

## Article

# A High-Quality Hybrid Mapping Model Based on Averaging Dense Sampling Parameters

Fanxiao Yi <sup>†</sup> , Weishi Li <sup>†</sup>, Mengjie Huang, Yingchang Du and Lei Ye <sup>\*</sup>

College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China; yifanxiao@nudt.edu.cn (F.Y.); liweishi1028@nudt.edu.cn (W.L.); huangmj@nudt.edu.cn (M.H.); duyinchang@nudt.edu.cn (Y.D.)

<sup>\*</sup> Correspondence: yelei@nudt.edu.cn

<sup>†</sup> These authors contributed equally to this work.

**Abstract:** Navigation map generation based on remote sensing images is crucial in fields such as autonomous driving and geographic surveying. Style transfer is an effective method for obtaining a navigation map of the current environment. However, there is lack of robustness of the map-style transfer model, resulting in unsatisfactory quality of the generated navigation maps. To address these challenges, we average the parameters of generators sampled from different iterations with a dense sampling strategy in the Generative Adversarial Network (CycleGAN). The results demonstrate that the training efficiency of our method on the MNIST and generation quality on the Google Map dataset are significantly superior to traditional style transfer methods. Moreover, it performs well in multi-environment hybrid mapping. Our method improves the generalization ability of the model and converts existing navigation maps to other styles of maps precisely. It can better adapt to different types of urban layout and road planning, bringing innovative solutions for traffic management and navigation systems.

**Keywords:** style transfer; remote sensing; model fusion; machine learning; deep learning



**Citation:** Yi, F.; Li, W.; Huang, M.; Du, Y.; Ye, L. A High-Quality Hybrid Mapping Model Based on Averaging Dense Sampling Parameters. *Appl. Sci.* **2024**, *14*, 335. <https://doi.org/10.3390/app14010335>

Academic Editor: Antonio Fernández-Caballero

Received: 30 November 2023

Revised: 25 December 2023

Accepted: 26 December 2023

Published: 29 December 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

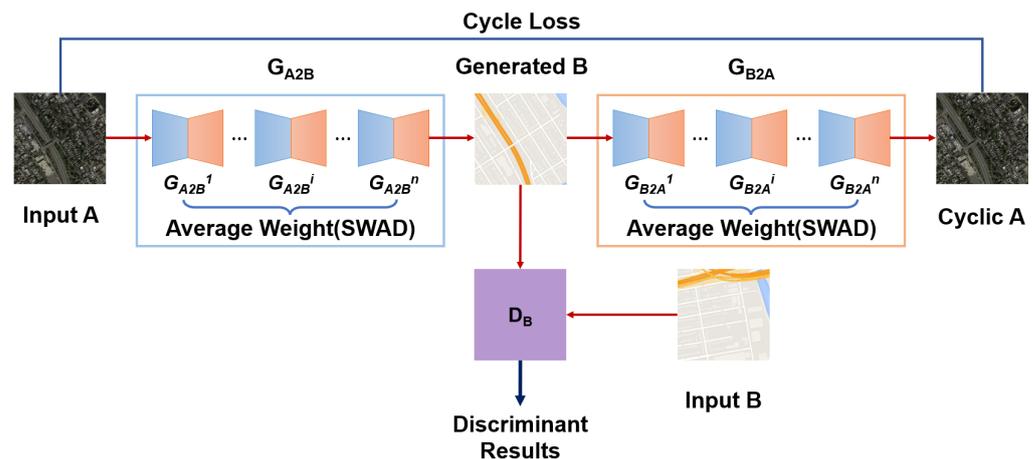
## 1. Introduction

Navigation maps [1–3], serving as a source of data, become more indispensable in fields such as autonomous driving [4], as more and more researchers use them for planning [5] and navigation decisions [6] in the context of autonomous vehicles. Point cloud data [7] and navigation systems [8] could acquire navigation maps, but they are time consuming, inefficient, and costly. In contrast, methods that use style transfer based on remote sensing images [9–11] to generate navigation maps are cost effective and offers real-time capability, which effectively address the aforementioned shortcomings. Style transfer [12,13] is one of the most significant parts of navigation map generation, which transforms images from one domain to another by using a style image to convert a given image without altering its original content. It extracts features from the content and style of the image separately, adjusting and optimizing the weights of neural networks, and then fuses them to reconstruct the final image. Currently, navigation maps are mainly used in urban environments and in off-road settings. Especially in some scenarios like autonomous vehicle testing, the generation of navigation maps based on remote sensing images can provide valuable off-road image samples and accurate geographic information and real-time updates [14–16].

Nevertheless, style transfer faces challenges such as insufficient precision and high time cost. In order to overcome the obstacles, we propose an improvement in optimization methods, which fuses the parameters of generators sampled from different iterations in CycleGANs [17]. The weight average (WA) of deep learning models leverages the strengths of different models to compensate for individual model deficiencies, giving rise to improved predictive outcomes [18,19]. However, the effect of CycleGAN combined

with WA is often not ideal. Considering that WA based on Dense Stochastic Weighted averaging (SWAD) [20] increases the diversity of samples and introduces cycle consistency constraints, we also average the weights of generators with a similar dense sampling strategy. The method improves the overall quality of style transfer and generalizability in off-road environments, which provides users with the best route planning and traffic information. In addition, it can also improve traffic efficiency and develop urban planning and future transportation.

Specifically, we improve the performance of CycleGAN by averaging the weights of generators sampled from different periods as shown in Figure 1. Image *A* is input into the network and transformed into *Generated B* (image *B*) by  $G_{A2B}$  (generators). Then, we establish a loss function between *Generated B* and image *B* via  $D_B$  (discriminators). *Cyclic A* is generated in reverse using  $G_{B2A}$  to establish the cycle loss, so as to maintain content consistency between *Generated B* and *A*. The SWAD is implemented within  $G_{A2B}$  and  $G_{B2A}$  by calculating weighted averages of model weights with the dense sampling strategy at different training stages. The method enables the model to efficiently extract features from off-road environments, improving its ability to generate navigation maps that blend both off-road and urban environments. Since each GAN model may have different training data and network architectures, the fusion process is able to mitigate the limitations of individual models and enhance the overall system’s robustness, which makes it more adaptable to various input conditions [21,22].



**Figure 1.** SWAD-CycleGAN fusion model network structure diagram. *A* denotes the A-class-style images including map information. *Generated B* denotes the final map with B-class-style images obtained by generators from A-style map information. *Cyclic A* denotes the final map with A-class-style images obtained by generators from *Generated B*.  $G_{A2B}$  represents the generator that converts A-class-style images into B-class-style images,  $G_{B2A}$  represents the generator that converts B-class-style images into A-class-style images, and  $D_B$  represents the discriminator for B-class images.  $G_{A2B}^i$  represents the model at round *i* in training.

The contributions made by this paper are as follows: we propose a novel method that averages the parameters of generators sampled from different iterations with the dense sampling strategy in CycleGANs. We conduct a large number of experiments to compare the performance of the method on datasets from Google Maps [23]. Additionally, we conduct separate training for urban and off-road environments. We confirm that the method not only conducts the convergence at a more rapid speed, but it also ensures the generation quality in both scenarios, ultimately resulting in the development of a hybrid-map-style transfer model. The remaining structure of this paper is organized as follows: Section 2 provides an overview of related work on GANs and the optimization methods employed. Section 3 gives priority to the mathematical formulation of the constructed models and the optimization methods adopted in our experiments. In Section 4, we present

comparative experiments and draw conclusions based on the results. Finally, we conclude our proposed method and offer prospects for future research.

## 2. Related Work

The work of map-style transfer focuses on transferring texture features in the early stage. However it does not explicitly address the style transfer task itself, resulting in less-than-ideal outcomes. With the development of deep learning technologies, some work [24] trains images with neural networks to make the image's style closer to the target style. But, these methods are computationally intensive and not suitable for real-time map generation, which tends to distortions in certain image data during transfer. Gatys et al. [25] simplify the style transfer problem as an optimization problem within a single neural network, of which the texture model based on deep image models was able to generate new images by matching feature representations of example images via an image search. However, it is not well suited for photos. Subsequently, Johnson et al. [26] expand on Gatys' work by replacing the input random noise image with a target image and incorporating a forward autoencoder network to model the style transfer process. Luan et al. [27] address the limitations of Gatys' model by constraining the output within a specific color space to suppress deformations and maintain the realism of photos. The constraint is fully differentiable, enabling better preservation of style information from reference images without introducing distortions caused by deformations. Li et al. [28] use feature transformations such as Whitening and Coloring Transform (WCT) to directly match the statistical properties of content feature statistics with those of style images in the deep feature space. Li et al. [29] improve the WCT-based method by introducing stylization and smoothing steps to tackle structural artifacts in the output of the WCT algorithm. However, the WCT-based approach still has the potential to produce spatial distortions or unrealistic artifacts. Yoo et al. [30] suggest incorporating wavelet transform into neural networks based on WCT, enabling features to retain both structural information and statistical properties of the VGG space, thereby addressing the aforementioned limitations. Recent work tends to use GANs to implement map-style transfer, especially CycleGAN [31]. However, there are challenges pertaining to image accuracy and real-time processing, which motivates us to enhance the CycleGAN for map-style transfer.

