

DIVERSE inpainting and editing with semantic conditioning

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

By
Hakan Sivük
September 2024

Diverse Inpainting and Editing with Semantic Conditioning

By Hakan Sivük

September 2024

We certify that we have read this thesis and that in our opinion it is fully adequate,
in scope and in quality, as a thesis for the degree of Master of Science.



Ayşegül Dünder Boral(Advisor)

Hamdi Dibekliolu

Ramazan Gökberk Cinbiş

Approved for the Graduate School of Engineering and Science:

Orhan Arıkan
Director of the Graduate School

ABSTRACT

DIVERSE INPAINTING AND EDITING WITH SEMANTIC CONDITIONING

Hakan Sivük
M.S. in Computer Engineering
Advisor: Ayşegül Dünder Boral
September 2024

Semantic image editing involves filling in pixels according to a given semantic map, a complex task that demands contextual harmony and precise adherence to the semantic map. Most previous approaches attempt to encode all information from the erased image, but when adding an object like a car, its style cannot be inferred only from the context. Models capable of producing diverse results often struggle with smooth integration between generated and existing parts of the image. Moreover, existing methods lack a mechanism to encode the styles of fully and partially visible objects differently, limiting their effectiveness. In this work, we introduce a framework incorporating a novel mechanism to distinguish between visible and partially visible objects, leading to more consistent style encoding and improved final outputs. Through extensive comparisons with existing conditional image generation and semantic editing methods, our experiments demonstrate that our approach significantly outperforms the state-of-the-art. In addition to improved quantitative results, our method provides greater diversity in outcomes. For code and a demo, please visit our project page at <https://github.com/hakansivuk/DivSem>.

Keywords: Semantic image editing, Conditional image inpainting, Conditional image outpainting, Generative Adversarial Networks.

ÖZET

SEMANTİK KOŞULLAMA İLE ÇEŞİTLİ TAMAMLAMA VE DÜZENLEME

Hakan Sivük
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Danışmanı: Ayşegül Dünder Boral
Eylül 2024

Semantik haritayı takip ederek piksellerin doldurulmasını içeren semantik görüntü düzenleme, hem bağlama uyum sağlanması hem de semantik haritalara uyulmasını gerektiren zorlu bir görevdir. Önceki yöntemlerin çoğu, silinen görüntülerden tüm bilgiyi kodlamaya çalışmaktadır. Ancak, bir araba gibi bir nesne sahneye eklendiğinde, yalnızca bağlamdan yola çıkarak stilinin kodlanması mümkün değildir. Farklı sonuçlar üretebilen modeller, resmin üretilen ve silinmemiş bölümleri arasında oluşan keskin sınırları engellemekte zorlanmaktadır. Ayrıca, önceki yöntemler, görünür ve kısmen görünür nesnelerin stillerini farklı şekillerde kodlayabilecek bir mekanizmaya sahip olmadıkları görülmektedir ve bu durum performanslarını sınırlandırmaktadır. Bu çalışmada, stil kodlama ve üretilen resimlerde tutarlılığı sağlamak amacıyla görünür ve kısmen görünür nesneleri yeni bir mekanizma ile kodlayabilen bir yaklaşım öneriyoruz. Önceki koşullu görüntü üretimi ve semantik görüntü düzenleme algoritmalarıyla kapsamlı karşılaştırmalar sonucunda yöntemimizin mevcut en iyi yöntemlere kıyasla önemli ölçüde iyileştirme sağladığını göstermektedir. Yöntemimiz sadece daha iyi nicel sonuçlar elde etmekle kalmayıp aynı zamanda sonuçlarda çeşitlilik de sunmaktadır. Yayınlanan kod ve demo için lütfen proje web sayfasına bakınız: <https://github.com/hakansivuk/DivSem>.

Anahtar sözcükler: Semantik resim düzenleme, Koşullu resim tamamlama, Koşullu resim genişletme, Üretici Çekişmeli Ağlar.

Acknowledgement

I am profoundly grateful to my advisor Ayşegül DüNDAR Boral whose support, insightful guidance, and encouragement have been crucial throughout the course of this thesis. Her expertise and insights have been instrumental in the completion of my work.

I am also deeply thankful to my thesis jury members, Hamdi Dibeklioglu and Ramazan Gökberk Cinbiş, for their constructive feedback, time, and effort in reviewing my research.

I would like to extend my sincere thanks to the Bilkent Image Generation Lab for their contribution to my academic development during the master's program. Their support and collaboration have played a significant role in my growth as a researcher.

Lastly, I would like to express my heartfelt appreciation to Bilkent University, which has provided me with an excellent education over the past eight years. The knowledge, experience, and opportunities I gained during this time have been invaluable in shaping both my personal and professional journey.

Contents

1	Introduction	1
2	Related Work	5
2.1	Image-to-Image Translation Algorithms	5
2.1.1	Generative Adversarial Network (GAN)	6
2.1.2	Diffusion Based Image Generation Models	6
2.2	Semantic Image Editing	7
2.2.1	Semantic Diffusion Models	9
2.3	Limitations of Existing Studies	10
3	Method	11
3.1	Style Encoding	11
3.2	Generator	15
3.3	Training Parameters	17

4 Experiments	19
4.1 Datasets	19
4.2 Evaluation Masks	20
4.3 Baselines	21
4.4 Metrics	23
4.5 Results	24
4.6 Ablation Study	26
4.7 Applications	27
5 Discussion	32
5.1 Comparison with the Diffusion Models	32
5.2 Applicability to Different Architecture	33
5.3 Limitations and Future Directions	34
5.3.1 Limitations	34
5.3.2 Future Directions	34
6 Conclusion	35
Bibliography	38

1.1 Our approach can eliminate items from a scene (using mask_removal to highlight them), create panoramas, fill in a semantic color map in various styles, and add items to a scene using different styles.

3.2 Comparisons of methods for free-form and outpainting masks on ADE20k dataset (The first rows are from ADE20k-Room, others are from ADE20k-Landscape). Our method achieves better results in terms of color consistency between the generated and existing pixels and sharpness in the generated parts. 18

3.3	Comparisons of methods for free-form, outpainting, and extension masks on Cityscapes dataset. Our method achieves better results in terms of color consistency between the generated and existing pixels and sharpness in the generated parts.	18
4.1	Comparisons of results generated by different methods on ADE20k-Room dataset for object addition. Our method can add objects with diverse styles which improves realism.	21
4.2	Comparisons of results generated by different methods on Cityscapes dataset for object addition. Our method can add objects with diverse styles which improves realism.	22
4.3	Ablation study visual results. Baseline stands for the version where all proposed methods are used.	26
4.4	Diverse object addition results generated with our model. Different style codes are used to inpaint the erased chair. Style codes are extracted from corresponding semantic classes of the style images.	29
4.5	Diverse editing results generated with our model. Different style codes are used to inpaint the erased image. Style codes are extracted from corresponding semantic classes of the style images.	30
4.6	Panorama images generated with our model. We can extend images with the given semantic map.	31

List of Tables

4.1	Comparisons of our framework with the state-of-the-art methods. The best results are highlighted with bold and the second best with <u>underline</u> . We do not calculate object addition scores for ADE20k-Landcape dataset, since it does not have instances. . . .	20
4.2	Ablation Study. FID scores are given for different configurations. We highlight the best results in bold	25

Chapter 1

Introduction

Semantic image editing involves the generation of consistent pixels with the input image based on semantic conditioning. The generated regions must blend seamlessly with the rest of the image while adhering strictly to the content provided by semantic maps. This task allows for the addition of objects, scene manipulation, object removal, as well as image outpainting and extension. It is also known as conditional inpainting and is considered very challenging, as the inpainted area must cohesively fit with the overall image and the provided semantic map.

Before, manual effort was required for semantic image editing, just like in many other image editing and inpainting tasks. However, today’s deep learning-based generative methods have achieved promising results without any human intervention [1, 2]. One of the primary difficulties in semantic image editing is maintaining a consistent context style between inpainted and unerased regions and seamlessly integrating those areas. Previous approaches have focused on semantic-based conditional image generation. For instance, methods like SPADE [3] can produce an image from scratch but lack a mechanism for encoding styles for visible and generated pixels. The same semantic-based style encoding is fed to instance normalization layers, and when training for the semantic image editing task, this results in unsatisfactory image editing outcomes. Another approach,

SEAN [4], learns styles for each semantic region while generating images. However, SEAN lacks a mechanism for encoding partially erased images, making it inefficient for inpainting tasks.

Alternative approaches for semantic image editing have been suggested, including SESAME [1] and SPMPGAN [2]. SESAME utilizes a SPADE architecture but lacks a mechanism to distinguish between valid and invalid pixels while encoding or to encode styles. SPMPGAN has demonstrated better results in semantic image editing than SESAME. It introduces a style-preserved modulation layer with a two-stage modulation method: the first stage injects style information from the semantic map, and the second stage injects encoded styles from the feature maps. Like SESAME, this work also lacks a mechanism for style encoding. A transformer-based approach, ASSET [5], has been proposed to achieve high-resolution image editing results. However, it suffers from boundary artifacts due to inconsistencies across generated and existing parts, particularly in dense scenes like Cityscapes and ADE20k-Room images [6]. SIEDOB [6] demonstrates that diverse object addition is achievable, but only through separately trained multiple sub-networks and only for a few selected foreground objects and not for semantic classes. In our research, we introduce a novel end-to-end framework. This framework is designed to ensure correct style encoding for partially erased and completely erased semantic classes, a unique feature that sets it apart from previous approaches. Our method not only overcomes these limitations but also demonstrates superior results compared to the previous approaches.

