electronics

MDPI

Article

RSCAN: Residual Spatial Cross-Attention Network for High-Fidelity Architectural Image Editing by Fusing Multi-Latent Spaces

Cheng Zhu [1,2], Guangzhe Zhao [1,2], Benwang Lin [1,2], Xueping Wang [1,2] and Feihu Yan [1,2,*]

[1] School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing 100044, China; 2108550021044@stu.bucea.edu.cn (C.Z.); zhaoguangzhe@bucea.edu.cn (G.Z.); linbenwang@stu.bucea.edu.cn (B.L.); wangxueping@bucea.edu.cn (X.W.)
[2] Beijing Key Laboratory of Robot Bionics and Function Research, Beijing 100044, China
* Correspondence: yanfeihu@bucea.edu.cn

Abstract: Image editing technology has brought about revolutionary changes in the field of architectural design, garnering significant attention in both the computer and architectural industries. However, architectural image editing is a challenging task due to the complex hierarchical structure of architectural images, which complicates the learning process for the high-dimensional features of architectural images. Some methods invert the images into the latent space of a pre-trained generative adversarial network (GAN) model, completing the editing process by manipulating this latent space. However, the task of striking a balance between reconstruction fidelity and editing efficacy through latent space mapping presents a formidable challenge. To address this issue, we propose a Residual Spatial Cross-Attention Network (RSCAN) for architectural image editing, which is an encoder model integrating multiple latent spaces. Specifically, we introduce the spatial feature extractor, which maps the image to the high-dimensional space F of the synthesis network, to enhance the spatial information retention and preserve the structural consistency of the architectural image. In addition, we propose the residual cross-attention to learn the mapping relationship between the low-dimensional space W and F space, generating modified features corresponding to the latent code and leveraging the benefits of multiple latent spaces to facilitate editing. Extensive experiments are performed on the LSUN Church dataset, and the experimental results indicate that our proposed RSCAN achieves significant improvements over the relevant methods in quantitative analysis metrics including the reconstruction quality, SSIM, FID, L2, LPIPS, PSNR, and editing effect ΔS, with enhancements of 29.49%, 17.29%, 8.81%, 11.43%, 11.26%, and 47.8%, respectively, thereby enhancing the practicality of architectural image editing.

Keywords: deep learning; image editing; architectural image; generative adversarial network; GAN inversion; latent space

Citation: Zhu, C.; Zhao, G.; Lin, B.; Wang, X.; Yan, F. RSCAN: Residual Spatial Cross-Attention Network for High-Fidelity Architectural Image Editing by Fusing Multi-Latent Spaces. *Electronics* **2024**, *13*, 2327. https://doi.org/10.3390/electronics13122327

Academic Editor: Ping-Feng Pai

Received: 13 May 2024
Revised: 4 June 2024
Accepted: 10 June 2024
Published: 14 June 2024

1. Introduction

Architectural images, as an important type of imagery, showcase the appearance and structure of architecture, possessing rich cultural and artistic value. With the development of deep learning and the increasing demand for digital tools in the construction industry, architectural image editing has emerged as a pivotal technology in digital architecture. By editing images, designers can achieve personalized modifications to the appearance, style, and interior layout of a building [1,2], thereby providing greater flexibility and creativity for project development and aesthetics. Furthermore, architectural image editing not only deepens our understanding and appreciation of the diversity of architectural art but also promotes deeper reflection on the cultural, historical, and social contexts of architectural styles. The evolution of generative adversarial networks (GANs) [3] has empowered automated architectural image editing, significantly impacting architecture

and urban planning. This automation greatly enhances the efficiency of architectural design and urban planning processes. Additionally, it enhances the characteristics of architectural styles, meets specific aesthetic needs, and promotes the dissemination and exchange of architectural art. However, challenges persist in architectural image editing, particularly in achieving precise edits within intricate and diverse structures while preserving the structural integrity and coherence. Addressing these challenges is crucial to advancing the application of architectural image editing and elevating the editing standards.

In recent years, architectural image editing methods have shifted from style transfer [4] and image translation [5] to GAN inversion. Although the methods of style transfer and image translation have achieved certain results, they are limited in their ability to manipulate only a subset of attributes within a single model, often resulting in imprecise control where only broad changes in color and overall style are achievable. With the improvement of the GAN generation quality, the use of pre-trained GAN models can effectively solve the above problems. The StyleGAN framework proposed by Karras [6] achieves advanced generation performance and can generate images with latent style vectors. Based on StyleGAN's excellent generation capabilities and powerful control capabilities, GAN inversion technology [7] has become popular, aiming to map images into the latent spaces of pre-trained GAN generators. Manipulating the latent space of StyleGAN can cause corresponding changes in the image without training additional networks. Using the powerful control capabilities of StyleGAN to manipulate real images represents a promising direction.

Inversion to different latent spaces affects the reconstruction quality and editing effect. Existing methods usually invert images into the W, $W+$, and F spaces. Karras [6] confirmed that the low-dimensional space W cannot facilitate the complete and accurate reconstruction of images. The GAN inversion problem can be viewed as a lossy data compression system [8]. According to rate distortion theory [9], inverting real-world images into low-dimensional latent codes will inevitably lead to information loss. According to information bottleneck theory [10], it is speculated that, since deep compression models tend to retain the common information of a domain, the lost information is mainly image-specific details. Therefore, some approaches invert images into higher-dimensional spaces [11] to preserve more spatial structural information. However, high-frequency details will be attached to the image during reconstruction and cannot be removed during editing, worsening the editing effect. Recent work mainly focuses on methods that invert images into multi-latent spaces [12]. However, the methods proposed by Li [13] involve complex steps, making training difficult. These methods still struggle to balance the reconstruction quality and editing effectiveness, rendering architectural image editing tasks challenging for practical application.

To solve the challenge of balancing the reconstruction quality and editing effect of intricate architectural images, we propose a Residual Spatial Cross-Attention Network (RSCAN), which inverts images into a fused latent space, reducing the drawbacks associated with separate latent spaces. The RSCAN comprises a spatial feature extractor module, a residual cross-attention module, and a synthesis network module. Compared to facial images, architectural images have more complex structures, making precise reconstruction crucial. Our approach seeks to capture the spatial structural details in the F space and facilitate effective editing in the W space. Thus, we fuse the two spaces while taking into account the advantages of both. For this purpose, we design a multi-level feature extraction network to align the image into the F space step by step. To ensure that manipulations performed in the W space do not affect the F space, we must learn the mapping relationship from the W space to the F space. Thus, we introduce the residual cross-attention module, where the characteristic of cross-attention allows one sequence to focus on another sequence. We extract the feature f from the F space and set them as query Q, and we set the variation Δw in the W space as the key K and the value V. Ultimately, the variations in the W space will guide the changes in the F space, facilitating the fusion of the two spaces. In order to achieve better alignment at the feature level and learn the correct W space manipulation changes, our training method is designed as self-supervised training, using the output of

the synthesis network as the input of the RSCAN and feeding the final predicted features back into the synthesis network. To avoid being limited to predefined edits, during training, we simulate edits by randomly inserting sampled latent codes.

Extensive experiments on the LSUN Church dataset show that our proposed RSCAN significantly outperforms related methods in key quantitative metrics, including the reconstruction quality (SSIM, FID, L2, LPIPS, PSNR) and editing effect (ΔS). Figure 1 shows the visual editing results of our method, achieving satisfactory effects in each attribute of editing. This verifies the successful fusion of the two spaces by our method, allowing the F space to attend to changes in the W space and enabling the image to undergo correct editing. Therefore, our method enhances the practicality of architectural image editing. Our main contributions can be summarized as follows.

1. We propose a multi-level spatial feature extractor module to map the image to the F space of the synthesis network, which enables us to more accurately reconstruct architectural images with many line details.
2. We fuse multi-latent spaces, including the high-dimensional feature space F, which excels in reconstruction, and the low-dimensional space W, which excels in editing, through the residual cross-attention module. By learning the mapping relationship from the W space to the F space, manipulations made in the W space preserve the original editing effects while ensuring correct changes in the features of the F space.
3. The self-supervised training method that we design can map images to the F space more rapidly and learn the correct direction of the W space variation for F space changes. On the LSUN Church dataset, our method outperforms existing methods in both qualitative and quantitative evaluations.

The structure of this paper is as follows. Section 2 reviews related work, while Section 3 elaborates on our proposed high-fidelity architectural image editing method. Section 4 details the experimental settings, including the datasets and evaluation metrics, and presents and analyzes the experimental results both quantitatively and qualitatively. Finally, Section 5 concludes the paper and discusses future work.

Figure 1. The inversion and editing results of our method RSCAN. The input image, inversion results, and editing results are displayed from left to right in each row. This approach achieves precise reconstruction and accurate editing effects when adding Gothic, Greek, and Byzantine architectural styles and cloud and glass elements.

2. Related Work

This section first introduces the definition and challenges of architectural image editing and presents the classification of the editing methods, including those based on neural style transfer, image translation, GAN inversion, and multimodal diffusion models. It then elaborates on the related techniques in the latent space of the GAN inversion method used in this study, points out the shortcomings of existing methods, and establishes research

objectives accordingly. Finally, it provides a detailed introduction to the residual network and cross-attention module techniques used in the proposed method.

