

Memory-guided Unsupervised Image-to-image Translation

Somi Jeong¹ Youngjung Kim² Eungbean Lee¹ Kwanghoon Sohn^{1*}

¹Department of Electrical & Electronic Engineering, Yonsei University, Seoul, Korea

²Agency for Defense Development (ADD), Daejeon, Korea

{somijeong, eungbean, khsohn}@yonsei.ac.kr, read12300@add.re.kr

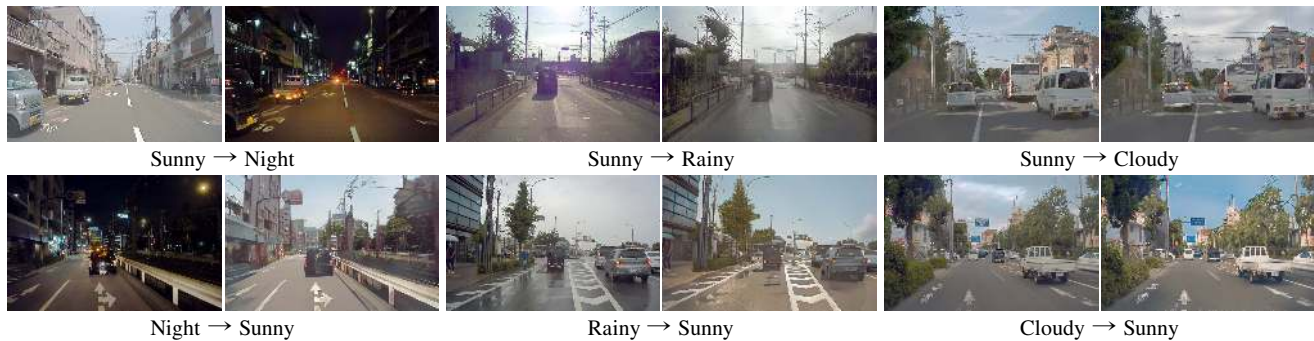


Figure 1: **Instance-level image-to-image translation.** We present a memory-guided unsupervised image-to-image translation method that performs diverse translation between two visual domains by leveraging a class-aware memory.

Abstract

We present a novel unsupervised framework for instance-level image-to-image translation. Although recent advances have been made by incorporating additional object annotations, existing methods often fail to handle images with multiple disparate objects. The main cause is that, during inference, they apply a global style to the whole image and do not consider the large style discrepancy between instance and background, or within instances. To address this problem, we propose a class-aware memory network that explicitly reasons about local style variations. A key-values memory structure, with a set of read/update operations, is introduced to record class-wise style variations and access them without requiring an object detector at the test time. The key stores a domain-agnostic content representation for allocating memory items, while the values encode domain-specific style representations. We also present a feature contrastive loss to boost the discriminative power of memory items. We show that by incorporating our memory, we can transfer class-aware and accurate style representations across domains. Experimental results demonstrate that our model outperforms recent instance-level methods and achieves state-of-the-art performance.

This research was supported by the Agency for Defense Development under the grant UD2000008RD.

*Corresponding author

1. Introduction

Unsupervised image-to-image (I2I) translation is the task of learning a mapping between unpaired images in diverse domains. It can be applied to a variety of applications, including attribute manipulation [3, 21], style transfer [43, 12], data augmentation [25, 11], and domain adaptation [30, 10]. Recent methods [49, 23, 16, 42, 47] achieved impressive results based on a cycle-consistency constraint that forces translated images to be mapped back to their original domain. However, they usually assume a deterministic one-to-one mapping between two domains, thus failing to capture the full distribution of possible outputs. Several methods [50, 13, 22, 8, 45] aim to model complex and multimodal distributions to generate diverse outputs. They postulate that the image representation can be disentangled into domain-invariant content and domain-specific style. However, they simply formulate I2I translation as a global translation problem and apply a global content/style to entire images, which is problematic when handling complex images with many disparate objects. Recently, INIT [38] and DUNIT [1] alleviated this problem by separately treating object instances and background with additional object annotations. During training, INIT [38] independently translates the instances using a separate reconstruction loss along with the global translation module. At test time, however, it only uses the global module and discards the instance-level information. DUNIT [1] integrates an object detector

within the I2I translation module and adds an instance-level encoder to extract instance-boosted features. Although it can leverage the object instances at test time, it is not flexible enough to model diverse local style variations. Furthermore, both methods require an off-the-shelf computationally expensive object detection module at test time.

Motivated by the aforementioned problems, in this paper, we introduce a novel instance-level I2I translation framework with an external memory module. Specifically, we propose a class-aware memory network that can accurately store and propagate local-style information across different visual domains. It comprises several class-wise memory matrices, and each matrix contains a set of key-values (items). The key is used to address relevant memory items with respect to queries, and covers a shared content space. Conversely, the values encode domain-specific style representations for its paired key. This memory module allows storing diverse styles for different object instances into memory items during training (*update*) and efficiently accessing them without an explicit object detector at test time (*read*). Furthermore, we present a feature contrastive loss to enhance the discriminative power of memory items. We show that, by incorporating our memory, the proposed method can capture the object details and reconstruct realistic images. Experimental results on standard benchmarks, including INIT [38], KITTI [7], and Cityscapes [4], demonstrate the effectiveness of our method, which outperforms state-of-the-art instance-level I2I translation methods. Furthermore, we demonstrate that our approach can be applied to domain adaptation detection tasks.

Our contributions can be summarized as follows:

- We propose a memory-guided unsupervised I2I translation (MGUIT) framework that stores and propagates instance-level style information across visual domains. To best of our knowledge, this is the first work that explores a memory network in I2I translation.
- We introduce a key-values memory structure to effectively record diverse style variations and access them during I2I translation. Our model does not require explicit object detection modules at test time. We also propose a feature contrastive loss to improve the diversity and discriminative power of our memory items.
- Our method produces realistic translation results while preserving instance details well; it outperforms recent state-of-the-art methods on standard benchmarks.

