Background-Focused Contrastive Learning for Unpaired Image-to-Image Translation

Mingwen Shao (mwshao@upc.edu.cn)
Minggui Han
Lingzhuang Meng
Fukang Liu

Research Article

Keywords: Image-to-image translation, Contrastive learning, Generative adversarial networks, Heat map

Posted Date: January 11th, 2024

DOI: https://doi.org/10.21203/rs.3.rs-3833664/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.
Background-Focused Contrastive Learning for Unpaired Image-to-Image Translation

Mingwen Shao¹,*, Minggui Han¹, Lingzhuang Meng¹, Fukang Liu¹

¹China University of Petroleum (East China), Qingdao Institute of Software, College of Computer Science and Technology, Qingdao, China, 266580.
*Corresponding author.

Contributing authors: mwshao@upc.edu.cn; s21070053@s.upc.edu.cn; lzhmeng1688@163.com; upcaishu@163.com;

Abstract

Contrastive learning for Unpaired image-to-image Translation (CUT) aims to learn a mapping from source to target domain with an unpaired dataset, which combines contrastive loss to maximize the mutual information between real and generated images. However, the existing CUT-based methods exhibit unsatisfactory visual quality due to the wrong locating of objects and backgrounds, particularly where it incorrectly transforms the background to match the object pattern in layout-changing datasets. To alleviate the issue, we present Background-Focused Contrastive learning for Unpaired image-to-image Translation (BFCUT) to improve the background's consistency between real and its generated images. Specifically, we first generate heat maps to explicitly locate the objects and backgrounds for subsequent contrastive loss and global background similarity loss. Then, the representative queries of objects and backgrounds rather than randomly sampling queries are selected for contrastive loss to promote reality of objects and maintenance of backgrounds. Meanwhile, global semantic vectors with less object information are extracted with the help of heat maps, and we further align the vectors of real images and their corresponding generated images to promote the maintenance of the backgrounds in global background similarity loss. Our BFCUT alleviates the wrong translation of backgrounds and generates more realistic images. Extensive experiments on three datasets demonstrate better quantitative results and qualitative visual effects.

Keywords: Image-to-image translation, Contrastive learning, Generative adversarial networks, Heat map

1 Introduction

The Image-to-Image (I2I) translation tasks are designed to transform the content of images from the source to the target domain, encompassing scenarios like translating images from horse to zebra [7, 19, 26, 42], label to photography [1, 30, 33, 57], and transitioning between rainy and clean images [5, 32, 44, 51, 56], among others. Recently, I2I translation has gathered significant
attention owing to its ability to produce realistic results and its wide applicability across various tasks, including deraining [5], dehazing [4, 31], and super-resolution [24, 45, 58].

Existing I2I translation methods [28, 34, 47, 52, 59] traditionally rely on paired datasets containing images from both the source and target domains to train the network for translating images from the source to target domain. For instance, Larsson et al. [28] and Zhang et al. [59] treat colorization as either a regression or classification task, constraining generated image pixels with paired labels. Long et al. [34] focus on pixel-wise classification utilizing paired human-annotated labels for semantic segmentation. However, it is challenging and demanding to collect paired datasets in real-world scenarios. Thus, there has been a shift towards unpaired I2I translation tasks that aim to learn the translation between the source and target domain without relying on paired images, which eliminates the requirement for paired data, making the training process more accessible and scalable.

To relieve the dependence on paired datasets, people adopt the generative-adversarial idea to guide the generator in producing realistic images in the target domain and present some methods [57, 62]. For instance, CycleGAN [62] introduces cycle-consistency loss to maintain content consistency between source and target images. However, the overly restrictive implicit one-to-one assumption in the cycle-consistency loss leads to monotonous results, especially when dealing with two domains containing diverse paired objects. In pursuit of enhanced image diversity, CUT [39] innovatively substitutes patch-wise contrastive learning for the cycle-consistency loss. The approach effectively enforces structure and texture consistency between the two domains, resulting in improved translation performance. Expanding on the framework established by CUT, several subsequent works [14, 15, 19, 48] aim to enhance its performance by introducing various modifications, such as changes to keys, queries, and other components. For example, QS-Attn [19] calculates the attention map of feature map pixels to identify distinct queries with less similar keys. NEGCUT [48] generates a set of hard negative keys for enhancing the effectiveness of negative keys. MCL [14] contrasts the results of the discriminator to mitigate overfitting, thereby strengthening the performance of both the generator and discriminator.

Despite the aforementioned improvements being proposed, there are still limitations in two aspects of the existing works. Firstly, several CUT-based works simply randomly sample patches as the queries, ignoring the different important degrees for different patches, which leads to paying excessive attention to color-monotonous, innocuous patches while neglecting
elaborate, significant patches. As indicated by the blue points in the “Queries” of Figure 1(a), the model focuses on too many innocuous points to accurately learn key features. Secondly, existing works inaccurately recognize the backgrounds in some cases, leading to wrongly translating the backgrounds to objects, as shown in the “Outputs” of Figure 1(a). In general, the premise of valid transformation is accurate positioning, which may struggle on some “layout-changing” datasets, where objects have significantly different sizes and locations in the images, as illustrated by the “Input” in Figure 1. Therefore, it is necessary to distinguish and learn the significant patches and accurately locate objects and backgrounds to translate images to the target domain.