2.1. Architectural Image Editing

Architectural image editing involves the automated manipulation of images through computer algorithms. It entails taking an original image as input and applying algorithmic calculations to generate a modified version. Architectural images, characterized by buildings and structures foregrounded against backgrounds of sky and vegetation, present unique challenges for such editing processes. Currently, research approaches to architectural image editing can be categorized into four main types: those based on neural style transfer, image translation, GAN inversion, and multi-modal diffusion models.

Since Gatys et al. [14] proposed style transfer, many methods using style transfer to edit architectural images have emerged. Luan et al. [4] proposed a photographic style transfer method that can produce realistic style transfer effects in outdoor scenes and indoor scenes, including the transfer of time, weather, seasons, and light. Chen et al. [15] used a segmentation network to separate the foreground and background of building images; it can convert a building image into dusk, early morning, evening, or noon. However, style transfer struggles to provide precise control over singular attributes, yielding only broad-scale color style conversions.

Image editing methods using image translation techniques, which aim to learn mapping relationships from the source domain to the target domain, can also produce satisfactory results. For architectural scene images, Sangkloy's Scribbler [5] takes indoor scene sketches as inputs and colors them under the guidance of user-specified strokes. Jiang et al. [1] proposed a two-step image translation framework that can convert sketches and architectural images into each other. However, editing methods using image translation often only support the conversion and editing of one or several attributes. If the editing of other attributes is required, another model needs to be trained.

The editing method based on GAN inversion [16] solves the above problems very well. The purpose of GAN inversion is to invert a given image back into the latent space of a pre-trained GAN model generator. The generator can then accurately reconstruct the image from the reverse code. GAN inversion involves manipulating real images by identifying controllable directions in the latent space, obviating the necessity for dedicated paired supervision data to train an independent network. Su et al. [17] deleted the low-level generator module, mapped the sketch directly to the middle layer of the generator, and realized the editing of the building through the sketch. In addition, Alaluf et al. [18] and Dinh et al. [19] achieved more accurate reconstruction by training a smaller network to generate weights for StyleGAN. However, the existing GAN inversion methods are still insufficient to accurately reconstruct architectural images.

Recently, diffusion models have demonstrated high-quality image generation based on text inputs [20]. These models denoise randomly sampled images through multiple iterations to generate realistic images. The image editing method based on multi-modal diffusion [21] involves editing images using a text-guided diffusion model, which provides better generation quality but also requires longer generation times and more expensive equipment, while still lacking in the controllability of the structure.

We have selected GAN inversion as the focus for our architectural image editing research. GAN inversion offers rapid computation and considerable flexibility. However, the current GAN inversion methods struggle in achieving a trade-off between the reconstruction quality and editing effectiveness. Addressing and improving upon these limitations has the potential to markedly increase the utility of architectural image editing.

2.2. Latent Space for GAN Inversion

GANs encompass various latent spaces, and, by manipulating the latent vectors within them, one can alter the generated images. Mapping images to different latent spaces yields distinct editing outcomes. In recent years, GAN inversion research has focused on these

latent spaces, and the design of mapping methods is crucial in addressing the trade-off between the reconstruction quality and editing effects in image editing. Table 1 presents a comparison of the GAN inversion methods utilizing different strategies in recent years.

GANs typically map values sampled from a simple distribution (such as a normal or uniform distribution) to generated images. This simple distribution is referred to as the Z space, which is applicable to all unconditional GAN models, such as BigGAN [3] and StyleGANs [6]. However, the constraint of the Z space, primarily based on a normal distribution, limits its representation capabilities and disentanglement of semantic attributes [22]. Apart from the general Z spaces of GANs, there also exist specialized latent spaces designed for StyleGANs [7], such as the W space, $W+$ space, S space, and F space. As StyleGAN models achieve advanced GAN image synthesis, the most advanced GAN inversion methods are conducted in the latent space of StyleGAN.

Table 1. Comparison of the proposed method with other GAN inversion image editing methods. The type includes learning-based (L), optimization-based (O), and hybrid (H).

Method	Publication	Type	Latent Space	Details	Weaknesses
Image2StyleGAN [23]	ICCV'19	O	W	Using optimization to embed images into the W+ space.	Time complexity is high, reconstruction quality is poor.
mGANPrior [22]	CVPR'20	O	Z	Invert images to the Z space and propose adaptive channel adjustment to improve reconstruction.	Artifacts are generated during editing.
PSP [24]	CVPR'21	L	W+	Using encoder to extract features and map them to the W+ space.	Poor reconstruction quality and the W+ space is far from the W space, losing a large number of editing effects.
E4E [16]	TOG'21	L	W+	Inversion to the W+ space uses adversarial training to position the W+ vectors closer to the W space.	The reconstruction quality is low.
StyleSpace [25]	CVPR'21	O	S	Explores the S space and proposes a method for the detection of decoupled control channels.	The S space still has difficulty in improving the reconstruction quality and reducing editing effects.
BDInvert [26]	ICCV'21	O	W+, F	Proposed a GAN inversion method for the F/W+ space.	Long computation time, does not support large-scale editing such as structure and pose.
PTI [27]	TOG'22	H	W+	Fine-tuned the generator and inverted it to the W space for reconstruction.	Long computation time, requires re-tuning for each input, and the tuning damages the generation quality.
HyperStyle [18]	CVPR'22	H	W	Optimized the modulation generator weights.	Reconstruction quality is greatly improved, but a large number of editing effects are lost.
HyperInverter [19]	CVPR'22	L	W	Inverted to the W space, using a hypernetwork to predict residual weights and restoring lost image details.	Reconstruction quality improves, but predicted weights are difficult to associate with the W space.
HFGI [8]	CVPR'22	L	W+, F	Achieved image-specific detail retention and editing using a distortion consultation branch.	Features retain too much spatial dependency, causing severe artifacts.
StyleRes [28]	CVPR'23	L	W, F	Learned residual features, using cycle consistency loss to learn feature editing transformation.	Too many encoders are designed, making training difficult and causing artifacts.
CLCAE [29]	CVPR'23	L	W, W+, F	Aligned images with the W space using contrastive learning, transforming W vectors to the W+ and F spaces with cross-attention.	Reconstruction is incomplete, guided by the W space to reconstruct W+ and F spaces, reducing editing effects.
Kai [12]	WACV'24	L	Z, F	Extended the Z space to Z+ and integrated it into advanced inversion algorithms such as F/W+.	The Z+ space is a lower-dimensional form of the W+ space, losing reconstruction and editing quality.
GradStyle [13]	arxiv'24	L	W+, F	Computed residual features, using selective attention mechanisms to align these details.	Original features focus on changes in editing features, making it more difficult to learn in two high-dimensional spaces.
Ours	2024	L	W, F	Extracted image features to the F space, using cross-attention to learn the variation values of the W space.	Cross-attention and modulation convolution use different calculation methods, making W space transfer incomplete.

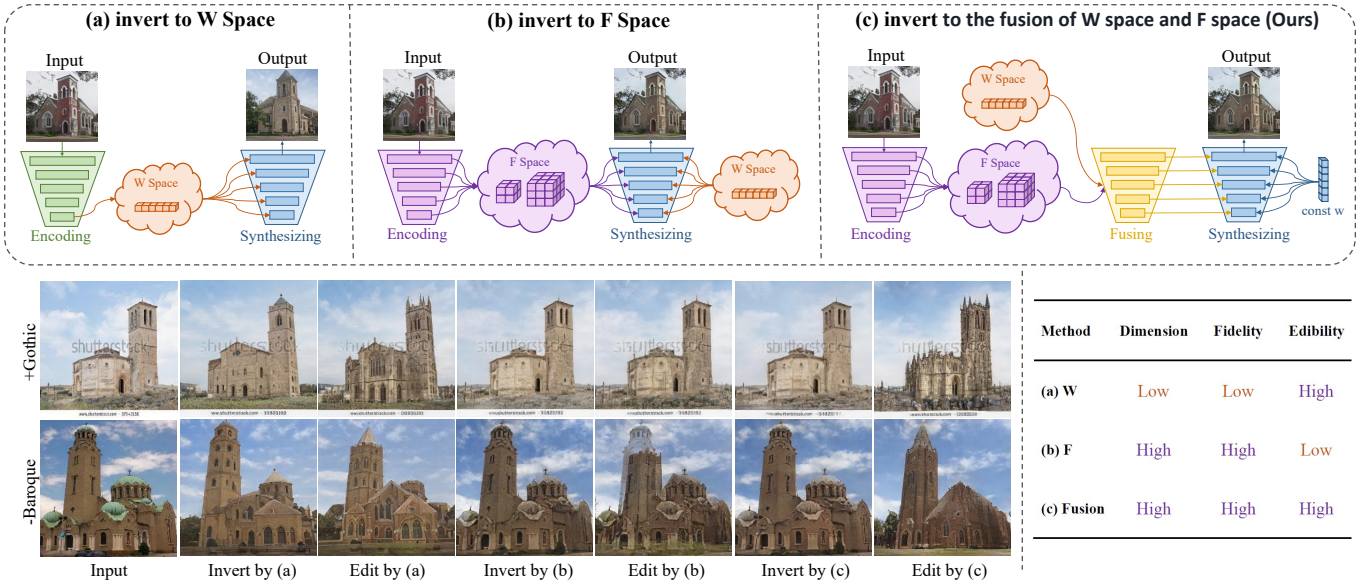
StyleGAN consists of a mapping network and a synthesis network. The mapping network uses an 8-layer multilayer perceptron (MLP) to map the z vector into a style vector w. The space where w is located is called the W space [23]. After changes, the W space

does not obey a certain known distribution, and it can better describe the distribution of the learned dataset and show more disentanglement [24]. However, the W space contains less information, so some methods extend the W space to the $W+$ space [24,27]. The $W+$ space applies different latent codes w to different layers of StyleGAN. Inverting the image to the $W+$ space can reduce the distortion, but it will also reduce the editing performance. The style space S [25] consists of style parameters s, where s is the style code obtained by the affine transformation of each latent code w, and its dimension is the same as w. Mapping to the S space incurs less distortion compared to the $W+$ space but sacrifices the editing performance; such distortion still cannot guarantee the structural consistency of the architectural image.