Despite the significant strides made in previous research, it's important to note that none of these models have been able to achieve diverse semantic image editing results while maintaining consistency between the generated and unerased portions. Most of these models attempt to encompass all the information necessary to produce an image from the erased input images and semantic maps. However, if a new object, such as a car, is introduced to a scene, or if a semantic class, such as the sky, is entirely erased for regeneration, the erased input image lacks the style information needed for successful generation. These models encounter difficulties in such situations and cannot replicate styles from reference images. Our work offers a novel approach to achieving diversity through style encodings

from reference images. As shown in Fig. 1.1, our framework can eliminate objects from a scene, introduce objects in diverse styles, create panoramas, and inpaint a semantic class again using different styles. It also accomplishes inpaintings of free-form masks with arbitrary shapes. In summary, our primary contributions are:

- Our proposal introduces a new training approach that merges the encoded styles from erased images for visible sections and the encoded styles from complete images when a semantic category or instance is entirely erased. Distinguishing partially and entirely erased images allows the fulfillment of the reconstruction goal during training and enables the extraction of diverse styles from different images for varied results during inference.
- We introduce a module aware of masks for encoding the styles of partially erased semantic classes or instances to ensure consistency in style extraction across masks of different sizes.
- We conduct a comparison of our method with various state-of-the-art conditional image inpainting methods, presenting results with a wide range of metrics. Our findings demonstrate the effectiveness of our framework, which exhibits significant improvements over the current state-of-the-art techniques.
- We illustrate practical applications of our framework, including object removal, the addition of diverse objects, panorama generation, and diverse semantic inpainting.

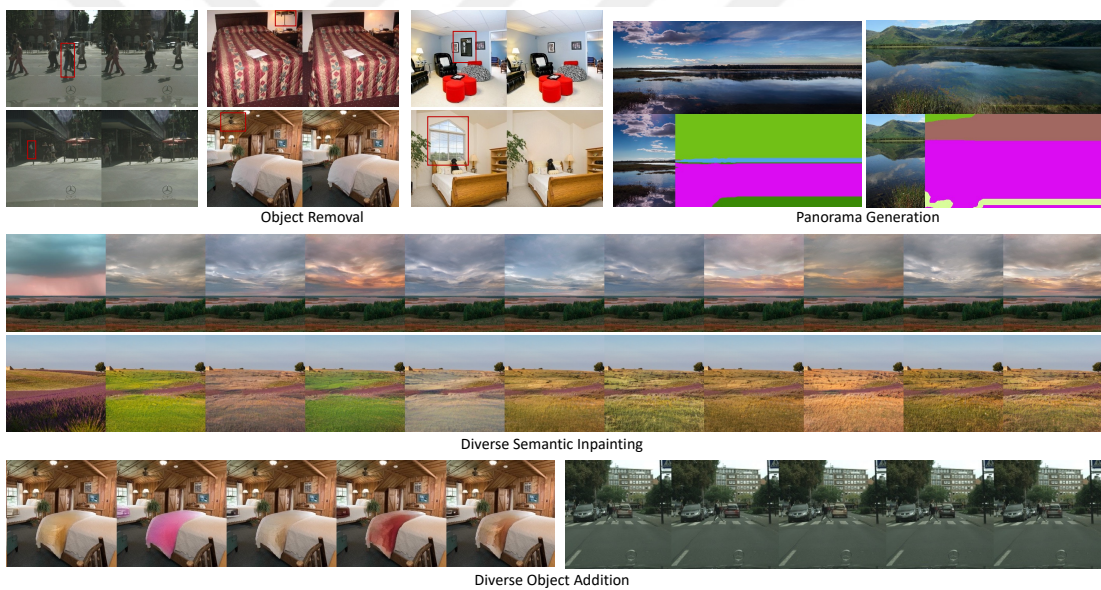


Figure 1.1: Our approach can eliminate items from a scene (using red boxes to highlight them), create panoramas, fill in a semantic class using various styles, and add items to a scene using different styles.

Chapter 2

Related Work

2.1 Image-to-Image Translation Algorithms

Image-to-image translation is transforming an image based on attributes such as image domain, style, and direction of the face images. Image-to-image translation techniques have many uses, including image colorization [7], image upsampling [8], texture prediction [9, 10, 11], and image domain editing [12, 13, 14, 3, 15, 4, 1, 16, 17]. Considering their wide range of applications, they have been popular in the field for a long time. Within image-to-image translation methods, generations can be based on different inputs, such as depth [18, 19, 20], key points [21, 22], edge maps [23, 24], and semantic maps [3, 15, 4, 1, 16, 25, 26]. Conditioning image generation on semantic maps is particularly popular due to their ease of use in painting, provision of pixel-level semantic information, and, therefore, reasonable user control over the generation process. A semantic map is a crucial component in image generation tasks. It is a mapping between pixels of an image and the corresponding semantic classes, which can be objects or background categories. While labeling a semantic map is labor-intensive, it provides fine-grained information that is essential for image generation tasks.

2.1.1 Generative Adversarial Network (GAN)

Generative Adversarial Networks (GANs) [27] have revolutionized the field of image generation since their introduction in 2014. The core concept of GANs involves the use of two models, namely the generator and the discriminator, which engage in a competitive process during training. The generator aims to create realistic images based on a given input, referred to as a latent code, while the discriminator’s role is to distinguish between real samples from an image dataset and the images generated by the generator. Through this adversarial process, the generator learns to produce images that are indistinguishable from real ones, effectively ”fooling” the discriminator.

Despite its widespread adoption and success, GANs are not without limitations. One such limitation is the unstable training process caused by the competitive nature of the generator and discriminator models. Despite this, GANs have been the dominant force in image generation tasks, as evidenced by their use in studies such as [3, 15, 4, 1, 16] until recently.

2.1.2 Diffusion Based Image Generation Models

Recently, diffusion models have emerged as strong contenders challenging the dominance of Generative Adversarial Networks (GANs) in the task of semantically conditioned image generation [28, 29, 30, 31]. These models are trained with extensive sets of text-image pairs and have demonstrated significant results in generating realistic images. The fundamental concept behind diffusion models involves denoising the input images. During training, the model aims to remove the Gaussian noise that has been added to the input images. As a result, post-training, the model is capable of generating high-quality images from a given noise input.

These models use text prompts [32, 33, 29, 28, 34] to describe the appearance of objects, although they may not be optimal for explaining coloring and style

information. For instance, object-level image editing with text prompts has been demonstrated in works such as [35, 36], where a text prompt is used with the input image to identify which pixels should be edited. While this eliminates the need for a mask input, it introduces challenges, especially for complex text prompts. Other works [37, 38, 39] use feature spaces in diffusion models. Although these models allow interpolation between images, extra effort is required to interpret and control the feature space. This situation negatively affects user control of the generated image. Therefore, it is arguable that some of the proposed GAN-based methods [4] provide better user control by relying on the appearance encoding from reference images.

Another challenge related to diffusion models is the size and number of networks. Most diffusion-based image generation studies combine multiple networks using a high number of parameters. This makes the training process harder, and considering the high inference times, it makes it difficult to use them in real-life applications.

2.2 Semantic Image Editing

Our focus in this study is on the task of semantic image editing, where image editing relies on segmentation maps. This task involves effectively adding missing pixels to an image based on the semantic map [1, 2]. As a result, our work is closely related to image inpainting methods. There has been extensive research on image inpainting, including inpainting with GAN inversion [40, 41, 42, 43, 44]. The GAN inversion method is an approach aimed at converting a given image into a latent code, which is then utilized by a GAN (Generative Adversarial Network) model to produce a corresponding image. This process results in the generation of an inpainted image. However, a drawback of this method is that the model lacks information about the erased area, thereby diminishing the user’s ability to have meaningful control over the inpainted images.

Also, these methods utilize StyleGAN models [45] trained on vast image

datasets representing various objects like faces and cars, which may not always be practical. StyleGAN uses a mapping network to translate latent codes into distinct style features. Adaptive Instance Normalization (AdaIN) is employed with these style codes for the generation process, effectively conditioning the model on the extracted style. It is noteworthy that in our study, we have demonstrated successful outcomes using training sets as small as two thousand images.

Other research has introduced methods like partial [46, 16] and gated [47] convolutions, as well as mask-aware transformers [48] to address the distinction between removed and existing pixels. This distinction is crucial for image inpainting task because of the varying mask ratio and shape. If there is no distinction between these pixels, mathematical calculations will be inconsistent for different mask ratios and shapes, which will negatively affect model performance. Our style encoding module shares the same principles as these methods. Furthermore, we leverage semantic and instance maps to encode styles differently for visible and hidden parts. Our work offers more functionalities than existing inpainting methods, as we can manipulate an image by altering the shapes of the objects, modifying the style of instances, removing objects from the scene, and outpainting of images with semantic control.

Semantic image editing presents a more significant challenge than conditional image generation or unconditional image inpainting due to the need for generated pixels to align with unerased/unmodified input pixels and semantic maps. Previous successful demonstrations of this task have employed GAN-based methods [1]. Another work, SPMPGAN [2], incorporates a context normalization module designed to maintain the context within an input image. Preserving context is crucial for generating realistic images because the unmasked region of the input image is directly transferred into the output image, while the masked region is filled in with the produced image. Failure to preserve context during image generation can result in inconsistencies between the masked and unmasked regions in the final output image. To address this, we have integrated this module into our architecture with some minor modifications, which will be explained in detail in the following sections. It should be noted that the existing GAN-based methods cannot generate diverse images, as they are not multi-modal and primarily focus

on generating novel objects in the scene by encoding visible parts.

A recent approach called ASSET [5] introduces a transformer-based architecture that builds on VQGAN [19], capable of producing diverse images through sparse attention mechanisms and a codebook in an autoregressive setting. Nevertheless, ASSET is affected by boundary artifacts between the visible and inpainted regions [6, 5]. Another method, SIEDOB [6], suggests generating foreground and background pixels separately and then combining them at the end, requiring the training of multiple individual networks and a fusion network to merge their outputs. SIEDOB can generate diverse results for foreground objects but not for semantic categories, necessitating many sub-networks and separate training for each. Our method is trained end-to-end and generates diverse foreground and background classes.

