ViT-based model is slightly better on MLP-Mixer [50]-based model in overall quality in Fig. 9(b), the object instance and style representation are faithfully preserved, which indicates that our method can be adopted in another Transformer backbone.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Kyungjune Baek, Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Hyunjung Shim. Rethinking the truly unsupervised image-to-image translation. In *ICCV*, pages 14154–14163, 2021.

[3] Deblina Bhattacharjee, Seungryong Kim, Guillaume Vizier, and Mathieu Salzmann. Dunit: Detection-based unsupervised image-to-image translation. In *CVPR*, pages 4787–4796, 2020.

[4] Ali Borji. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 179:41–65, 2019.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020.

[6] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *CVPR*, pages 12299–12310, 2021.

[7] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, pages 8789–8797, 2018.

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016.

[9] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *CVPR*, pages 1601–1610, 2021.

[10] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *arXiv preprint arXiv:2106.04803*, 2021.

[11] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *TPAMI*, 38(2):295–307, 2015.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[13] Aviv Gabbay and Yedid Hoshen. Scaling-up disentanglement for image translation. *arXiv preprint arXiv:2103.14017*, 2021.

[14] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, pages 2414–2423, 2016.

[15] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361. IEEE, 2012.

[16] Abel Gonzalez-Garcia, Joost Van De Weijer, and Yoshua Bengio. Image-to-image translation for cross-domain disentanglement. *arXiv preprint arXiv:1805.09730*, 2018.

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014.

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.

[19] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, pages 6626–6637, 2017.

[20] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, pages 1989–1998. PMLR, 2018.

[21] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, pages 1501–1510, 2017.

[22] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, pages 172–189, 2018.

[23] Drew A Hudson and C Lawrence Zitnick. Generative adversarial transformers. *arXiv preprint arXiv:2103.01209*, 2021.

[24] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 36(4):1–14, 2017.

[25] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *CVPR*, pages 5001–5009, 2018.

[26] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017.

[27] Somi Jeong, Youngjung Kim, Eungbean Lee, and Kwanghoon Sohn. Memory-guided unsupervised image-to-image translation. In *CVPR*, pages 6558–6567, 2021.

[28] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two transformers can make one strong gan. *arXiv preprint arXiv:2102.07074*, 1(3), 2021.

[29] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *ICML*, pages 5156–5165. PMLR, 2020.

[30] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, pages 1646–1654, 2016.

[31] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv:1703.05192*, 2017.

[32] Taekyung Kim, Minki Jeong, Seunghyeon Kim, Seokeon Choi, and Changick Kim. Diversify and match: A domain adaptive representation learning paradigm for object detection. In *CVPR*, pages 12456–12465, 2019.

[33] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, pages 35–51, 2018.

[34] Kwonjoon Lee, Huiwen Chang, Lu Jiang, Han Zhang, Zhuowen Tu, and Ce Liu. Vitgan: Training gans with vision transformers. *arXiv preprint arXiv:2107.04589*, 2021.

[35] Hanxiao Liu, Zihang Dai, David R So, and Quoc V Le. Pay attention to mlps. *arXiv preprint arXiv:2105.08050*, 2021.

[36] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *NeurIPS*, pages 700–708, 2017.

[37] Yahui Liu, Enver Sangineto, Yajing Chen, Linchao Bao, Haoxian Zhang, Nicu Sebe, Bruno Lepri, Wei Wang, and Marco De Nadai. Smoothing the disentangled latent style space for unsupervised image-to-image translation. In *CVPR*, pages 10785–10794, 2021.

[38] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[39] Luke Melas-Kyriazi. Do you even need attention? a stack of feed-forward layers does surprisingly well on imagenet. *arXiv preprint arXiv:2105.02723*, 2021.

[40] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[41] Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Instagan: Instance-aware image-to-image translation. *arXiv preprint arXiv:1812.10889*, 2018.

[42] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[43] Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. *arXiv preprint arXiv:2007.15651*, 2020.

[44] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, pages 2536–2544, 2016.

[45] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *ICML*, pages 5301–5310. PMLR, 2019.

[46] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 28:91–99, 2015.

[47] Adrian Lopez Rodriguez and Krystian Mikolajczyk. Domain adaptation for object detection via style consistency. *arXiv preprint arXiv:1911.10033*, 2019.

[48] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *NeurIPS*, 29:2234–2242, 2016.

[49] Zhiqiang Shen, Mingyang Huang, Jianping Shi, Xiangyang Xue, and Thomas S Huang. Towards instance-level image-to-image translation. In *CVPR*, pages 3683–3692, 2019.

[50] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al.

Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021.

[51] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. Resmlp: Feedforward networks for image classification with data-efficient training. *arXiv preprint arXiv:2105.03404*, 2021.

[52] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357. PMLR, 2021.

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[54] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtv2: Improved baselines with pyramid vision transformer. *arXiv preprint arXiv:2106.13797*, 2021.

[55] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021.

[56] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 13(4):600–612, 2004.

[57] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *arXiv preprint arXiv:2106.14881*, 2021.

[58] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *arXiv preprint arXiv:2105.15203*, 2021.

[59] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *ICCV*, pages 2849–2857, 2017.

[60] Dong Zhang, Hanwang Zhang, Jinhui Tang, Meng Wang, Xiansheng Hua, and Qianru Sun. Feature pyramid transformer. In *ECCV*, pages 323–339. Springer, 2020.

[61] Pan Zhang, Bo Zhang, Dong Chen, Lu Yuan, and Fang Wen. Cross-domain correspondence learning for exemplar-based image translation. In *CVPR*, pages 5143–5153, 2020.

[62] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, pages 649–666. Springer, 2016.

[63] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999*, 2017.

[64] Long Zhao, Zizhao Zhang, Ting Chen, Dimitris N Metaxas, and Han Zhang. Improved transformer for high-resolution gans. *arXiv preprint arXiv:2106.07631*, 2021.

[65] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. The spatially-correlative loss for various image translation tasks. In *CVPR*, pages 16407–16417, 2021.

[66] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, pages 6881–6890, 2021.

[67] Xingran Zhou, Bo Zhang, Ting Zhang, Pan Zhang, Jianmin Bao, Dong Chen, Zhongfei Zhang, and Fang Wen. Cocosnet v2: Full-resolution correspondence learning for image translation. In *CVPR*, pages 11465–11475, 2021.

[68] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2223–2232, 2017.

[69] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.