

Cascaded Cross MLP-Mixer GANs for Cross-View Image Translation

Bin Ren¹
bin.ren@unitn.it

Hao Tang²
hao.tang@vision.ee.ethz.ch

Nicu Sebe¹
niculae.sebe@unitn.it

¹ DISI
University of Trento

² Computer Vision Lab
ETH Zurich

Abstract

It is hard to generate an image at target view well for previous cross-view image translation methods that directly adopt a simple encoder-decoder or U-Net structure, especially for drastically different views and severe deformation cases. To ease this problem, we propose a novel two-stage framework with a new Cascaded Cross MLP-Mixer (CrossMLP) sub-network in the first stage and one refined pixel-level loss in the second stage. In the first stage, the CrossMLP sub-network learns the latent transformation cues between image code and semantic map code via our novel CrossMLP blocks. Then the coarse results are generated progressively under the guidance of those cues. Moreover, in the second stage, we design a refined pixel-level loss that eases the noisy semantic label problem with more reasonable regularization in a more compact fashion for better optimization. Extensive experimental results on Dayton [40] and CVUSA [42] datasets show that our method can generate significantly better results than state-of-the-art methods. The source code and trained models are available at <https://github.com/Amazingren/CrossMLP>.

1 Introduction

Cross-view image translation is a task that aims at synthesizing new images at the target view from the source view. It has achieved lots of interest due to it can be applied to city scene synthesis [48] and 3D object translation [44] from both the computer vision and virtual reality communities [12, 19, 25, 29, 33, 38, 42, 47, 48, 50]. This task is usually investigated by Convolutional Neural Networks (CNNs) based encoder-decoder structure [44, 48] or Generative Adversarial Networks (GANs) based methods [25]. However, these methods were all designed to generate new view images that share a degree of overlap on both appearances and geometry information with the original ones. Unlike previous methods mentioned above, Regmi *et al.* [29] proposed a conditional GAN which jointly learns the generation in both the image domain and its corresponding semantic domain, and the semantic predictions are further utilized to supervise the image generation. To further improve the performance, Tang *et al.* [33] proposed SelectionGAN, a coarse-to-fine method which built on three complicated U-Net generators and a selection module.

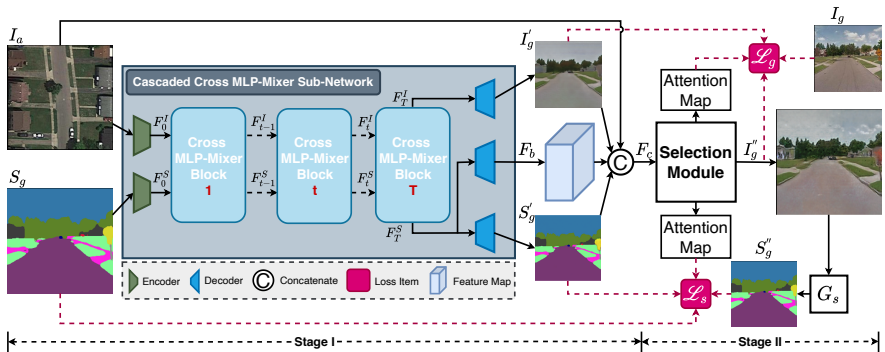


Figure 1: Overview of the proposed Cascaded Cross MLP-Mixer GANs.

Although these two methods performed interesting explorations, we observe that there are three main factors that can account for the reason why the quality of their generated images is still far from satisfactory. Firstly, previous methods generate the target view image just in an one-shot manner directly, which ignores some important details during the translation. Secondly, the interaction between the source view image and the target view semantic map lacks a deep exploration since it is critical for a model to learn a precise mapping pattern, especially for the case that there is a drastic gap between source view and target view in both appearance and geometry. Thirdly, the inevitable noisy semantic label problem needs to be further addressed since it will mislead the image generation because the semantic label maps are not accurate enough for all the pixels when they are usually produced from pretrained semantic models.

To this end, we propose a two-stage framework with a novel Cascaded Cross MLP-Mixer (CrossMLP for short) sub-network in the first stage (see Figure 1) and a refined pixel-level loss in the second stage. Within the CrossMLP sub-network, we stack several novel Cross MLP-Mixer (CrossMLP) blocks to produce a coarse image and a coarse semantic map at target view. This procedure is just like a human drawer is painting a picture [4], progressively. What's more, to dig out the latent translation cues between the source view image and the target view semantic map, we design the CrossMLP module (see Figure 2) within each CrossMLP block in the generator of a GAN. It interactively model the mutual latent long-range dependencies which consist of both the appearance and the structure information within each view, and the latent correspondences between two inputs can also be captured via its two consecutive multi-layer perceptrons (MLP). Then the latent cues are used to guide the transformation from source view to target view.

In the second stage, we propose the refined pixel-level loss. On one hand, it can ease the noisy semantic label problem by more reasonable regularization to the overall optimization loss. On the other hand, the formulation of our refined pixel-level loss becomes more compact with lower redundancy. Consequently, the generated images at target views become more realistic. Finally, extensive experimental results demonstrate that our method achieves new state-of-the-art results on both Dayton [40] and CVUSA [47] datasets.