2. Related Work

Image-to-image translation. The seminal work of Pix2Pix [15] achieved impressive results in I2I translation tasks using paired images based on conditional generative adversarial networks (GANs) [28]. To reduce the difficulty in collecting the image pairs, various unsupervised I2I translation approaches [49, 23, 16, 42, 47] have been

proposed. They mainly regularize ill-posed training procedure by adopting a cycle consistency constraint, which enforces the translated image from source to target domain to be mapped back to the source domain. Because they model a deterministic one-to-one mapping, they failed to generate diverse outputs. To tackle this limitation, some methods have extended it into multi-modal/multi-domain mapping [50, 13, 22, 8, 45]. Based on the assumption that images can be disentangled into shared content and separate style representations, they apply various learning strategies to enhance their generalization capabilities, such as weight sharing [22, 13], variational autoencoder [13, 8], and normalization layer [43, 6, 12]. Unfortunately, they show poor results when translating images with multiple instances because they do not consider instance-level information.

Instance-level image-to-image translation Very recently, several efforts have been dedicated to achieving instance-level I2I translation [29, 38, 1]. InstaGAN [29] performs the instance-level image translation using the object segmentation masks as extra supervision while maintaining the background. On the other hand, INIT [38] and DUNIT [1] focus on translating instances and backgrounds simultaneously, which is the same objective as our work. INIT [38] employs the instance and global styles separately to guide the generation of target domain objects directly. In inference, however, it uses the global style only, thus neglecting the instance style. DUNIT [1] incorporates the object detector and I2I translation to extract the instance-boost feature representations. Since the global and instance features are unified using a global style, its translated results may lose the inherent instance characteristic. Different from the aforementioned methods, we aim to infer the instance style in both training and testing time to produce more realistic results. To this end, we adopt the novel memory networks, which store the style information during training and read the appropriate style representation for inference.

Memory networks. Memory network [44, 40] is a learnable neural network module, which stores information in external memory and reads the relevant contents from the memory. The Key-Value Memory Networks [27] was introduced, which exploits a key-value structured memory for reading documents. Given a query, the key is used to retrieve relevant memories, and its corresponding values are returned. Thanks to its high flexibility that it records different knowledge in the key and value, it has been widely adopted in solving various vision problems such as natural language processing [20, 5], movie understanding [31], visual tracking [46], and video object segmentation [32, 26].

Inspired by [27], we introduce a key-values structured memory, modified to be suitable for I2I translation. Recently, DM-GAN [51] adopts a dynamic memory network to generate a high-quality image from text descriptions. They select the relevant value by comparing the key mem-

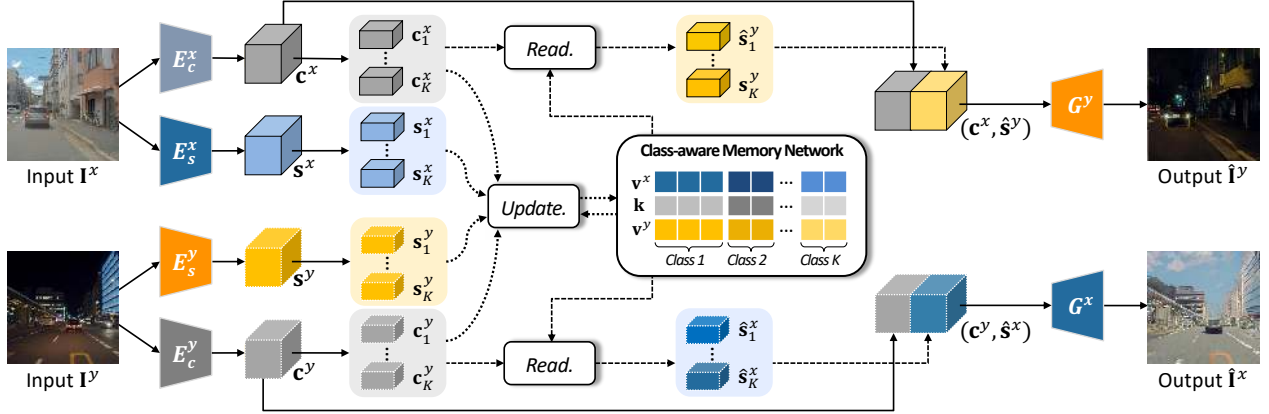


Figure 2: **The overview of the proposed architecture.** The content and style encoders extract content \mathbf{c}^x and style \mathbf{s}^x features from the input image \mathbf{I}^x and they are clustered by object class $\{(\mathbf{c}_1^x, \mathbf{s}_1^x), \dots, (\mathbf{c}_K^x, \mathbf{s}_K^x)\}$. The class-aware memory network consists of key-values memory items $(\mathbf{k}, \mathbf{v}^x, \mathbf{v}^y)$ assigned to each object class and uses $(\mathbf{c}_k^x, \mathbf{s}_k^x)$ to read and update memory items. The generator takes the enhanced style feature maps $\hat{\mathbf{s}}^y$ retrieved from memory and generates the image $\hat{\mathbf{I}}^y$.

ory with the input text, and it is used to generate the image. In contrast, we employ the key-values memory to store domain-agnostic content representations and domain-specific style representations.

3. Proposed Method

We denote by \mathcal{X} and \mathcal{Y} two visual domains, e.g., sunny and night (or rainy). Our objective is to learn a multi-modal mapping between \mathcal{X} and \mathcal{Y} by accurately storing and propagating class-aware style information. To this end, we introduce a novel memory network along with an I2I network to explicitly explain the objects. The memory network contains several memory items; each memory item stores class-aware feature representations. The features from the I2I encoders, i.e., *queries*, are used to *read* and *update* class-aware features in the memory. The I2I generator then inputs them to reconstruct the final translated image. An overview of our framework is illustrated in Fig. 2. We assume that, during training time, we can access the ground-truth object annotations (bounding box and class) to update the memory items assigned for each class. At test time, however, no object annotations are required given that we can retrieve the appropriate memory items through the read operations. Next, we comprehensively describe the components of the MGUIT framework.