WA is a technique which fuses the deep learning models and aims to improve the model. Some recent work [32] uses Stochastic Weighted Averaging (SWA) [33] in CycleGANs [34,35], which enhances the capabilities of generators. Inspired by checkpoint averaging [36,37], Izmailov et al. [33] propose SWA. Using the mean of multiple models on the training trajectory can help find a better-performing network with no additional computational overhead compared to training a single model. When SGD convergence is good, the checkpoint of SWA sampling is concentrated around the optimal solution. Using SWA operations for these checkpoints helps optimize the training network. Similar to FGE [38–40], using the cyclic learning rate [33,41], models that are spatially close to each other but produce different predictions can be collected. Some recent work proposes variations on SWA that allow SWA to be adapted to more complex application scenarios and have more efficient rates. We tend to combine SWAD and SWA with CycleGANs to improve the performance of the navigation maps.

## 3. Methods

### 3.1. Cyclic Adversarial Generation Network Based on SWAD Optimization Method

CycleGAN realizes mapping from one domain to another via generators and discriminators. In this section, we propose a method that averages the parameters of generators sampled from different iterations with the dense sampling strategy in CycleGANs. First, we illustrate the framework of SWAD-CycleGAN, as shown in Figure 2. CycleGAN involves four models:  $G_{A2B}$ ,  $G_{B2A}$ ,  $D_A$  and  $D_B$ .  $G_{A2B}$  transforms *Real A* inputs into *Cyclic B* outputs, while  $G_{B2A}$  handles the transformation of *Cyclic B* inputs into *Cyclic A* outputs. The concept of cycle consistency in adversarial networks involves both forward cycle consistency

and backward cycle consistency. Forward cycle consistency refers to the fact that when an input from the source domain passes through a generator and a reverse generator, it should map back to an output within the source domain while maintaining consistency with the original input. Similarly, backward cycle consistency ensures that an input from the target domain, after going through the reverse generator and generator, should map back to an output within the target domain, remaining consistent with the original input. The cycle consistency loss helps in preventing the adversarial networks from learning mapping functions  $G$  and  $F$ , while learning mapping functions  $G$  and  $F$  tends to make the source and target domains have the same distribution. Despite the network having the capacity to map source domain images to any image in the target domain and generate a distribution equivalent to that of the target domain, cycle consistency loss is beneficial to constraining this behavior. Additionally, the identity loss function ensures that when A-class-style maps are inputted into  $G_{A2B}$ , the generated output resembles B-class-style images and vice versa. However, this may not always hold true, so the identity loss function is designed to impose constraints on this aspect.

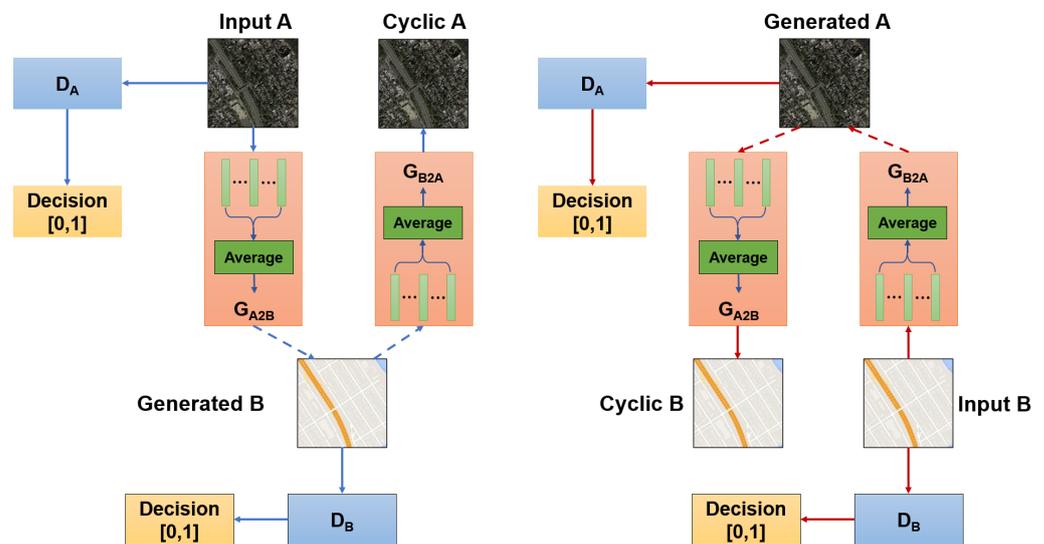


Figure 2. Forward experimental model and reverse experimental model of SWAD-CycleGAN.

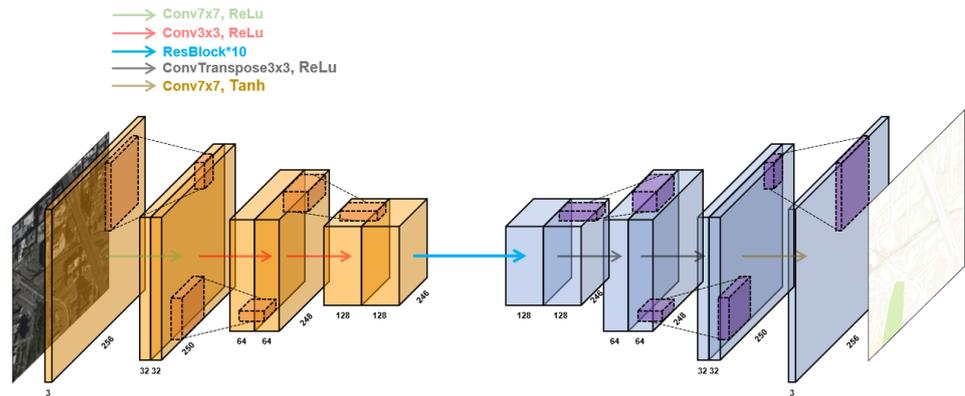
$D_A$  and  $D_B$  are responsible for discriminating between the authenticity of A and B. Additionally, there are four types of loss functions:  $G_{loss}$ ,  $D_{loss}$ ,  $Cycle_{loss}$ , and  $identity_{loss}$ . The loss function  $G_{loss}$  for the Generator as Equation (1), with a target to make the generated images closely align with the distribution of target images, ultimately “deceiving” the discriminator.

$$\mathcal{L}_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{data}(y)} [\log D_Y(y)] + \mathbb{E}_{x \sim p_{data}(x)} [\log(1 - D_Y(G(x)))] \quad (1)$$