StyleGAN also has a feature space F with a wider dimension, which is composed of the spatial features f obtained after each layer of convolution of the synthesis network. The F space is a hierarchical space with dimensions surpassing those of the image itself, enabling the representation of high-frequency details. Song et al. [11] utilized a feature extractor similar in structure to the discriminator to achieve image reconstruction. Mapping to the F space will lose less information than mapping to the W, $W+$, and S spaces so that the image structure and details are best preserved, but this preservation also further reduces the editing ability.

Moreover, certain methods [26] aim to invert images into diverse latent spaces simultaneously, thereby enhancing both the reconstruction quality and editing effects. HFGI [8] treats the difference between the reconstructed image in the W space and the input image as overlooked specific information, projecting it into a higher-dimensional space for improved reconstruction. StyleRes [28] allows low-dimensional latent codes to transform high-dimensional features through period consistency constraint models. CLCAE [29] inverts the image into the W space and uses cross-attention to guide the W space to be optimized towards the $W+$ space and F space. These attempts not only further optimize the reconstruction quality, but also have better editing effects. Nonetheless, these methods still rely on inversion to the W space. They typically begin with an initial inversion and reconstruction in the W space and then learn the distortion between the source image and the reconstructed image to refine the details in the F space. During editing, the F space features must be aligned with the W space to prevent artifacts. Overall, these methods are complex, involving three steps that lead to insufficient space fusion, poor reconstruction quality, and artifacts during editing.

We propose a more efficient approach by mapping images directly into the F space, bypassing the initial inversion in the W space. By learning how changes in the W space affect the F space, we achieve transformations directly in the F space, simplifying the process. Unlike traditional methods that perform reconstruction and editing simultaneously in the W space, our method inputs only the changes from the W space (excluding reconstruction information), allowing the W space to focus on editing and the F space to focus on reconstruction. This division of labor simplifies network training and accelerates the fusion of multiple latent spaces. Figure 2 illustrates the advantages and disadvantages of inverting images into the W space and the F space and demonstrates how fusing both spaces can combine their strengths to achieve high-fidelity architectural image editing.



Method	Dimension	Fidelity	Edibility
(a) W	Low	Low	High
(b) F	High	High	Low
(c) Fusion	High	High	High

Figure 2. Comparison of methods for inversion to different latent spaces. When images are inverted to the *W* space, as shown in (**a**), the reconstructed images have structural distortion but decent editing effects. As shown in (**b**), more spatial information can be learned in the *F* space, but it may produce artifacts during editing. As shown in (**c**), choosing a method that fuses the spaces can achieve good results in both aspects.

2.3. Residual Network and Cross-Attention Mechanism

The residual network (ResNet) [30] is a deep neural network architecture that introduces skip connections to address the degradation problem in the training of deep networks. Skip connections allow gradients to propagate directly to earlier layers, mitigating the vanishing gradient problem and facilitating the training of deeper networks. ResNet enhances the feature extraction capabilities and network expressiveness through more complex topological structures and connection methods. We opt for the residual network to design our spatial feature extractor, which can better learn image features, and the concept of skip connections is utilized throughout the overall model design, accelerating model convergence.

The cross-attention mechanism is an attention mechanism that is widely used in the Transformer architecture [31] to capture the correlations between two different sequences, enabling richer information interaction. DETR [32] introduced a Transformer structure to achieve end-to-end object detection, where the cross-attention mechanism facilitates information interaction between image features and object queries, thereby enhancing the detection performance. CrossViT [33] uses the cross-attention mechanism to fuse image features at different scales, achieving excellent results in image classification tasks. This mechanism enhances the integration of multi-scale features by introducing interactions between feature maps at different scales. In image editing tasks, the cross-attention mechanism can be used to establish connections between different representations in the latent space, merging multiple latent spaces to improve the quality and consistency of the generated images. For instance, CLCAE [29] attempts to fuse two spaces using the cross-attention mechanism, but primarily performs inversion in the low-dimensional latent space, with only guidance in the high-dimensional latent space, resulting in limited practical improvements. By combining residual networks with the cross-attention mechanism, we can simultaneously extract rich image features and focus on the changes in the features induced by the editing vectors, achieving high-fidelity architectural image editing and improving the precision and effectiveness of the editing process.

3. Method

In this section, we describe the proposed high-fidelity architectural image editing method. We first present an overview and the objectives of the model. Subsequently, we elaborate on our proposed spatial feature extractor module and residual cross-attention module. Lastly, we provide detailed information about the training process.

3.1. Overview

Figure 3 illustrates an overview of the proposed method RSCAN, which comprises three main components: a spatial feature extractor module, a residual cross-attention module, and a synthesis network module.

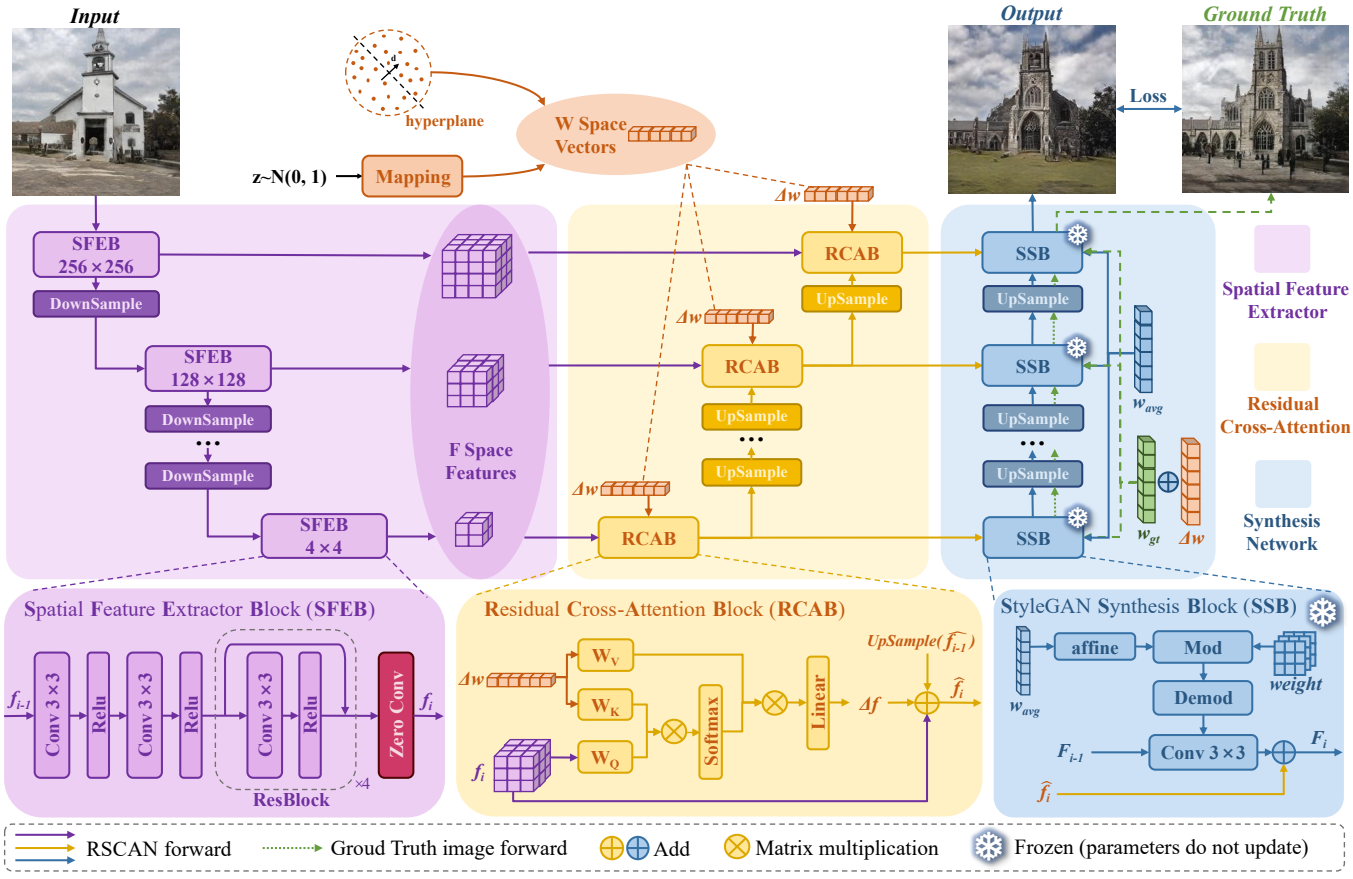


Figure 3. The overall framework of RSCAN, which consists of a spatial feature extractor module, a residual cross-attention module, and a synthesis network module. The input to the spatial feature extractor module is the image, while the input to the residual cross-attention module is the extracted spatial features and the variation Δw in the w vector. Finally, we residually connect the sum of the outputs of the two modules to the synthesis network to generate the output image. The synthesis network is a pre-trained StyleGAN2 synthesis network with frozen parameters.

