# Advancing Image Inpainting: From GANs to Diffusion Models

by **Yongle Zhang**

Thesis submitted in fulfilment of the requirements for the degree of

*Doctor of Philosophy*

under the supervision of Professor Qiang Wu

School of Electrical and Data Engineering

Faculty of Engineering and IT

University of Technology Sydney

December 16, 2025

# Certificate of Original Authorship

I, Yongle Zhang, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Electrical and Data Engineering at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution.

This research was supported by an Australian Government Research Training Program (RTP) Scholarship doi.org/10.82133/C42F-K220.

Signature: Production Note: Signature removed prior to publication.

Date: December 16, 2025

# Abstract

Image inpainting, the task of reconstructing corrupted regions in images to achieve visually coherent results, is pivotal in computer vision, enabling applications such as photo restoration, object removal, and creative design. Despite significant advancements, existing methods struggle with complex real-world corruptions involving diverse semantic content, intricate structural details, and precise spatial control. This thesis addresses three critical challenges in image inpainting, spanning generative adversarial network (GAN)-based and diffusion-based frameworks, to advance the field toward solutions that restore semantic plausibility, structural fidelity, and spatial consistency, thereby meeting the demands of diverse applications.

The first challenge in GAN-style inpainting involves restoring multiple corrupted semantic regions, such as facial features with distinct class-level content. This is complicated by the need for semantic coherence across diverse regions, as existing methods that rely on implicit learning of semantics (e.g., GANs trained on large datasets to infer patterns) struggle with effective restorations, while methods that use explicit guidance (e.g., from pre-trained models or predicted semantic maps) often propagate errors from inaccurate initial predictions. To address this, a dual-task generative framework is proposed that jointly refines semantic segmentation predictions and texture restoration, iteratively correcting segmentation errors using restored textures and guiding inpainting with refined semantics.

The second challenge, also in GAN-style inpainting, focuses on reconstructing structurally intricate regions, such as repetitive patterns in building facades, where semantic priors alone are insufficient for capturing fine-grained spatial details. Existing methods incorporate structural priors (e.g., sketches) through direct or modulated feature fusion, but they lack dynamic and global adaptation to evolving inpainting features, leading to inconsistencies or artifacts. An adaptive multi-modal framework is introduced, inspired by human drawing processes, which dynamically integrates structural and semantic priors to ensure both structural accuracy and

semantic plausibility throughout the progressive inpainting process.

The third challenge, specific to Diffusion-style inpainting, concerns recovering partially occluded objects with precise posture control. This task is hindered by the limitations of text-only guidance in conveying pixel-aligned spatial attributes, while existing visual guides ignore residual cues from uncorrupted object regions, leading to pose mismatches. This thesis solves this via a dual-path visual control module, which explicitly models interactions between uncorrupted visual cues and guided sketches, then integrates guided sketches into the diffusion-based inpainting process. This spatial bridge ensures that sketch-controlled generated regions seamlessly connect to existing structures, achieving precise posture control and consistency. Additionally, two novel datasets, CUB-sketch and MSCOCO-sketch, are introduced to benchmark posture-aware inpainting.

# Dedication

I would like to dedicate this thesis to my loving family.

# Acknowledgements

At the end of my PhD journey, I can still clearly remember the scenes of my first arrival in Sydney and my first visit to UTS to collect my student card. Time seems to fly, yet the entire journey has been filled with countless unforgettable experiences—just as the lyrics say, "A year seems like a day, and a day seems like a year."

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Qiang Wu. I have learned a great deal from him throughout my PhD, which I divide into three distinct periods. The first phase began with my initial application for his PhD position and preparation for the IELTS language tests, followed by a one-year delay caused by the COVID-19 pandemic. Throughout this time, Professor Wu provided unwavering support. In the second phase, while I was unable to enter Australia due to travel restrictions, I studied remotely in China for seven months. During this period, Professor Wu guided me on how to conduct literature reviews and determine research topics. The third phase commenced when I finally arrived in Sydney and began my on-campus studies. He patiently taught me how to conduct research and solve challenging problems. In every meeting, he would carefully delve into the details of my recent work and the issues I encountered. He also showed me how to revise a paper repeatedly until it met submission standards. From Professor Wu, I came to truly understand how a mentor teaches students to fish rather than simply giving them a fish. His knowledge and personal guidance have paved the way for my future research career and how to conduct scientific research. Life's path is long, and I am grateful to have met my mentor during my golden years.