Based on the above analysis, to relieve the problem of indiscreet query selection and unreliable background localization, we propose Background-Focused Contrastive learning for Unpaired image-to-image Translation (BFCUT) to generate images with realistic backgrounds and better visual quality. Specifically, to precisely distinguish between objects and backgrounds, we leverage a pre-trained model of contrastive learning to calculate a heat map with the help of Squeeze-and-Excitation (SE) [18] module that learns to weight channels about objects, as shown in the “Heat maps” of Figure 1(b). Then we identify the pixels that more tend to be undisputed objects or backgrounds in the heat map and utilize these pixels as queries for the patch-wise contrastive loss. The step enhances the learning process for the translation of both objects and backgrounds. Simultaneously, we process the feature maps of the real and corresponding fake image with the heat map and compress the results, to

Fig. 1 Layout-changing datasets. Objects in certain datasets exhibit diverse sizes and layouts, making some objects challenging to be recognized. (a) The evenly distributed queries and the generated images of CUT [39]. (b) Our heat maps, selected queries and the generated images.
produce two global semantic vectors that emphasize the background in both real and fake images.

Then, we pull the vectors close to promote the maintenance of backgrounds. Ultimately, we introduce perceptual loss [22] in the model to enhance the visual performance of the fake images and the loss contributes to an overall improvement in fake images.

Our contributions are summarized as follows:

• We present BFCUT to enhance the visual quality of generated images that have more realistic backgrounds and objects in I2I translation tasks.

• Representative pixels of objects and backgrounds are selected as queries for patch-wise contrastive learning to reinforce the model’s ability to learn objects and backgrounds more effectively.

• We align global semantic vectors to facilitate the accurate localization of objects and backgrounds, which mitigates erroneous locations and translations of the backgrounds.

• Extensive experiment results show the effectiveness of our BFCUT and superior performance compared to the baseline.

2 Related work

We present an overview of related works in contrastive learning, I2I translation, and heat map in this section.

2.1 Contrastive learning

Contrastive learning has achieved considerable success in recent years, particularly in the domain of unsupervised learning [2, 3, 6, 10, 17, 29, 36, 49, 50, 54, 55]. The fundamental concept behind contrastive learning involves providing a query sample $q$ and a set of key samples $\{k\}$, comprising a positive key $k_+$ and some negative keys $\{k\}_-$. The network is then tasked with distinguishing $k_+$ from $\{k\}_-$ through contrastive loss, typically employing the InfoNCE loss [37], as shown in Eq. (1). The primary objective is to pull positive pairs close while pushing negative pairs apart in feature space.

$$l_{NCE}(q, k_+, \{k\}_-) = -\log \frac{\exp(q \cdot k_+/\tau)}{\exp(q \cdot k_+/\tau) + \sum_{k \in \{k\}_-} \exp(q \cdot k/\tau)}, \quad (1)$$

where $\tau$ is the temperature hyper-parameter and set to 0.07.

Some recent works [3, 6, 10, 17, 29, 55] leverage instance-wise contrastive loss, which treats feature vectors of entire images as samples. For example, MoCo [17] considers feature vectors from different augmented views of the same image as positive sample pairs, accompanied by a large-sized queue of negative keys. PLC [29] introduces prototypes
of images to learn the hierarchical semantic structure of the dataset. Moreover, SimCLR [3] demonstrates that introducing a nonlinear transformation between the encoder and the loss enhances the encoder’s capabilities by preserving valuable information for downstream tasks. In contrast, some works [2, 36, 49, 50, 54] adopt pixel-wise contrastive loss, selecting pixels in feature maps as samples for contrastive loss. For example, PixPro [54] and VADeR [36] choose negative keys based on their spatial location in the image, while SetSim [50] and DenseCLIP [49] select pixels that are semantically similar to queries. Additionally, several pixel-wise methods [2, 49, 54] incorporate instance-wise contrastive loss to align global information across the entire images. These studies demonstrate that instance-wise contrastive loss and pixel-wise contrastive loss complement each other. Combining these two losses can improve the performance of pixel-wise downstream tasks, like semantic segmentation. Building on these findings, we introduce the global background similarity loss to improve pixel-wise tasks in I2I translation. Simultaneously, we adjust the global semantic vectors to reduce their inclusion of object-related information, allowing us to protect the backgrounds from wrong translations.