The spatial feature extractor module extracts features of different dimensions for the high-quality reconstruction of architectural images. These features are then input into the cross-attention module, along with changes in different latent vectors w, to obtain the predicted feature variations. Finally, the predicted residual feature $\hat{f}$ is connected to the synthesis network module with a locked parameter, and the final image is output through the synthesis network.

Specifically, given randomly sampled Gaussian noise z, inputting it into the mapping network M of StyleGAN yields a latent code $w_{gt} = M(z)$. Modifying the latent code w_{gt} to obtain w_{edit} results in a change value Δw. By inputting w_{gt} and w_{edit} separately

into the synthesis network, one obtains an image and its edited version, denoted as I_{gt} and I_{edit}, respectively. Inputting I_{gt} to the spatial feature extractor module produces feature f. Feature f is aligned in the F space during training in the synthesis network. f and the change in latent code Δw serve as inputs to the residual cross-attention module. Subsequently, feature f is set as the query, and Δw is set as the key and value. The module outputs the sum of feature f and the change in feature Δf, denoted as $\hat{f}$. Finally, $\hat{f}$ is input into the synthesis network, where the latent code is the average latent code w_{avg}. The objective is to learn a function R, such that, given inputs I_{gt} and Δw, it outputs the edited image I_{edit} of I_{gt}. The overall optimization objective is as follows:

$$\min_{R}\big(G\big(w_{avg}, R\big(G\big(w_{gt}\big), \Delta w\big)\big), G\big(w_{gt} + \Delta w\big)\big). \tag{1}$$

3.2. Spatial Feature Extractor Module

As shown in the bottom-left corner of Figure 3, the spatial feature extractor module F is constructed as a pyramid structure from spatial feature extractor blocks, which maps the image to the F space of the synthesis network. Given an input image I, the feature extractor module downsamples from 256×256 to 4×4, resulting in multi-level spatial features $f = F(I)$. Each block i of the feature f_i is defined as follows:

$$f_i = ZeroConv(ResBlock(Conv(f_{i-1}))), \tag{2}$$

where $i \in \{1, 2, 3, \cdots, 13\}$. When $i = 1$, f_0 represents the input image I. $ZeroConv(\cdot)$ denotes a convolutional layer with weights and biases initialized to 0, facilitating training initialization, and each update step tends to be closer to the true value. $ResBlock(\cdot)$ represents a residual convolutional block comprising 4 convolutional layers. $Conv(\cdot)$ refers to a convolutional block containing a convolutional layer followed by a ReLU activation layer.

3.3. Residual Cross-Attention Module

Mapping the image to the F space of the synthesis network preserves the spatial information of the image effectively, but when there is a need to manipulate the latent code to edit the image, the following equation can be derived:

$$\begin{aligned} G(w_{gt}) &\approx G(w_{avg}, F(I)), \\ G(w_{gt} + \Delta w) &\neq G(w_{avg} + \Delta w, F(I)), \end{aligned} \tag{3}$$

where the learned image spatial features f_i by F cannot vary according to the changes in w, thus leaving artifacts on the image.

The concept of cross-attention seeks to enable one vector to attend to another vector. Therefore, as shown at the bottom of Figure 3, we introduce the residual cross-attention module, which consists of residual cross-attention blocks with progressive upsampling, and it can learn the changes in feature f with respect to the changes in the latent code Δw. We set f as the query vector (Q) and Δw as the key (K) and value (V).

$$\begin{aligned} \Delta f = CrossAttention(f, \Delta w) = Softmax\left(\frac{QK^T}{\sqrt{d}}\right)V, \\ Q = W_Q f, K = W_K \Delta w, V = W_V \Delta w, \end{aligned} \tag{4}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{512 \times 512}$, and the feature dimension is 512. Softmax is used as the activation function.

Finally, by residually connecting the edited spatial feature $\hat{f}$ to the corresponding dimension of the synthesis network block, the generated image can be edited according to the changes in the latent code w. At this point,

$$G(w_{gt} + \Delta w) \approx G(w_{avg} + \Delta w, R(I, \Delta w)). \tag{5}$$

We obtain the new synthesized network features for each resolution block i and connect them to the synthesis network. As shown in the bottom-right corner of Figure 3, the output features of each synthesis network block are continuously upsampled to obtain the output image:

$$\begin{aligned} F_i &= \hat{f}_i + G^i(F_{i-1}), \\ \hat{f}_i &= f_i + \Delta f_i + UpSample(\hat{f}_{i-1}), \\ G^i(F_{i-1}) &= ModulatedConv(affine(w_{avg}), F_{i-1}), \end{aligned} \tag{6}$$

where $i \in \{1, 2, 3, \cdots, 13\}$, and UpSample denotes upsampling. $ModulatedConv(\cdot, \cdot)$ is the modulated convolutional layer, and $affine(\cdot)$ is a linear affine function. When $i = 0$, F_0 is a fixed-dimensional feature with dimensions $512 \times 4 \times 4$, initialized randomly. When $i = 13$, F_{13} represents the output image.

3.4. Training Details

As shown in Figure 4a, to better facilitate the learning of the feature changes corresponding to the variations in the latent code w, the variation Δw is set to a randomly sampled latent code $M(z)$ generated by random Gaussian noise z. Different editing directions d are introduced randomly, drawn from the editing method InterFaceGAN [34]. Initially, a large number of images are generated through StyleGAN, and an attribute predictor is employed to score each image. The top and bottom 1000 images are selected as positive and negative samples, respectively, based on their scores. A support vector machine (SVM) is trained using their corresponding latent codes w, which yields the decision boundary for a particular attribute in StyleGAN. The normal vector d of this boundary represents the editing direction. Thus, the expression for Δw is given as

$$\Delta w = \alpha M(z) + \sigma d, \tag{7}$$

where $\alpha \in (0, 1), \sigma \in (-10, 10)$.

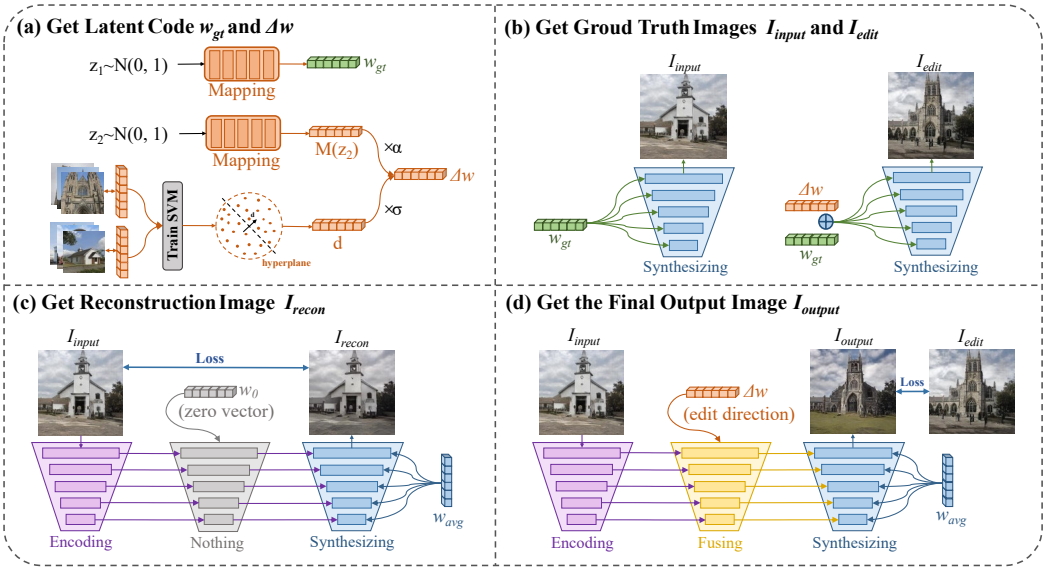


Figure 4. The training procedure of RSCAN. (**a**) First, obtain the real latent code w_{gt} and the editing vector Δw. (**b**) Input the latent code into the synthesis network to obtain the RSCAN input image I_{input} and the real edited image I_{edit}. (**c**) Input I_{input} into RSCAN; input the zero vector w_0 into the residual cross-attention module, disabling its editing function; and output the reconstructed image I_{recon}, aligning it with I_{input}. (**d**) Replace w_0 with Δw to input into the residual cross-attention module, obtaining the edited output image I_{output} and aligning it with I_{edit}.

The goal of RSCAN is to align the features of the predicted image with the F space of the synthesis network and, at the same time, be able to learn the corresponding image spatial feature change Δf when editing the latent code. Therefore, we must align the generated edited image and the real edited image and ensure that when $\Delta w = 0$, the reconstructed image I_{recon} is aligned with the input image. Additionally, a substantial amount of data is necessary to facilitate the cross-attention module's effective learning of information from the W space. Consequently, we employ a self-supervised approach, wherein all inputs are images produced by the synthesis network, as depicted in Figure 4. Thus, 4 images, I_{input}, I_{edit}, I_{recon}, I_{output}, and 2 latent codes, w_{gt}, Δw, need to be generated. Here, I_{input} and I_{edit} are generated by the synthesis network, $I_{input} = G(w_{gt})$, $I_{edit} = G(w_{gt} + \Delta w)$. I_{recon} and I_{output} are the outputs of the entire RSCAN structure, $I_{recon} = G(w_{avg}, R(I_{input}, \Delta w_0))$, $I_{output} = G(w_{avg}, R(I_{input}, \Delta w))$, where Δw_0 is a zero vector with dimensions of 14×512. During training, we simultaneously align the reconstructed image pair $\langle I_{recon}, I_{input} \rangle$ and the edited image pair $\langle I_{output}, I_{edit} \rangle$.