Second, I extend my sincere gratitude to my co-supervisor, Dr. Jingsong Xu, for his valuable suggestions and support in parts of my research. I would also like to sincerely thank the classmates, friends, and collaborators I met during my PhD. Thanks to Professor Jian Zhang (UTS) for his research support; to my graduate classmate Ruotong Hu for collaborating with

me during her visiting study at UTS and for being my meal companion; to Yimin Liu for our collaboration and shared work during her visiting study at UTS; and to Shangyi Sun for joining me for meals and kindly providing his computing resources for my research. I am also thankful to Junjie Huang, Xinyuan Liu, Yuenian Chen, Mingfei Tong, Zhaoqi Cui, Ruohong Cao, and others for going out together, sharing meals, and cooking as a group. Without them, my life in Sydney would have been much lonelier and less colorful. Finally, thanks to my graduate classmate Yingyu Wang for our shared rental life, which brought familiarity and convenience to me in Sydney.

Finally, I would like to thank my parents, who have supported me all the way from my undergraduate studies to my PhD.

Yongle Zhang

December 16, 2025

Sydney, Australia

# List of Publications

## Journal Papers

J-1. **Yongle Zhang**, Yimin Liu, Ruotong Hu, Qiang Wu, Jian Zhang, "Mutual Dual-Task Generator with Adaptive Attention Fusion for Image Inpainting," IEEE Transactions on Multimedia, 2023. This paper is presented in Chapter 3.

J-2. **Yongle Zhang**, Yimin Liu, Hao Fan, Ruotong Hu, Jian Zhang, Qiang Wu, "Consistent Image Inpainting with Pre-Perception and Cross-Perception Collaborative Processes," IEEE Transactions on Image Processing, 2025. This paper is presented in Chapter 4.

J-3. **Yongle Zhang**, Yimin Liu, Yan Huang, Qiang Wu, "Recovering Partially Corrupted Objects via Sketch-Guided Bidirectional Feature Interaction," submitted to IEEE Transactions on Image Processing, 2025 (arXiv: `https://arxiv.org/abs/2503.07047v2`). This paper is presented in Chapter 5.

J-4. **Yongle Zhang**, Yimin Liu, Qiang Wu, "Boundary-Constrained Object Inpainting with Object-Visual-Aware Textual Prompts," In writing, 2025.

## Co-Authored Papers

J-1. Ruotong Hu, Xianzhi Wang, Xiaojun Chang, **Yongle Zhang**, Yeqi Hu, Xinyuan Liu, Shusong Yu, "CStrCRL: Cross-View Contrastive Learning Through Gated GCN With Strong Augmentations for Skeleton Recognition," IEEE Transactions on Circuits and Systems for Video Technology, 2023.

J-2. Ruotong Hu, Xianzhi Wang, Xiangqian Ding, **Yongle Zhang**, Xiaowei Xin, Wei Pang, Shusong Yu, "Unsupervised Domain Adaptation for Skeleton Recognition With Fourier Analysis," IEEE Internet of Things Journal, 2024.

J-3. Yimin Liu, Meibin Qi, **Yongle Zhang**$^{*}$(*Corresponding*), Qiang Wu, Jingjing Wu, Shuo Zhuang, "Improving Consistency of Proxy-Level Contrastive Learning for Unsupervised Person Re-Identification," IEEE Transactions on Information Forensics and Security, 2024.

J-4. Yimin Liu, Meibin Qi, **Yongle Zhang**, Wenbo Xu, Qiang Wu, "Camera-aware Embedding Refinement for Unsupervised Person Re-Identification," Knowledge-Based Systems, 2025.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Background