2.2 Image-to-image translation

I2I translation aims to translate images from the source to the target domain while maintaining the content of the source images. This translation is typically achieved through the use of GAN [13], which engages in adversarial training between a generator and a discriminator, taking advantage of their ability to model complex high-dimensional distributions. Some earlier GAN-based approaches [21, 38, 46] incorporate reconstruction loss using paired datasets to enhance the quality of generated images. For instance, Pix2Pix [21] leverages L1 loss to produce accurate images, minimizing blur in the generated outputs. Other methods [38, 46] share similar processes, utilizing paired labels to constrain the generated outputs. However, due to the challenges and sometimes impracticality of collecting paired images, some approaches [9, 20, 23, 62] have introduced unpaired I2I translation methods based on the concept of cycle-consistency [62]. These methods perform training with both-way mappings simultaneously: an image is translated from the source to the target domain and then back, enforcing the constraint that the generated image and its original image must be pixel-wise identical. This implies a one-to-one mapping between the target domain and the source domain. However, the strictness of this assumption may limit the network’s ability to learn a more effective mapping.

To overcome the aforementioned limitations of the strict assumption, CUT [39] innovatively...
replaces the cycle-consistency [62] loss by patchwise contrastive loss (same as the pixel-wise contrastive loss in contrastive learning [6, 10, 17]), which aims to maximize the mutual information between real and generated images. CUT extracts feature maps from different layers of a generator and randomly samples pixels from the feature maps to act as queries and keys in calculating the contrastive loss.

Based on CUT [39], DCLGAN [15] adopts a dual learning framework that employs two separate generators for precise feature extraction and decoding in each domain. Drawing inspiration from CUT, NEGCUT [48] generates hard negative keys using negative generators for contrastive loss. In a different approach, Santa [53] views the source and the target domain as a continuous domain, generating images along the shortest path. QS-Attn [19] enhances relation strength among patches in real and generated images by selecting notable queries through an attention matrix in a query-key manner. In our method, we choose a subset of queries based on the weighted heat map to prioritize representative pixels of objects and backgrounds for more focused attention on noticeable patches. Meanwhile, in most methods [15, 19, 39, 48, 53], the contrastive loss focuses solely on pixel-wise features, overlooking the influence of global information. To enhance training efficiency and performance, we introduce a global background similarity loss with the restrained objects to improve the consistency of backgrounds and overall visual performance, which supplements global semantic information by combining with instance-wise contrastive loss.

### 2.3 Heat map

Zhou et al. [61] first adopt the Class Activation Map (CAM) heat map for a specific category to achieve discriminative object localization. The CAM heat map is calculated by the weighted summing of channels in feature maps, which obtains weights about the selected category from the classifier. A classifier-independent approach, proposed by N. Komodakis and S. Zagoruyko [27], computes a heat map by summing up or averaging the last feature map of the encoder along channels, which has been discussed for its effectiveness.

Many works utilize this method to get a heat map and employ the heat map to locate objects and backgrounds. For instance, J. Choe and H. Shim [8] learn to classify images after removing the pixel with the maximum value in the heat map, which reduces reliance on specific pixels during the classification process. Peng et al. [40] utilize the channel-wise summing heat map to determine the object’s location and crop patches around the object. Similarly, Wang et al. [50] eliminate noisy backgrounds by leveraging channel-wise summing heat maps and contrasting pixel sets of objects.

Different from the above works, our I2I translation task involves only one category during the
training-testing-applying period, thus the weights of the channels remain similar across different images from the same domain. To obtain the weights, we integrate a trainable SE module after a pre-trained model of contrastive learning. The SE module enables our network to learn the weights of the feature map’s channels for improving the accuracy of the heat map.

3 Method

In this section, we introduce the pipeline and architecture of our BFCUT and then delve into the details.

3.1 Architecture

The overall architecture of our framework is illustrated in Figure 2. Given two domains $\mathcal{X} \subset \mathbb{R}^{H \times W \times C}$ and $\mathcal{Y} \subset \mathbb{R}^{H \times W \times C}$, along with unpaired instances $X = \{x \in \mathcal{X}\}$ and $Y = \{y \in \mathcal{Y}\}$, we aim to learn the mapping $G : \mathcal{X} \rightarrow \mathcal{Y}$. Here, $X$ and $Y$ represent the image sets of the source and target domains, respectively. Each $x$ is an image from domain $\mathcal{X}$, similarly for $y$.

As depicted in Figure 2, our BFCUT comprises two branches. The top half is the heat map generating network including a pre-trained encoder $Enc$ and a trainable $SE$ module. In the bottom half, we have the image translating network including the generator $G$, discriminator $D$, and several projectors denoted as $H$.

For the heat map generating network, the encoder $Enc$ produces a feature map containing both spatial and semantic information. Subsequently, the $SE$ module learns to weight the channels of the feature map, facilitating precise recognition of class-specific objects in the source domain. Next, we get the activation-based spatial heat map $M$ by averaging the feature map processed by $SE$ along the channel dimension. Finally, we reverse it by subtraction to get the reversed heat map $M_r$ that highlights backgrounds.