The contributions of this paper are summarized as follows,

- We propose a two-stage framework that can progressively generate the target view image via the proposed CrossMLP blocks. To the best of our knowledge, the proposed model is the first MLP-based cross-view image translation framework.
- We propose a novel CrossMLP module to dig out the latent mapping cues between the source view image and the target view semantic map. The latent cues can guide the

cross-view translation with better geometry structure and appearance information.

- We propose a refined pixel-level loss, which can reduce the redundancy of loss items and ease the noisy semantic label problem in the cross-view translation task.
- Extensive experiments demonstrate the effectiveness of our method and show new state-of-the-art results on most of the evaluation metrics for both datasets.

2 Related Work

Generative Adversarial Networks (GANs). As one of the most used techniques of the generative model, GANs are a unique approach for learning desired data distribution to generate new samples [8]. Generally, there is a generator with a discriminator that exists in GANs. The former tries to produce photo-realistic target images to fool the latter, while the latter tries its best to figure out whether a sample is real or fake. Though lots of works have shown their capability in generating realistic-looking images [0, 16, 51], it is still challenging for vanilla GANs to produce images in a controlled manner. As an extension solution to this problem, conditional GANs (CGANs) are proposed by incorporating conditional information [8] such as class labels [6, 41, 46], attention maps [18, 57], human skeleton [34, 35, 51], and semantic maps [21, 22, 26, 52, 56]. In this paper, our work focuses on cross-view image generation with semantic maps as conditional information.

Image-to-Image Translation aims to learn a parametric mapping between input and output data. Isola *et al.* realized this by its supervised Pix2pix method to learn a translation function between input and results based on CGANs. Zhu *et al.* [49] introduced CycleGAN, which using the cycle-consistency loss to solve the unpaired image translation problem. In addition, there are lots of works like [0, 23, 24, 43] that all adopted attention mechanism to improve the generation performance. Moreover, based on the self-attention strategy, Transformer based methods have become popular relied on their ability for modeling global long-range dependencies in vision tasks compared to conventional CNN methods recently [14, 50]. However, due to Transformer is built on the self-attention mechanism, when the size of the feature map increases, the computing and memory overheads increase quadratically. To ease this problem, inspired by MLP-Mixer [49] which closely resembles the Transformer model in an efficient way and has shown its powerful ability on discriminative vision tasks [5]. We propose the novel CrossMLP in this paper which is based entirely on multi-layer perceptrons (MLPs) for both spatial locations and feature channels module to address the generative task. Therefore it can model the long-range dependencies via a more simple alternative in GAN.

Cross-View Transformation. Most existing works on cross-view transformation mainly focus on synthesizing novel views of the same object, such as chairs, tables, or cars [3, 4, 58]. Another group of works [45, 43] explore the cross-view scene image translation problem in which there exists a large degree of overlapping in both appearances and geometry structures. However, when facing the drastically different view and severe formation cases which involve huge geometry mismatch, large-scale uncertainty, and obvious appearance difference, it's really challenging for existing methods. Zhai *et al.* [47] firstly proposed a CNN-based network for generating panoramic ground-level images from aerial images at the location. Krishna *et al.* [49] proposed a GAN-based structure X-Fork and an X-Seq to address the aerial to street-view image translation via an extra semantic segmentation map. To improve the performance, Tang *et al.* [33] proposed SelectionGAN, a coarse-to-fine framework based on three U-Net generators and a Selection module. However, nearly all the methods mentioned above generate the results in a hard one-shot way, the final generated results are still

far from satisfactory due to the drastic difference between source and target views. Hence, we propose a novel progressive generation fashion via several cascaded CrossMLP blocks for this task. This strategy allows our method can interactively produce complex images at target view with much better generation quality step by step.

3 Cascaded Cross MLP-Mixer GANs

3.1 Overview

Notations. The overall structure of the proposed Cascaded Cross MLP-Mixer GAN is depicted in Figure 1. There are two stages in this framework. The first one takes as input both one image I_a at source view and one semantic map S_g at the target view, aiming to progressively produce a coarse image I'_g at target view, a coarse target semantic map S'_g , and a feature map F_b which contains the latent transformation cues between I_a and S_g . The second stage firstly takes F_c , a combination feature originated from I_a , I'_g , S'_g , and F_b , as input. Then F_c is sent to the selection module proposed by Tang *et al.* [B3], note that this module is just used for producing the final cross-view image I''_g and uncertainty maps U to multiple optimization losses, which is our secondary focus. Moreover, I''_g is pass through G_s , a U-net based structure, for recovering a more detailed semantic map S''_g . Finally, all image related items are used to construct the refined pixel-level image loss \mathcal{L}_g while all semantic maps related items are used to construct the refined pixel-level semantic map loss \mathcal{L}_s , through which a more fine-grained synthesized cross-view image can be produced.