3.5. Loss Function

We employ both the reconstruction loss and editing loss to align the output images with the input images, and the total loss function of the training process is expressed as

$$L_{total} = \lambda_{edit} L_{edit} + \lambda_{recon} L_{recon}, \tag{8}$$

where L_{edit} represents the editing loss, designed to ensure that the model can effectively learn the corresponding image changes for variations in w. The expression for L_{edit} is as follows:

$$L_{edit} = \lambda_{L_2} L_2 + \lambda_{L_{lp}} L_{lp} + \lambda_{L_{spatial}} L_{spatial} + \lambda_{L_{adv}} L_{adv}, \tag{9}$$

where L_2 is the L_2 norm loss function, used to measure the pixel similarity, and L_{lp} is the global perceptual loss. L_2 and L_{lp} are defined as follows:

$$\begin{aligned} L_2 &= \left\| I_{edit} - I_{output} \right\|_2, \\ L_{lp} &= \left\| F_{lp}(I_{edit}) - F_{lp}(I_{output}) \right\|_2, \end{aligned} \tag{10}$$

where F_{lp} represents the perceptual feature extractor [35], using the pre-trained version of AlexNet [36]. To enable the image to express better spatial features at each feature layer, we add another spatial matching loss to the objective function:

$$L_{spatial} = \frac{1}{N} \sum_i \left\| G^i \left(w_{edit} - G^i \left(w_{avg}, R^i \left(I_{output} \right) \right) \right) \right\|_2, \tag{11}$$

where $G^i(\cdot)$ is the spatial feature output by the i-th convolution of the pre-trained StyleGAN synthesis network, and $R^i(\cdot)$ is the residual space of the dimension corresponding to $G^i(\cdot)$.

In order to make the edited image more realistic, the adversarial loss L_{adv} is also added. The original StyleGAN uses the adversarial loss to guide network convergence, so this study uses the pre-trained StyleGAN discriminator to guide the encoder to converge to the original intermediate space. L_{adv} and the discriminator loss are defined as follows:

$$\begin{aligned} L_{adv} &= -\mathbb{E}_{I_{output}} \left[\log \left(D \left(I_{output} \right) \right) \right], \\ L_D &= -\mathbb{E}_{I_{edit}} [\log(D(I_{edit}))] - \mathbb{E}_{I_{output}} \left[\log \left(D \left(1 - I_{output} \right) \right) \right] \\ &\quad + \mathbb{E}_{I_{edit}} \left[\left\| \nabla_{I_{edit}} D(I_{edit}) \right\|_2^2 \right], \end{aligned} \tag{12}$$

where D is the discriminator, which is initialized using the pre-trained StyleGAN discriminator weights. The discriminator D is trained together with RSCAN in an adversarial manner. Finally, R1 regularization is further applied to the D loss [37].

To ensure better reconstruction quality, the image reconstruction loss L_{recon} is added, so that the output image is reconstructed as the input image when the latent code is not changed. L_{recon} is defined as follows:

$$\begin{aligned} L_{recon} = \lambda_{L_2} L_2(I_{input}, I_{recon}) + \lambda_{L_{lp}} L_{lp}(I_{input}, I_{recon}) + \lambda_{L_{spatial}} L_{spatial}(I_{input}, I_{recon}) \\ + \lambda_{L_{grad}} L_{grad}, \\ L_{grad} = \left\| \nabla(I_{input}) - \nabla(I_{recon}) \right\|_2, \end{aligned} \tag{13}$$

where $\nabla(\cdot)$ is the image gradient. The image gradient can well represent the edge of the object in the image. Preserving the good gradient properties of the image ensures the reality and fidelity of the image to a certain extent [38].

4. Experiment

In this section, we conduct extensive experiments on the proposed Residual Spatial Cross-Attention Network (RSCAN) to validate its effectiveness in high-fidelity architectural image editing. We begin by introducing the dataset and evaluation metrics used in our experiments, followed by a description of the experimental setup. Subsequently, we qualitatively and quantitatively compare the RSCAN method with other existing methods in terms of the reconstruction quality and editing effects, highlighting its advantages. The comparison methods include PSP, E4E, HyperStyle, HyperInverter, and CLCAE. Additionally, we perform ablation studies to verify the effectiveness of the proposed modules. The experimental results of image blending are also presented, further demonstrating the potential of the RSCAN method in practical applications.

4.1. Datasets and Evaluation Metrics

We use the LSUN Church [39] dataset to train the synthesis network, and we use a randomly sampled z to input the generated church data obtained by the synthesis network as the training set of RSCAN. The LSUN Church dataset is a large-scale, high-quality collection of church images, comprising 126,226 images of various resolutions. It is widely used in architectural image editing tasks, where it enables the training of high-quality generators, thereby enhancing the quality of the generated images. Furthermore, the extensive range of the LSUN Church dataset, which covers diverse architectural styles and scenes, significantly contributes to validating the effectiveness and robustness of our method when dealing with a variety of architectural images. We unify the image size to 256×256 to facilitate training. Additionally, to enable the editing of architectural style attributes, we use the architectural style dataset [40] to train the ResNet50 [30] style classifier to obtain the style editing vector. The architectural style dataset comprises 10,113 images spanning 25 architectural styles. Its diverse range of style types and abundant image resources provide us with a rich set of samples for style editing, which contributes to enhancing the accuracy and diversity of the editing process.

The editing method in our research follows the principle of reconstruction first and then editing, so the evaluation indicators are divided into the reconstruction quality and editing effect. We employ a variety of evaluation metrics to comprehensively assess the reconstruction quality, including the pixel-level L2 distance, peak signal-to-noise ratio (PSNR) [41], structural similarity index (SSIM) [42], learning perceptual image patch similarity (LPIPS) [35], and Fréchet inception distance (FID) [43] indicators. The evaluation metrics are introduced as follows:

- Pixel-level L2 distance: It measures image differences by calculating pixel-level discrepancies.
- PSNR: It is based on the L2 distance and evaluates the image quality through the ratio of the peak signal to noise power.
- SSIM: It provides a holistic assessment of the image quality considering the brightness, contrast, and structure, taking into account the structural information of the image.

- LPIPS: It utilizes a pre-trained neural network to simulate the human visual system's perception, capturing detailed differences in images.
- FID: It assesses the overall quality and style consistency of images at a higher level by comparing the distance between the generated images and real images in the feature space.

Furthermore, we test the editing effects of the different methods under the same editing magnitude. In this experiment, we input the edited images into style classifiers and attribute predictors to obtain scores for the corresponding editing attributes. To objectively compare the impacts of the editing, we calculate the absolute differences between the attribute scores of the edited images and the reconstructed images, denoted as ΔS [19].

4.2. Experiment Setting

We utilize the Adam [44] optimizer; the learning rate is set to 0.0001, $\alpha = 0.5$, $\beta = 0.999$ during training; the training batch size is set to 8; and the training runs for 200,000 iterations. The loss function hyperparameters are set as follow: $\lambda_{L_2} = 1.0$, $\lambda_{L_{lp}} = 0.8$, $\lambda_{L_{grad}} = 0.6$, $\lambda_{L_{spatial}} = 1.0$, $\lambda_{L_{adv}} = 0.15$, $\lambda_{L_{recon}} = 0.5$. As shown in Figure 5, we choose to use a grid search to validate the settings of $\lambda_{L_{grad}}$ and $\lambda_{L_{spatial}}$. For other hyperparameters and learning rates, we adopt the values from E4E [24] and also use a grid search to verify their effectiveness.

The experiment was conducted using Python 3.7 and PyTorch 1.8. In the training, a Tesla V100 SMX3 graphics card equipped with 32 GB of video memory was utilized. During the test, a 3060 Laptop graphics card with 6 GB of video memory was employed.

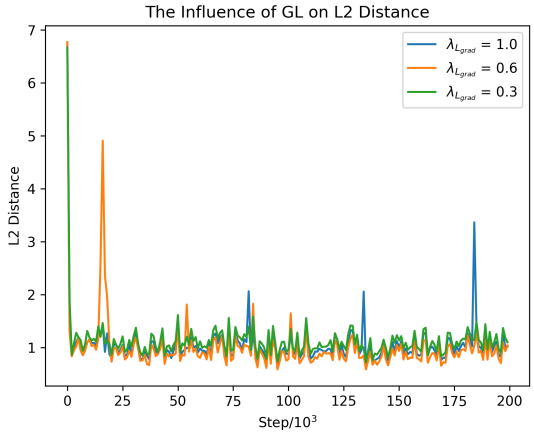


Figure 5. Comparative visual analysis of $\lambda_{L_{grad}}$ with different values.

4.3. Comparisons with Other Methods in Terms of Reconstruction Quality

4.3.1. Qualitative Evaluation

Figure 6 shows the reconstruction results: the first column shows the input image, and the remaining six columns show the reconstruction results of different methods. Our proposed approach excels in preserving intricate architectural details, evident in instances such as the small window on the castle's left and the square structure outlined on the right, highlighted within the red frame in the first row. Notably, the PSP and E4E methods only provide rudimentary outlines of the building, inaccurately placing and numbering the windows, while HyperInverter produces a cluttered amalgamation of windows. For the horizontal lines on the main structure of the church, our method can successfully restore the straight horizontal lines. Although HyperStyle achieves this, the picture is blurry. CLCAE has a good effect in color restoration, but, in some details, such as the lines on the top of the castle, the building cannot be completely reconstructed.