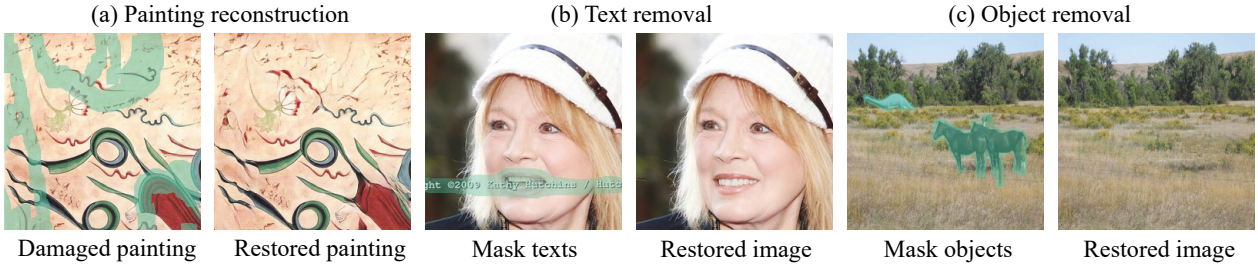



Figure 1.1: Image inpainting has various applications, including (a) painting reconstruction, (b) text removal, and (c) object removal.

Image inpainting [1], the technique of reconstructing missing or corrupted regions of an image such that the restored content is visually indistinguishable from the original to a human observer, remains a fundamental task in computer vision [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. Its applications span various domains, including the restoration of deteriorated or aged photographs and artworks [14], [15], the removal of undesired elements such as watermarks, logos, and subtitles [16], [17], and the seamless erasure or replacement of unwanted objects [3], [18], [19], [20]. As illustrated in Fig. 1.1, image inpainting can effectively reconstruct damaged paintings, erase overlaid text, or remove distracting foreground objects (e.g., the horse), producing visually plausible and coherent results.

Ideally, a robust image inpainting algorithm should be capable of flexibly handling arbitrary corruptions, ensuring that the reconstructed content closely matches the original image in both

(a) Multi-semantic region damage (b) Structure-rich region damage (c) Large-area object damage

Figure 1.2: Examples of corrupted images with diverse types of damage: (a) multi-semantic region damage (e.g., eye, nose, and mouth), (b) structure-rich region damage within the brick-pattern building category, and (c) large-area object damage.

structural layout and semantic meaning. Achieving this involves significant challenges. First, in scenarios where the corrupted area spans multiple semantic regions with distinct class-level content (e.g., occlusion of all facial features, such as the eyes, nose, and mouth, as depicted in Fig. 1.2 (a)), the network must learn to generate semantically coherent and spatially consistent features that account for each class. Second, when dealing with semantically corrupted yet structurally intricate regions (e.g., a building facade with repetitive brick patterns, as shown in Fig. 1.2 (b)), generating high-frequency, structurally precise details within the inpainted regions becomes particularly demanding.

These two challenges pose significant technical barriers for deep learning-based inpainting networks, particularly those built on the generative adversarial networks (GANs) paradigm [21]. GANs typically frame image inpainting as a conditional image generation problem: a generator is trained to synthesize the missing region conditioned on the undamaged parts of the image, while a discriminator evaluates the authenticity of the generated content [9], [22]. Numerous studies have proposed GAN-based models that demonstrate impressive results [9], [10], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], and this thesis refers to such approaches collectively as GAN-style inpainting.

More recently, Diffusion models [13], [36] have emerged as a compelling alternative, leading to the development of diffusion-style inpainting methods. These models progressively refine randomly sampled Gaussian noise through multiple denoising steps to generate high-quality images. The iterative generation process enables better preservation of fine-grained details and allows more flexible conditioning on external information, such as textual descriptions [13], [37],

[38], [39], [40], [41]. Diffusion-style methods have exhibited notable strengths in addressing large or fully masked object regions, as illustrated in Fig. 1.2 (c), which depicts a largely masked object scenario. However, when objects are partially occluded, inpainting is required to reconstruct the missing parts of the subjects to be consistent with the non-occluded parts in terms of both subject posture and the existing texture information. However, existing diffusion-based methods lack spatial reasoning capabilities and are therefore unable to guarantee plausible object posture.

This thesis aims to systematically investigate the aforementioned challenges in image inpainting across both GAN-style and diffusion-style frameworks.

For GAN-style inpainting, this thesis discusses two primary challenges:

- The challenge of restoring multiple corrupted semantic regions.
- The challenge of restoring complex structures while maintaining semantic plausibility.

For Diffusion-style inpainting, the key challenge under investigation is:

- The challenge of recovering the partially occluded objects using the diffusion model.