3.1. Image-to-Image Translation Network

We basically follow the DRIT [22] architecture¹. Our architecture consists of two coupled content encoders $E_c = \{E_c^x, E_c^y\}$, style encoders $E_s = \{E_s^x, E_s^y\}$, and generators $\{G^x, G^y\}$ in each domain, \mathcal{X} or \mathcal{Y} . For adversarial learning, it additionally contains domain discriminators $\{D^x, D^y\}$ to determine whether the image is from its original domain,

¹We thus omit unnecessary details to avoid repetition.

and a content discriminator D_c . As in [22], we decompose an image \mathbf{I} into a domain-agnostic content space $\mathbf{c} \in \mathcal{C}$ and a domain-specific style space $\mathbf{s} \in \mathcal{S}$, where $(\mathbf{c}^x, \mathbf{c}^y) = (E_c^x(\mathbf{I}^x), E_c^y(\mathbf{I}^y))$ and $(\mathbf{s}^x, \mathbf{s}^y) = (E_s^x(\mathbf{I}^x), E_s^y(\mathbf{I}^y))$. The existing I2I methods [22, 13, 8] simply swap \mathbf{s} from both domains ($\mathcal{X} \leftrightarrow \mathcal{Y}$) to produce $\hat{\mathbf{I}}^y = G^y(\mathbf{c}^x, \mathbf{s}^y)$ (and vice versa for $\hat{\mathbf{I}}^x$). This strategy performs a global-style translation over the entire image, making the results for complex scenes with multiple objects less realistic. In contrast, we use an external class-aware memory network M that records diverse intra- and inter-class style variations simultaneously. Through a read operation, the memory M takes \mathbf{c} as query maps and outputs the enhanced style feature maps $\hat{\mathbf{s}}$. Finally, the generators reconstruct the translated images by combining \mathbf{c} and $\hat{\mathbf{s}}$:

$$\hat{\mathbf{I}}^x = G^x(\mathbf{c}^y, \hat{\mathbf{s}}^x), \quad \hat{\mathbf{I}}^y = G^y(\mathbf{c}^x, \hat{\mathbf{s}}^y). \quad (1)$$

Next, we describe how to read the appropriate style and update M according to the object classes.

3.2. Class-aware Memory Network

The memory network contains N memory items to store class-aware feature representations. We assign N_k items to each class, where $\sum_{k=1}^K N_k = N$ and K is the total number of classes (including the background). N_k is the parameter used to model the intra-class variation, which can vary according to the class. For example, we can assign 4 and 6 memory items to “car” and “background” classes, respectively, for a total of $N = 10$ memory items. Each item consists of a pair of $1 \times 1 \times C$ vectors $(\mathbf{k}, \mathbf{v}^x, \mathbf{v}^y)$, where C is the number of channels. \mathbf{k} denotes the shared **key** used to address items, and also encodes the domain-agnostic content representations. Similarly, **values** $(\mathbf{v}^x, \mathbf{v}^y)$ store domain-specific style representations for the paired \mathbf{k} . This **key-values** memory structure allows recording diverse

style variations into memory items and accessing them during I2I translation without an off-the-shelf object detector.

Given the object annotations, we first cluster (\mathbf{c}, \mathbf{s}) into a set of features $\{(\mathbf{c}_1, \mathbf{s}_1), \dots, (\mathbf{c}_K, \mathbf{s}_K)\}$ to train the memory network. We feed the class-wise cluster $(\mathbf{c}_k, \mathbf{s}_k)$ to only read/write the corresponding N_k memory items, as shown in Fig. 3. Next, the subscript k is omitted for simplicity, but we note that $(\mathbf{c}_k, \mathbf{s}_k)$ are only applied to the corresponding N_k items assigned to class k .

Read. To read the appropriate style values, we compute the similarity between each \mathbf{c}_p and \mathbf{k} , resulting in a read-weight matrix α^x (or y) $\in \mathbb{R}^{P \times N}$:

$$\alpha_{p,n}^x = \frac{\exp(d(\mathbf{c}_p^x, \mathbf{k}_n))}{\sum_{n'=1}^N \exp(d(\mathbf{c}_p^x, \mathbf{k}_{n'}))}, \alpha_{p,n}^y = \frac{\exp(d(\mathbf{c}_p^y, \mathbf{k}_n))}{\sum_{n'=1}^N \exp(d(\mathbf{c}_p^y, \mathbf{k}_{n'}))}, \quad (2)$$

where \mathbf{c}_p denotes individual features ($p = 1, \dots, P$) of size $1 \times 1 \times C$, and P is the total number of pixels in \mathbf{c} . $d(\cdot, \cdot)$ is defined using cosine similarity as follows:

$$d(\mathbf{c}_p, \mathbf{k}_n) = \frac{\mathbf{c}_p \mathbf{k}_n^\top}{\|\mathbf{c}_p\|_2 \|\mathbf{k}_n\|_2}. \quad (3)$$

Inspired by [27], we read the memory item by taking a weighted average of the cross-domain values:

$$\hat{\mathbf{s}}_p^x = \sum_{n'=1}^N \alpha_{p,n'}^y \mathbf{v}_{n'}^x, \quad \hat{\mathbf{s}}_p^y = \sum_{n'=1}^N \alpha_{p,n'}^x \mathbf{v}_{n'}^y. \quad (4)$$

This step is repeated for all \mathbf{c}_p (or y), and produces an enhanced and aggregated style feature map $\hat{\mathbf{s}}^y$ (or x)². Through (4), our model can transfer class-aware and spatially varying style information across domains ($\mathcal{X} \leftrightarrow \mathcal{Y}$) by referring to their content characteristics. The translated images ($\hat{\mathbf{I}}^x, \hat{\mathbf{I}}^y$) can be obtained according to (1).