For the image translating network, the generator $G$ is responsible for translating images from domain $\mathcal{X}$ to domain $\mathcal{Y}$, while discriminator $D$ determines whether the images belong to the corresponding domains. Regarding the generator $G$, the first half (encoder $G_{enc}$) encodes features, and the second half (decoder $G_{dec}$) generates fake images based on these features. The projectors $H$ are two-layer MLPs, projecting the feature vectors to enhance the performance of correlative patch-wise contrastive loss and global background similarity loss functions.

As illustrated in the lower half of Figure 2, considering a real image $x$ from domain $\mathcal{X}$ and its corresponding fake image $\hat{y}$ generated by $G$ in domain $\mathcal{Y}$, we obtain two stacks of feature maps $\{F^l\}_L$ and $\{\hat{F}^l\}_L$ by encoding $x$ and $\hat{y}$ using $G_{enc}^l$, i.e. $F^l_L = \{C_{enc}^l(x)\}_L$. Here, $l \in 1, 2, \ldots, L$ represents the index of the layers in the encoder, and...
Fig. 2 The architecture of our model. BFCUT consists of several key components. The pre-trained encoder, denoted as $\text{Enc}$, processes the real image to generate a feature map. Subsequent modules further process this feature map to produce a reversed heat map. In the lower half, the generator $G$ translates real images to fake images, and the discriminator $D$ distinguishes between real and fake images. Feature maps are extracted from $G_{\text{enc}}$ and utilized for NCE and Glo Loss. These $H$ are projectors that make vectors more suitable for losses. The term “real $X$” denotes images from the given dataset, while “fake $Y$” refers to the generated images in domain $\mathcal{Y}$. The $G_{\text{enc}}^l$ represents the first $l$ layers of the encoder.

Based on the method of queries of maximums and minimums selecting explained in Section 3.3, we choose some pixels of feature maps as the queries and keys of contrastive loss, in which some pixels are randomly selected while others are carefully selected according to refined heat map $M$. All the selected pixels are then projected by $H^l$ to obtain two stacks of feature vectors $\{z^l\}$ and $\{\hat{z}^l\}$, where $z^l = H^l(G_{\text{enc}}^l(x))$ and $z^l \in \mathbb{R}^{C_l}$. The $\{\hat{z}^l\}$ serve as queries while the $\{z^l\}$ are regarded for keys in the patch-wise contrastive loss. Subsequently, as outlined in the global background similarity loss explained in Section 3.4, the deepest feature maps of $G_{\text{enc}}$ are weighted by reversed heat map $M_r$ and then processed through Global Average Pooling (GAP) and $H^g$. This process yields the extraction of global semantic vectors $g$ and $\hat{g}$, represented as $g = H^g(G_{\text{enc}}(x))$, where $G_{\text{enc}}$ represents the operation of averaging the
feature map along the spatial dimensions (i.e., width and height), and \( \odot \) represents element-wise multiplication and \( g \in \mathbb{R}^{C_g} \). The global background similarity loss then enforces the constraint that \( g \) and \( \hat{g} \) should be similar.

In the subsequent subsections, we will delve into the details of BFCUT and elaborate on its processes in three parts.

### 3.2 Heat map generating

As shown in the upper half of Figure 2, the heat map generating network comprises \( Enc \) and \( SE \).

We generate the heat map by a ResNet-50 \( Enc \) that is pre-trained with a method of contrastive learning \[6\] and frozen during training, along with a trainable \( SE \) module. The pre-trained module significantly reduces training time and memory requirements.

Specifically, a feature map is first generated by \( Enc \), then the trainable \( SE \) module is employed to learn weights that enhance the channels associated with objects and diminish the channels related to the backgrounds. The refined heat map \( M \) is obtained through GAP as follows:

\[
M = GAP_C(SE(Enc(x))), \quad (2)
\]

where \( GAP_C() \) denotes global average pooling along the channel dimension and \( M \in \mathbb{R}^{H_m \times W_m} \).

The refined heat map \( M \) provides a clearer indication of the object and background in images from the source domain. To constrain the objects in the feature map during the global background similarity loss, we reverse the heat map by subtracting the heat map from its maximum, as follows:

\[
M_r = Max(M) - M + \epsilon, \quad (3)
\]

where \( Max() \) signifies obtaining the maximum across the entire refined heat map, and \( \epsilon \) is a small number introduced to prevent the occurrence of zero. The reversed heat map \( M_r \) places greater emphasis on the backgrounds. Typically, the size of \( M_r \) is \( 8 \times 8 \) for an input of \( 256 \times 256 \), which is smaller than the stack of feature maps \( \{\hat{F}^l\}_L \).