Encoder and Decoder. As shown in Figure 1, the input image I_a from the source view and the semantic map S_g from one target view are firstly encoded by two similar feature extraction encoders that consist of N down-sampling convolutional layers ($N=2$ in our case). On the output side of our proposed cascaded MLP-Mixer generation, both the coarse output image I'_g and the coarse output semantic map S'_g at the target view are recovered by two similar decoders from feature F_T^I and F_T^S via N deconvolutional layers. What's more, for the intermediate feature map F_b , we recover it from F_T^S only via $N/2$ deconvolutional layers.

3.2 Cascaded Cross MLP-Mixer Sub-Network

As the core of the generator in the first stage, the cascaded cross MLP-Mixer sub-network is consisting of several CrossMLP blocks shown in Figure 1. Starting from the initial image code F_0^I and semantic map code F_0^S . The CrossMLP sub-network progressively updates these two codes through a sequence of CrossMLP blocks. Then the final image code F_T^I is taken to decode the coarse output image I'_g at target view. While the final semantic code F_T^S is token to decode for both the feature map F_b and the semantic map S'_g . All CrossMLP blocks have identical structures, and each carries out one step of update. Consider the t -th block depicted in Figure 2, whose inputs are F_{t-1}^I and F_{t-1}^S . Each block comprises two pathways, called the image pathway and the semantic map pathway. With deep interactions via CrossMLP module, these two pathways update F_{t-1}^I and F_{t-1}^S to F_t^I and F_t^S , respectively. In the following, we describe the detailed update process in three parts and justify their designs.

CrossMLP Module. From a basic level, the cross-view image translation is to moving patches from the location induced by the source view image to the location induced by the

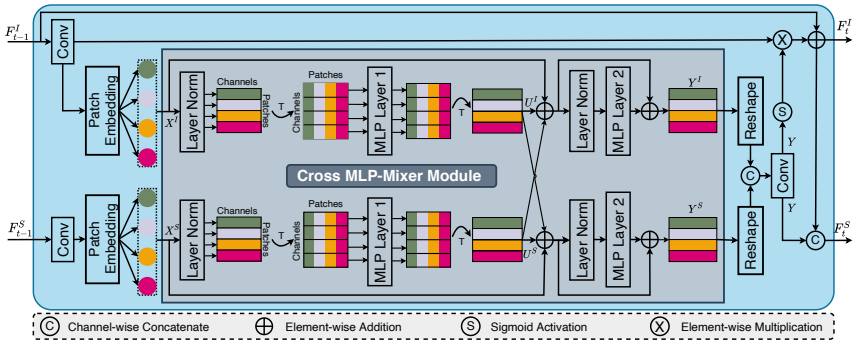


Figure 2: Architecture of the proposed Cross MLP-Mixer block.

target view semantic map. So how to let the translation knows the latent cues that where to sample source patches and where to put target patches is extremely significant. Therefore, we design the novel CrossMLP module for modeling the global long-range relation between source view image code and target view semantic map code to realize such cues.

Modern deep vision architectures generally consist of layers that mix features (i) at a given spatial location, (ii) between different spatial locations, or both at once [49]. CNNs realize this via their convolutional kernels and larger kernels can perform both. In Vision Transformers or other attention-based architecture, self-attention layers allow both (i) and (ii). Inspired by [49], we design the CrossMLP architecture which firstly conducts cross-location (token-mixing) operations (ii) for one kind of information only, then conduct per-location (channel-mixing) operations (i) for both kinds of information. Note both (i) and (ii) are implemented with only MLPs in our CrossMLP and we summarize the architecture of our CrossMLP module in Figure 2.

Specifically, given two input features F_{t-1}^I and F_{t-1}^S with spatial shape $H \times W$ (H and W indicate the height and width of a given feature). Both of them are firstly embedded to a sequence of S non-overlapping feature map patches ($S = HW/P^2$, P is the patch size) and each one is projected to a desired hidden dimension C for generating the 2D real-valued input table $X_I \in \mathbb{R}^{S \times C}$ and $X_S \in \mathbb{R}^{S \times C}$. Each CrossMLP consists of 7 layers and each layer consist of two MLP blocks. The first one is the token-mixing MLP while another is the channel-mixing MLP. The former acts on columns of X_I and X_S , which is similar to the original MLP-Mixer proposed by Tolstikhin *et al.* [49] as follows:

$$\begin{aligned} U_{*,i}^I &= X_{*,i}^I + W_2^I \sigma(W_1^I \text{LayerNorm}(X^I)_{*,i}), \text{ for } i = 1, \dots, C \\ U_{*,i}^S &= X_{*,i}^S + W_2^S \sigma(W_1^S \text{LayerNorm}(X^S)_{*,i}), \text{ for } i = 1, \dots, C \end{aligned} \quad (1)$$

The latter is shared across all rows but accompanied with another code by addition in a mutually cross interactive manner shown in Figure 2. This operation can be described as follows:

$$\begin{aligned} Y_{j,*}^I &= U_{j,*}^I + W_4^I \sigma(W_3^I \text{LayerNorm}(U^I + U^S)_{j,*}), \text{ for } j = 1, \dots, S \\ Y_{j,*}^S &= U_{j,*}^S + W_4^S \sigma(W_3^S \text{LayerNorm}(U^S + U^I)_{j,*}), \text{ for } j = 1, \dots, S \end{aligned} \quad (2)$$

Here σ means element-wise nonlinearity (GELU [41]), W indicates the linear learnable weight of MLP. Based on this fashion, each kind of the information will be processed by one MLP layer along with the columns first, then each processed information accompanied with another information will be further strengthened mutually by the second MLP layer along with the rows.