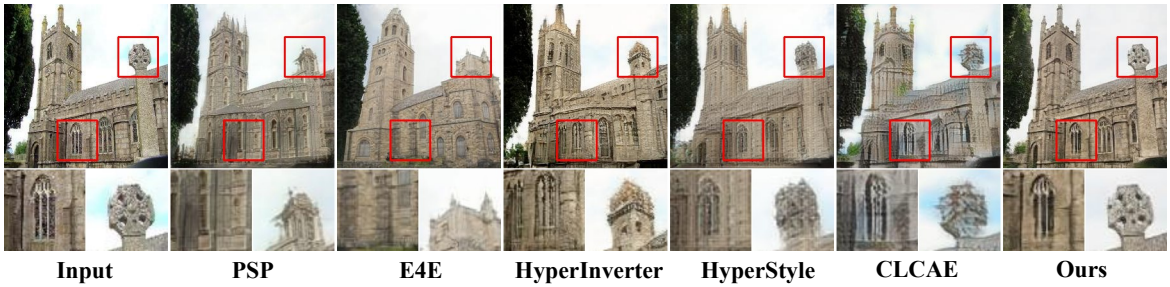


Figure 6. Qualitative reconstruction quality comparison of our method with existing works. The second row is the enlarged result of the red box in the first row.

4.3.2. Quantitative Evaluation

Table 2 shows a quantitative comparison of the reconstruction quality of different methods. Our method significantly outperforms other methods in this regard. RSCAN produces higher results than the other five methods in terms of the L2, PSNR, SSIM, LPIPS, and FID. For the SSIM and FID, its scores are 29.49% and 17.29% higher than those of the next best method, respectively. Regarding the L2, LPIPS, and PSNR, they are improved by 8.81%, 11.43%, and 11.26% respectively. This quantitative analysis demonstrates the effectiveness of the proposed method in terms of the architectural image reconstruction quality.

Table 2. Quantitative evaluation of reconstruction quality. The bold values represent the best performances.

	PSNR↑	**SSIM↑**	**FID↓**	**L2↓**	**LPIPS↓**
PSP [24]	17.6227	0.4464	40.9037	0.1621	0.2279
E4E [16]	15.9481	0.4175	41.9608	0.1991	0.3163
HyperInverter [19]	17.0909	0.4511	35.2321	0.1671	0.1773
HyperStyle [18]	19.4163	0.4999	39.6117	0.1284	0.1303
CLCAE [29]	19.4931	0.5628	51.3415	0.1353	0.1493
Ours	**21.6889**	**0.7288**	**29.1387**	**0.1171**	**0.1154**

4.4. Comparisons with Other Methods in Terms of Editing Effects

The evaluation of the editing effects in our study encompasses two dimensions: architectural style editing and architectural element editing. InterfaceGAN [34] is employed as the editing method. The style editing uses Resnet50 [30] for style prediction to obtain the editing vector. Four architectural styles—Gothic, Greek, Baroque, and Byzantine—are evaluated. Element editing uses the attribute predictor [45] for scoring to obtain the edit vector. Three attributes of glass, clouds, and trees are considered in the evaluation.

4.4.1. Qualitative Evaluation

Figure 7 shows the visual results of style editing, showing a total of two cases and four styles. In style editing, HyperStyle solely introduces color modifications, while HyperInverter exhibits subtle alterations. PSP and CLCAE cannot maintain consistency in the architectural structure and produce distorted and blurred artifacts. E4E and our RSCAN can achieve good editing effects among the four attributes, but E4E fails to change the Gothic and Byzantine architectural styles well in the second and third rows of the first sample. The spire at the top of the tower is due to E4E inverting the image into the $W+$ space, which results in the loss of the editing effect. Our method successfully converts the spire into the two types of Gothic architecture and a dome representing Byzantine architecture. As shown in Figure 8, in the element editing, PSP, CLCAE, and E4E can add clouds and reduce trees in the image, but the former two produce blur in the house part after editing. When adding glass, an extra window is opened at the top of the building in our method. When adding cloud and tree attributes, our method can achieve successful editing and ensure that no other changes occur to the original building. At the same time, our method creates a smaller distortion. When editing the building, it also retains the integrity of the building

structure, which is more consistent with the effect of modification on the original image. In contrast, other methods with lower reconstruction quality may hinder the editing based on the semantics of the original image.

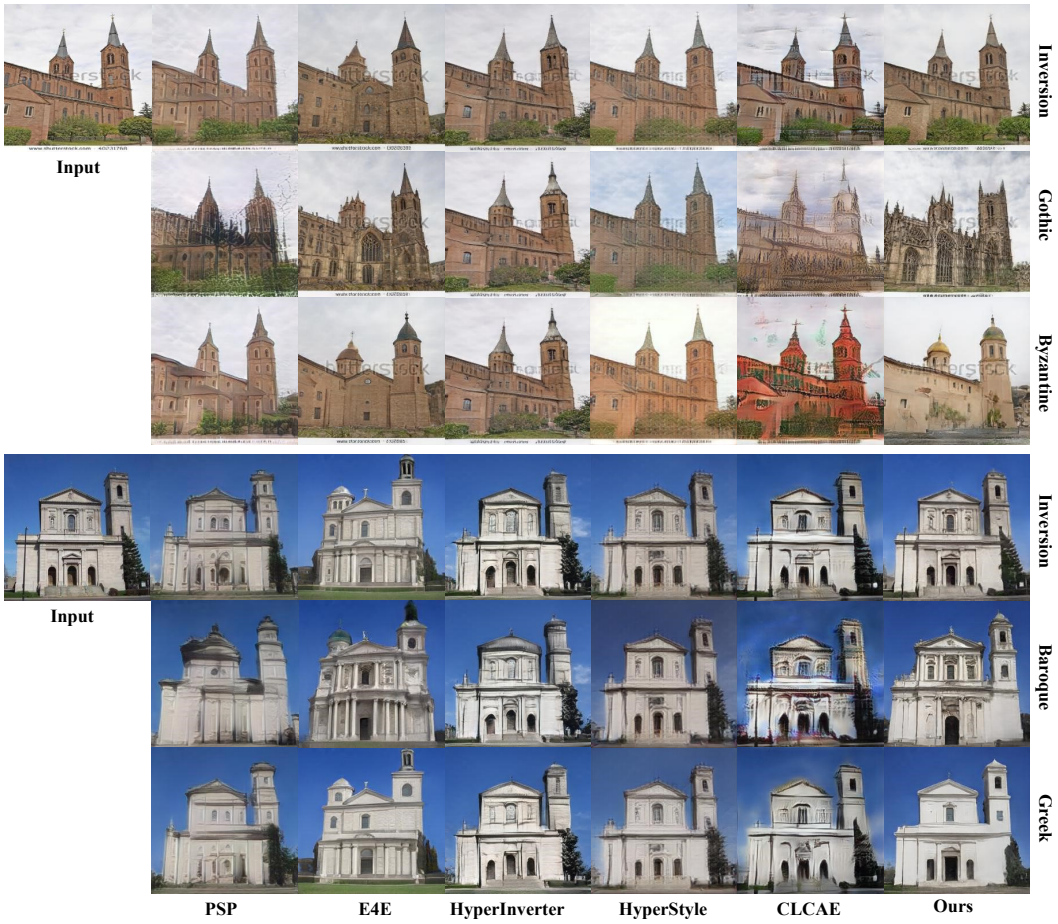


Figure 7. Qualitative style editing comparison of our method with existing works.

Figure 8. Qualitative element editing comparison of our method with existing works.

4.4.2. Quantitative Evaluation

As shown in Table 3, our method achieves the highest editing scores ΔS for both style editing and element editing, while E4E obtains the second highest values, consistent with the qualitative evaluation. Our approach exhibits superior reconstruction quality and editing effects because we invert the image features into the F space, ensuring the best reconstruction quality, and we utilize cross-attention to learn the mapping relationship between the W space and F space, replacing the role of the modulation layers in synthesis networks, thus ensuring consistent editing effects. Regarding the total editing score, our method improves the style editing and element editing by 25.0% and 70.6%, respectively, compared with the other methods.

Table 3. Quantitative evaluation of architectural style editing and element editing effects. The bold values represent the best performances.