### 3.3 Contrastive loss with representative queries

For CUT \[39\], the network randomly samples \( S \) pixels in each selected layer \( \hat{F}^l \) as queries. The pixel at the corresponding position in \( F^l \) serves as the positive key for each query, while the other \( S-1 \) pixels act as negative keys. Building upon this, we enhance the contrastive loss by replacing \( 2N \) of the randomly selected \( S \) pixels with \( 2N \) carefully selected pixels.

Specifically, we begin by extracting the feature maps \( F^l \) and \( \hat{F}^l \) from the \( l \)-th layer, which are larger than the refined heat map. Next, to emphasize pixels related to objects and backgrounds in the contrastive loss, we propose the
Fig. 3 The refined heat map along with the reddish carefully selected (i.e. “Picked” in the figure) queries and the blue randomly selected (i.e. “Sampled” in the figure) queries. In the carefully selected queries, the red circles correspond to the maximums, and the pink octagons represent the minimums in the refined heat map. These carefully selected queries demonstrate the accurate localization of objects and backgrounds.

The method named queries of maximums and minimums selecting. In the method, we select the first \( N \) maximums and the first \( N \) minimums pixels from the refined heat map, these selected pixels more tend to be the doubtless objects or backgrounds. We then magnify the smaller heat map to match the larger feature maps, ensuring the correct positioning of corresponding pixels.

To prevent aggregation around the first minimum and maximum pixels, we adopt the approach of first selecting and then magnifying, rather than the inverse method of first magnifying and then selecting. Subsequently, we randomly sample \( 2N \) pixels from the feature maps, which are evenly distributed across the entire feature map. These \( S \) pairs of pixel are then projected by \( H^l \) to obtain sets of feature vectors \( \{z^l\} \) and \( \{\hat{z}^l\} \in \mathbb{R}^{S \times C_l} \).

Finally, we apply the contrastive loss given by Eq. (1) to each query. Our overall patch-wise contrastive loss is then formulated as follows:

\[
L_{NCE}(G, H, X) = \mathbb{E}_{x \sim X} \sum_{l=1}^{L} \sum_{s=1}^{S} L_{NCE}(z^l_s, \hat{z}^l_s, \{z^l\}_{S-s}),
\]

where \( \hat{z}^l_s \) and \( z^l_s \) denote the \( s \)-th query and its positive key in the \( l \)-th layer, and \( \{z^l\}_{S-s} \) represents the other \( S-1 \) selected negative keys for \( \hat{z}^l_s \).

We select the queries of maximums that correspond to the objects to focus more attention...
on the objects and choose the queries of minimums that are located on the backgrounds to ensure unchanged backgrounds in the generated images. The results of queries of maximums and minimums selecting are presented in the third row of Figure 3. As depicted in this figure, the selected maximum pixels (red circles) are located on the objects, the selected minimum pixels (pink octagons) are situated on the backgrounds, and the randomly selected pixels (blue dots) are evenly distributed across the images. The aligned distribution of selected pixels is indicative of the method of first selecting and then magnifying.

3.4 Global background similarity loss

For the global background similarity loss, we contrast vectors containing global semantic information from the deepest feature maps. We introduce the loss with dual objectives: one is to enhance the performance of the pixel-wise contrastive loss, while the other is to improve the consistency of the backgrounds between real and its fake images.

The encoder of the generator extracts common features from both domains while eliminating domain-specific features. In a sense, the feature maps of the real image and its generated image represent that an input undergoes two different data augmentations, implying the process of contrastive learning. Thus, the complementary nature of instance-wise and pixel-wise contrastive losses also applies to our method. To obtain suitable global semantic vectors with less object information for the instance-wise contrastive loss, we mask the deepest feature maps of the encoder with the reversed heat map $M_r$ and apply GAP along the spatial dimension to the feature maps. However, despite the encoder’s efforts to eliminate domain-specific features, the object information unavoidably lingers at corresponding pixels in the feature map. So we utilize the projectors $H^g$ to calculate $g$ and $\hat{g}$ after GAP for further filtration, i.e., $g = H^g(\text{GAP}_S(M_r \odot G_{\text{enc}}(x)))$. Subsequently, we enforce their similarity by the instance-wise contrastive loss, as shown in Eq. (5). The negative keys are sourced from the negative queue, consisting of global semantic vectors from other real images generated in previous iterations.

$$L_{\text{Glo}}(G, H^g, X) = \mathbb{E}_{x \sim X} \text{NCE}(\hat{g}; g, \{g\}_{\text{Queue}}),$$

(5)

where $\{g\}_{\text{Queue}}$ represents the queue of negative keys, containing 256 vectors. The $H^g$ filters out unnecessary information that might be useful for the decoders but is not relevant for the global background similarity loss, which ensures that sufficient information is retained for the translation task.
3.5 Total loss

We introduce perceptual loss [22] to enhance the visual quality, which measures the dissimilarity between the features of real and corresponding fake images, expressed as follows:

\[ L_{Per}(G, X) = \mathbb{E}_{x \sim X} \text{MSE}(VGG_{3,8,15}(G(x)), VGG_{3,8,15}(x)), \]  

(6)

where \( \text{MSE}() \) denotes the Mean Squared Error (MSE) loss, and \( VGG_{3,8,15} \) refers to the 3rd, 8th, and 15th feature maps of VGG16 [41], pre-trained on ImageNet [11].