Image Pathway Update. After we get the latent cues Y^I and Y^S about where to sample source patches and where to put patches. We design the image pathway update to let the cues guide the coarse image generation. Firstly, both Y^I and Y^S will be recovered from sequence shape $S \times C$ to the image shape $H \times W$, then a concatenation operation is adopted to combine them together. These steps can be depicted as follows:

$$Y = \text{cat}(\text{reshape}(Y^I), \text{reshape}(Y^S)). \quad (3)$$

Then Y is transferred to an attention map denoted as M_{Att} , which incorporates both the image code at source view and semantic map code at target view. The value of M_{Att} is in the range of $(0, 1)$ and indicates the importance of every element in the image code. Mathematically:

$$M_{Att} = \text{Sigmoid}(\text{Conv}(Y)), \quad (4)$$

where Conv is a convolutional layer with kernel size 1×1 to fully exchange information among channels. Sigmoid here is to map the value after the convolutional layer to the range of $(0, 1)$. Finally, we update the image code as follows:

$$F_t^I = F_{t-1}^I + M_{Att} \otimes \text{Conv}(F_{t-1}^I), \quad (5)$$

where \otimes denotes element-wise multiplication, which ensures image code F_t^I at one certain location are either preserved or suppressed. Moreover, the addition to F_{t-1}^I constructs a residual connection [10] helps preserve the original image appearance information. It is critical especially when the number of the CrossMLP block goes higher.

Semantic Map Pathway Update. To make the semantic map code can match the updated image code at the same feature level, we adopt channel-wise concatenation between F_t^I and Y as follows:

$$F_t^S = \text{cat}(F_t^I, Y). \quad (6)$$

By doing so, both image code and semantic map code are activated mutually for generating more detailed images at the target view.

3.3 Refined Pixel-level Loss

This noisy semantic label problem has been investigated in [10], and it showed that weighing multiple loss functions by considering the uncertainty of each task can add regularization to the overall optimization objectives. This strategy was also adopted by SelectionGAN [53], which constructed 4 "tasks" with 4 fake-real pairs forming the uncertainty-guided pixel-level loss \mathcal{L}_p to overcome this problem as follows:

$$\mathcal{L}_p \leftarrow \frac{\mathcal{L}_p(\text{Fake}, \text{Real})}{U} + \log U = \frac{|\text{Fake} - \text{Real}|}{U} + \log U, \quad (7)$$

where \mathcal{L}_p denotes a pixel-level loss and U denotes the relevant uncertainty map. *Fake* and *Real* indicate generated results and ground truth items. We notice that in the SelectionGAN [53], it need to compute the original pixel-level loss \mathcal{L}_p 4 times for 4 (*Fake*, *Real*) pairs, there are (I'_g, I_g) , (I''_g, I_g) , (S'_g, S_g) , and (S''_g, S_g) . We argue this approach is complicated and redundant. Moreover, it ignores the relation between generated image pair (I'_g, I''_g) and semantic map pair (S'_g, S''_g) .

Hence, we grouped these 4 fake-real pairs into 2 kinds based on whether the pair is image related or semantic map related. This makes the form of the loss become more compact than the one used in [43]. In addition, we add a new square item to constraint the difference between generated items for both stages. The refined pixel-level loss for both \mathcal{L}_g and \mathcal{L}_s share the same form as :

$$\mathcal{L}_p \leftarrow \frac{|Fake' - Real| + |Fake'' - Real|}{U} + \log U + (Fake' - Fake'')^2, \quad (8)$$

The first item of our refined loss can reduce the distance between generated and the ground truth images for both stages at once, while the third square item further strengthen the consistency between the generated results from both stages.

3.4 Model Training

Parameter-Sharing Discriminator. We extend the vanilla discriminator original in [43] to another parameter-sharing structure. In the first stage, this structure takes the real image I_a and the generated image I'_g or the ground-truth image I_g as input. The discriminator D learns to tell whether a pair of images from different domains is associated with each other or not. In the second stage, it accepts the real image I_a and the generated image I''_g or the real image I_g as input. This pairwise input encourages D to discriminate the diversity of image structure and capture the local-aware information.