Update. To enrich the memory items, we also select and store class-aware features into the memory while removing redundant features from the memory. Similar to the read operation, we calculate an update-weight matrix β^x (or y) $\in \mathbb{R}^{P \times N}$ between \mathbf{c} and \mathbf{k} :

$$\beta_{p,n}^x = \frac{\exp(d(\mathbf{c}_p^x, \mathbf{k}_n))}{\sum_{p'=1}^P \exp(d(\mathbf{c}_{p'}^x, \mathbf{k}_n))}, \beta_{p,n}^y = \frac{\exp(d(\mathbf{c}_p^y, \mathbf{k}_n))}{\sum_{p'=1}^P \exp(d(\mathbf{c}_{p'}^y, \mathbf{k}_n))}, \quad (5)$$

where we apply the softmax function along the \mathbf{c} -direction, as opposed to (2). The update-weight matrix β is used to assign the extracted content \mathbf{c} and style features \mathbf{s} to the relevant memory item. The items $(\mathbf{k}_n, \mathbf{v}_n^x, \mathbf{v}_n^y)$ are updated using $(\mathbf{c}_p, \mathbf{s}_p)$ weighted by β as follows:

$$\begin{aligned} \hat{\mathbf{k}}_n &= \|\mathbf{k}_n + \sum_{p'=1}^P \beta_{p',n}^x \mathbf{c}_{p'}^x + \sum_{p'=1}^P \beta_{p',n}^y \mathbf{c}_{p'}^y\|_2, \\ \hat{\mathbf{v}}_n^x &= \|\mathbf{v}_n^x + \sum_{p'=1}^P \beta_{p',n}^x \mathbf{s}_{p'}^x\|_2, \\ \hat{\mathbf{v}}_n^y &= \|\mathbf{v}_n^y + \sum_{p'=1}^P \beta_{p',n}^y \mathbf{s}_{p'}^y\|_2. \end{aligned} \quad (6)$$

²Specifically, $(\mathbf{s}_1, \dots, \mathbf{s}_k)$ are separately processed with the corresponding N_k memory items assigned to class k (see Fig. 3), and then merged into $\hat{\mathbf{s}}$ using their original coordinates in \mathbf{s} .

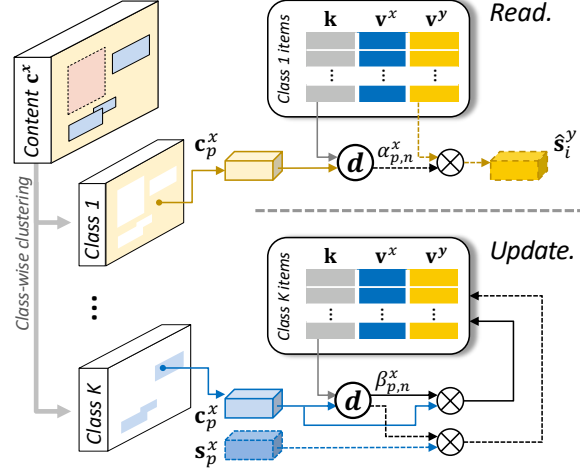


Figure 3: **Read and update operations for training.** We cluster features by class, and the read and update operations are processed class-wisely. For read, we compute a read-weight $\alpha_{p,n}^x$ between each \mathbf{c}_p^x and all memory keys \mathbf{k} in (2). The aggregated style feature $\hat{\mathbf{s}}_p^y$ is retrieved by taking a weighted average of the cross-domain values as (4). For update, we compute an update-weight $\beta_{p,n}^x$ in (5), and update the key and values as (6). (and vice versa for domain \mathcal{Y})

We utilize both $(\mathbf{c}_p^x, \mathbf{c}_p^y)$ to update \mathbf{k}_n because it records the shared content representations. In contrast, the domain-specific values $(\mathbf{v}_n^x, \mathbf{v}_n^y)$ are updated individually. We train the memory with a large number of images and ground-truth object annotations, thus enabling the most representative and discriminative features to be stored.

At test time, we compute α for all memory keys \mathbf{k} without considering class information and retrieve the style values using (2) and (4). We find that this strategy still works well because our memory is discriminatively trained using ground-truth object annotations.

3.3. Loss Functions

3.3.1 Image-to-image translation network

Following DRIT [22], we adopt several loss functions to facilitate proper image reconstruction as follows.

Reconstruction loss makes the translated image similar to its original image [49, 23], which regularizes the ill-posed unsupervised I2I translation problem. It consists of two terms, namely self-reconstruction \mathcal{L}^{self} and cycle-reconstruction \mathcal{L}^{cyc} , which are expressed as

$$\begin{aligned} \mathcal{L}^{self} &= \mathbb{E}_{x,y} [\|G^x(\mathbf{c}^x, \hat{\mathbf{s}}^x) - \mathbf{I}^x\|_1 + \|G^y(\mathbf{c}^y, \hat{\mathbf{s}}^y) - \mathbf{I}^y\|_1], \\ \mathcal{L}^{cyc} &= \mathbb{E}_{x,y} [\|G^x(\hat{\mathbf{c}}^y, \hat{\mathbf{s}}^x) - \mathbf{I}^x\|_1 + \|G^y(\hat{\mathbf{c}}^x, \hat{\mathbf{s}}^y) - \mathbf{I}^y\|_1], \end{aligned} \quad (7)$$

where $(\hat{\mathbf{c}}^x, \hat{\mathbf{c}}^y)$ denotes the content features from $(\hat{\mathbf{I}}^x, \hat{\mathbf{I}}^y)$.

Adversarial loss aims to minimize the distribution discrepancy between two different features, widely used in

GANs [9, 28]. We adopt two adversarial loss functions: the content adversarial loss \mathcal{L}_c^{adv} between \mathbf{c}^x and \mathbf{c}^y , and the domain adversarial loss \mathcal{L}_d^{adv} between \mathcal{X} and \mathcal{Y} .

KL loss \mathcal{L}^{KL} makes the style representation to be close to a prior Gaussian distribution.

Latent regression loss \mathcal{L}^{latent} enforces the mappings between the style and the image to be invertible.

3.3.2 Class-aware memory network

It is important to store representative and discriminative class-aware features in the memory. To this end, we propose a feature contrastive loss function.