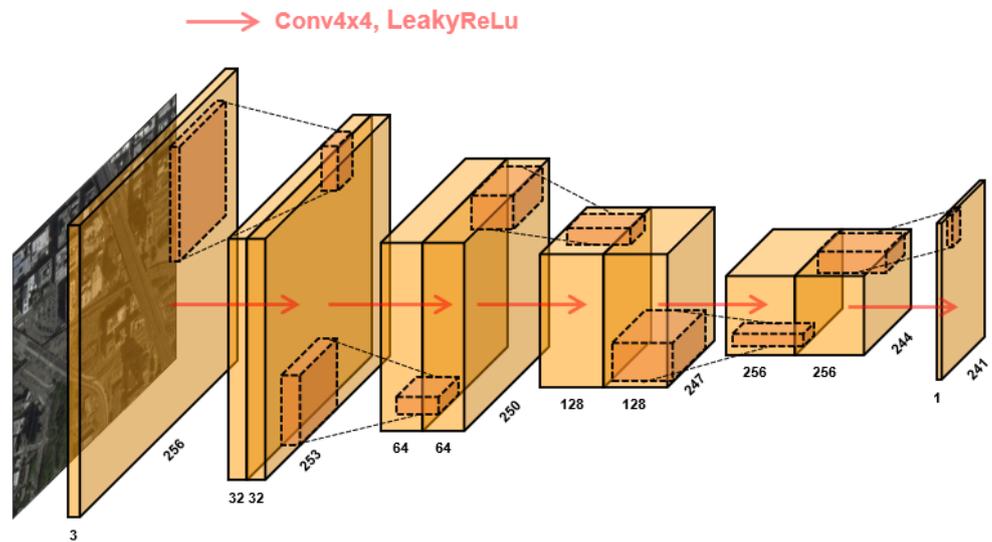
where  $p_{data}(y)$  denotes probability distribution on the output.  $G$  and  $D$  denote the generators and discriminators. On the other hand, the loss function for the Discriminator ( $D_{loss}$ ) has the opposite objective, namely the distinguishment of the images generated by the Generator. The cycle loss ( $Cycle_{loss}$ ) function is depicted as Equation (2):

$$\mathcal{L}_{cyc}(G, F) = \mathbb{E}_{x \sim p_{data}(x)} [\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{data}(y)} [\|G(F(y)) - y\|_1]. \quad (2)$$

The generator and discriminator we designed are based on residual networks, and the specific internal network structures are shown in Figures 3 and 4. While previous work has mainly laid stress on improving loss functions, our target is to enhance the optimization methods for the four models and compare them with several state-of-the-art techniques.



**Figure 3.** Neural network structure of the generators. ResBlock is a residual module composed of multiple convolutional layers and activation layers.



**Figure 4.** Neural network structure of the discriminators.

### 3.2. Network Optimization

Definitely, CycleGAN itself serves as a generative adversarial network for image transformation, which realizes mapping from one domain to another via two generators and two discriminators. We can fuse the output of two generators of CycleGAN to produce better results.

In this paper, we not only adopt SWAD to solve the instability and relieve mode oscillation problems in GAN model training, but also boost the robustness of the model. By averaging multiple best points in the training process, the model can better adapt to the unseen data. There are several generator networks in the GAN network, and through the SWAD strategy, the parameters of these generators can be averaged and the multiple generators can be “fused” into a unified generator model. While this reduces the difficulty in choosing the best generator, it also makes the model more stable and reliable. The SWA is based on averaging multiple points on the SGD trajectory with a periodic learning rate [33] as Equation (3):

$$\begin{aligned} \alpha(i) &= (1 - t(i))\alpha_1 + t(i)\alpha_2, \\ t(i) &= \frac{1}{c}(\text{mod}(i - 1, c) + 1). \end{aligned} \tag{3}$$

where  $\alpha$  denotes the learning rate at  $i$  iteration, and  $c$  denotes the number of iterations. An improvement in generalization is accessible, only by averaging some points on the optimized trajectory during the training process. SWA significantly improves the training

of many state-of-the-art deep neural networks on a series of important baselines, with essentially no overhead. Considering that SWA averages the latter part of the network, the result cannot accurately approximate the ideal minimum on the high-dimensional parameter space. Therefore, we use SWAD [20] to gather enough weights in each iteration.

#### 4. Experiments

The workstation configuration used for the experiments in this article consists of an AMD Ryzen 9 6900HX with 16 GB RAM. To evaluate the effectiveness of the SWAD method on general GAN networks, we first conduct tests on the MNIST [42] to ensure comprehensiveness. Subsequently, we evaluate the effectiveness and benefits of the SWAD-CycleGAN model on Google Maps [23]. Additionally, we conduct experiments in both urban and off-road environments to assess the generalization ability of SWAD-CycleGAN. The experimental results for these two environments are presented in this section.

We randomly split the data in each domain into two splits: 20% for an independent test set, and the remaining 80% for 10 times 10-fold cross validation to set hyperparameters. After that, the training set and validation set are combined and retrained to obtain the final model. Finally, we use the test set to assess the model's generalization ability.

##### 4.1. Dataset and Hyperparameters

In the integrity testing phase of this study, we use the MNIST dataset which is composed of numbers handwritten by 250 different people, as shown in Table 1. During the experimental phase, we utilize RS maps and navigation maps provided by Google Maps which were sampled from in and around New York City. The Google Maps datasets also include urban and off-road environmental data from various locations, categorized into RS maps (class-A-style data) and navigation maps (class-B-style data), each with its respective training and testing subsets.

**Table 1.** The introduction of datasets.

Phase	Source	Image Size	Structure
Integrity Testing	MNIST	$28 \times 28$	$\left\{ \begin{array}{l} \text{train} \\ \text{test} \end{array} \right.$
Experiment	Google Maps	$256 \times 256$	$\left\{ \begin{array}{l} \text{train} \left\{ \begin{array}{l} \text{A class} \\ \text{B class} \end{array} \right. \\ \text{test} \left\{ \begin{array}{l} \text{A class} \\ \text{B class} \end{array} \right. \end{array} \right.$

The foundational neural network architecture used in this paper is the Residual Network [43] (ResNet). Consequently, the input size for the data was set at  $256 \times 256$ . For the training set, in order to ensure comprehensive training, the original data was initially magnified by a factor of 1.12 and subsequently randomly cropped to  $256 \times 256$  dimensions. The data images were then subjected to random flips and finally normalized. Conversely, there were no such constraints applied to the testing set. The data was randomly split into a test set comprising 20% of the total data. The remaining 80% was divided into 10 parts. In each iteration, we randomly select nine parts as the training set, and the remaining part was used as the validation set, which ensures that each part was used as the validation set in turn. We repeat the process 10 times and average the results. The hyperparameters were adjusted based on the final results, as shown in Table 2.

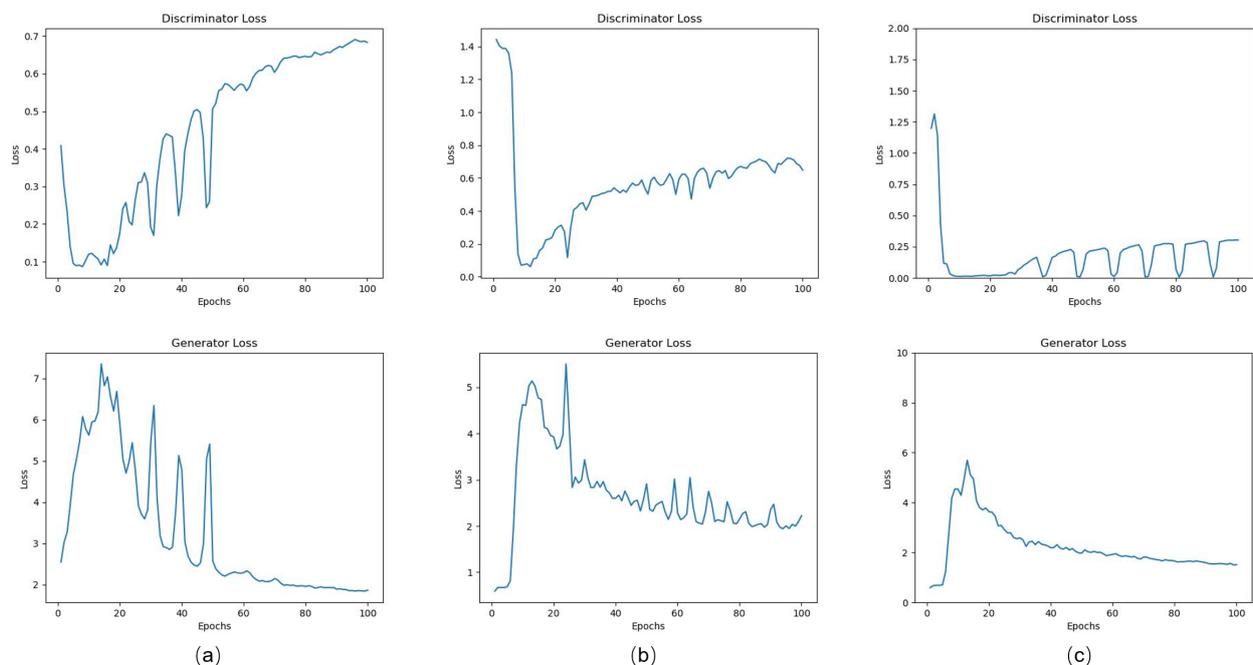
**Table 2.** Hyperparameters employed in SWAD and traditional methods.