2.2.1 Semantic Diffusion Models

Several recent studies have extensively explored diffusion-based image inpainting methods [49, 28, 50, 51, 52], but these methods do not offer semantic control over the generation. Another approach by [53] utilizes edge maps as structural information, which helps prevent semantic inconsistency between the masked and original areas, even though the model does not directly condition on a semantic map. [54, 55, 56] tackle semantic editing task with diffusion-based models. The work [54] demonstrates that the content structure can be manipulated using diffusion models through separately encoded semantic maps and appearance codes. They use the updated semantic map and the appearance codes based on the editing when generating the edited image. Their model requires generating a semantic class from scratch, and even when the partial content is erased, it does not guarantee the preservation of valid pixels. We compare this model with our work by using their inpainting setup demonstrated in the paper. [55] uses diffusion model conditioning on a semantic map to synthesize images. It allows object addition by specifying the edited semantic region with the added object. It uses skip connections between the encoder and decoder to preserve the context

within the input image. In our work, we use a similar strategy to preserve the context. Since it is an image synthesis model, it does not preserve the consistency between the original and inpainted pixels for partially masked objects. Therefore, we only compare this model with our model for object addition setup. [56] uses diffusion model conditioning on a semantic map too. Similarly, we only compare this model with our model for object addition setup.

2.3 Limitations of Existing Studies

Considering the recent studies and their limitations mentioned above, there is some gap in the field that should be filled for better performance. The current studies struggle to provide diversity without disrupting the reality of the produced images. Generating diverse and realistic images is the most crucial goal of the image inpainting task. Therefore, this issue is noteworthy. Another issue is not differentiating partially and entirely erased instances in the masked images. Ignoring the difference between those cases negatively affects the training process and results in inconsistencies in mathematical calculations.

Because of these limitations, we aim to provide an image editing framework that can generate diverse results without adding boundary artifacts to the produced images. We also want to distinguish partially and entirely erased instances, as well as masked and original pixels. In this work, we propose a novel architecture and training pipeline to achieve diverse semantic editing. The training objective includes image reconstruction and adversarial losses. The architecture has a novel way of encoding styles from reference and input images.

Chapter 3

Method

Our approach involves an innovative style encoding process, training that supports style learning, and a generator that produces the desired images. Each of these components will be detailed in this section.

3.1 Style Encoding

The style encoding module aims to encode styles from erased images. However, simply encoding an erased image may not always be sufficient. For instance, if an entire semantic class like the sky or a specific instance such as a car is completely erased, there would be no style information for these parts in the erased images. Prior studies like SPMPGAN [2] and SESAME [1] attempt to encode all information from the erased images. In our research, we encode those styles from the original image during training, as illustrated in Fig. 3.1.

Our style encoder is based on SEAN [4] but has significant differences to tailor the style encoding concept for the inpainting task. SEAN encodes styles from unerased images to generate outputs. It uses an encoder with symmetric convolutional downsampling and upsampling layers. The encoder’s output has the same spatial size as the input image. A region-wise averaging pooling is applied

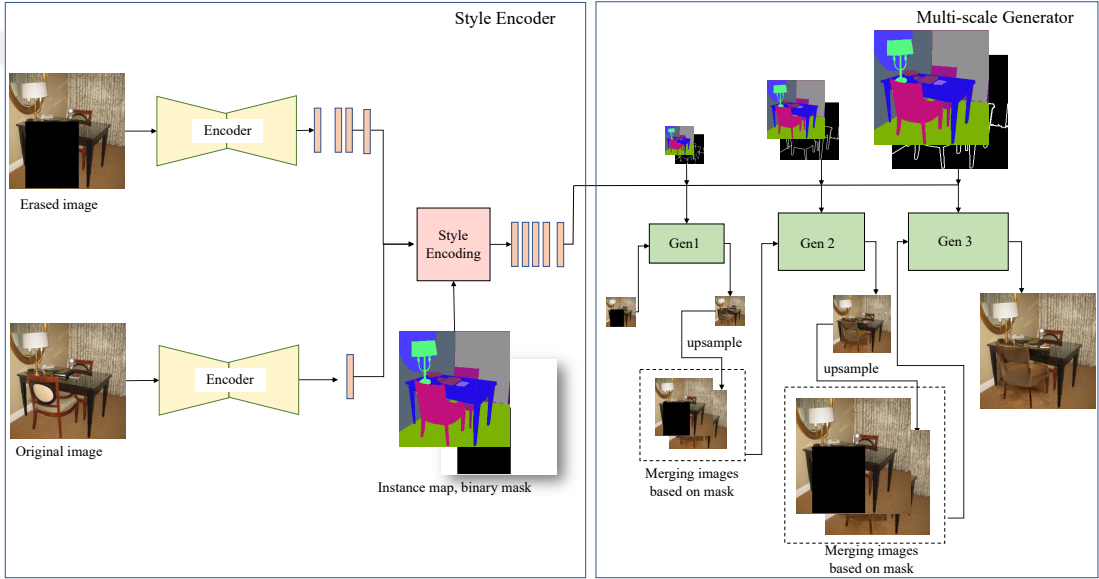


Figure 3.1: The overall pipeline for the diverse semantic image editing. During training, we encode both the erased and original images. The original image is encoded because if a semantic or instance ID is completely erased, then its style is extracted from the original image for the training. The encoded styles, instance maps, and binary mask are fed into the style encoding module which is explained in more detail in Sec. 3.1. The outputs of the style encoding are used in the normalization layers of the generator. The normalization layers additionally take semantic maps, edge maps, and binary masks as inputs. As for the generator, we use a multi-scale generator that refines the predictions at each stage as explained in more depth in Sec. 3.2.

based on semantic maps for stuff (e.g., sky, road) and instance maps for objects (e.g., person, car), a technique also used in pix2pixHD [23]. This pooling creates a bottleneck by removing spatial information from the encoded features. The encoded styles are broadcasted to the original resolution based on the semantic and instance maps which are used in the generator’s adaptive normalization layers.

In our scenario, the pixels associated with a semantic or instance ID may be entirely erased, partially erased, or not erased at all. We encode the styles from the erased input images for the parts that are not erased and partially erased. However, when the input images are masked, some pixels are initially set to 0. This has a significant impact-the use of these features corresponding to the masked pixels in the average operation of instance-wise pooling influences the calculations. Therefore, utilizing instance-wise pooling directly in the image inpainting domain will lead to inconsistency in the style code calculation.

To ensure consistent style code calculation across various masking ratios, we suggest an algorithm in which the average operation is exclusively carried out on the valid area corresponding to the object. It means that only the values from the valid area are used to calculate the mean. Furthermore, we enhance the adaptability of our approach by training a normalization layer based on the valid ratio of the region. This training process ensures that the normalization layer is optimized for the specific characteristics of the region. For instance, if 100 out of 500 pixels are erased for the sky pixels, the valid ratio is 0.2. We utilize the valid ratio as input to the two linear layers of size 1×1 . This input is then processed by a sigmoid operation, which provides the necessary scale and shift to be applied to the style codes. This approach allows the network to treat the partially-erased and not-erased parts differently, as they convey distinct information. If the network determines that the valid ratio information is not useful, the shift and scale can be adjusted to ineffective values (shift=0, scale=1) using the sigmoid operation. However, our observation shows that the network chooses to utilize this information, leading to improved results. We present these results in our Ablation studies.

There may also be objects entirely covered by the mask. In this situation,

the number of valid pixels for these objects and semantic maps is 0. The same input image is used without any mask during training for this scenario, as shown in Fig. 3.1. It is used without a mask to capture the style of that object or semantic class. The remaining styles are derived from the erased input image. In other words, we encode the style code from the original image and perform instance-wise pooling. The valid ratio for these cases is 1, and normalization is applied accordingly. With this algorithm, the calculated style codes are more numerically consistent with the masked input image, and fully masked objects do not negatively impact training performance since there is no information in the masked image that can be encoded to the style code for these objects. It is important to note that in these tasks, previous studies [1, 2] have focused on training for full image reconstruction, even when the input does not contain the information needed to reconstruct the target completely. Our approach gives the network full access to the image’s information and can better support the complete image reconstruction objective.

We offer styles for the completely erased objects and semantic maps from our training images for inference time. This capability enables us to create diverse images using the provided reference styles.

In summary, the Style encoding module displayed in Fig. 3.1 utilizes instance, binary maps, and styles encoded from erased and original images. If a semantic or instance ID has zero valid pixels, the module takes the style from the original image for that particular object. The Style encoding module also conducts normalization based on the valid pixel ratio. The style encoding module receives the binary mask and instance maps to calculate the valid pixel ratio and determine whether an instance or semantic class is completely erased. The generated styles are then utilized in the generator, which will be explained in the following section.

Finally, the architectural specifications of the encoder used in the style encoder are as follows: The encoder starts with a 3×3 convolution, followed by 4 sequential downsampling blocks, reducing the feature map dimensions to 16×16 . After the downsampling blocks, a 3×3 convolution is applied to increase the channel size to 512. In the decoder, 4 upsampling operations, each with a factor of

2, are performed, and each is followed by a 3×3 convolution. The feature maps between the upsampling and convolution layers are concatenated with skip connections from the encoder. Finally, a 3×3 convolution is performed to set the channel size to the style length. Instance normalization, denoted as (+IN), is used throughout except for the final convolution block. The encoder’s channels increase in the sequence of $\{16, 32, 64, 128, 256, 512\}$ and decrease as $\{512, 256, 256, 256, \text{style-length}\}$. After passing through the encoder, the output undergoes the instance-wise pooling operation. The style-length is defined as 128, and other alternatives are presented in Ablation Studies.

3.2 Generator

We employ a hierarchical generator model inspired by SPMPGAN [2]. This model comprises three layers of generators: Gen1, Gen2, and Gen3. The input resolutions for the style encoding and Gen3 are 256×256 . This includes the erased images (I_3), binary (M_3), and semantic maps (S_3). Also, our target image has a resolution of 256×256 . However, Gen1 starts by creating a smaller image. Specifically, the erased input image, mask, and semantic maps are reduced to 64×64 , denoted as I_1, M_1, S_1 . The output of Gen1, O_1 , is then enlarged and combined with I_2 based on the visibility masks using the expression $O_1 \cdot M_1 + I_2 \cdot (1 - M_1)$. This aspect is depicted as merging images based on the mask in Fig. 3.1.