Feature contrastive loss For each feature \mathbf{c}_p (or \mathbf{s}_p), we define its nearest item \mathbf{k}_{p+} (or \mathbf{v}_{p+}) as a positive sample, and the others as negative samples. The distances to the positive/negative samples are penalized as follows:

$$\begin{aligned}\mathcal{L}_k^{con} &= -\sum_{p=1}^P \log \frac{\exp(\mathbf{c}_p \cdot \mathbf{k}_{p+}/\tau)}{\sum_{n=1}^N \exp(\mathbf{c}_p \cdot \mathbf{k}_n/\tau)}, \\ \mathcal{L}_v^{con} &= -\sum_{p=1}^P \log \frac{\exp(\mathbf{s}_p \cdot \mathbf{v}_{p+}/\tau)}{\sum_{n=1}^N \exp(\mathbf{s}_p \cdot \mathbf{v}_n/\tau)},\end{aligned}\quad (8)$$

for both domains, \mathcal{X} and \mathcal{Y} . τ is a temperature parameter that controls the distribution concentration level.

This is conceptually similar to feature separateness loss in [33], which encourages the queries to be close to the nearest item and separates individual items in the memory. However, they only consider the second-nearest item as a negative sample using triplet loss [39]. Thus, the selection method of the second-nearest item has a high impact on the training efficiency and final performance. By contrast, the proposed feature contrastive loss compares all items in the memory. It is more effective for learning good feature representations and clustering in an unsupervised manner.

As a summary, the full objective function is as follows:

$$\begin{aligned}\min_{E_c, E_s, G} \max_{D, D_c} & \lambda^{self} \mathcal{L}^{self} + \lambda^{cyc} \mathcal{L}^{cyc} + \lambda_c^{adv} \mathcal{L}_c^{adv} \\ & + \lambda_d^{adv} \mathcal{L}_d^{adv} + \lambda^{KL} \mathcal{L}^{KL} + \lambda^{latent} \mathcal{L}^{latent} \\ & + \lambda_k^{con} \mathcal{L}_k^{con} + \lambda_v^{con} \mathcal{L}_v^{con},\end{aligned}\quad (9)$$

where the λ s control the importance of each term.

4. Experiments

4.1. Experimental Settings

Implementation Details. Our networks were implemented based on DRIT³ with PyTorch [34] and trained on one single NVIDIA TITAN RTX GPU. Every network weights of each layer are initialized by a Gaussian distribution with a zero mean and a standard deviation of

³<https://github.com/HsinYingLee/DRIT>

0.001. The Adam solver [18] was employed for optimization, where $\beta_1 = 0.9$, $\beta_2 = 0.999$. The batch size was set to 1. The initial learning rate was set to 0.0001 and 1, kept for first 30 epochs, and linearly decayed to zero over the next 30 epochs. We set the number of memory items as 20 and its channel C as 256. We resize the short side of images to 360 pixels and crop it to 360×360 to train our framework. The hyperparameters $\{\lambda^{self}, \lambda^{cyc}, \lambda_c^{adv}, \lambda_d^{adv}, \lambda^{KL}, \lambda^{latent}\}$ for I2I translation network are set the same as DRIT [50], and $\{\lambda_k^{con}, \lambda_v^{con}\}$ are empirically determined 1 and 0.5. Our code will be made publicly available.

Datasets. We conduct experiments on three datasets.

(1) *INIT dataset* [38] consists of 155K street scene images, including 4 domain categories (sunny, night, rainy, and cloudy). It provides instance bounding box and object class annotations for car, person, and traffic sign. We set the number of memory items for each class as 5, 3, and 2, and for the background as 10. Following INIT [38], we use 85% images for training and 15% images for testing. We conduct three translation experiments for sunny \leftrightarrow night, sunny \leftrightarrow rainy, and sunny \leftrightarrow cloudy.

(2) *KITTI object detection benchmark* [7] and *Cityscapes dataset* [4] are used to demonstrate that our method can help with domain adaptation. KITTI benchmark [7] contains 7,481 images for training and 7,518 images for testing, and it provides the bounding boxes for 6 object classes. Cityscapes dataset [4] is widely exploited for semantic segmentation, which consists of 5,000 images with pixel-level annotations for 30 classes. These datasets are used to conduct the domain adaptation for object detection (KITTI \rightarrow Cityscapes case). To integrate two datasets' the object classes, we set the common 4 object classes as person, car, truck, and bicycle. We build 3 memory items for each class, including 8 background memory items.

Compared methods. We perform the evaluation on the following methods.

- CycleGAN [49] and UNIT [23] are the typical unsupervised I2I translation methods.
- MUNIT [13] and DRIT [22] are the multi-modal unsupervised I2I translation methods that are extensions of CycleGAN [49] and UNIT [23]. Especially, we exploit DRIT [22] as our baseline model.
- INIT [38] and DUNIT [1] are the existing instance-level unsupervised I2I methods. These methods are compared only for quantitative evaluation and not included in the qualitative comparison, since their code (parameters) is not publicly available.

Evaluation metrics. Following the experimental protocol of existing unsupervised I2I translation methods, we evaluate our methods with Inception Score (IS) [37], Conditional Inception Score (CIS) [13], and LPIPS Metric [48].