Hyperparameters	SGD <sup>1</sup>	Adam <sup>1</sup>	SWAD <sup>1</sup>	SWA <sup>2</sup>	PSWA <sup>2</sup>	SGD <sup>2</sup>	SWAD <sup>2</sup>
Learning rate	-	-	$1 \times 10^{-3}$	$1 \times 10^{-4}$	$2 \times 10^{-4}$	$2 \times 10^{-4}$	$2 \times 10^{-4}$
Batch size	-	-	64	2	2	2	2
Number of epochs	-	-	100	100	100	100	100

<sup>1</sup> Integrity testing phase. <sup>2</sup> Experiment phase.

#### 4.2. Integrity Testing of the SWAD Method on the GAN Network

Prior to the formal experiments, we conduct integrity testing on the SWAD optimization method to assess its feasibility and reliability. The integrity testing is performed using the MNIST datasets. We compare the performance of two conventional optimization methods, Stochastic Gradient Descent [44] (SGD) and Adaptive Moment Estimation [45] (Adam), with that of SWAD on the MNIST train set. Figure 5 presents the discriminator and generator loss based on three different optimization methods: Adam, SGD, and SWAD. Generated data is displayed every 10 epochs, with 16 samples randomly selected from the generated datasets. Over the course of 100 epochs, the SWAD method effectively reduces loss fluctuations, resulting in a steady decline. The SWAD method achieves a significantly lower discriminator loss and a slight decrease in generator loss, which performs exceptionally well on a GAN network. Generated data is displayed every 10 epochs, with 16 samples randomly selected from the generated datasets. Moreover, Figure 6 illustrates that by the 40th epoch, discernible handwritten numbers can already be observed. This suggests that the SWAD method rapidly converges and produces clear images, greatly improving training efficiency and image quality. In summary, the SWAD optimization method demonstrates faster convergence and a better ability to identify the optimal solution in gradient descent. This results in shorter training time while maintaining the quality of generated images. The method has successfully passed the integrity test and can be further applied to CycleGAN in a multi-environment in subsequent experiments.



**Figure 5.** Loss reduction functions of various optimization methods under integrity testing. (a) The discriminator and generator loss of Adam. (b) The discriminator and generator loss of SGD. (c) The discriminator and generator loss of SWAD.

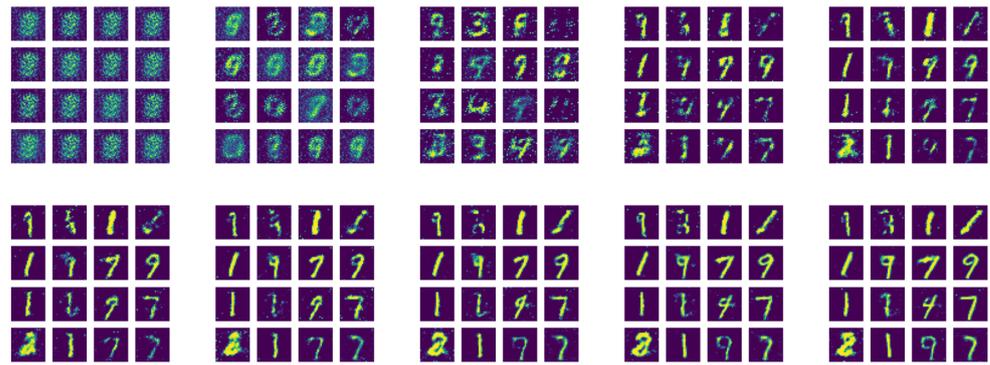


Figure 6. The generation results of SWAD on MNIST after 100 rounds of training (results sampled every 10 rounds).

### 4.3. Evaluation Indicators

The performance of the GAN’s generators is evaluated based on the ability to deceive the discriminator with generated “fake maps”. This iterative process aims to achieve a “Nash equilibrium”, which is a non-cooperative game equilibrium between the generator and discriminator. To explore this, we conduct experiments using a “controlled variable method” where optimization algorithms are applied separately to the two types of models mentioned above, as well as simultaneously to both. The results indicate that applying the optimization algorithm to both models does not significantly improve their performance.

The quantitative indexes of navigation map generation quality include positioning accuracy, map coverage rate, update speed, path planning accuracy, real-time traffic information accuracy, etc. Generally, *FID* (Frechet inception distance score) [46] and *IS* (inception score) [47] are the main indicators used to evaluate the image quality generated by GANs. *FID* calculates the distance between feature vectors of real and generated images. A lower *FID* score indicates a closer similarity between the two images or more similar distributions. In the best case scenario, the *FID* score is 0.0, which represents identical images. Another indicator used is the *IS* (inception score) [47], which can be understood as a measure of image clarity or resolution during training. Since the primary goal of the generative model is to produce realistic images, the *IS* score reflects the clarity of the generated images. For a clear image, the values of vector  $y$  in one dimension are larger, indicating a higher probability of belonging to a certain class, while values in other dimensions are smaller. In professional terms, a clear image has lower entropy in  $P(y | x)$  (higher entropy indicates greater chaos or uncertainty in the values of a random variable). The *IS* indicator is shown as Equation (4):

$$IS(G) = \mathbb{E}[Ex \sim PgKL(P(y | x)P(y))]. \tag{4}$$

In brief, the larger the *IS* index of the generated image, the clearer and higher the resolution, which is obviously expected by the generated model.

### 4.4. Experiment Results

#### 4.4.1. Experiment Results of Prevailing and Proposed Methods on Google Maps Dataset

In the initial phase, we conduct 200 rounds of training. The training results show that after approximately 100 rounds, all indicators display minimal fluctuations, indicating convergence. Therefore, we present the results and various parameter metrics after 100 rounds of training. To ensure the completeness of the model, we conduct forward and reverse experiments for each set of methods. This means that the model should be able to transform from satellite map-style class-A data to navigation map-style class-B-style data and vice versa. Table 1 presents the parameter values for the various methods used in the four comparative experiments. *IS-B* refers to the inception score of the generated class-B-style data in the forward process (satellite map to navigation map). *IS-A* refers to

the inception score of the generated class-A-style data in the reverse process (navigation map to satellite map).

First, from Table 3, the combination of CycleGAN with SWAD performs best than SWA and other methods. Compared to SWA, when SWAD is deployed on the generators, the *FID* indicator decreases by 55.8% and the *IS-B* indicator has increased by 28.9%. It means that the gap of SWAD between the generated data and real data decreases significantly compared to SWA or SGD. At the same time, the quality of both A-class-style and B-class-style generated data improves considerably, which is a favorable outcome for GANs. Moreover, when PSWA [37] is deployed on both the generators, it improves the *IS-A* indicator, but it performs less well on other indicators such as *FID*.

**Table 3.** The performance indicators based on prevailing and proposed methods on test dataset.

Methods	<i>FID</i>	Decreasing Rate of <i>FID</i> <sup>1</sup>	<i>IS-A</i>	<i>IS-B</i>
SWA	194.970	0	4.328	2.334
SWA (generators and discriminators) <sup>2</sup>	331.962	−70.3%	3.897	2.097
PSWA	246.689	−26.5%	4.756	1.796
SGD	89.147	54.3%	3.696	3.004
SWAD	86.274	55.8%	3.892	3.008

<sup>1</sup> Using SWA (only generators) as a reference. <sup>2</sup> Other methods only use SWA for generators by default.

Second, the method works better when deployed on generators than deployed on both generators and discriminators. When SWAD is deployed only on the generators, the gap between generated data and real data decreases significantly compared to the previous experiments. Although it is still larger than the baseline, the quality of generated A-class data is even higher, and the quality of generated B-class data also improves considerably. This indicates that deploying SWAD on the generator alone yields better results than deploying it simultaneously. When SWAD is simultaneously deployed on both the generator and discriminator, it amplifies the gap between generated data and real data. This is because the model inherently involves an adversarial iterative process, where the generator and discriminator compete with each other to achieve training effectiveness. Deploying the optimization method on both sides inevitably leads to one side being weaker and the other stronger, which is clearly not conducive to training. Hence, the elevated value of this indicator is understandable.

Third, compared to SGD, the indicators of SWA related to the quality of generation are better, which indicates that SWDA improves the resolution of generated images. Additionally, the effect of SWAD on *FID*, *IS-A*, and *IS-B* is higher than SGD, but other SWA variants still fall short of the SGD indicators. This further highlights that the network's adversarial performance is not solely dependent on the optimization method.