The resulting image is sent to Gen2, and after a comparable process, Gen3 produces our ultimate image. These generators feature an encoder-decoder architecture with skip connections that eliminate the information bottleneck. By integrating skip connections, the decoder can retrieve high-resolution features from the encoder, aiding in the preservation of fine details and contextual information.

Intermediate-generated and erased images are input for the convolutional architecture within the generators. The encoded styles, semantic maps, and edge

maps are input through normalization layers. Styles are encoded once for use across multi-scale generators. The normalization layer we utilize combines context encoding [2], semantic layout, binary mask, and encoded styles. Styles are distributed to the original input resolution based on the semantic and instance IDs of pixels. Semantic layouts, binary masks, edge maps, and styles are initially combined through the channel dimension. Convolutional layers encode two sets of parameters from them: $(\gamma_{s1}, \beta_{s1})$ and $(\gamma_{s2}, \beta_{s2})$. The context modulation parameters, (γ_c, β_c) , are derived from feature maps before applying instance normalization. This approach ensures that context styles encoded through the convolutional layers are conveyed by the feature maps via context modulation [2]. The fusion of these two modulations is done as follows:

$$\gamma_f = (1 + \gamma_{s2}) \cdot \gamma_c + \beta_{s2} \quad (3.1)$$

$$\beta_f = (1 + \gamma_{s1}) \cdot \beta_c + \beta_{s1} \quad (3.2)$$

After the instance normalization is applied to the features, the parameters β_f and γ_f are used to shift and scale, respectively.

The generators’ architectural details are outlined below: For the encoder, a 3×3 convolution is followed by 5, 6, and 7 consecutive convolution blocks for Gen1, Gen2, and Gen3, which conduct downsampling. Each downsampling operation reduces the feature map dimensions by 2. After these downsampling blocks, a final 3×3 convolution is carried out in the encoder. The decoder sections have 5, 6, and 7 consecutive convolution blocks for Gen1, Gen2, and Gen3. The channel size is set to 3, corresponding to the RGB images, through a final 3×3 convolution. Within these convolution blocks, after increasing the feature map dimensions by 2, the encoder’s skip connections are concatenated with the feature maps. The concatenated feature maps are input into the gated convolution layer. Apart from this final convolution block, contextual and style normalization are utilized in addition to instance normalization (denoted as +IN). For Gen 1, the channel progression is as follows: $\{32, 64, 128, 128, 256, 256, 512\}$ in the encoder and $\{256, 128, 64, 32, 32, 3\}$ in the decoder. For Gen 2, the channel progression is as

follows: $\{32, 64, 128, 128, 256, 256, 512, 512\}$ in the encoder and $\{512, 256, 128, 64, 32, 32, 3\}$ in the decoder. For Gen 3, the channel progression is as follows: $\{32, 64, 128, 128, 256, 256, 512, 512, 1024\}$ in the encoder and $\{512, 256, 256, 128, 64, 32, 32, 3\}$ in the decoder.

3.3 Training Parameters

Our model undergoes end-to-end training using a mix of adversarial and reconstruction loss functions. To facilitate adversarial learning, we assign a discriminator to each generator. The discriminators consist of convolutional layers with a filter size of 5×5 . The smallest discriminator that guides Gen1 comprises 4 layers, while the others consist of 5 and 6 layers, respectively. Within the discriminator, the channel size increases from 64 by a factor of 2 at each layer until it reaches 512. Finally, it decreases to 3. We utilize Leaky ReLU as the activation function and apply a stride of 2.

The loss functions we use include adversarial loss (L_{adv}), perceptual loss (L_P), and L1 distance losses (L_1) at every scale. The complete loss function utilized to train the model can be expressed as:

$$L = L_{adv} + 10 \times L_P + L_1$$

Following the SPMPGAN [2], we adopt these parameters without making any adjustments. By modifying the architectures without adjusting any parameters, we are able to achieve improved scores. Our model is trained on a single GPU for 500 epochs with a batch size of 3, using $\beta_1 = 0.5$, $\beta_2 = 0.999$, weight decay of 0.0001, and a learning rate of 0.0001.

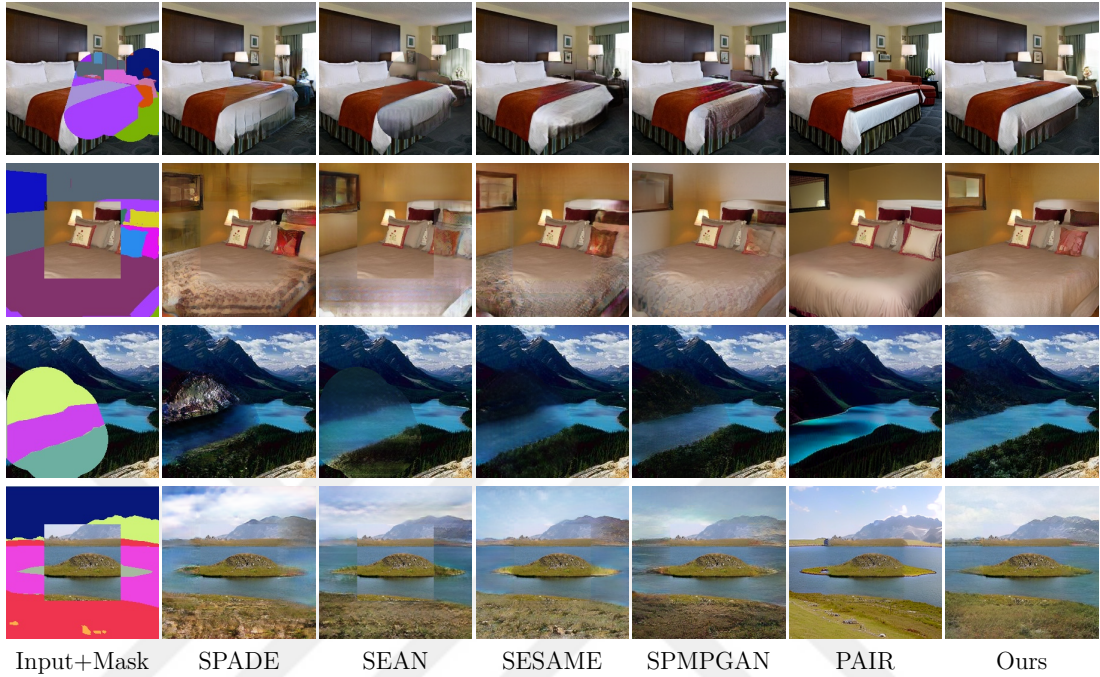


Figure 3.2: Comparisons of methods for free-form and outpainting masks on ADE20k dataset (The first rows are from ADE20k-Room, others are from ADE20k-Landscape). Our method achieves better results in terms of color consistency between the generated and existing pixels and sharpness in the generated parts.

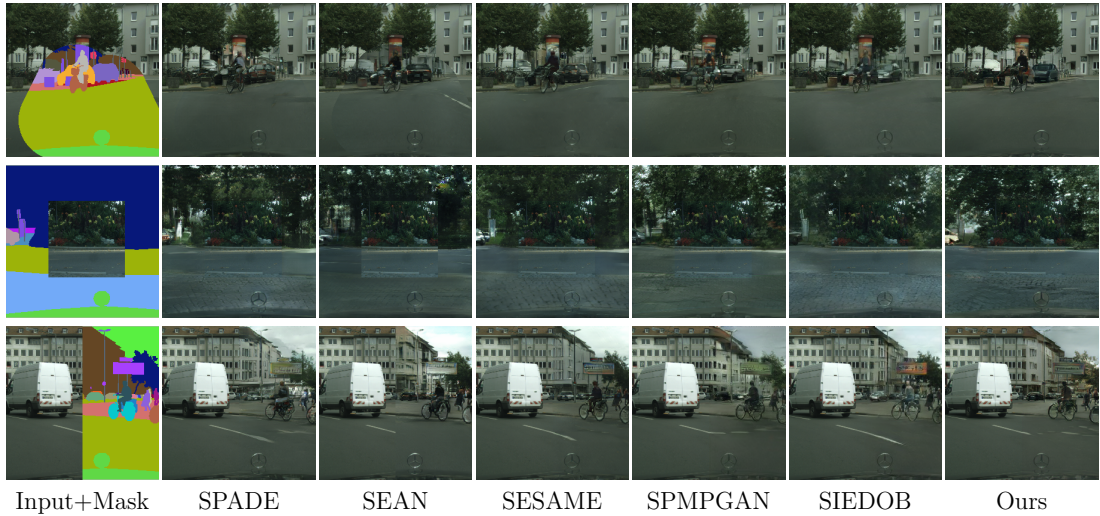


Figure 3.3: Comparisons of methods for free-form, outpainting, and extension masks on Cityscapes dataset. Our method achieves better results in terms of color consistency between the generated and existing pixels and sharpness in the generated parts.

Chapter 4

Experiments

4.1 Datasets

We conduct experiments on three different datasets.

ADE20k-Room. In line with [2], we select a subset of the ADE20k dataset [57] consisting of Bedroom, Hotel Room, and Living Room, referred to as ADE20k-Room. It comprises 2246 images for training and 225 for testing. Although the ADE20k dataset offers 150 semantic labels, we end up with 117 semantic labels for this subset.

ADE20k-Landscape. We choose the landscape subclass from the ADE20k dataset for our experiments, following [2]. The training and testing sets include 1695 images and 153 images, respectively. This subset includes 81 unique semantic classes.

Cityscapes. Our experiments are carried out on the Cityscapes [58] datasets, containing both instance and semantic segmentation labels. The Cityscapes dataset comprises 2,975 training images and 500 validation images of urban street scenes, including 35 semantic classes and 9 instance classes.