Similar to CUT [39], we employ adversarial loss [12] to guide the generator in translating realistic images that align with the distribution of the target domain:

\[ L_{GAN}(G, D, X, Y) = \mathbb{E}_{x \sim X} \text{log}(1 - D(G(x))) + \mathbb{E}_{y \sim Y} \text{log}(D(y)). \]  

(7)

The total loss is the weighted sum of the aforementioned losses as follows:

\[ L_{all}(G, H, X, Y) = \lambda_{NCE} L_{NCE}(G, H, X) + \lambda_{Glo} L_{Glo}(G, H^g, X) + \lambda_{Per} L_{Per}(G, X) + \lambda_{GAN} L_{GAN}(G, D, X, Y) + \lambda_{idt} L_{idt}(G, H, Y), \]  

(9)

where those \( \lambda \)s are trade-off hyper-parameters that balance different losses, and we set \( \lambda_{NCE} = 1 \), \( \lambda_{Glo} = 1 \), \( \lambda_{Per} = 0.1 \), \( \lambda_{GAN} = 1 \) and \( \lambda_{idt} = 1 \) by default.

4 Experiments

In this section, we introduce the experiment setup for our method and show the problems that FID may be misguided by the object-style pattern in some cases. Then, we show the effectiveness of our method through experimental results and delve into the impact of individual components.
Fig. 4 Misguided FID. We substitute a portion of the generated images of NEGCUT [48] with incorrectly translated images from other models [19, 39, 53] and measure the FID before and after substitution.

4.1 Experiment setup

Training Details. We follow most of the hyper-parameters of the existing method [39] to train our BFCUT. We utilize the LSGAN loss [35] for adversarial training of the generator and discriminator. Our network is trained for 400 epochs using the Adam optimizer [25] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$, unless otherwise specified. The initial learning rate is set to 0.0002 for the first half of the total epochs and then linearly decays to 0 in the residual epochs. We employ a ResNet-based [16] generator with a ResNet-based PatchGAN [21] as the discriminator. The batch size is set to 1, and instance normalization [43] is employed. All training and testing images are resized in $286 \times 286$ resolution and then cropped to $256 \times 256$. We select the $0^{th}$, $4^{th}$, $8^{th}$, $12^{th}$, and $16^{th}$ feature maps for patchwise contrastive loss and the $20^{th}$ feature map for global background similarity loss. The number of negative keys $S$ is set to 256, consistent with other CUT-based works [14, 15, 39, 48], while both the maximum and minimum numbers, $N$, are set to 8.

Datasets. We evaluate our BFCUT and baseline on three datasets:

(i) Horse $\rightarrow$ Zebra [62] is a subset of ImageNet [11] that exclusively comprises the horse and zebra classes in various scenes. The training set consists of 1334 zebra images and 1607 horse images, while the testing set comprises 140 zebra images and 120 horse images. We abbreviate this dataset as H$\rightarrow$Z in the following tables.

(ii) Apple $\rightarrow$ Orange [62] is another subset of ImageNet [11], focusing exclusively on the apple and orange classes in various scenes. The training set is formed with 1019 orange images and 995 apple images, while the testing set includes 248 orange images and 266 apple images. We abbreviate this dataset as A$\rightarrow$O in the following tables.

(iii) Cat $\rightarrow$ Dog is a subset of the AFHQ [9], including of 9892 high-quality face images of cats and dogs. The training set is split into 4739 dog images and 5153 cat images, while the testing set includes 500 dog images and 500 cat images.
Table 1 Quantitative comparisons of Horse $\rightarrow$ Zebra, Apple $\rightarrow$ Orange and Cat $\rightarrow$ Dog. The best results in each column are **bold**, while the second-best are *italic*. The "*" represents the operation of masking images with the binary images before the implementation of LPIPS.

<table>
<thead>
<tr>
<th>Method</th>
<th>FID↓</th>
<th>LPIPS*↓</th>
<th>sec/iter↓</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>53.5</td>
<td>141.2</td>
<td>34.6%</td>
<td>26.3%</td>
</tr>
<tr>
<td>CUT [39]</td>
<td>45.5</td>
<td>197.9</td>
<td>76.5</td>
<td>34.6%</td>
</tr>
<tr>
<td>FastCUT [39]</td>
<td>73.4</td>
<td>195.6</td>
<td>94.0</td>
<td>36.6%</td>
</tr>
<tr>
<td>NEGCUT [48]</td>
<td>39.6</td>
<td>187.8</td>
<td>55.9</td>
<td>43.3%</td>
</tr>
<tr>
<td>QS-attn [19]</td>
<td>41.1</td>
<td>176.2</td>
<td>72.8</td>
<td>39.9%</td>
</tr>
<tr>
<td>Santa [53]</td>
<td><strong>36.2</strong></td>
<td><strong>156.5</strong></td>
<td><strong>52.1</strong></td>
<td>48.9%</td>
</tr>
</tbody>
</table>

Due to the abundance of training images, we train our BFCUT on this dataset for 200 epochs. We abbreviate this dataset as $C \rightarrow D$ in the following tables.