Figure 7 demonstrates the loss reduction process of various methods during the training process. Each column corresponds to a different optimization method. Within each column, the top section represents the loss when discriminating A-class images ( $D_A$ ), the middle section represents the loss when discriminating B-class-style images ( $D_B$ ), and the bottom section represents the total loss of the generator. To compare the convergence speed and oscillation level of different optimization methods during training, we perform statistical analysis on the total loss of generator shown at the bottom of Figure 7 and document the results in Table 4. In order to standardize the convergence criteria, we establish the following guidelines:

1. Convergence is considered when the amplitude of the loss curve does not exceed 10% of the maximum loss value, and the training round at this point is identified as the convergence time.
2. We record the oscillation counts equal to or greater than 10% of the maximum loss value to assess the extent of oscillation during training.

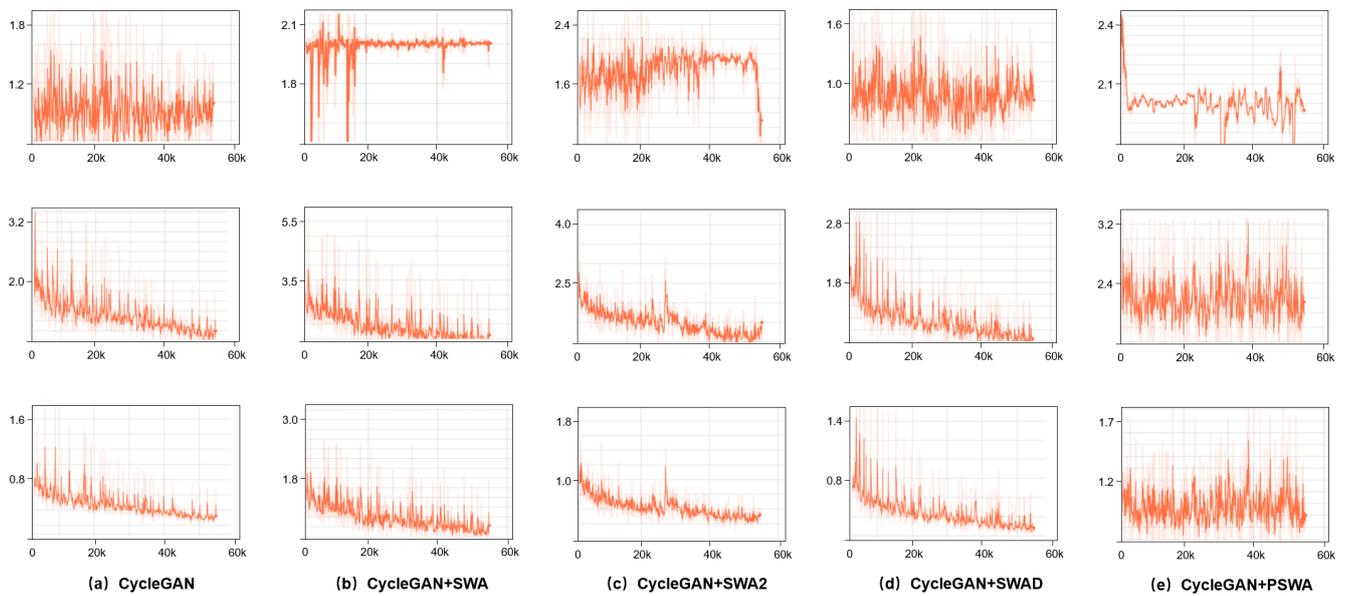
**Table 4.** The convergence rounds and oscillation counts for each optimization method.

Methods	Convergence Rounds	Oscillation Counts
SWA	42 k	40
SWA (generators and discriminators)	35 k	35
PSWA	48 k	51
SGD	36 k	46
SWAD	26 k	36

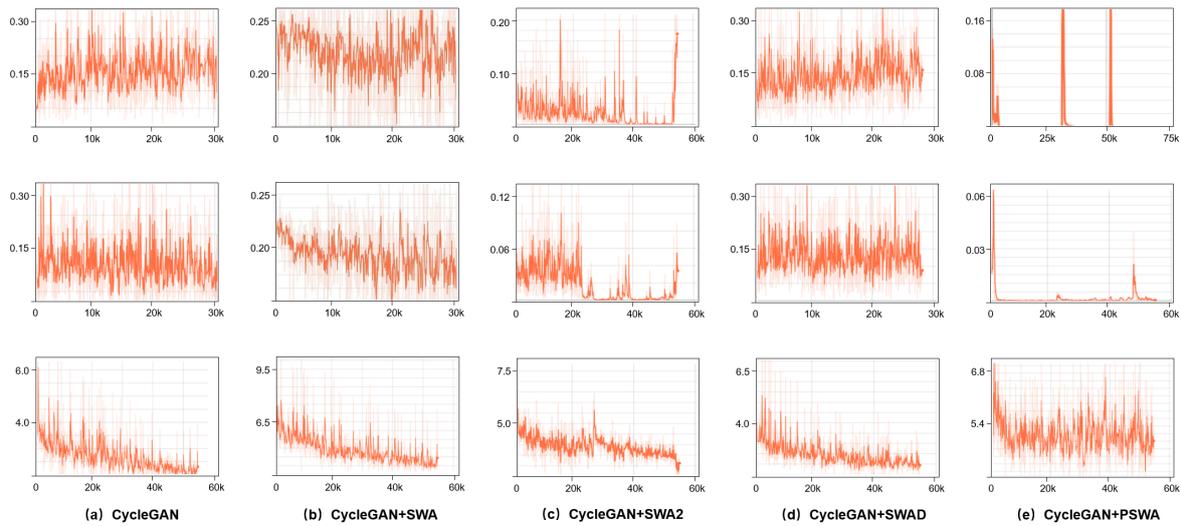
As shown in Table 4, the convergence speed of SWA is nearly identical to that of SGD, but it significantly reduces the degree of oscillation. On the other hand, the PSWA method not only converges slower but also intensifies the oscillation. In contrast, SWAD demonstrates better performance by significantly accelerating the convergence speed and reducing the degree of oscillation, leading to a more stable training process.

Figure 7 indicates that the SWA is more effective in identifying the optimal solution compared to the baseline, enabling it to reach the lowest gradient more rapidly. However, this also leads to a higher generator loss for the same batch compared to the baseline. In panel (d), both the generator and discriminator exhibit a more balanced performance, ultimately surpassing the baseline. The experimental results further demonstrate that the SWA optimization method effectively addresses the two longstanding issues encountered in map-style transfer tasks: time-consuming processes and poor quality.

The total loss of the generator consists of three components:  $G_{GAN_{loss}}$  (composed solely of the loss from the  $G_{A2B}$  and  $G_{B2A}$  generators),  $G_{cycle_{loss}}$ , and  $G_{identity_{loss}}$ . The loss functions for these components are depicted in Figure 8.



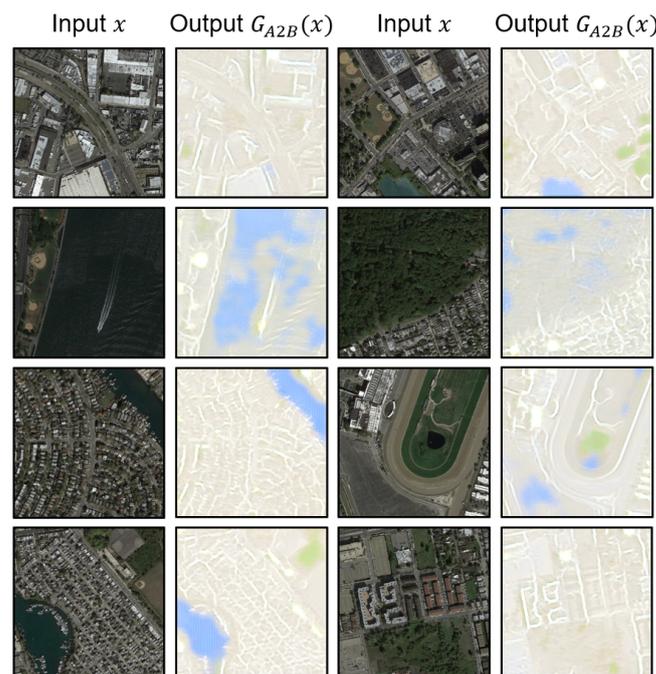
**Figure 7.** Training loss for each optimization method. (Top) loss of discriminator A; (Middle) loss of discriminator B; and (Bottom) total loss of generator.