	Cityscapes											
	Free-Form			Extension			Outpainting			Adding Objects		
Methods	FID	mIoU	Acc.	FID	mIoU	Acc.	FID	mIoU	Acc.	FID	mIoU	Acc.
SPADE [3]	18.70	<u>57.94</u>	93.77	33.10	56.47	93.63	41.37	55.99	93.65	3.92	58.14	93.80
SEAN [4]	21.27	57.81	<u>93.79</u>	35.38	56.92	<u>93.87</u>	43.06	56.08	93.70	4.08	58.42	93.81
SESAME [1]	<u>17.60</u>	59.16	94.07	<u>32.06</u>	59.66	94.25	<u>39.94</u>	60.86	94.36	<u>3.67</u>	58.55	93.84
SPMPGAN [2]	18.76	57.14	93.61	34.53	55.93	93.55	41.88	56.52	93.55	3.74	58.91	93.84
SIEDOB [6]	18.71	56.78	93.60	34.33	56.14	93.53	43.21	<u>56.54</u>	93.46	4.17	58.65	93.79
Ours	17.03	57.89	93.72	31.05	<u>57.09</u>	<u>93.73</u>	39.53	56.44	<u>93.73</u>	3.49	<u>58.87</u>	<u>93.83</u>
	ADE20k-Room											
SPADE [3]	22.64	25.70	82.39	37.63	25.91	82.07	47.69	26.00	81.72	17.00	26.81	81.65
SEAN [4]	25.74	26.52	81.84	36.92	25.21	81.09	47.05	25.07	81.11	15.39	27.63	82.03
SESAME [1]	21.35	<u>27.94</u>	83.43	34.77	<u>27.94</u>	<u>83.52</u>	44.61	<u>28.54</u>	83.84	15.33	28.48	82.56
SPMPGAN [2]	<u>19.09</u>	27.29	82.96	<u>32.63</u>	27.70	83.36	40.82	28.28	<u>84.02</u>	<u>14.46</u>	<u>28.68</u>	<u>82.88</u>
PAIR [54]	25.63	26.51	81.60	34.38	24.85	81.13	39.62	24.93	80.52	21.71	27.20	81.82
ControlNet [56]	-	-	-	-	-	-	-	-	-	32.69	23.74	77.75
Semantic Diffusion [55]	-	-	-	-	-	-	-	-	-	18.15	28.44	83.05
Ours	18.40	28.09	<u>83.38</u>	32.05	28.07	83.80	<u>40.13</u>	28.71	84.47	13.50	28.80	82.80
	ADE20k-Landscape											
SPADE [3]	32.51	28.89	75.02	59.62	24.51	74.80	78.61	22.82	75.81	-	-	-
SEAN [4]	44.96	27.35	74.00	66.68	24.12	73.91	89.97	21.66	74.78	-	-	-
SESAME [1]	31.88	28.95	75.29	57.15	<u>26.38</u>	75.00	74.38	26.81	78.59	-	-	-
SPMPGAN [2]	<u>25.22</u>	<u>29.10</u>	75.78	<u>52.03</u>	26.26	75.82	<u>69.29</u>	25.15	77.45	-	-	-
PAIR [54]	47.91	28.30	72.90	71.41	27.76	71.49	83.51	28.36	70.93	-	-	-
Ours	24.19	29.24	<u>75.52</u>	49.28	24.08	<u>75.19</u>	64.80	<u>27.16</u>	<u>78.43</u>	-	-	-

Table 4.1: Comparisons of our framework with the state-of-the-art methods. The best results are highlighted with **bold** and the second best with underline. We do not calculate object addition scores for ADE20k-Landscape dataset, since it does not have instances.

4.2 Evaluation Masks

To evaluate the model, we generate predetermined masks for testing and use them to assess all models. There are different types of masks, each with a specific role in evaluating the model from various perspectives.

Free-Form. Free-form masks can have any shape, offering greater flexibility than standard masks, which typically have preset shapes like rectangles or circles. This flexibility allows for more realistic and visually coherent inpainting results, as the model is trained to produce content that aligns with the context of the surrounding image.

Extension. Extension masks are used to remove the left portion of an image, requiring the model to fill in the missing pixels using only the information from the right half of the image. This setup also enables the creation of panoramas.

Outpainting. These masks retain the central quarter of an image, compelling the model to generate the extensions in the borders using only the central parts of the image.

Adding Objects. These masks are created by randomly choosing an instance within the image and then creating the smallest possible rectangular mask to cover that selected instance. This technique is used for Cityscapes and ADE20k-Room datasets containing various instances. However, it is not applicable to the ADE20k-Landscape dataset, which consists of semantic classes.



Figure 4.1: Comparisons of results generated by different methods on ADE20k-Room dataset for object addition. Our method can add objects with diverse styles which improves realism.

4.3 Baselines

We have compared our method with SPADE [3], SEAN [4], SESAME [1], SPMPGAN [2], SIEDOB [6], Semantic-Diffusion [55], PAIR-Diffusion [54], and ControlNet [56]. SPADE and SEAN were developed for tasks related to generating images. For inpainting, we trained them by providing the erased images as input and setting the unerased images as the target. Similarly, Semantic-Diffusion and ControlNet are proposed for semantic image synthesis. We use the pretrained models for object addition setup. Since these are originally image synthesis models, they are not designed to fill erased regions. However, for object addition

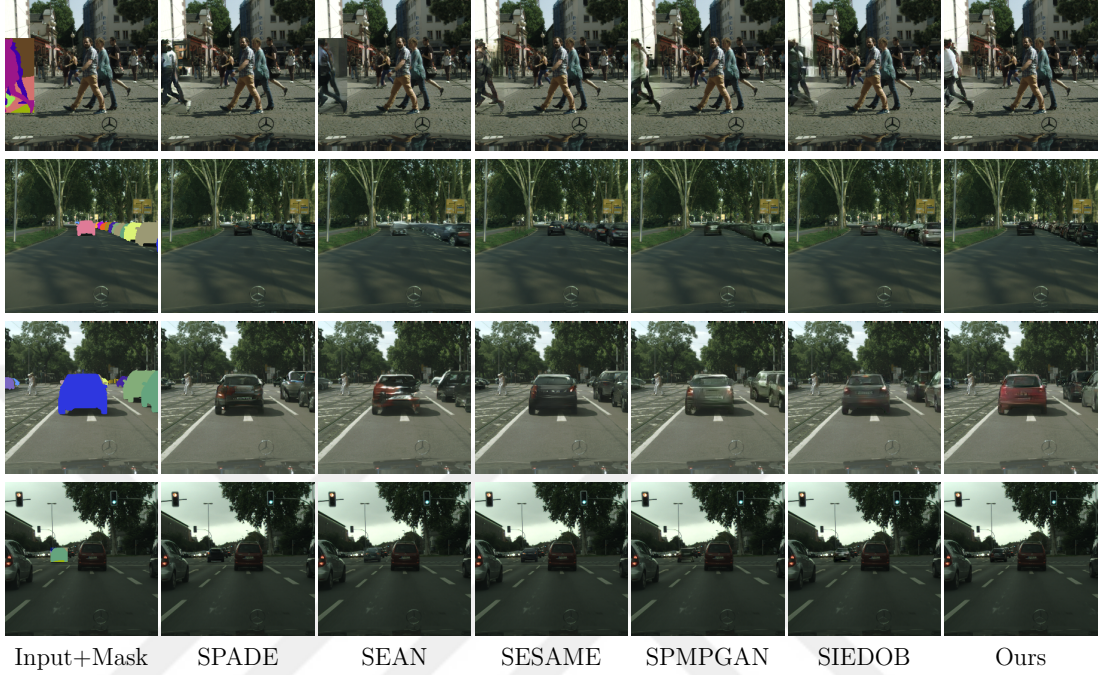


Figure 4.2: Comparisons of results generated by different methods on Cityscapes dataset for object addition. Our method can add objects with diverse styles which improves realism.

setup, it can be used by providing a semantic map edited with the added object. Because of these reasons, we compare it with our model only for object addition setup. Three methods, SESAME, SPMPGAN, and SIEDOB, have been suggested for semantic-based image inpainting. These methods accept erased images as input. While SESAME adopts a similar structure to SPADE, SPMPGAN employs a multi-scale architecture with a context-preserving normalization layer. Both methods are deterministic and are unable to produce varied images. SIEDOB introduces separate handling of foreground and background pixels and involves training an instance inpainting network, a style-diversity object generator, a background generator, and a fusion network to combine their outputs. SIEDOB can create diverse outcomes for foreground objects but not for semantic categories. We have assessed the pretrained models of SPMPGAN and SIEDOB using our validation images. We have trained the other methods on our datasets, utilizing the code released by the original authors. Finally, PAIR-Diffusion is proposed for object-level image editing with semantic map conditioning. It was not originally designed for inpainting, and for partially erased objects, it will

cause inconsistency by carrying the valid parts to the output image. We used the inpainting setup proposed in their source code. Also, reference based object addition can be performed as an inpainting task with a fully erased object and a reference image. Therefore, we also compare this model with ours for the object addition task. Additionally, since it allows reference-based semantic editing, we compare it with our model for scene editing examples.

4.4 Metrics

The performance of the suggested image inpainting model is assessed using three commonly used metrics for this task [1] as well as a user survey. The first metric is the Fréchet Inception Distance (FID), while the other two are the mean Intersection over Union (mIoU) and Accuracy (Acc.).

FID measures the difference between the distribution of the generated and real images based on their feature representations extracted by a pretrained Inception network [59]. A smaller distance indicates higher quality in the generated images.

The scores for **mIoU** and **Acc.** are obtained by running a well-trained segmentation model on the generated images. For the Cityscapes dataset, we utilize the DRN model [60], and for the ADE20k dataset, we employ UperNet [61]. The predicted maps from the segmentation networks are compared with the ground-truth labels to measure mIoU and pixel-level accuracy. This metric assesses how well the generator adheres to the conditional segmentation map.