Overall Optimization Objective. The total optimization loss is as:

$$\min_{\{G_i, G_s, G_a\}} \max_{\{D\}} \mathcal{L} = \sum_{i=1}^2 \lambda_i \mathcal{L}_p^i + \mathcal{L}_{cGAN} + \lambda_{TV} \mathcal{L}_{TV}, \quad (9)$$

where \mathcal{L}_p^i indicate our proposed refined pixel-level loss for both image and semantic map information, \mathcal{L}_{cGAN} is the classic adversarial loss as follows:

$$\mathcal{L}_{cGAN} \left(I_a, I_g^{(\cdot)} \right) = \mathbb{E}_{I_a, I_g^{(\cdot)}} [\log D(I_a, I_g)] + \mathbb{E}_{I_a, I_g^{(\cdot)}} \left[\log \left(1 - D \left(I_a, I_g^{(\cdot)} \right) \right) \right], \quad (10)$$

where $I_g^{(\cdot)}$ means both the result I'_g from the first stage and I''_g from the second stage share the same adversarial loss. \mathcal{L}_{TV} is the total variation regularization [43] on the I''_g as follows:

$$\mathcal{L}_{TV}(I''_g) = \sum_{m,n} \left| I''_{g[m+1,n]} - I''_{g[m,n]} \right| + \left| I''_{g[m,n+1]} - I''_{g[m,n]} \right|, \quad (11)$$

where m and n indicate the coordinate of the pixels of I''_g . λ_i and λ_{TV} are the trade-off parameters to control the relative importance of different objectives, and they are set to 0.5 and 1, respectively. The training is performed by solving the min-max optimization problem.

Implementation Details. Besides the proposed cascaded CrossMLP sub-network, we also utilize the selection module proposed in [43]. In addition, G_s is a shallow U-Net [43] architecture based generator and the filters in the first convolutional layer is set to 3. What's more, We adopt PatchGAN [43] for the discriminator D . About the training details, we follow the optimization method in [8] to optimize the proposed Cascaded Cross MLP-Mixer GAN, i.e., one gradient descent step on discriminator and generators alternately. We first train cascaded CrossMLP sub-network, selection module, and G_s with D fixed, and then train D with others

Table 1: Quantitative comparisons on the Dayton dataset.

Method	Accuracy (%) \uparrow				Inception Score \uparrow			SSIM \uparrow	PSNR \uparrow	SD \uparrow	KL \downarrow
	Top-1		Top-5		all	Top-1	Top-5				
Pix2pix [13]	6.80*	9.15*	23.55*	27.00*	2.8515*	1.9342*	2.9083*	0.4180*	17.6291*	19.2821*	38.26 \pm 1.88*
X-Fork [29]	30.00*	48.68*	61.57*	78.84*	3.0720*	2.2402*	3.0932*	0.4963*	19.8928*	19.4533*	6.00 \pm 1.28*
X-Seq [29]	30.16*	49.85*	62.59*	80.70*	2.7384*	2.1304*	2.7674*	0.5031*	20.2803*	19.5258*	5.93 \pm 1.32*
SelectionGAN [53]	42.11 \dagger	68.12 \dagger	77.74 \dagger	92.89 \dagger	3.0613 \dagger	2.2707 \dagger	3.1336 \dagger	0.5938\dagger	23.8874\dagger	20.0174\dagger	2.74 \pm 0.86 \dagger
CrossMLP (Ours)	47.65	78.59	80.04	94.64	3.3466	2.2941	3.3783	0.5599	23.6232	19.6688	2.33 \pm 0.80
Real Data	-	-	-	-	3.7196 \dagger	2.3626 \dagger	3.8998 \dagger	-	-	-	-

Table 2: Quantitative comparisons on the CVUSA dataset.

Method	Accuracy (%) \uparrow				Inception Score \uparrow			SSIM \uparrow	PSNR \uparrow	SD \uparrow	KL \downarrow
	Top-1		Top-5		all	Top-1	Top-5				
Pix2pix [13]	7.33*	9.25*	25.81*	32.67*	3.2771*	2.2219*	3.4312*	0.3923*	17.6578*	18.5239*	59.81 \pm 2.12*
X-Fork [29]	20.58*	31.24*	50.51*	63.66*	3.4432*	2.5447*	3.5567*	0.4356*	19.0509*	18.6706*	11.71 \pm 1.55*
X-Seq [29]	15.98*	24.14*	42.91*	54.41*	3.8151*	2.6738*	4.0077*	0.4231*	18.8067*	18.4378*	15.52 \pm 1.73*
SelectionGAN [53]	41.52 \dagger	65.51 \dagger	74.32 \dagger	89.66 \dagger	3.8074 \dagger	2.7181 \dagger	3.9197 \dagger	0.5323\dagger	23.1466 \dagger	19.6100\dagger	2.96 \pm 0.97 \dagger
CrossMLP (Ours)	44.96	69.96	76.98	91.91	3.8392	2.8035	3.9757	0.5251	23.1532	19.5799	2.69 \pm 0.94
Real Data	-	-	-	-	4.8741 \dagger	3.2959 \dagger	4.9943 \dagger	-	-	-	-

fixed. Our method is trained and optimized in an end-to-end fashion. We employ Adam [20] with momentum terms $\beta_1=0.5$ and $\beta_2=0.999$ as our solver. The initial learning rate for Adam is 0.0002. The network initialization strategy is Xavier [10], weights are initialized from a Gaussian distribution with standard deviation 0.2 and mean 0.