**Figure 8.** Three components of the total loss in the generator. **(Top)** GAN loss; **(Middle)** cycle loss; and **(Bottom)** identity loss.

#### 4.4.2. Style Transfer Result on Google Maps Dataset

We select urban and off-road environment datasets for group training. After completing the training, we input class-A-style data or class-B-style data into the model to generate the corresponding transformed image. In the following sections, we display and analyze the output results of the PSWA and SWAD methods. These results consist of 16 randomly selected outputs from over a thousand results obtained in urban and off-road environments. The relationship between the input and output is as follows: when the class-A-style test set is inputted, a class-B-style converted image is generated, and when the class-B-style test set is inputted, a class-A-style converted image is generated. Figures 9 and 10 illustrate the forward migration process using the PSWA method, while Figure 11 depicts the reverse migration process using the PSWA method. Additionally, Figures 12 and 13 demonstrate the forward migration process using the SWAD method, and Figure 14 showcases the reverse migration process using the SWAD method.



**Figure 9.** The forward transfer process in urban environments based on PSWA.

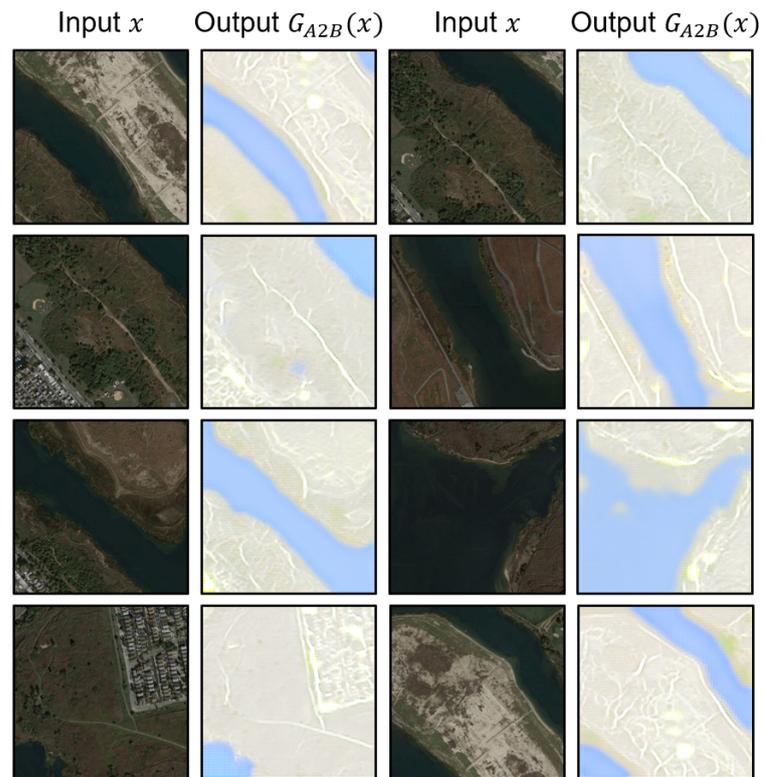


Figure 10. The forward transfer process in off-road environments based on PSWA.

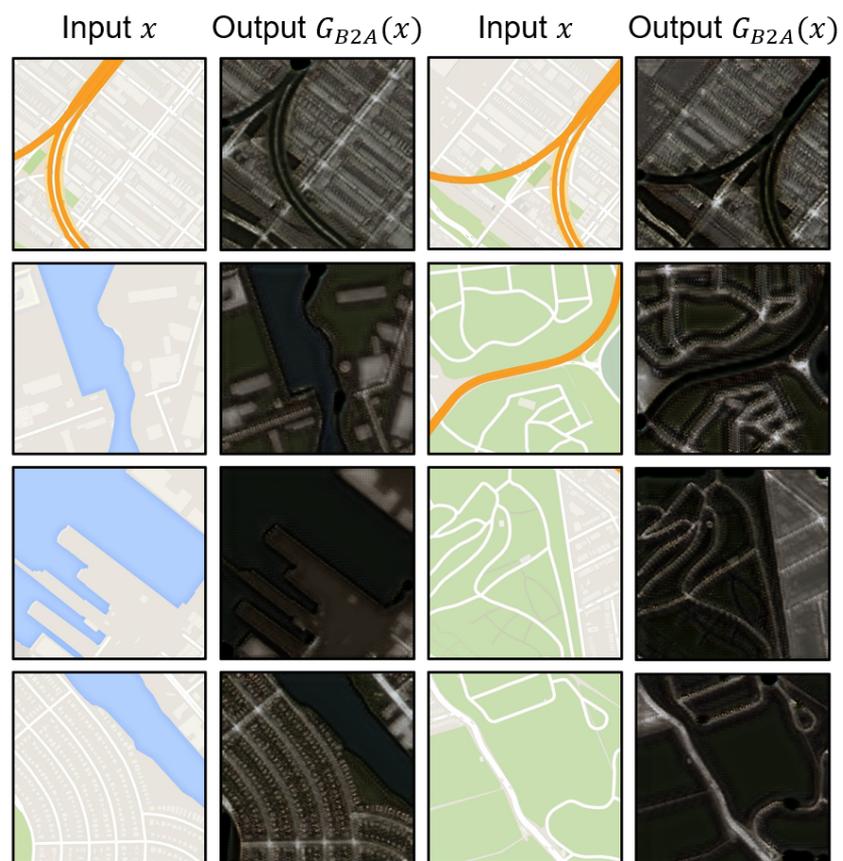


Figure 11. The reverse transfer process based on PSWA.

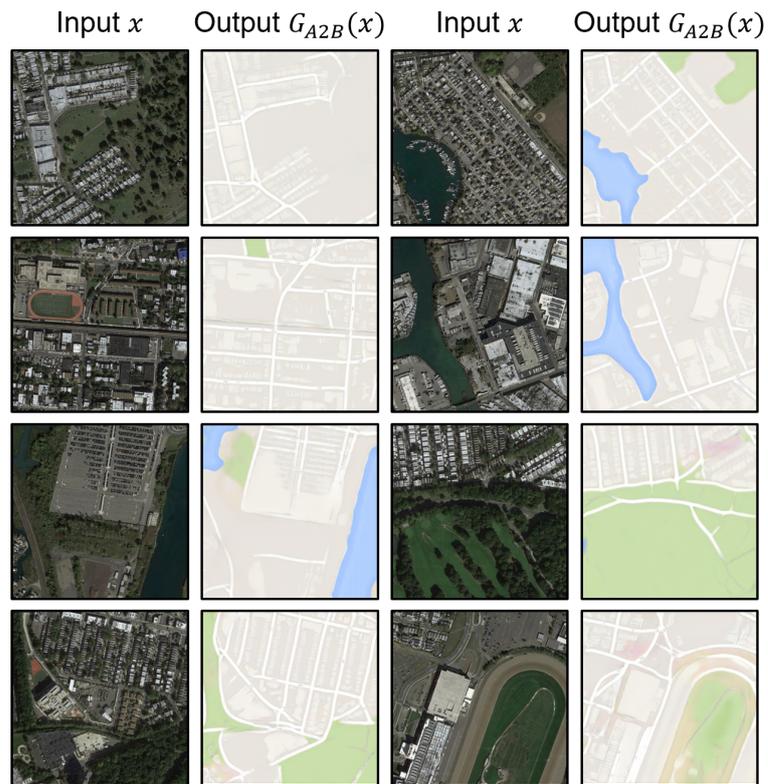


Figure 12. The forward transfer process in urban environments based on SWAD.