User Study involves 30 samples evaluated by 10 users. To generate these 30 samples, we use the initial 10 images from the validation set of each dataset without bias. We conduct an A/B test, presenting users with input images and generated images obtained through our method and SPMPGAN. Due to the expenses of user study and the superior performance of SPMPGAN over prior methods, we limit our user study to this comparison. The left-right order is randomized to ensure fair evaluations. Users are asked to assess whether the

generated portions harmonize with the rest of the image while accurately following the content provided as semantic maps.

4.5 Results

We present the quantitative results in Table 4.1 for the Cityscapes, ADE20k-Room, and ADE20k-Landscape datasets. Pretrained models by SIODEB on the Cityscapes dataset are included in our quantitative results. All the methods achieve similar mIoU and accuracy results, indicating their ability to capture semantic content. However, they exhibit varying quality outputs as measured by FIDs. Methods designed for semantic-based image generation, such as SPADE, SEAN, Semantic-Diffusion, PAIR-Diffusion, and ControlNet yield noticeably poorer FID results than others. On the other hand, SESAME, SPMPGAN, and SIEDOB produce better scores. Our method consistently achieves the best FID scores across all datasets and mask types, aligning with the visual results.

Qualitative results for free-form and outpainting masks on ADE20k-Room and ADE20k-Landscape images are illustrated in Fig. 3.2. SPADE, SEAN, and SESAME generate images with distinct boundaries between the visible and inpainted pixels. SPMPGAN achieves improved blending between the generated and existing pixels, although inconsistencies are still present in many examples. As depicted in the figure, our method best preserves the style of partially erased semantic maps. For instance, in the third example, our method accurately fills the remaining lake area with a color similar to the existing pixels, which the other methods fail to achieve. Additionally, free-form and outpainting mask results are shown for Cityscapes images in Fig. 3.3. Our method demonstrates improved color consistency and more distinct foreground objects, as seen in the first row with the cyclist.

The qualitative outcomes for replacing all objects from the same semantic category or adding a single instance are displayed in Fig. 4.1 using ADE20k images.

		ADE20k-Landscape		
Methods		Free-Form	Extension	Outpainting
w/o Style		25.22	52.03	69.29
w/o Valid Ratio		25.95	51.23	66.58
w/o Multi-scale Gen.		27.03	52.48	69.38
Style Dim.	64	25.22	50.12	66.20
Style Dim.	256	25.14	50.62	66.27
Ours		24.19	49.28	64.80

Table 4.2: Ablation Study. FID scores are given for different configurations. We highlight the best results in **bold**.

These visuals demonstrate scenarios in which all instances of the same class are replaced in the scene, or a single instance of a class is added to the scene. In these visual results, our approach is the only one capable of integrating style information for each instance, particularly impacting scenarios where multiple instances are added to a scene. For instance, the third column of Fig. 4.1 illustrates the addition of the curtain class with two instances, represented with different colors. Other methods encounter difficulties generating realistic two-piece curtains, resulting in muddled textures. In contrast, our outcomes are distinct, with clear boundaries and styles for the two-piece curtains. Similar behavior is evident in other instances as well. It is worth noting that we do not compare with SIEDOB on ADE20k images as the model has not been released. Nevertheless, SIEDOB treats the bed, chest, lamp, chair, and table as foreground objects and categorizes curtains as background. Consequently, SIEDOB would not deliver diverse styles for these instances.

In Fig. 4.2, we compare all methods for the object addition and semantic replacement tasks using Cityscapes images. We observe that SIEDOB generates lower-quality results when presented with object additions in the form of a rectangular box, as exemplified in the first instance. These visual results are also reflected in SIEDOB’s FID scores for adding objects, as shown in Table 4.1. While SIEDOB is marginally superior to SPMPGAN in most scenarios, it performs worse than SPMPGAN when presented with objects enclosed by the smallest possible rectangle. We also notice that SIEDOB excels at generating

high-quality car instances in unobstructed scenarios but struggles when cars occlude each other, as demonstrated in the second row featuring side-parked cars. For such instances, our method yields superior results compared to competing approaches.

Finally, in our user study, our method is preferred over SPMPGAN 65% of the time (50% is a tie) due to our method’s superior color consistency and instance generation. These findings align with the FID metrics in Table 4.1.

4.6 Ablation Study

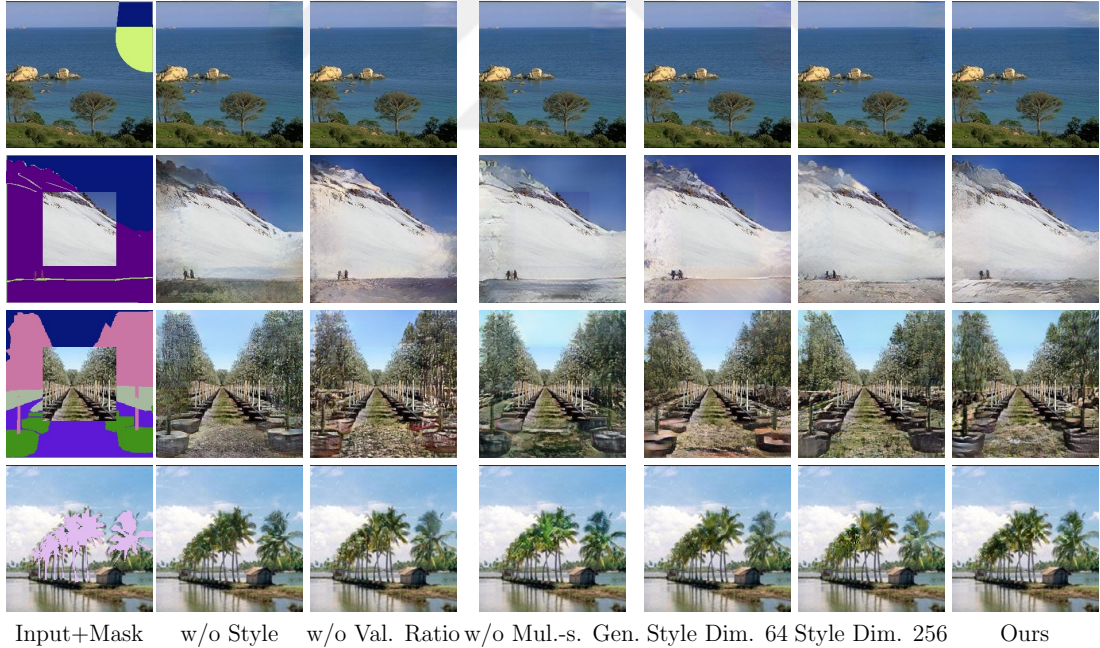


Figure 4.3: Ablation study visual results. Baseline stands for the version where all proposed methods are used.

In Table 4.2, we present an ablation study conducted on the ADE20k-Landscape dataset due to its smaller size and faster training. We monitor the FID scores, which indicate the quality of the generated images. Visual results are also provided in Fig. 4.3.

Initially, we present the outcomes without incorporating style encoding. Removing the style encoding mechanism, resulting in the SPMPGAN model, produces inferior results and is deterministic. Next, we confirm the efficacy of the style encoding approach, considering the valid ratio mentioned in Section 3.1. Our findings demonstrate that the mechanism, which considers the valid ratio and generates shift and scale parameters for the style, enhances the results.

We then assess the utility of the multi-scale generator in our framework. We exclusively train the Gen3 architecture for these trials using the original resolution as input. Its performance falls short of the final model. We also conduct an ablation study on different dimensions of styles extracted via style encoding, determined by the last channel size of the style encoder network. The results are comparable, but we observe that a style dimension of 128 yields the best score. Smaller dimensions may not provide sufficient information for style encoding, while larger dimensions could lead to overfitting by containing excessive information.

4.7 Applications

Our proposed method encompasses a wide range of image editing capabilities within a single framework. It enables us to remove objects from a scene and introduce new ones. One key distinction between our approach and previous ones is our ability to incorporate new objects with diverse styles. Examples of these outcomes are depicted in Fig. 4.4. Likewise, we can alter the style of a semantic category, as illustrated in Fig. 4.5. We provide the resultant outputs and images used as style references for the respective semantic categories. As evidenced by Fig. 4.5, the generated semantic styles resemble the semantic regions from the style image.

Additionally, our method is proficient in creating panoramic views through recursive image outpainting, as demonstrated in Fig. 4.6. To augment the visual results shown in Fig. 4.6, we have gathered images from Flickr to showcase

more intriguing outputs. Our dataset comprises 7389 images, partitioned into training and validation sets, with 6651 images in the training set and 738 images in the validation set. We have used UperNet for image segmentation and trained a model on this dataset. It offers a broader array of landscape variations in contrast to the ADE20k-Landscape dataset, which we used for comparison with previous methodologies. Fig. 4.6 displays images from the validation set. To produce realistic segmentation maps, we have employed a technique that involves extending the segmentation map through mirroring and repeating the border pixels. The resulting outputs exhibit high-quality and realistic details.

Furthermore, we present a demonstration of the model trained on the Flickr dataset, which is accessible through the web page provided in the Abstract.



Figure 4.4: Diverse object addition results generated with our model. Different style codes are used to inpaint the erased chair. Style codes are extracted from corresponding semantic classes of the style images.