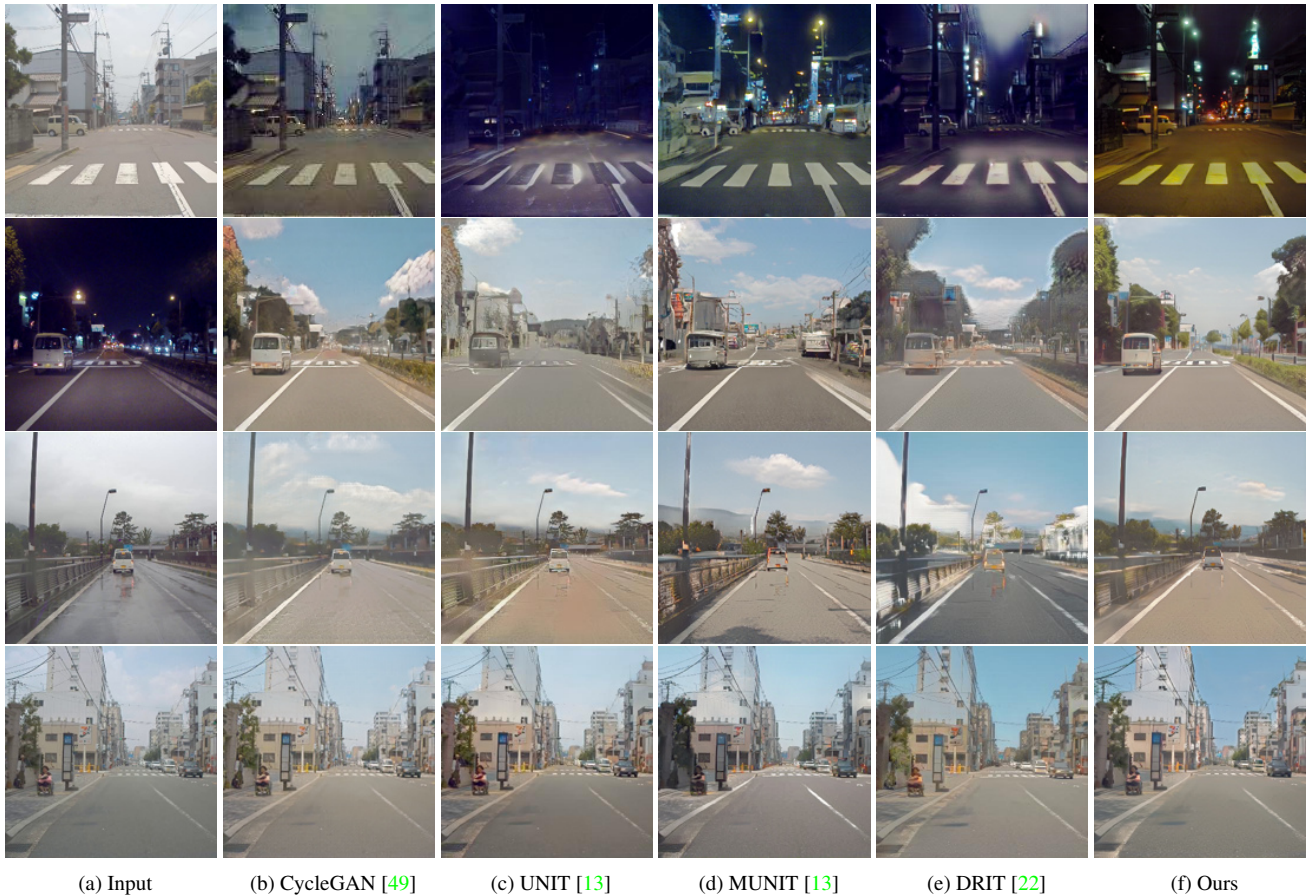


Figure 4: **Qualitative comparison of existing I2I translation methods.** (Top to bottom) sunny→night, night→sunny, rainy→sunny, and cloudy→sunny results. Our results preserve object details well and look realistic. Best viewed in color.

4.2. Comparison to state-of-the-art

Qualitative evaluation. Fig. 4 shows a qualitative comparison of the state-of-the-art methods. We observe that the multi-modal I2I methods MUNIT [13] and DRIT [22] fails to capture instance details and boundaries well. As these methods do not have any access to semantic information, they tend to translate instances to the other semantic styles (e.g., translating buildings into the sky). Our method produces the most visually appealing images with more vivid details. Thanks to the proposed class-aware memory network, it shows high capacity to better understand the semantic instances and employ the appropriate local style representation for object classes. We compare the translated results with instance-level I2I method DUNIT [1] in Fig. 5. Our result yields sharper and distinctive instances and more realistic images. Lastly, we visualize the multimodal translated results in Fig. 6. We use the stored key k in the memory and randomly sampled values (v^x, v^y) . It can be observed that the degree of color (e.g. road, sky) changes across these images.

User study. We conducted a user study to compare subjective quality of the translated results. For each translation,

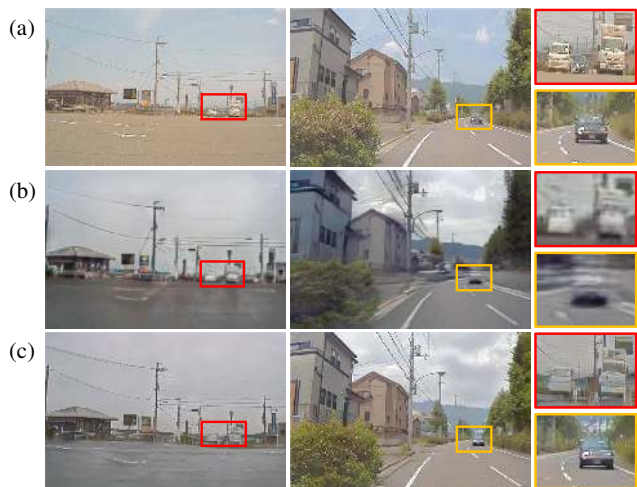


Figure 5: **Visual comparison of DUNIT [1].** (a) Input, (b) DUNIT, (c) Ours. We show the results for sunny→rainy in the first column and sunny→cloudy in the second column. Note that the results are taken from DUNIT paper.

	CycleGAN [49]		UNIT [23]		MUNIT [13]		DRIT [22]		INIT [38]		DUNIT [1]		Ours	
	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	1.176	1.271
night→sunny	0.012	1.023	0.027	1.024	1.036	1.051	1.024	1.099	1.045	1.080	1.083	1.108	1.115	1.130
sunny→rainy	0.011	1.073	0.097	1.075	1.012	1.146	1.007	1.207	1.036	1.152	1.029	1.225	1.092	1.213
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	1.052	1.218
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	1.136	1.489
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	1.112	1.254

Table 1: **Quantitative evaluation on INIT dataset [38].** We perform bidirectional translation for each domain pair. We measure CIS and IS (higher is better). Our results attain the best results.