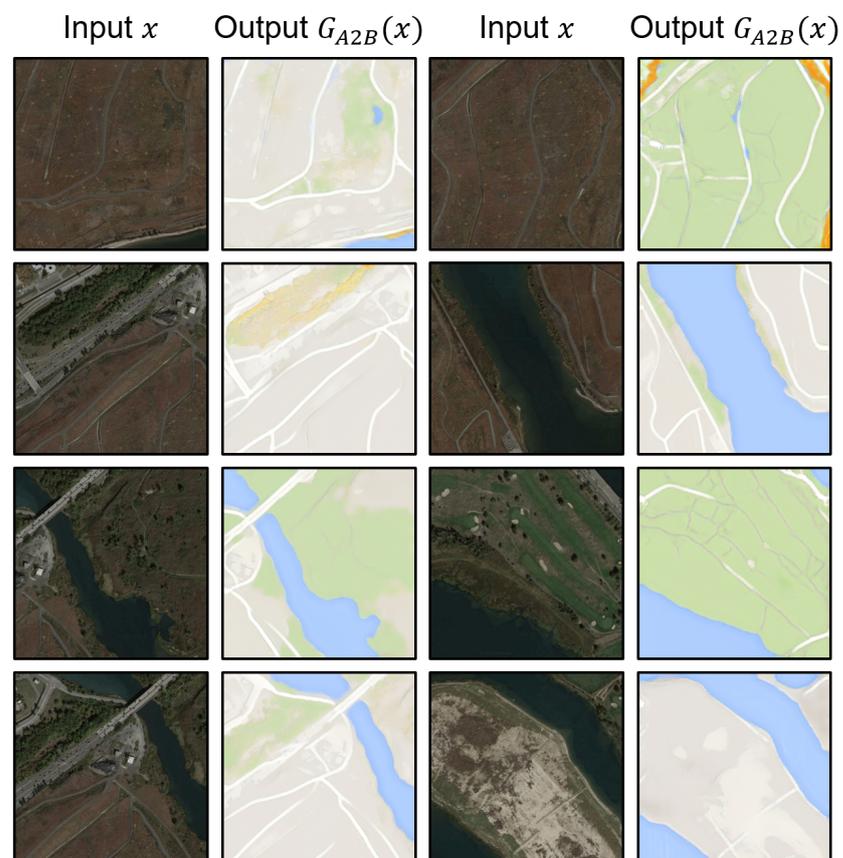
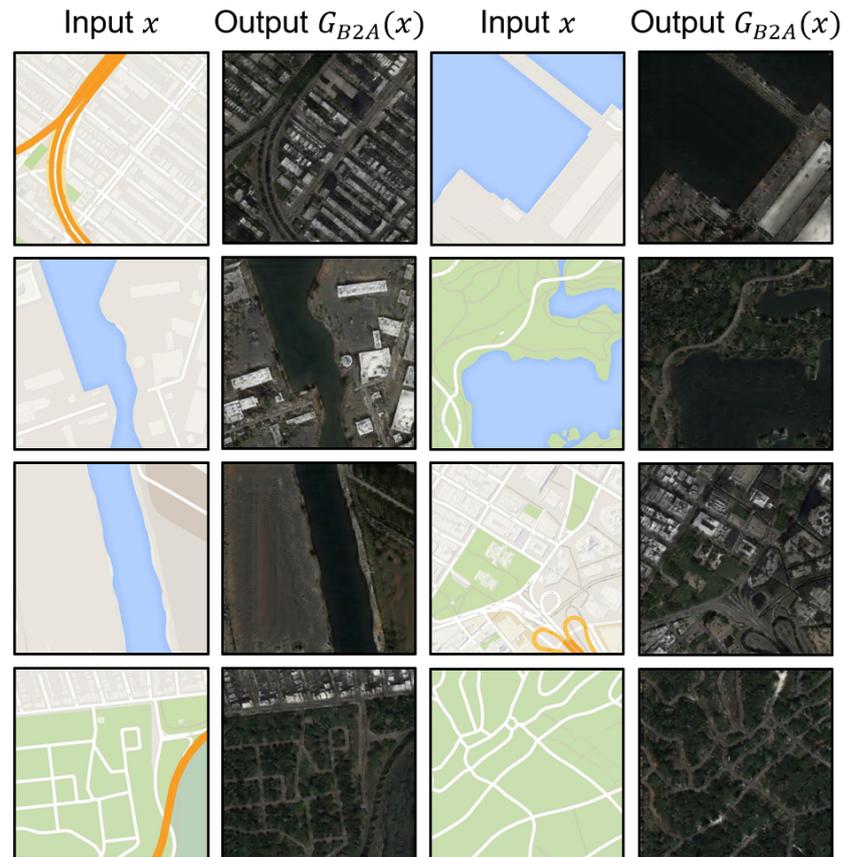


Figure 13. The forward transfer process in off-road environments based on SWAD.



**Figure 14.** The reverse transfer process based on SWAD.

As observed in Figures 9 and 10, it is apparent that the forward migration effect of PSWA is not satisfactory. The map outline drawn appears blurred, and the migration effect of certain key elements is subpar. Overall, this can be attributed to the fact that although PSWA partially addresses the issue of over-fitting in ordinary SWA, it still lacks sufficient training, resulting in a decrease in the clarity of the generated images. Furthermore, the data adaptability to off-road environments is limited, and the mapping accuracy is low, which does not meet the requirements for hybrid mapping in multiple environments. Nevertheless, it is evident that PSWA exhibits a strong resemblance to the source domain, thereby demonstrating its potential to enhance the model's robustness.

Figure 11 illustrates the reverse migration process of the PSWA method. The outlines of the buildings in the map appear to be clearer, although still somewhat blurry. The migration effect is more pronounced for data with significant differences in characteristics, but the distinction between the reverse generation of data with similar characteristics is low. Specifically, there is not a significant differentiation between rivers and green spaces in the data.

From Figures 12 and 13, we can observe that the SWAD method effectively addresses the issue of over fitting without compromising the quality of generation. Furthermore, the clarity of the SWAD method surpasses the generation effect of the PSWA method. In urban environments, distinct features such as vegetation, lakes, and rivers are adequately represented in the remote sensing images and exhibit good transfer results. Boundaries like roads and sidewalks are also well reflected in the navigation maps. In off-road environments, the elements in the remote sensing images are relatively uniform, mainly consisting of deserts or channels. As a result, the transfer effect is good, essentially achieving the goal of generating hybrid-transfer maps between urban and off-road environments.

When inputting class-B-style images, it should be possible to perform a reverse style transfer back to class-A-style images. Therefore, the results of the reverse transfer are shown in Figure 14. It can be observed that although some empty areas in the remote

sensing images can only be filled with houses or other objects generated by the algorithm after the transfer, the overall segmentation lines and block restoration are relatively good, resulting in a satisfactory outcome.

## 5. Conclusions

In this paper, we propose a novel method that averages the parameters of generators sampled from different iterations with the dense sampling strategy in CycleGANs. Two networks of generators converse images from one domain to another. The fusion of generators improves the conversion quality, increases sample diversity, balances generator strengths and weaknesses, and reduces problems such as mode crashes and oscillations. In future work, we will generalize the method under different datasets and scenarios to adapt to migration tasks in different environments. In sum, the fusion of GAN generators is an efficient method to further improve the performance of image conversion in CycleGAN, bringing more diverse and authentic results. However, the computing time and resource consumption will increase significantly in large-scale occasions. Also, the adjustment of parameters may increase the cost of training. The combination of WA and the generation of navigation maps based on style transfer has a potential future. With the development of deep learning and generative models, there may be a more efficient and accurate WA technique of GANs in the future, making navigation image generation more intelligent and realistic. In addition, combined with other technologies such as reinforcement learning and augmented reality, navigation image generation may further explore the fields of application and provide a more rich and interactive navigation experience.

**Author Contributions:** Conceptualization, L.Y. and F.Y.; methodology, F.Y. and W.L.; software, F.Y. and W.L.; validation, F.Y., W.L. and M.H.; formal analysis, F.Y. and W.L.; investigation, Y.D. and M.H.; resources, F.Y. and W.L.; data curation, F.Y.; writing—original draft preparation, F.Y. and W.L.; writing—review and editing, W.L.; visualization, F.Y. and W.L.; supervision, L.Y.; project administration, L.Y.; funding acquisition, L.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available on request from the author Fanxiao Yi (yi\_fanxiao@foxmail.com). The data are not publicly available due to our laboratory's confidentiality agreement and policies.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Huang, C.; Mees, O.; Zeng, A.; Burgard, W. Visual language maps for robot navigation. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 10608–10615.
2. Huang, C.; Mees, O.; Zeng, A.; Burgard, W. Audio visual language maps for robot navigation. *arXiv* **2023**, arXiv:2303.07522.
3. Mao, J.H.; Yang, J.; Shao, R.P.; Wang, W.Z. Research on the construction of a BIM-based model for cross-floor indoor navigation maps. In *Frontiers in Civil and Hydraulic Engineering*; CRC Press: Boca Raton, FL, USA, 2023; Volume 1, pp. 372–378.
4. Grigorescu, S.; Trasnea, B.; Cocias, T.; Macesanu, G. A survey of deep learning techniques for autonomous driving. *J. Field Robot.* **2020**, *37*, 362–386. [[CrossRef](#)]
5. Jiang, Z.; Zhang, X.; Wang, P. Grid-Map-Based Path Planning and Task Assignment for Multi-Type AGVs in a Distribution Warehouse. *Mathematics* **2023**, *11*, 2802. [[CrossRef](#)]
6. Yamaguchi, T.; Kuwano, A.; Koyama, T.; Okamoto, J.; Suzuki, S.; Okuda, H.; Saito, T.; Masamune, K.; Muragaki, Y. Construction of brain area risk map for decision making using surgical navigation and motor evoked potential monitoring information. *Int. J. Comput. Assist. Radiol. Surg.* **2023**, *18*, 269–278. [[CrossRef](#)] [[PubMed](#)]
7. Zhang, Q.; Liu, X. Robot indoor navigation point cloud map generation algorithm based on visual sensing. *J. Intell. Syst.* **2023**, *32*, 20220258. [[CrossRef](#)]
8. Tanwar, J.; Sharma, S.K.; Mittal, M. Designing obstacle's map of an unknown place using autonomous drone navigation and web services. *Int. J. Pervasive Comput. Commun.* **2023**, *19*, 154–169. [[CrossRef](#)]