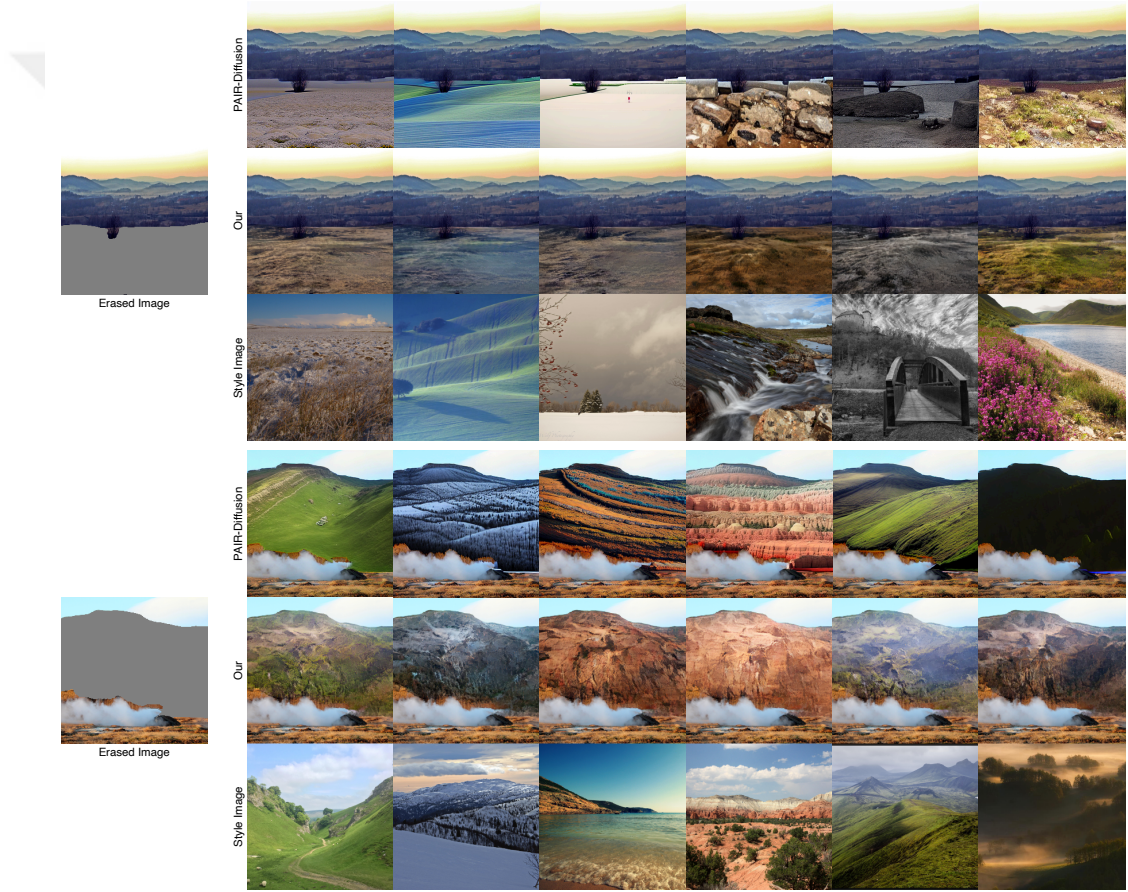


Figure 4.5: Diverse editing results generated with our model. Different style codes are used to inpaint the erased image. Style codes are extracted from corresponding semantic classes of the style images.



Figure 4.6: Panorama images generated with our model. We can extend images with the given semantic map.

Chapter 5

Discussion

In this thesis, we proposed a novel framework that can encode style codes from visible or partially visible instances and use these style codes to generate realistic and diverse outputs. We presented our results to demonstrate the capabilities of our framework, which includes object addition/removal, semantic image inpainting, and semantically conditioned panorama generation. Our model encodes style codes with a mask-aware strategy that provides consistency over calculations, which is essential for image inpainting tasks considering the changing mask shapes. We provide extensive comparisons in Sec. 4 to show the effectiveness of our method. Our ablation study shows that using all proposed strategies together gives the best result among combinations. Also, our method outperforms compared models in both visual and quantitative results.

5.1 Comparison with the Diffusion Models

We compare our model with diffusion models [54], [55], [56] for object addition and scene editing. Considering the visual results, these models could not preserve the input’s general context. Although they generate promising results for the masked region, the generated area is not consistent with the unmasked area. On

the contrary, our results have integrity between these areas while using the style of the given reference image.

Besides, as shown in Fig. 4.4, PAIR-Diffusion could not generate realistic chair samples. Generated chairs are highly consistent with the given reference images. However, it is inconsistent with the input image and the given mask for several structural aspects, such as the direction of the chair and its shape.

Last but not least, diffusion models are computationally expensive, both memory-wise and operation-wise. They require significant resources for training and testing. Therefore, it is also hard to use these models during inference time. This situation limits the usability of diffusion models in real-life applications. On the other hand, since it is GAN-based, our model is faster and computationally cheaper than diffusion models.

5.2 Applicability to Different Architecture

Adapting to various mask ratios is crucial in the image inpainting task. This ability not only enhances the quality of the generated image but also provides better user control. As demonstrated in the results in Sec. 4.6, a valid ratio significantly improves FID results for various mask types, offering a promising outlook. The performance of mask-aware calculation is pivotal for achieving consistent results, especially for inputs with different mask ratios, including fully masked ones.

In this thesis, we proposed a GAN-based model for semantic image editing tasks. However, the mask-aware style encoding module should not be used only with GAN-based models. We can apply this module to transformer-based and diffusion models with minor modifications to have complete semantic control over the generated image. The applicability of this method shows that our method is separate from the proposed architecture and can be used in general semantic image inpainting tasks.

5.3 Limitations and Future Directions

5.3.1 Limitations

In this thesis, we do not use separate models for encoding and decoding style codes for semantic classes. It is more convenient than using separate models for each semantic class since our architecture can adapt to different datasets with different semantic classes. However, this approach has some downsides as well. We used Cityscapes [58], and ADE-20K [2] datasets to train our model. These are relatively small datasets; therefore, our model might not train on a sufficient number of samples for some of the semantic classes. Therefore, this situation might result in unbalanced quality between different semantic classes. While the model can give realistic results for some semantic classes, it might give bad results for others.

5.3.2 Future Directions

To prevent the abovementioned situation, our model can be trained on larger and semantically balanced datasets. In that way, the model can learn weights that give better overall results for most semantic classes. Another possible direction is to use our mask-aware style encoding approach with a diffusion model, considering the rapid development and promising results of diffusion models in image editing tasks. As discussed in Sec. 5.2, our methods can easily be used with different architectures, and using diffusion models with larger datasets might improve our results.

Chapter 6

Conclusion

This thesis proposes a novel framework for image editing with semantic style codes. It comprises a mask-aware encoder and multi-scale generator architecture with skip connections. While the encoder extracts semantic style codes, the multi-scale generator uses these style codes with semantic maps to generate the output image. The encoder uses valid pixel ratio information from the mask to provide a consistent calculation for various mask shapes and ratios. Instead of explicitly using this information, we let the model learn how to utilize it. We also perform average pooling only on the valid pixels. This mask-aware approach is essential, considering different mask shapes and ratios are used in image inpainting task.

Extracting semantic style codes also allows the editing of semantic regions based on a given reference image. The style code of the instance that we want to edit is extracted from the reference image and used to generate the output image. In this way, we can get diverse results with different inputs. We also have a context normalization module for preserving the context of the image while conditioning the provided semantic map and structural information. This module is vital for getting realistic results since it allows the model to produce realistic results without any boundary artifacts in the image.

To show the effectiveness of our work, we extensively compare our model with

other models, including diffusion-based algorithms. We include a wide range of image inpainting and editing methods as baseline models to make an insightful evaluation. To make a fair comparison, we trained some baseline models for the image inpainting task by providing masked images instead of original ones. We compared our model with others using metrics such as FID, mIoU, and accuracy. Each of these metrics provides various insights to evaluate our results. Similarly, we used different mask types to understand different aspects of our model. Based on the quantitative results, it can be seen that our model, along with the methods proposed in this thesis, outperforms other models. In addition to quantitative results, visual results are also provided. Since image inpainting and editing tasks are highly related to the quality of the generated images, these visuals are crucial to evaluating our model’s performance. The visual results show that our model can generate diverse and realistic images without boundary artifacts between the edited and the original areas.

Besides, each proposed method is experimented with in an ablation study to show its effectiveness individually. Ablation study results show that our model learns to use valid ratio information effectively. Another insight is that the style encoding module improves our results while being the source of the diversity in our results. Finally, the optimal style code length is found based on our experiments. It is observed that increasing the length does not always result in better performance.

Based on the insights we get from our results, our study fills some of the gaps in the field we mentioned in Sec. 2. Previous studies experience difficulty in getting diverse results without boundary artifacts between regions. We overcome this issue in our study thanks to the context normalization module that preserves the context in the image and the style encoding module that extracts style codes from instances with different styles. In addition to that, we trained our model with relatively small datasets compared to previous studies in the field. Our model is computationally and memory-wise small, which makes it easy to use in real-life applications. Finally, we differentiate the usage of the partially and entirely erased instances, which was missing in some of the studies we mentioned.

Our model has several applications, including diverse object addition/editing, scene editing with different styles, and panorama generation. We can add objects to a scene with a desired style and edit styles of objects in the image based on the given reference image. In addition to objects, we can edit background classes in the scene, such as sky, mountain, and ground. Another capability of our model is generating panorama images with a small portion of an image and a semantic map prepared by mirroring and extending the corresponding semantic map.

We also present limitations of our model that can be resolved with further studies. Despite these limitations, we proposed a novel approach to semantic image inpainting and editing with superior results.

Bibliography

- [1] E. Ntavelis, A. Romero, I. Kastanis, L. V. Gool, and R. Timofte, “Sesame: Semantic editing of scenes by adding, manipulating or erasing objects,” in *European Conference on Computer Vision*, pp. 394–411, Springer, 2020.
- [2] W. Luo, S. Yang, H. Wang, B. Long, and W. Zhang, “Context-consistent semantic image editing with style-preserved modulation,” in *European Conference on Computer Vision*, pp. 561–578, Springer, 2022.
- [3] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, “Semantic image synthesis with spatially-adaptive normalization,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2337–2346, 2019.
- [4] P. Zhu, R. Abdal, Y. Qin, and P. Wonka, “Sean: Image synthesis with semantic region-adaptive normalization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5104–5113, 2020.
- [5] D. Liu, S. Shetty, T. Hinz, M. Fisher, R. Zhang, T. Park, and E. Kalogerakis, “Asset: autoregressive semantic scene editing with transformers at high resolutions,” *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 1–12, 2022.
- [6] W. Luo, S. Yang, X. Zhang, and W. Zhang, “Siedob: Semantic image editing by disentangling object and background,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1868–1878, 2023.