Figure 6: **Results of multimodal image translation.** We use randomly sampled style values to generate (left) sunny image → (right) night images.

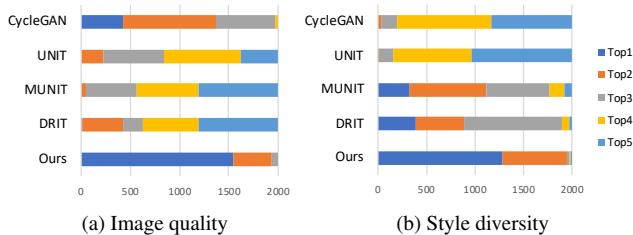


Figure 7: **User study results.** Our method is most preferred for image quality and style diversity both.

we randomly select 10 images from INIT validation to set up a total of 80 images for comparison. From 25 participants, we asked to rank all the methods in terms of the image quality and style diversity of the translated image, and we received a total of 2,000 votes. Fig. 7 shows the results, and our method ranks first in 77.9% for the image quality and 64.5% for the style diversity.

Quantitative evaluation. Table 1 shows the IS [37] and CIS [13], and Table 2 shows average LPIPS metric [48]. The IS measures the diversity of output images based on the Inception V3 model [41]. The CIS quantifies the quality and diversity of output conditioned on a single image. Additionally, the LPIPS metric [48] measures the translation diversity by calculating the similarity between two different deep features from the pre-trained AlexNet [19]. The results indicate significant performance gains with our method in all metrics. It further highlights the contribution of class-aware memory network to the improved performance.

Domain adaptation for object detection. We test our method for the domain adaptive object detection. Using

Method	sunny →night	sunny →rainy	sunny →cloudy	Average
CycleGAN [49]	0.016	0.008	0.011	0.012
UNIT [23]	0.067	0.062	0.068	0.066
MUNIT [13]	0.292	0.239	0.211	0.247
DRIT [22]	0.231	0.173	0.166	0.190
INIT [38]	0.330	0.267	0.224	0.274
DUNIT [1]	0.338	0.298	0.225	0.287
Ours	0.346	0.316	0.251	0.304
Real images	0.573	0.489	0.465	0.509

Table 2: **Quantitative evaluation with average LPIPS metric.** The LPIPS metric calculates the diversity scores.

Method	KITTI → Cityscapes				
	Pers.	Car	Truc.	Bic.	mAP
DT [14]	28.5	40.7	25.9	29.7	31.2
DAF [2]	39.2	40.2	25.7	48.9	38.5
DARL [17]	46.4	58.7	27.0	49.1	45.3
DAOD [36]	47.3	59.1	28.3	49.6	46.1
DUNIT w/o IC [1]	56.2	59.5	24.9	48.2	47.2
DUNIT w/ IC [1]	60.7	65.1	32.7	57.7	54.1
Ours	58.3	68.2	33.4	58.4	54.6

Table 3: **Quantitative results for domain adaptive detection.** We report per-class AP for KITTI→Cityscapes case.

the Faster-RCNN [35] trained on images in the source domain, we evaluate the detection performance of the translated images from source to target domain. Following DUNIT [1], we conduct experiments on the KITTI object detection benchmark [7] as the source domain and Cityscapes dataset [4] as the target domain. We compare the performance to state-of-the-art domain adaptation methods [14, 2, 17, 36] and DUNIT [1] with instance consistency loss (w/ IC) and without (w/o IC). Note that the instance consistency loss enforces the consistency constraints between results detected from original and translated image. We report the mean average precision (mAP) for the detected objects in Tab. 3. Our method performed well in the domain adaptive object detection tasks without explicitly using the object detection network. Unlike DUNIT [1] that improves performance by applying direct constraints on detected results, our method can recognize the semantic information contained in images thanks to our highly dis-



Figure 8: **Qualitative evaluation for ablation study.** Our full configuration with class-aware memory and contrastive loss produces a realistic and well-preserved image.

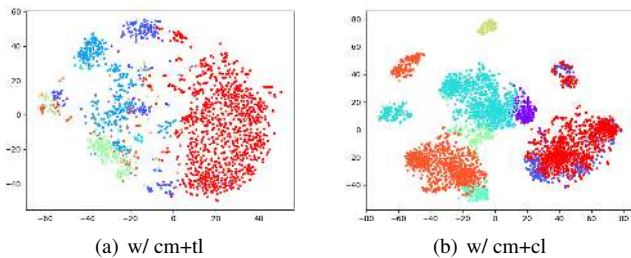


Figure 9: **t-SNE visualization for the content features.** The same colored points indicate the content features addressed to the same memory item.

criminative class-aware memory network. Consequently, it allows the image that are translated into an appropriate object style while preserving its inherent semantic information. Furthermore, it demonstrates that our method can realize more complex domain adaptation tasks.

4.3. Ablation study

We examine the impact of i) single memory (*sm*) without considering object class vs. class-aware memory (*cm*) and ii) feature triplet loss (*tl*) vs. feature contrastive loss (*cl*). We conduct the ablation studies on sunny \leftrightarrow night case in INIT [38], which is apt to show the effectiveness of individual components. We compare the results of 4 cases; (a) w/ single memory + triplet loss, (b) w/ single memory + contrastive loss, (c) w/ class-aware memory + triplet loss, and (d) w/ class-aware memory + contrastive loss. The qualitative and quantitative results are shown in Fig. 8 and Table 4.

Effectiveness of class-aware memory. The results using the single memory (in Figure 8 (b), (c)) cannot preserve the instance boundaries well, and even small instances disappear into the background. On the other hand, the results using the class-aware memory (in Figure 8 (d), (e)) show clear and well-preserved instance structures. The quantitative results from Table 4 also indicate that the translated images using the class-aware memory are more realistic.