9. Ma, L.; Liu, Y.; Zhang, X.; Ye, Y.; Yin, G.; Johnson, B.A. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS J. Photogramm. Remote Sens.* **2019**, *152*, 166–177. [[CrossRef](#)]
10. Salvo, C.; Vitale, A. A Remote Sensing Method to Assess the Future Multi-Hazard Exposure of Urban Areas. *Remote Sens.* **2023**, *15*, 4288. [[CrossRef](#)]
11. Wang, L.; Gao, R.; Li, C.; Wang, J.; Liu, Y.; Hu, J.; Li, B.; Qiao, H.; Feng, H.; Yue, J. Mapping Soybean Maturity and Biochemical Traits Using UAV-Based Hyperspectral Images. *Remote Sens.* **2023**, *15*, 4807. [[CrossRef](#)]
12. Jing, Y.; Yang, Y.; Feng, Z.; Ye, J.; Yu, Y.; Song, M. Neural style transfer: A review. *IEEE Trans. Vis. Comput. Graph.* **2019**, *26*, 3365–3385. [[CrossRef](#)]
13. Wang, P.; Li, Y.; Vasconcelos, N. Rethinking and improving the robustness of image style transfer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 124–133.
14. Ayyalasomayajula, R.; Arun, A.; Wu, C.; Sharma, S.; Sethi, A.R.; Vasisht, D.; Bharadia, D. Deep learning based wireless localization for indoor navigation. In Proceedings of the 26th Annual International Conference on Mobile Computing and Networking, London, UK, 21–25 September 2020; pp. 1–14.
15. Fernández, C.; Munoz-Bulnes, J.; Fernández-Llorca, D.; Parra, I.; Garcia-Daza, I.; Izquierdo, R.; Sotelo, M.A. High-level interpretation of urban road maps fusing deep learning-based pixelwise scene segmentation and digital navigation maps. *J. Adv. Transp.* **2018**, *2018*, 2096970. [[CrossRef](#)]
16. Golroudbari, A.A.; Sabour, M.H. Recent Advancements in Deep Learning Applications and Methods for Autonomous Navigation—A Comprehensive Review. *arXiv* **2023**, arXiv:2302.11089.
17. Lee, Y.W.; Kim, J.S.; Park, K.R. Ocular Biometrics with Low-Resolution Images Based on Ocular Super-Resolution CycleGAN. *Mathematics* **2022**, *10*, 3818. [[CrossRef](#)]
18. Xu, C.; Shu, J.; Zhu, G. Multi-Feature Dynamic Fusion Cross-Domain Scene Classification Model Based on Lie Group Space. *Remote Sens.* **2023**, *15*, 4790. [[CrossRef](#)]
19. Singh, S.P.; Jaggi, M. Model fusion via optimal transport. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 22045–22055.
20. Cha, J.; Chun, S.; Lee, K.; Cho, H.C.; Park, S.; Lee, Y.; Park, S. Swad: Domain generalization by seeking flat minima. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 22405–22418.
21. Li, J.; Hong, D.; Gao, L.; Yao, J.; Zheng, K.; Zhang, B.; Chanussot, J. Deep learning in multimodal remote sensing data fusion: A comprehensive review. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *112*, 102926. [[CrossRef](#)]
22. Ghassemian, H. A review of remote sensing image fusion methods. *Inf. Fusion* **2016**, *32*, 75–89. [[CrossRef](#)]
23. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
24. Gatys, L.; Ecker, A.S.; Bethge, M. Texture synthesis using convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 262–270.
25. Gatys, L.A.; Ecker, A.S.; Bethge, M. Image style transfer using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2414–2423.
26. Johnson, J.; Alahi, A.; Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Proceedings, Part II 14; Springer: Cham, Switzerland, 2016; pp. 694–711.
27. Luan, F.; Paris, S.; Shechtman, E.; Bala, K. Deep photo style transfer. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4990–4998.
28. Li, Y.; Fang, C.; Yang, J.; Wang, Z.; Lu, X.; Yang, M.H. Universal style transfer via feature transforms. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 385–395.
29. Li, Y.; Liu, M.Y.; Li, X.; Yang, M.H.; Kautz, J. A closed-form solution to photorealistic image stylization. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 453–468.
30. Yoo, J.; Uh, Y.; Chun, S.; Kang, B.; Ha, J.W. Photorealistic style transfer via wavelet transforms. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9036–9045.
31. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2223–2232.
32. Li, B.; Li, P.; Liu, B.; Li, M. A High-Precision Underwater Target Detection Method Based on Cascade Neural Network and Edge Computing. *CN116758406A*, 15 September 2023.
33. Izmailov, P.; Podoprikin, D.; Garipov, T.; Vetrov, D.; Wilson, A.G. Averaging weights leads to wider optima and better generalization. *arXiv* **2018**, arXiv:1803.05407.
34. Son, D.M.; Kwon, H.J.; Lee, S.H. Enhanced Night-to-Day Image Conversion Using CycleGAN-Based Base-Detail Paired Training. *Mathematics* **2023**, *11*, 3102. [[CrossRef](#)]
35. Krstanović, L.; Popović, B.; Janev, M.; Brkljač, B. Feature Map Regularized CycleGAN for Domain Transfer. *Mathematics* **2023**, *11*, 372. [[CrossRef](#)]
36. Chen, H.; Lundberg, S.; Lee, S.I. Checkpoint ensembles: Ensemble methods from a single training process. *arXiv* **2017**, arXiv:1710.03282.
37. Guo, H.; Jin, J.; Liu, B. Stochastic weight averaging revisited. *Appl. Sci.* **2023**, *13*, 2935. [[CrossRef](#)]

38. Garipov, T.; Izmailov, P.; Podoprikin, D.; Vetrov, D.P.; Wilson, A.G. Loss surfaces, mode connectivity, and fast ensembling of dnns. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 8789–8798.
39. Huang, G.; Li, Y.; Pleiss, G.; Liu, Z.; Hopcroft, J.E.; Weinberger, K.Q. Snapshot ensembles: Train 1, get m for free. *arXiv* **2017**, arXiv:1704.00109.
40. Neklyudov, K.; Molchanov, D.; Ashukha, A.; Vetrov, D. Variance networks: When expectation does not meet your expectations. *arXiv* **2018**, arXiv:1803.03764.
41. Mandt, S.; Hoffman, M.D.; Blei, D.M. Stochastic gradient descent as approximate bayesian inference. *arXiv* **2017**, arXiv:1704.04289.
42. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [[CrossRef](#)]
43. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
44. Robbins, H.; Monro, S. A stochastic approximation method. *Ann. Math. Stat.* **1951**, *22*, 400–407. [[CrossRef](#)]
45. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
46. Obukhov, A.; Krasnyanskiy, M. Quality assessment method for GAN based on modified metrics inception score and Fréchet inception distance. In *Proceedings of the Software Engineering Perspectives in Intelligent Systems: Proceedings of 4th Computational Methods in Systems and Software 2020*; Springer: Cham, Switzerland, 2020; Volume 14, pp. 102–114.
47. Chong, M.J.; Forsyth, D. Effectively unbiased fid and inception score and where to find them. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6070–6079.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.