$\Delta S \uparrow$	Style Editing					Element Editing			
	Gothic	**Greek**	**Byzantine**	**Baroque**	**Total**	**Trees**	**Clouds**	**Glass**	**Total**
PSP [24]	2.2884	1.2974	1.2148	0.7145	5.5151	0.0131	0.0125	0.0151	0.0407
E4E [16]	3.3226	1.6265	2.5017	1.7826	9.2334	0.0912	0.0586	0.1045	0.2543
HyperInverter [19]	1.3079	0.0679	0.8607	0.3201	2.5566	0.0569	0.0251	0.0291	0.1111
HyperStyle [18]	1.3246	0.4062	0.6361	0.3893	2.7562	0.0622	0.0165	0.0258	0.1045
CLCAE [29]	1.4427	0.4762	1.4154	0.6469	3.9812	0.0191	0.0389	0.0548	0.1128
Ours	**4.1377**	**2.0002**	**3.3117**	**2.0877**	**11.5373**	**0.2003**	**0.0784**	**0.1551**	**0.4338**

4.5. Ablation Study

4.5.1. Impact of Mapping Space and Loss Function on Reconstruction Quality

We compared the approaches of mapping images to the $W+$ space, methods lacking the reconstruction loss (w/o RL), methods lacking the gradient loss (w/o GL), and methods preserving all functionalities in RSCAN. As shown in Table 4 and Figure 9, mapping images to the $W+$ space results in more distortion compared to mapping to the F space, yielding only partial reconstruction in terms of contours and colors. Following mapping to the F space, the absence of the reconstruction loss leads the model to prioritize the learning of image features, consequently neglecting the reconstruction quality, as seen in the windows of the second-row building and the leaves on the left. As shown in Figure 10, in self-supervised training, there exists a domain gap between the synthetically generated fake images and real ones. With the absence of the gradient loss, the reconstruction of the fake images improves while the loss of image features decreases, whereas the reconstruction of the real images deteriorates. Incorporating the gradient loss enhances the reconstruction of the architectural contours and line details, reducing the influence of differences in data content and minimizing the domain gap between the generated and real images. Although fluctuations occur in the loss of the real images, there is no increasing trend, and the overall values are lower. Thus, while learning the image features, the reconstruction quality is preserved.

Table 4. Ablation study of reconstruction quality. The bold values represent the best performances.

	PSNR↑	**SSIM↑**	**FID↓**
W+ Space	15.9481	0.4175	41.9608
w/o RL	17.2613	0.4813	38.6894
w/o GL	18.5846	0.5546	32.7428
Ours	**21.6889**	**0.7288**	**29.1387**

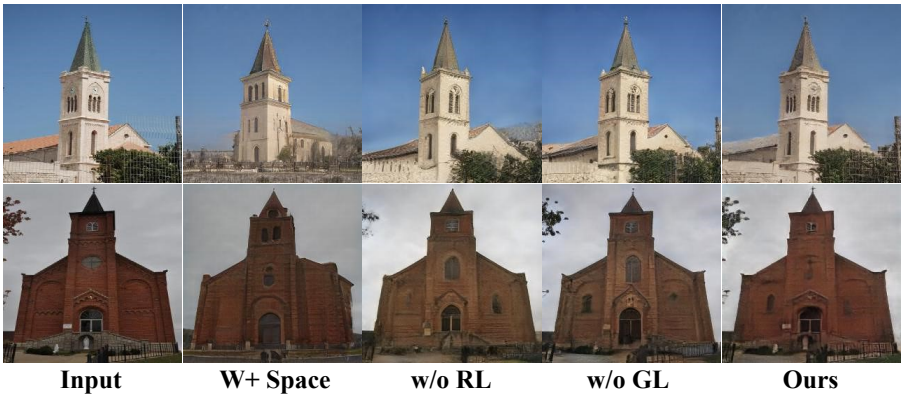


Figure 9. Visual examples of ablation study on reconstruction quality.

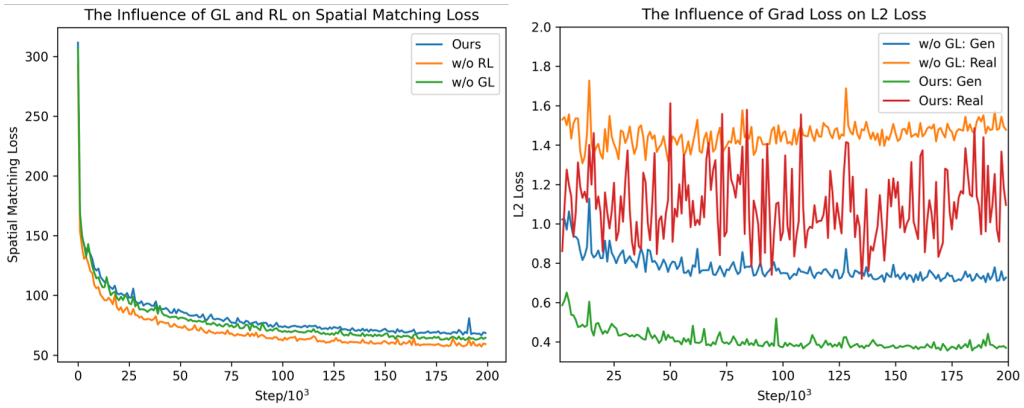


Figure 10. Spatial matching loss and L2 loss lacking different loss functions.

4.5.2. Impact of Residual Cross-Attention on Editing Effects

In order to verify the effectiveness of the residual cross-attention module, we remove it and directly edit the w vector in the W space. As shown in Figure 11 and Table 5, directly manipulating the latent vector in the W space has little impact on the image. This is because altering the latent vectors in the synthesis network does not affect the residual reconstruction features of the image, resulting in a reduction in the overall feature variation in the synthesis network. Reconstructed features may exhibit artifacts in unintended areas, as seen in the second row, where the circular roofs only slightly change in color but remain attached to the image. After adding residual cross-attention, the reconstructed features vary according to the changes in the latent vectors, leading to the disappearance of artifacts and a greater magnitude of feature variation. Editing the latent vectors in the residual cross-attention and W space at the same time will produce more obvious changes in the image, such as fewer trees in the first row, darker Gothic buildings, and larger windows.

Table 5. Ablation study of editing effects. The bold values represent the best performances.

$\Delta S \uparrow$	W Space	Attention	Ours
Style	3.8752	10.8075	**11.5373**
Element	0.1204	0.3927	**0.4338**

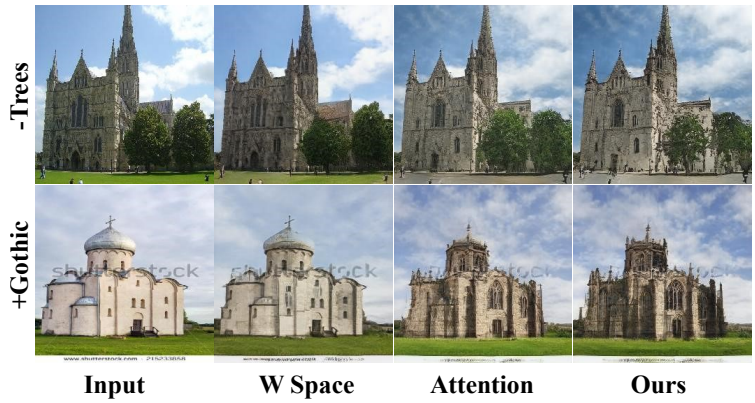


Figure 11. Visual examples of ablation study on editing effects.

4.6. Image Blending

In this section, we propose a novel method to interpolate between real images and reference style images. A commonly used method involves first finding the corresponding latent codes through GAN inversion, computing the linear interpolation between the latent codes of the real and reference images, and then obtaining the interpolated image through the GAN model. Based on our model, cross-attention can only learn the change in w corresponding to f, so we introduce a new interpolation method and input the latent code corresponding to the reference style image into the residual cross-attention at the same time, and the w mean of the original w latent space is interpolated with the reference latent code. Figure 12 shows some qualitative results comparing our method with methods that only interpolate latent codes. It can be observed that our method not only reconstructs the input image with the correct details (e.g., windows, building outlines) but also produces smoother and more realistic transition images during the blending process.

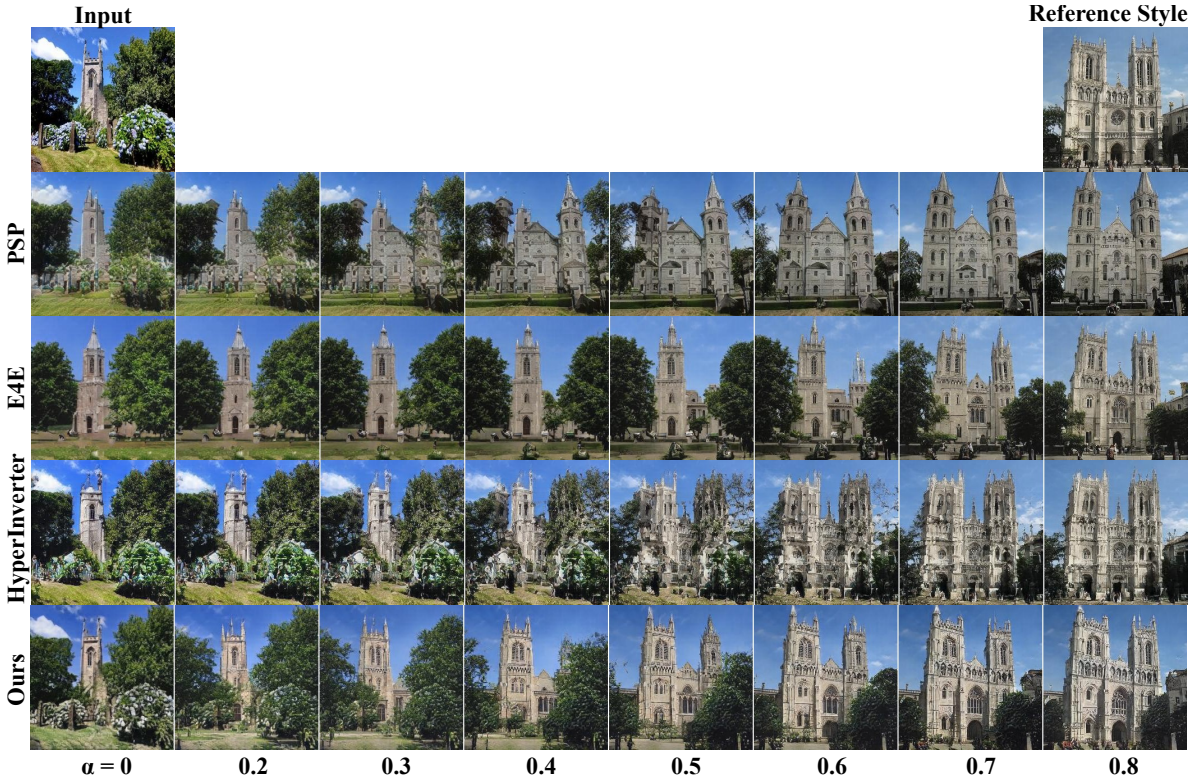


Figure 12. Qualitative evaluation of image blending.