We evaluate our model’s performance against other works on Horse $\rightarrow$ Zebra and Apple $\rightarrow$ Orange to validate its effectiveness in preserving background and enhancing visual quality in layout-changing datasets. Additionally, we present qualitative and quantitative results on Cat $\rightarrow$ Dog to demonstrate comparable outcomes in layout-unchanging datasets.

**Evaluation Metrics.** To quantitatively evaluate the visual quality of generated images, we utilize the Fréchet Inception Distance (FID) score, a widely-used metric in I2I translation tasks [9, 14, 15, 20, 23, 39, 48, 62]. FID measures the distribution distance between real images and the fake generated images of same domain in human perception. The lower FID suggests that the fake images are more realistic and visually similar to the real images. We also measure the Learned Perceptual Image Patch Similarity (LPIPS) [60] score between the real image and its fake image from different domains, which measures the perceptual distance between two images. Before the implementation of LPIPS, we mask the real and its generated images with the binary image to compare the fidelity of the backgrounds. The binary image is produced by a reversed heat map and has zero value in objects and one value in backgrounds. We denote it as LPIPS* in the tables to distinguish it from the normal implementation. A lower LPIPS* implies greater perceptual similarity of background.

### 4.2 Misguided FID

Conventional evaluation metrics like FID usually used to evaluate the overall visual quality of the dataset of generated images, may fail on certain images with obvious yet monotonous characteristics. For instance, images may be classified as zebra based on the presence of zebra stripes, with little attention to the content. To demonstrate this, we intentionally substitute a portion of the generated images from NEGCUT [48] with...
Fig. 5. Qualitative comparisons on Horse $\rightarrow$ Zebra. We compare the generated images of our BFCUT with other works, including CUT [39], FastCUT [39], NEGCUT [48], QS-Attn [19] and Santa [53]. BFCUT exhibits superior visual effects on both background and objects compared to the other methods.

incorrectly translated images from other models [19, 39, 53] to misguide FID, which proves that FID may be cheated by the inadvertently translated backgrounds, as shown in Figure 4. In this figure, we show a part of the substituted images and their corresponding generated images of NEGCUT, we can see that the FID is reduced after the substitution. To measure whether there is a better background, we mask the real and its fake images with the binary mask produced by heat maps and measure their LPIPS, which is denoted as LPIPS* to distinguish the normal LPIPS. The results are shown in the following tables, the lower
Fig. 6 Qualitative comparisons on Apple → Orange. We compare the generated images of our BFCUT with other works, including CUT [39], FastCUT [50], NEGCUT [48], QS-Attn [19] and Santa [53]. BFCUT exhibits superior visual effects on both background and objects compared to the other methods.

LPIPS* shows the effectiveness of our method in background maintenance.

4.3 Comparison with other works

We compare our approach with several existing methods, including CUT [39], FastCUT [39], NEGCUT [48], QS-Attn [19], and Santa [53], focusing on three distinct translation tasks: Horse → Zebra, Apple → Orange, and Cat → Dog. All of these methods are based on the CUT framework. Quantitative results are presented in Table 1, which shows that our BFCUT has relatively poor FID and the best LPIPS* in three tasks compared with other works, which implies the best
Fig. 7 Qualitative comparisons on Cat → Dog. We compare the generated images of our BFCUT with other works, including CUT [39], FastCUT [39], NEGCUT [48], QS-Attn [19] and Santa [53]. Our BFCUT demonstrates comparable visual results to other methods.

maintenance of background and accords with our observation that FID may be misguided in some situations. The qualitative results presented in Figures 5 and 6 demonstrate that our BFCUT generates images that seem more realistic compared to the outputs of other methods. Notably, BFCUT exhibits less distortion in the background across different layouts. Similar outputs with other methods are depicted in Figure 7, indicating that our approach does not compromise the visual quality of the layout-unchanging dataset. Additional images are compared in Appendix A for a more comprehensive demonstration of effectiveness.
4.4 Ablation experiments

Following the training settings, we conduct ablation experiments by forbidding components on the Horse → Zebra dataset. The results of the ablation study are presented in Table 2 and Figure 8, which explores the impact of each component.