Method	sunny \rightarrow night			night \rightarrow sunny	
	LPIPS	CIS	IS	CIS	IS
w/ sm+tl	0.287	1.061	1.189	1.037	1.080
w/ sm+cl	0.310	1.094	1.206	1.062	1.107
w/ cm+tl	0.328	1.156	1.253	1.101	1.103
w/ cm+cl	0.346	1.176	1.271	1.115	1.130

Table 4: **Ablation study on memory types and memory losses.** Our full configuration shows the best performance.

Effectiveness of feature contrastive loss. We observe that the results using the feature contrastive loss (in Figure 8 (c), (e)) are more vivid and represent a style that is appropriate for each instance compared to the results using the feature triplet loss (in Figure 8 (b), (d)). To investigate its effect, we visualize the distribution of the content features, which are learned with the triplet loss in Fig. 9 (a) and with the contrastive loss in Fig. 9 (b). Specifically, we project the embedded content features from the test images into 2-dimensional space using t-SNE [24]. The color indicates the memory items, which means that the points with the same color are mapped to the same item. The contrastive loss is more effective in separating and clustering the feature semantically. Therefore, it enhances the diversity and discriminative power of our memory items.

5. Conclusion

We present an instance-level unsupervised image-to-image translation framework with a class-aware memory network. It consists of a set of key-values that store shared content and domain-specific style representations, used to explicitly reason style representations. To this end, we introduce feature contrastive loss to increase the diversity and discriminative power of our memory items. This allows obtainin object-preserved and high-quality translated outputs without the additional use of extra object detection modules. Extensive experiments show that our method achieves state-of-the-art performance.

References

- [1] Deblina Bhattacharjee, Seungryong Kim, Guillaume Vezier, and Mathieu Salzmann. Dunit: Detection-based unsupervised image-to-image translation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.
- [2] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.
- [3] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.
- [4] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.
- [5] Michał Daniłuk, Tim Rocktäschel, Johannes Welbl, and Sebastian Riedel. Frustratingly short attention spans in neural language modeling. In *Int. Conf. Learn. Represent.*, 2017.
- [6] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. In *Int. Conf. Learn. Represent.*, 2017.
- [7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2012.
- [8] Abel Gonzalez-Garcia, Joost Van De Weijer, and Yoshua Bengio. Image-to-image translation for cross-domain disentanglement. In *Adv. Neural Inform. Process. Syst.*, 2018.
- [9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Adv. Neural Inform. Process. Syst.*, 2014.
- [10] 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 *Int. Conf. Mach. Learn.*, 2018.
- [11] Sheng-Wei Huang, Che-Tsung Lin, Shu-Ping Chen, Yen-Yi Wu, Po-Hao Hsu, and Shang-Hong Lai. Auggan: Cross domain adaptation with gan-based data augmentation. In *Eur. Conf. Comput. Vis.*, 2018.
- [12] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Int. Conf. Comput. Vis.*, 2017.
- [13] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Eur. Conf. Comput. Vis.*, 2018.
- [14] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.
- [15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.
- [16] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *Int. Conf. Mach. Learn.*, 2017.
- [17] Taekyung Kim, Minki Jeong, Seunghyeon Kim, Seokeon Choi, and Changick Kim. Diversify and match: A domain adaptive representation learning paradigm for object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.
- [18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Int. Conf. Learn. Represent.*, 2015.
- [19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Adv. Neural Inform. Process. Syst.*, 2012.
- [20] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Roman Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *Int. Conf. Mach. Learn.*, 2016.
- [21] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.
- [22] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Eur. Conf. Comput. Vis.*, 2018.
- [23] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Adv. Neural Inform. Process. Syst.*, 2017.
- [24] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.
- [25] Giovanni Mariani, Florian Scheidegger, Roxana Istrate, Costas Bekas, and Cristiano Malossi. Bagan: Data augmentation with balancing gan. *arXiv preprint arXiv:1803.09655*, 2018.
- [26] Jiayu Miao, Yunchao Wei, and Yi Yang. Memory aggregation networks for efficient interactive video object segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.
- [27] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. In *EMNLP*, 2016.
- [28] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.
- [29] Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Instagan: Instance-aware image-to-image translation. In *Int. Conf. Learn. Represent.*, 2019.
- [30] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.
- [31] Seil Na, Sangho Lee, Jisung Kim, and Gunhee Kim. A read-write memory network for movie story understanding. In *Int. Conf. Comput. Vis.*, 2017.

- [32] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Int. Conf. Comput. Vis.*, 2019.
- [33] Hyunjong Park, Jongyoun Noh, and Bumsub Ham. Learning memory-guided normality for anomaly detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.
- [34] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.
- [35] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2016.
- [36] Adrian Lopez Rodriguez and Krystian Mikolajczyk. Domain adaptation for object detection via style consistency. *arXiv preprint arXiv:1911.10033*, 2019.
- [37] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Adv. Neural Inform. Process. Syst.*, 2016.
- [38] Zhiqiang Shen, Mingyang Huang, Jianping Shi, Xiangyang Xue, and Thomas S Huang. Towards instance-level image-to-image translation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.
- [39] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Int. Conf. Comput. Vis.*, 2015.
- [40] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Adv. Neural Inform. Process. Syst.*, 2015.
- [41] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.
- [42] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. In *Int. Conf. Learn. Represent.*, 2017.
- [43] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.
- [44] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. In *Int. Conf. Learn. Represent.*, 2015.
- [45] Dingdong Yang, Seunghoon Hong, Yunseok Jang, Tianchen Zhao, and Honglak Lee. Diversity-sensitive conditional generative adversarial networks. In *Int. Conf. Learn. Represent.*, 2019.
- [46] Tianyu Yang and Antoni B Chan. Learning dynamic memory networks for object tracking. In *Eur. Conf. Comput. Vis.*, 2018.
- [47] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *Int. Conf. Comput. Vis.*, 2017.
- [48] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.
- [49] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Int. Conf. Comput. Vis.*, 2017.
- [50] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Adv. Neural Inform. Process. Syst.*, 2017.
- [51] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.