### 1.1.1 Challenges in Restoring Multiple Corrupted Semantic Regions in GAN-style Inpainting

In GAN-style image inpainting, a major difficulty arises when the corrupted regions contain multiple subareas, each associated with a different semantic class. Fig. 1.2 (a) provides an example where random occluded areas cover multiple critical facial features such as the eyes, nose, and mouth. Even for human observers, accurately inferring and reconstructing fine-grained textures with appropriate semantics is not trivial. Existing works tackling this problem generally fall into two broad categories:

1. **Approaches for enforcing implicit semantic consistency**: These leverage GANs to implicitly learn semantic features from large-scale datasets, using adversarial training to generate plausible content [9], [10], [22], [24], [26], [30], [42], [43], [44], [45].
2. **Approaches for enforcing explicit semantic consistency**: These apply pre-trained semantic feature constraints [46], [47], [48] or introduce explicit semantic priors [23], [25], [28], [49] to enhance the semantic consistency of inpainted results.

The methods of the first category rely heavily on the capacity of GANs to extract implicit semantic representations from data, enabling the model to generate coherent content even in the presence of substantial occlusions. However, the ability of such methods to handle large free-form corruptions that affect multiple semantic categories is inherently limited. In contrast, semantic-guided methods, i.e., the methods of the second category, have shown promise in leveraging explicit cues to improve inpainting quality. For instance, feature constraint approaches introduce supervision from pretrained networks such as VGG [46], [50] or StyleGAN [51], [52] to focus on contextual semantics in uncorrupted areas [10], [53]. These constraints enhance the model's ability to generate semantically faithful content in complex regions.

Another way for enforcing semantic consistency involves generating intermediate semantic priors, such as semantic segmentation maps [54], [55], from the corrupted image. These predicted priors then guide the subsequent inpainting process to ensure semantic alignment. However, generating accurate segmentation maps from severely corrupted inputs is itself a non-trivial problem. Errors in the predicted semantic priors may propagate and negatively impact the final inpainting results. This raises an important question: how can semantic errors in image inpainting be mitigated to enhance the effectiveness of subsequent guidance? To address this issue, this thesis proposes a dual-task generative framework that jointly optimizes semantic segmentation and image inpainting. Inspired by prior segmentation-guided methods [23], [28], the proposed approach enables mutual refinement of segmentation priors and texture restoration, thereby improving robustness against initial prediction errors in heavily corrupted regions.

### 1.1.2 Challenges in Restoring Complex Structures and Reasonable Semantics in GAN-style Inpainting

While explicit semantic priors help ensure correct category-level guidance in image inpainting, they often lack sufficient detail to recover structurally intricate features that demand fine-grained spatial details within each semantic region. For example, restoring the texture of a brick wall, as illustrated in Fig. 1.2 (b), requires not only semantic knowledge that the region belongs to the "building" class, but also the ability to replicate intricate local patterns, such as the alignment, repetition, and perspective of the bricks. Semantic segmentation maps, although useful for conveying category-level layouts, are inherently limited in their capacity to preserve these intra-class structural characteristics.

To overcome this limitation, recent research has introduced structural priors—such as Canny edges [56], gradient maps, contours, or hand-drawn sketches—into the inpainting process [27], [29], [57], [58], [59], [60]. These structure-aware methods typically follow a pipeline similar to semantic-guided models: first, a structure generator produces the edge or sketch representation; second, a texture generator leverages this structural input to guide the reconstruction of image details. These methods aim to enhance the spatial precision and local consistency of the inpainted content by enforcing alignment with guided structural features.

However, structure guidance and semantic segmentation serve complementary roles: the former provides low-level structural cues, while the latter focuses on high-level category layouts and class relationships. As such, semantic information dictates what class content should be present, while structural information constrains how it should be spatially organized. Therefore, combining both types of guidance in a coherent and adaptive manner is critical for restoring content that is both semantically plausible and structurally accurate. This thesis identifies two mainstream approaches to fusing such guidance information:

1. **Direct feature fusion**: Structural and semantic guidance features are directly concatenated with corrupted images or internal features within the inpainting network [23], [58], [59], [60], [61], [62]. This approach enables the network to process all available information simultaneously but lacks mechanisms to prioritize or adaptively balance guidance signals.

2. **Modulated feature fusion**: Guidance information is injected through transformation functions, such as conditional normalization or feature gating, to modulate the image inpainting process in a spatially aware manner [25], [33], [49], [63], [64], [65], [66].