- [7] X. Dong, W. Li, X. Wang, and Y. Wang, “Cycle-cnn for colorization towards real monochrome-color camera systems,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 10721–10728, 2020.
- [8] G. Kim, J. Park, K. Lee, J. Lee, J. Min, B. Lee, D. K. Han, and H. Ko, “Unsupervised real-world super resolution with cycle generative adversarial network and domain discriminator,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 456–457, 2020.
- [9] A. Dundar, J. Gao, A. Tao, and B. Catanzaro, “Fine detailed texture learning for 3d meshes with generative models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2023.
- [10] A. Dundar, J. Gao, A. Tao, and B. Catanzaro, “Progressive learning of 3d reconstruction network from 2d gan data,” *arXiv preprint arXiv:2305.11102*, 2023.
- [11] S. F. Altindis, A. Meric, Y. Dalva, U. Gudukbay, and A. Dundar, “Refining 3d human texture estimation from a single image,” *arXiv preprint arXiv:2303.03471*, 2023.
- [12] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, “Stargan v2: Diverse image synthesis for multiple domains,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.
- [13] T. Xiao, J. Hong, and J. Ma, “Elegant: Exchanging latent encodings with gan for transferring multiple face attributes,” in *Proceedings of the European conference on computer vision (ECCV)*, pp. 168–184, 2018.
- [14] Y. Dalva, S. F. Altindis, and A. Dundar, “Vecgan: Image-to-image translation with interpretable latent directions,” *Proceedings of the European conference on computer vision (ECCV)*, 2022.
- [15] A. Dundar, K. Sapra, G. Liu, A. Tao, and B. Catanzaro, “Panoptic-based image synthesis,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8070–8079, 2020.

- [16] G. Liu, A. Dundar, K. J. Shih, T.-C. Wang, F. A. Reda, K. Sapra, Z. Yu, X. Yang, A. Tao, and B. Catanzaro, “Partial convolution for padding, inpainting, and image synthesis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
- [17] Y. Dalva, H. Pehlivan, O. I. Hatipoglu, C. Moran, and A. Dundar, “Image-to-image translation with disentangled latent vectors for face editing,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.
- [18] L. Zhang and M. Agrawala, “Adding conditional control to text-to-image diffusion models,” *arXiv preprint arXiv:2302.05543*, 2023.
- [19] P. Esser, R. Rombach, and B. Ommer, “Taming transformers for high-resolution image synthesis,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.
- [20] X. Xu, J. Guo, Z. Wang, G. Huang, I. Essa, and H. Shi, “Prompt-free diffusion: Taking” text” out of text-to-image diffusion models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8682–8692, 2024.
- [21] A. Dundar, K. J. Shih, A. Garg, R. Pottorf, A. Tao, and B. Catanzaro, “Unsupervised disentanglement of pose, appearance and background from images and videos,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3883–3894, 2021.
- [22] T.-C. Wang, A. Mallya, and M.-Y. Liu, “One-shot free-view neural talking-head synthesis for video conferencing,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10039–10049, 2021.
- [23] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, “High-resolution image synthesis and semantic manipulation with conditional gans,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8798–8807, 2018.
- [24] I. Najdenkoska, A. Sinha, A. Dubey, D. Mahajan, V. Ramanathan, and F. Radenovic, “Context diffusion: In-context aware image generation,” *arXiv preprint arXiv:2312.03584*, vol. 5, 2023.

- [25] A. Andonian, S. Osmany, A. Cui, Y. Park, A. Jahanian, A. Torralba, and D. Bau, “Paint by word,” *arXiv preprint arXiv:2103.10951*, 2021.
- [26] Y. Zeng, Z. Lin, J. Zhang, Q. Liu, J. Collomosse, J. Kuen, and V. M. Patel, “Scenecomposer: Any-level semantic image synthesis,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22468–22478, 2023.
- [27] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in neural information processing systems*, vol. 27, 2014.
- [28] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.
- [29] T. Wang, T. Zhang, B. Zhang, H. Ouyang, D. Chen, Q. Chen, and F. Wen, “Pretraining is all you need for image-to-image translation,” in *arXiv*, 2022.
- [30] R. Mokady, A. Hertz, K. Aberman, Y. Pritch, and D. Cohen-Or, “Null-text inversion for editing real images using guided diffusion models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6038–6047, 2023.
- [31] S. Li, B. Zeng, Y. Feng, S. Gao, X. Liu, J. Liu, L. Li, X. Tang, Y. Hu, J. Liu, *et al.*, “Zone: Zero-shot instruction-guided local editing,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6254–6263, 2024.
- [32] J. Singh, S. Gould, and L. Zheng, “High-fidelity guided image synthesis with latent diffusion models,” *arXiv preprint arXiv:2211.17084*, 2022.
- [33] O. Avrahami, T. Hayes, O. Gafni, S. Gupta, Y. Taigman, D. Parikh, D. Lischinski, O. Fried, and X. Yin, “Spatext: Spatio-textual representation for controllable image generation,” *arXiv preprint arXiv:2211.14305*, 2023.

- [34] N. Li, Q. Liu, K. K. Singh, Y. Wang, J. Zhang, B. A. Plummer, and Z. Lin, “Unihuman: A unified model for editing human images in the wild,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2039–2048, 2024.
- [35] C. Liu, X. Li, and H. Ding, “Referring image editing: Object-level image editing via referring expressions,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13128–13138, 2024.
- [36] G. Couairon, J. Verbeek, H. Schwenk, and M. Cord, “Diffedit: Diffusion-based semantic image editing with mask guidance,” *arXiv preprint arXiv:2210.11427*, 2022.
- [37] C. Mou, X. Wang, J. Song, Y. Shan, and J. Zhang, “Dragondiffusion: Enabling drag-style manipulation on diffusion models,” *arXiv preprint arXiv:2307.02421*, 2023.
- [38] J. Guo, X. Xu, Y. Pu, Z. Ni, C. Wang, M. Vasu, S. Song, G. Huang, and H. Shi, “Smooth diffusion: Crafting smooth latent spaces in diffusion models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7548–7558, 2024.
- [39] J. Jeong, M. Kwon, and Y. Uh, “Training-free content injection using h-space in diffusion models,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5151–5161, 2024.
- [40] A. B. Yildirim, H. Pehlivan, B. B. Bilecen, and A. Dundar, “Diverse inpainting and editing with gan inversion,” *IEEE International Conference on Computer Vision*, 2023.
- [41] Y. Yu, L. Zhang, H. Fan, and T. Luo, “High-fidelity image inpainting with gan inversion,” in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVI*, pp. 242–258, Springer, 2022.
- [42] W. Wang, L. Niu, J. Zhang, X. Yang, and L. Zhang, “Dual-path image inpainting with auxiliary gan inversion,” in *Proceedings of the IEEE/CVF*

- Conference on Computer Vision and Pattern Recognition*, pp. 11421–11430, 2022.
- [43] O. Tov, Y. Alaluf, Y. Nitzan, O. Patashnik, and D. Cohen-Or, “Designing an encoder for stylegan image manipulation,” *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–14, 2021.
 - [44] H. Pehlivan, Y. Dalva, and A. Dundar, “Styleres: Transforming the residuals for real image editing with stylegan,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1828–1837, 2023.
 - [45] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019.
 - [46] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, “Image inpainting for irregular holes using partial convolutions,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 85–100, 2018.
 - [47] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, “Free-form image inpainting with gated convolution,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4471–4480, 2019.
 - [48] W. Li, Z. Lin, K. Zhou, L. Qi, Y. Wang, and J. Jia, “Mat: Mask-aware transformer for large hole image inpainting,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10758–10768, 2022.
 - [49] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, “RePaint: Inpainting using denoising diffusion probabilistic models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11461–11471, 2022.
 - [50] A. B. Yildirim, V. Baday, E. Erdem, A. Erdem, and A. Dundar, “Inst-inpaint: Instructing to remove objects with diffusion models,” *arXiv preprint arXiv:2304.03246*, 2023.

- [51] B. Yang, S. Gu, B. Zhang, T. Zhang, X. Chen, X. Sun, D. Chen, and F. Wen, “Paint by example: Exemplar-based image editing with diffusion models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18381–18391, 2023.
- [52] C. Corneanu, R. Gadde, and A. M. Martinez, “Latentpaint: Image inpainting in latent space with diffusion models,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 4334–4343, 2024.
- [53] H. Liu, Y. Wang, B. Qian, M. Wang, and Y. Rui, “Structure matters: Tackling the semantic discrepancy in diffusion models for image inpainting,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8038–8047, 2024.
- [54] V. Goel, E. Peruzzo, Y. Jiang, D. Xu, N. Sebe, T. Darrell, Z. Wang, and H. Shi, “Pair-diffusion: Object-level image editing with structure-and-appearance paired diffusion models,” *arXiv preprint arXiv:2303.17546*, 2023.
- [55] W. Wang, J. Bao, W. Zhou, D. Chen, D. Chen, L. Yuan, and H. Li, “Semantic image synthesis via diffusion models. arxiv 2022,” *arXiv preprint arXiv:2207.00050*, 2022.
- [56] L. Zhang, A. Rao, and M. Agrawala, “Adding conditional control to text-to-image diffusion models,” 2023.
- [57] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, “Scene parsing through ade20k dataset,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [58] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.
- [59] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” *Advances in neural information processing systems*, vol. 30, 2017.

- [60] F. Yu, V. Koltun, and T. Funkhouser, “Dilated residual networks,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [61] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, “Unified perceptual parsing for scene understanding,” in *European Conference on Computer Vision (ECCV)*, 2018.