5. Conclusions

The proposed method RSCAN achieves the high-fidelity editing of architectural images by introducing a feature extractor, residual cross-attention, and synthesis network modules. Different from the traditional GAN inversion method that directly encodes the image into the *W* space of StyleGAN, our method inverts the image into the *F* space with more spatial information and learns the mapping relationship between the *F* space and *W* space through cross-attention. We address the issues of spatial information loss and poor reconstruction quality encountered by traditional methods, while preserving the editing capabilities of the original *W* space. The experimental results demonstrate significant improvements over existing methods across multiple evaluation metrics, addressing the issues of poor reconstruction and editing effects. Therefore, the architectural image editing method that we propose provides an efficient and accurate image editing solution for the field of digital architecture. Our method also has certain limitations: due to the differing operational principles of cross-attention and the originally modulated convolution, our residual cross-attention cannot learn the comprehensive mapping relationship between the *F* space and *W* space. If an input latent vector outside the domain seen by the cross-attention module is provided, there will be a decline in the editing performance. In the future, we will focus on further optimizing the algorithm to enhance the editing efficiency and quality, as well as exploring the model's scalability to other datasets.

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

Funding: This work was funded by the National Natural Science Foundation of China (No. 62176018) and the Beijing University of Civil Engineering and Architecture Research Capacity Promotion Program for Young Scholars (X23024).

Data Availability Statement: The LSUN Church dataset is publicly available at https://github.com/fyu/lsun (accessed on 10 June 2024). The architectural style dataset is publicly available at https://www.kaggle.com/datasets/dumitrux/architectural-styles-dataset (accessed on 10 June 2024). Our code is publicly available via https://github.com/ChhhZzz/RSCAN (accessed on 10 June 2024).

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

References

1. Jiang, S.; Yan, Y.; Lin, Y.; Yang, X.; Huang, K. Sketch to building: Architecture image translation based on GAN. *J. Phys. Conf. Ser.* **2022**, *2278*, 012036. [CrossRef]
2. Nauata, N.; Hosseini, S.; Chang, K.H.; Chu, H.; Cheng, C.Y.; Furukawa, Y. House-gan++: Generative adversarial layout refinement network towards intelligent computational agent for professional architects. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13632–13641.
3. Brock, A.; Donahue, J.; Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. *arXiv* **2018**, arXiv:1809.11096.
4. 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.
5. Sangkloy, P.; Lu, J.; Fang, C.; Yu, F.; Hays, J. Scribbler: Controlling deep image synthesis with sketch and color. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5400–5409.
6. Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; Aila, T. Analyzing and improving the image quality of stylegan. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 8110–8119.
7. Xia, W.; Zhang, Y.; Yang, Y.; Xue, J.H.; Zhou, B.; Yang, M.H. Gan inversion: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 3121–3138. [CrossRef] [PubMed]
8. Wang, T.; Zhang, Y.; Fan, Y.; Wang, J.; Chen, Q. High-fidelity gan inversion for image attribute editing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11379–11388.
9. Shannon, C.E. Coding theorems for a discrete source with a fidelity criterion. *IRE Nat. Conv. Rec* **1959**, *4*, 1.

10. Tishby, N.; Zaslavsky, N. Deep learning and the information bottleneck principle. In Proceedings of the 2015 IEEE Information Theory Workshop (ITW), Seattle, WA, USA, 3 November 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1–5.
11. Song, Q.; Li, G.; Wu, S.; Shen, W.; Wong, H.S. Discriminator feature-based progressive GAN inversion. *Knowl.-Based Syst.* **2023**, *261*, 110186. [CrossRef]
12. Katsumata, K.; Vo, D.M.; Liu, B.; Nakayama, H. Revisiting Latent Space of GAN Inversion for Robust Real Image Editing. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2024; pp. 5313–5322.
13. Li, H.; Huang, M.; Zhang, L.; Hu, B.; Liu, Y.; Mao, Z. Gradual Residuals Alignment: A Dual-Stream Framework for GAN Inversion and Image Attribute Editing. *arXiv* **2024**, arXiv:2402.14398.
14. 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, 27–30 June 2016; pp. 2414–2423.
15. Chen, Y.; Vu, T.A.; Shum, K.C.; Yeung, S.K.; Hua, B.S. Time-of-Day Neural Style Transfer for Architectural Photographs. In Proceedings of the 2022 IEEE International Conference on Computational Photography (ICCP), Pasadena, CA, USA, 1–5 August 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–12.
16. Tov, O.; Alaluf, Y.; Nitzan, Y.; Patashnik, O.; Cohen-Or, D. Designing an encoder for stylegan image manipulation. *Acm Trans. Graph.* **2021**, *40*, 1–14. [CrossRef]
17. Su, W.; Ye, H.; Chen, S.Y.; Gao, L.; Fu, H. Drawinginstyles: Portrait image generation and editing with spatially conditioned stylegan. *IEEE Trans. Vis. Comput. Graph.* **2022**, *29*, 4074–4088. [CrossRef] [PubMed]
18. Alaluf, Y.; Tov, O.; Mokady, R.; Gal, R.; Bermano, A. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 18511–18521.
19. Dinh, T.M.; Tran, A.T.; Nguyen, R.; Hua, B.S. Hyperinverter: Improving stylegan inversion via hypernetwork. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11389–11398.
20. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 10684–10695.
21. Kawar, B.; Zada, S.; Lang, O.; Tov, O.; Chang, H.; Dekel, T.; Mosseri, I.; Irani, M. Imagic: Text-based real image editing with diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 6007–6017.
22. Gu, J.; Shen, Y.; Zhou, B. Image processing using multi-code gan prior. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 3012–3021.
23. Abdal, R.; Qin, Y.; Wonka, P. Image2stylegan: How to embed images into the stylegan latent space? In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4432–4441.
24. Richardson, E.; Alaluf, Y.; Patashnik, O.; Nitzan, Y.; Azar, Y.; Shapiro, S.; Cohen-Or, D. Encoding in style: A stylegan encoder for image-to-image translation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 2287–2296.
25. Wu, Z.; Lischinski, D.; Shechtman, E. Stylespace analysis: Disentangled controls for stylegan image generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 12863–12872.
26. Kang, K.; Kim, S.; Cho, S. Gan inversion for out-of-range images with geometric transformations. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 13941–13949.
27. Roich, D.; Mokady, R.; Bermano, A.H.; Cohen-Or, D. Pivotal tuning for latent-based editing of real images. *Acm Trans. Graph.* **2022**, *42*, 1–13. [CrossRef]
28. Pehlivan, H.; Dalva, Y.; Dundar, A. Styleres: Transforming the residuals for real image editing with stylegan. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 1828–1837.
29. Liu, H.; Song, Y.; Chen, Q. Delving stylegan inversion for image editing: A foundation latent space viewpoint. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 10072–10082.
30. 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.
31. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30* , 6000–6010.
32. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 213–229.
33. Chen, C.F.R.; Fan, Q.; Panda, R. Crossvit: Cross-attention multi-scale vision transformer for image classification. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 357–366.
34. Shen, Y.; Gu, J.; Tang, X.; Zhou, B. Interpreting the latent space of gans for semantic face editing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9243–9252.

35. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 586–595.
36. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25* , 1097–1105. [CrossRef]
37. Mescheder, L.; Geiger, A.; Nowozin, S. Which training methods for gans do actually converge? In Proceedings of the International Conference on Machine Learning, (PMLR), Stockholm, Sweden, 10–15 July 2018; pp. 3481–3490.
38. Mechrez, R.; Shechtman, E.; Zelnik-Manor, L. Photorealistic style transfer with screened poisson equation. *arXiv* **2017**, arXiv:1709.09828.
39. Yu, F.; Seff, A.; Zhang, Y.; Song, S.; Funkhouser, T.; Xiao, J. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv* **2015**, arXiv:1506.03365.
40. Xu, Z.; Tao, D.; Zhang, Y.; Wu, J.; Tsoi, A.C. Architectural style classification using multinomial latent logistic regression. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part I 13; Springer: Berlin/Heidelberg, Germany, 2014; pp. 600–615.
41. Almohammad, A.; Ghinea, G. Stego image quality and the reliability of PSNR. In Proceedings of the 2nd International Conference on Image Processing Theory, Tools and Applications, Paris, France, 7–10 July 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 215–220.
42. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]
43. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Adv. Neural Inf. Process. Syst.* **2017**, *30* , 6629–6640.
44. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
45. Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; Torralba, A. Places: A 10 million image database for scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 1452–1464. [CrossRef] [PubMed]