In Table 2 and Figure 8, “Per” represents the perceptual loss $L_{Per}$, “Glo” represents the global background similarity loss $L_{Glo}$, “Qmm” represents the method of queries of maximums and minimums selecting in patch-wise contrastive loss $L_{NCE}$. From the table, it is evident that these components worsen the FID, indicating the lessening of wrongly translated backgrounds. And we conclude from the LPIPS* that “Glo” has the best effect in alleviating the wrongly translated background, and the combination of these components has the best overall impact. The results in Figure 8 show that the “Per” polishes up the fake images, and the “Glo” and “Qmm” help hold the
background and make the objects realistic. The combined efforts of all the components result in the best visual quality.

5 Conclusion

In this paper, we present Background-Focused contrastive learning for Unpaired image-to-image Translation (BFCUT) to maintain backgrounds and enhance the visual quality of generated images. Specifically, BFCUT first accurately locates objects and backgrounds for contrastive loss and global background similarity loss by generating the heat map with the heat map generating network. Subsequently, representative queries related to objects and backgrounds are selected for contrastive loss to generate lifelike objects and keep backgrounds from translation. Concurrently, we align the global semantic vectors of the real and its generated image in global background similarity loss to promote the maintenance of the backgrounds. Extensive experiments show the effectiveness of our BFCUT.

Acknowledgments. This work was supported by National Key Research and development Program of China (2021YFA1000102), and in part by the grants from the National Natural Science Foundation of China (Nos. 62376285, 62272375, 61673396), Natural Science Foundation of Shandong Province, China (No. ZR2022MF260).

Declarations

We have made our code publicly available at BFCUT. The experimental results can be replicated by training with the provided code. However, in some cases, it may be necessary to adjust certain hyperparameters according to the information provided in Section 4. Additionally, the experimental data of other works used for comparison comes from the corresponding papers as mentioned in the paper.

Appendix A More Contrastive Experiments

We show more qualitative experiments on three datasets in Figures A1, A2, and A3 respectively for further comparison.

References


[doi:10.1109/CVPR.2018.00917].


<table>
<thead>
<tr>
<th>Input</th>
<th>Ours</th>
<th>CUT</th>
<th>FastCUT</th>
<th>NEGCUT</th>
<th>QS-Attn</th>
<th>Santa</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Input Image" /></td>
<td><img src="image2" alt="Ours Image" /></td>
<td><img src="image3" alt="CUT Image" /></td>
<td><img src="image4" alt="FastCUT Image" /></td>
<td><img src="image5" alt="NEGCUT Image" /></td>
<td><img src="image6" alt="QS-Attn Image" /></td>
<td><img src="image7" alt="Santa Image" /></td>
</tr>
<tr>
<td><img src="image8" alt="Input Image" /></td>
<td><img src="image9" alt="Ours Image" /></td>
<td><img src="image10" alt="CUT Image" /></td>
<td><img src="image11" alt="FastCUT Image" /></td>
<td><img src="image12" alt="NEGCUT Image" /></td>
<td><img src="image13" alt="QS-Attn Image" /></td>
<td><img src="image14" alt="Santa Image" /></td>
</tr>
<tr>
<td><img src="image15" alt="Input Image" /></td>
<td><img src="image16" alt="Ours Image" /></td>
<td><img src="image17" alt="CUT Image" /></td>
<td><img src="image18" alt="FastCUT Image" /></td>
<td><img src="image19" alt="NEGCUT Image" /></td>
<td><img src="image20" alt="QS-Attn Image" /></td>
<td><img src="image21" alt="Santa Image" /></td>
</tr>
<tr>
<td><img src="image22" alt="Input Image" /></td>
<td><img src="image23" alt="Ours Image" /></td>
<td><img src="image24" alt="CUT Image" /></td>
<td><img src="image25" alt="FastCUT Image" /></td>
<td><img src="image26" alt="NEGCUT Image" /></td>
<td><img src="image27" alt="QS-Attn Image" /></td>
<td><img src="image28" alt="Santa Image" /></td>
</tr>
<tr>
<td><img src="image29" alt="Input Image" /></td>
<td><img src="image30" alt="Ours Image" /></td>
<td><img src="image31" alt="CUT Image" /></td>
<td><img src="image32" alt="FastCUT Image" /></td>
<td><img src="image33" alt="NEGCUT Image" /></td>
<td><img src="image34" alt="QS-Attn Image" /></td>
<td><img src="image35" alt="Santa Image" /></td>
</tr>
</tbody>
</table>

**Fig. A1** Qualitative results on Horse → Zebra. We show the generated images to compare our BFCUT with other works, including CUT [39], FastCUT [39], NEGCUT [48], QS-Attn [19] and Santa [53].
Fig. A2  Qualitative results on Apple $\rightarrow$ Orange. We show the generated images to compare our BFCUT with other works, including CUT [39], FastCUT [39], NEGCUT [48], QS-Attn [19] and Santa [53].
Fig. A3 Qualitative results on Cat $\rightarrow$ Dog. We show the generated images to compare our BFCUT with other works, including CUT [39], FastCUT [39], NEGCUT [48], QS-Attn [19] and Santa [53].