4 Experiments

Datasets. We perform all experiments on two datasets: (i) For the Dayton dataset [40], we follow the same setting of [29] and select 76,048 images and create a train/test split of 55,000/21,048 pairs. And we reset the resolution to 256×256 ; (ii) The CVUSA dataset [22] consists of 35,532/8,884 train/test image pairs. Following [29, 47], the aerial images are center-cropped to 224×224 and resized to 256×256 . For the ground-level images and corresponding segmentation maps, we also resize them to 256×256 .

Parameter Settings. For a fair comparison, we adopt the same training setup as in [13, 29, 53] for both datasets. All images are scaled to 256×256 , and we enabled image flipping and random crops for data augmentation. Similar to [29], our method on the Dayton dataset is trained for 35 epochs. For the CVUSA dataset, we follow the same setup as in [29, 47], and train our network for 30 epochs. Our method is implemented in PyTorch [27].

Evaluation Metrics. Inception Score, top-k prediction accuracy, and KL score are employed for the quantitative analysis. These metrics evaluate the generated images from a high-level feature space. We also employ Structural-Similarity (SSIM), Peak Signal-to-Noise Ratio (PSNR), and Sharpness Difference (SD) to measure the pixel-level similarity.

4.1 State-of-the-Art Comparisons

We compare our Cascaded Cross-Mixer GAN with four state-of-the-art methods Pix2pix [13], X-Fork [29], X-Seq [29], and SelectionGAN [53] both quantitatively and qualitatively.

Quantitative Comparisons. The results are shown in Tables 1 and 2. Note that (*, \dagger) means these results are reported in [29] and [53]. We can see the significant improvement of our method in both tables compared to other state-of-the-art methods on most of the metrics. Note that for metrics like SSIM, PSNR, and SD, our method also achieve the second-best results with only a bit lower on both datasets. However, the qualitative results presented in Figure 3 show our method can generate more realistic and sharper results than others.

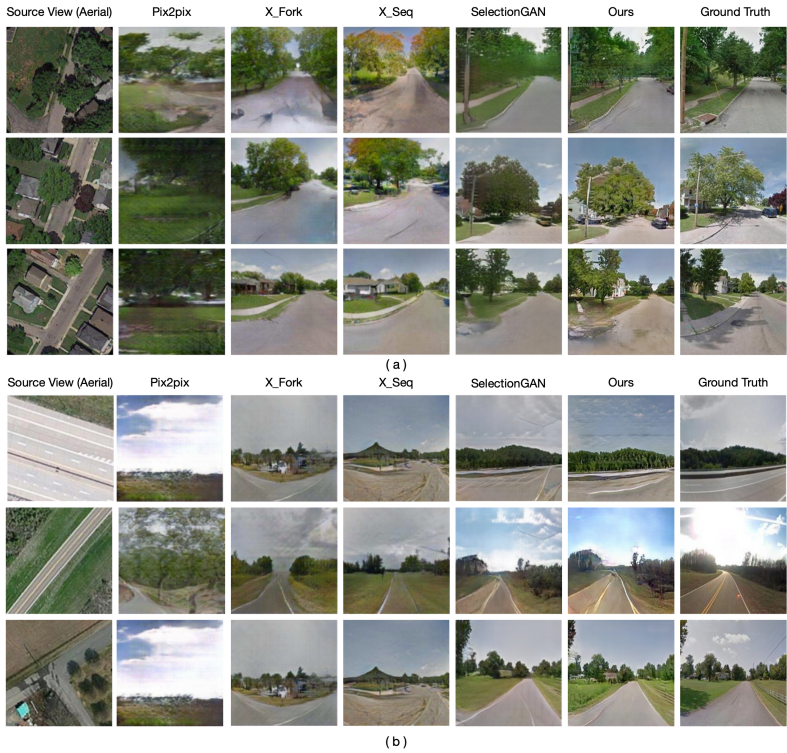


Figure 3: Qualitative results of different methods on (a) Dayton and (b) CVUSA datasets.

Qualitative Comparisons. The qualitative results on both Dayton and CVUSA datasets are shown in Figure 3. It’s clear that the proposed Cascaded Cross-Mixer GAN can generate clearer and sharper scenes with more details but fewer artifacts on trees (see the first row in Figure 3), clouds, roads(see the first and the fifth rows in Figure 3), shadows (see the third row Figure 3), sky (see the second and sixth rows in Figure 3), than other methods.

4.2 Ablation Analysis

To reduce the training time, we randomly select 1/3 samples from the whole Dayton dataset to form the ablation study dataset Dayton-Ablation with around 18,334/7,017 samples. We design five baselines (B1, B2, B3, B4, B5) as shown in Table 3 and Table 4. The first four baselines are used to figure out how the number of CrossMLP blocks affects the experimental results without the proposed refined loss. They use 3, 5, 7, and 9 blocks, respectively. The last one B5 is built on B4, but with our refined pixel-level loss for optimization during training. All the baseline models are trained and tested by the same configuration settings.

From the numerical results presented in Table 3. We conclude that with the increase of the number of cascaded CrossMLP blocks, the performance is also improved. What’s more, from the comparison between B4 and B5 in Table 4, it’s clear that the refined pixel-level loss can boosts the numerical results obviously. We also provide more intuitive visual results in our supplementary materials. To further validate the effectiveness of the proposed refined pixel-level loss, another ablation experiment is conducted based on SelectionGAN [B3], and results presented in Table 4 show that SelectionGAN can achieve much better performance

Table 3: Ablation results on the Dayton-Ablation dataset.

Method	Accuracy (%) \uparrow				Inception Score \uparrow			KL \downarrow
	Top-1		Top-5		all	Top-1	Top-5	
	B1 (3 Blocks)	43.68	72.84	74.58	91.23	3.2513	2.2051	
B2 (5 Blocks)	44.35	73.40	77.21	91.60	3.1825	2.1625	3.1889	2.803 \pm 0.93
B3 (7 Blocks)	42.58	70.67	74.85	90.77	3.2988	2.2190	3.3317	2.878 \pm 0.91
B4 (9 Blocks)	43.41	70.73	76.84	91.77	3.3241	2.2570	3.3340	2.725 \pm 0.87
Real Data	-	-	-	-	3.8246	2.5668	3.9119	-

Table 4: Quantitative comparisons on the Dayton-Ablation dataset to validate the effectiveness of our proposed refined pixel-level loss.

Method	Accuracy (%) \uparrow				Inception Score \uparrow			SSIM \uparrow	PSNR \uparrow	SD \uparrow	KL \downarrow
	Top-1		Top-5		all	Top-1	Top-5				
B4	43.41	70.73	76.84	91.77	3.3241	2.2570	3.3340	0.4814	21.0154	19.1099	2.725 \pm 0.87
B5 (B4 + Our Loss)	44.08	73.46	76.44	92.28	3.4140	2.3197	3.4444	0.5515	22.9443	19.5192	2.699 \pm 0.90
B5 + BAM [28]	42.51	72.96	72.35	89.86	3.2756	2.1811	3.2703	0.5451	22.7198	19.4813	3.130 \pm 1.03
SelectionGAN [28]	41.60	71.21	75.20	92.01	3.2252	2.2387	3.2459	0.4952	21.1408	19.3979	2.852 \pm 0.93
[28] + Our Loss	42.75	71.47	75.70	92.39	3.2664	2.3101	3.2909	0.5442	21.1408	19.6984	2.704 \pm 0.82
Real Data	-	-	-	-	3.8246	2.5668	3.9119	-	-	-	-

on all the evaluation metrics with our proposed refined loss.

In addition, Qiu *et al.* [28] also proposed a MLP based bilateral augmentation module (BAM) which is depicted somewhat similar to our CrossMLP module. Actually, they are totally different. Compared to our CrossMLP module, (1) For each module of BAM, only one input is from previous module and another is always the original 3D coordinates; (2) All MLP operations in BAM are in the channel-wise manner; (3) The cross interaction is realised by concatenation operation for two feature at different semantic level. Most importantly, we replace the CrossMLP module with the BAM (with slight modification to make it suitable for the data of our task) to see if it can achieve better performance based on B5. The results in Table 4 shows that our CrossMLP can outperform BAM by a large margin in all evaluation metrics. Hence we conclude the proposed CrossMLP module is more suitable and effective for the drastic cross-view image translation task.

5 Conclusion

We propose a novel two-stage Cascaded Cross MLP-Mixer GAN to progressively address the challenging cross-view image translation task, particularly when there exists little or no overlap between source view and target view. In particular, our method can generate more detailed coarse results step by step in the first stage via our proposed cascaded CrossMLP blocks. What’s more, within each block, the latent transformation cues can be well established by the CrossMLP module. Then the cross-view translation operation will be guided by those latent cues with better visual quality. In addition, we propose a refined pixel-level loss which can provide a further constraint between the coarse and final result images with a more compact formulation fashion, the inaccurate semantic labels problem can be also eased for better optimization with more reasonable regularization within the overall optimization objective. Extensive experimental results on two public datasets demonstrate that our method obtains much better results than the state-of-the-art.

Acknowledgments. This work was supported by the EU H2020 AI4Media No. 951911 project and by the PRIN project PREVUE.

References

- [1] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2018.
- [2] Xinyuan Chen, Chang Xu, Xiaokang Yang, and Dacheng Tao. Attention-gan for object transfiguration in wild images. In *ECCV*, 2018.
- [3] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*, 2016.
- [4] Alexey Dosovitskiy, Jost Tobias Springenberg, Maxim Tatarchenko, and Thomas Brox. Learning to generate chairs, tables and cars with convolutional networks. *IEEE TPAMI*, 39(4):692–705, 2017.
- [5] 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. In *ICLR*, 2020.
- [6] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational u-net for conditional appearance and shape generation. In *CVPR*, pages 8857–8866, 2018.
- [7] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feed-forward neural networks. In *ICAIS*, 2010.
- [8] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.
- [9] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. In *ICML*, pages 1462–1471, 2015.
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.
- [11] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.
- [12] Rui Huang, Shu Zhang, Tianyu Li, Ran He, et al. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In *ICCV*, 2017.
- [13] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.
- [14] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two transformers can make one strong gan. *arXiv preprint arXiv:2102.07074*, 2021.
- [15] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

- [16] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019.
- [17] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, pages 7482–7491, 2018.
- [18] Junho Kim, Minjae Kim, Hyeonwoo Kang, and Kwang Hee Lee. U-gat-it: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. In *ICLR*, 2019.
- [19] Taeksoo Kim, Moon-su Cha, Hyunsoo Kim, Jungkwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *ICML*, 2017.
- [20] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- [21] Gaowen Liu, Hao Tang, Hugo Latapie, and Yan Yan. Exocentric to egocentric image generation via parallel generative adversarial network. In *ICASSP*, pages 1843–1847, 2020.
- [22] Gaowen Liu, Hao Tang, Hugo Latapie, Jason Corso, and Yan Yan. Cross-view exocentric to egocentric video synthesis. In *ACM MM*, 2021.
- [23] Shuang Ma, Jianlong Fu, Chang Wen Chen, and Tao Mei. Da-gan: Instance-level image translation by deep attention generative adversarial networks. In *CVPR*, 2018.
- [24] Youssef Alami Mejjati, Christian Richardt, James Tompkin, Darren Cosker, and Kwang In Kim. Unsupervised attention-guided image-to-image translation. In *NeurIPS*, 2018.
- [25] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *CVPR*, 2017.
- [26] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, pages 2337–2346, 2019.
- [27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.
- [28] Shi Qiu, Saeed Anwar, and Nick Barnes. Semantic segmentation for real point cloud scenes via bilateral augmentation and adaptive fusion. In *CVPR*, pages 1757–1767, 2021.
- [29] Krishna Regmi and Ali Borji. Cross-view image synthesis using conditional gans. In *CVPR*, pages 3501–3510, 2018.
- [30] Bin Ren, Hao Tang, Fanyang Meng, Runwei Ding, Ling Shao, Philip HS Torr, and Nicu Sebe. Cloth interactive transformer for virtual try-on. *arXiv preprint arXiv:2104.05519*, 2021.

-
- [31] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *ICCV*, pages 4570–4580, 2019.
- [32] Hao Tang and Nicu Sebe. Layout-to-image translation with double pooling generative adversarial networks. *IEEE TIP*, 2021.
- [33] Hao Tang, Dan Xu, Nicu Sebe, Yanzhi Wang, Jason J Corso, and Yan Yan. Multi-channel attention selection gan with cascaded semantic guidance for cross-view image translation. In *CVPR*, pages 2417–2426, 2019.
- [34] Hao Tang, Song Bai, Philip HS Torr, and Nicu Sebe. Bipartite graph reasoning gans for person image generation. In *BMVC*, 2020.
- [35] Hao Tang, Song Bai, Li Zhang, Philip HS Torr, and Nicu Sebe. Xinggan for person image generation. In *ECCV*, pages 717–734, 2020.
- [36] Hao Tang, Dan Xu, Yan Yan, Philip HS Torr, and Nicu Sebe. Local class-specific and global image-level generative adversarial networks for semantic-guided scene generation. In *CVPR*, pages 7870–7879, 2020.
- [37] Hao Tang, Hong Liu, Dan Xu, Philip HS Torr, and Nicu Sebe. Attentiongan: Unpaired image-to-image translation using attention-guided generative adversarial networks. *IEEE TNNLS*, 2021.
- [38] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Multi-view 3d models from single images with a convolutional network. In *ECCV*, 2016.
- [39] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, et al. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021.
- [40] Nam N Vo and James Hays. Localizing and orienting street views using overhead imagery. In *ECCV*, pages 494–509. Springer, 2016.
- [41] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004.
- [42] Scott Workman, Richard Souvenir, and Nathan Jacobs. Wide-area image geolocalization with aerial reference imagery. In *ICCV*, pages 3961–3969, 2015.
- [43] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*, 2018.
- [44] Jimei Yang, Scott E Reed, Ming-Hsuan Yang, and Honglak Lee. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In *NeurIPS*, 2015.
- [45] Xiaochuan Yin, Henglai Wei, Xiangwei Wang, Qijun Chen, et al. Novel view synthesis for large-scale scene using adversarial loss. *arXiv preprint arXiv:1802.07064*, 2018.
- [46] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. In *CVPR*, pages 9459–9468, 2019.

- [47] Menghua Zhai, Zachary Bessinger, Scott Workman, and Nathan Jacobs. Predicting ground-level scene layout from aerial imagery. In *CVPR*, pages 867–875, 2017.
- [48] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *ECCV*, 2016.
- [49] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.
- [50] Xinge Zhu, Zhichao Yin, Jianping Shi, Hongsheng Li, and Dahua Lin. Generative adversarial frontal view to bird view synthesis. In *3DV*, 2018.
- [51] Zhen Zhu, Tengting Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. Progressive pose attention transfer for person image generation. In *CVPR*, pages 2347–2356, 2019.