However, both strategies lack dynamic and global utilization of the guidance information throughout the inpainting process, overlooking its correlation with the evolving inpainting feature representations. As reconstruction progresses, the degree of missing information changes dynamically, requiring adaptive guidance to maintain optimal performance. Consequently, these methods may provide excessive or insufficient guidance at various stages, resulting in semantic inconsistencies or structural artifacts in the final output.

This remains an open challenge: is there a feasible approach to enhance structure and semantics in image inpainting through dynamic and adaptive guidance? To address this limitation, this thesis draws inspiration from human drawing processes [67], [68], [69], where structural

outlines and semantic understanding are iteratively refined alongside texture rendering. Accordingly, this thesis proposes an adaptive multi-modal inpainting framework that not only integrates structural and semantic priors but also dynamically modulates them based on the inpainting progress. This approach enables more accurate restoration of semantically coherent and structurally consistent content in challenging scenarios.

### 1.1.3 Challenges in Recovering Partially Occluded Objects in Diffusion-style Inpainting

Textual prompts offer a versatile and expressive means to specify high-level semantic concepts, such as object appearance, category, or general spatial configuration, in image inpainting tasks, particularly for largely or fully occluded object regions (see Fig. 1.2(c)). In diffusion-based inpainting, these prompts guide the denoising trajectory via cross-modal attention mechanisms. However, despite the richness of linguistic descriptions, text-only conditioning struggles to control spatially precise attributes, such as posture, orientation, and pixel-level geometry of corrupted objects. Consequently, achieving accurate posture consistency between the object's non-occluded parts and the generated regions from the inpainting process remains a critical yet underexplored challenge in text-guided diffusion-based inpainting. Existing approaches can be broadly classified into two categories:

1. **Text-only inpainting**: These models utilize text embeddings derived from vision-language models such as CLIP [70] and incorporate them into the diffusion process using cross-attention mechanisms across all layers [13], [38], [41], [71], [72], [73], [74], [75], [76]. While effective in steering generation toward semantically appropriate results, these approaches cannot impose precise spatial constraints.

2. **Visual-guided inpainting**: These methods introduce sketches, contours, or keypoints as auxiliary inputs to encode structural intent [39], [77], [78]. The visual prompts provide an additional spatial prior that can complement textual instructions.

Current text-guided diffusion models frequently produce reconstructions with incorrect poses or distorted shapes. This difficulty arises because natural language operates at an abstract and symbolic level, whereas posture control in image inpainting requires spatially grounded, pixel-aligned information. Even prompts that explicitly describe an object's orientation (e.g., "a horse facing left with a raised front leg") may lead to results that deviate from the intended spatial

configuration due to the inherent limitations of text in conveying pixel-aligned information.

While visual guidance improves structure recovery, most existing models treat such guidance as one-way visual control: the sketch or edge is directly injected into the model without considering how it interacts with uncorrupted object regions or fragments. This approach is well-suited for fully occluded objects, where no residual information exists, but performs poorly in partially occluded scenarios. In such cases, uncorrupted fragments provide pose-relevant cues, such as local orientation, symmetry, or part relationships, which, if ignored, lead to reconstructions lacking spatial continuity or alignment with the original posture. In real-world applications, partial occlusion is more common than total occlusion, making it essential for inpainting systems to leverage all available spatial context. This includes both externally provided visual priors (e.g., sketches) and in-image residual structures (e.g., uncorrupted object parts). Hence, achieving precise posture control requires not just pixel-grained visual prompts, but an interactive mechanism that unifies internal information and external guidance.

To address this challenge, this thesis proposes a dual-path visual control module that explicitly models the connection between visible object regions and guided sketches. Unlike one-way visual prompts, this design allows mutual information exchange between known (uncorrupted) and unknown (corrupted) regions before sketch-based control is applied to the corrupted area. The module jointly reasons over both paths to ensure that the generated posture is consistent with the visible structure and aligns with the intended sketch. Furthermore, two novel datasets are introduced to evaluate posture-aware inpainting capabilities. These datasets include a range of partially and largely occluded objects, paired with user-style sketches and descriptive prompts, enabling systematic benchmarking of partial-object inpainting.

## 1.2 Research Problems and Contributions

1. **Joint Optimization of Semantic Refinement and Image Texture Restoration**:

   Image corruptions or missing regions in real-world scenarios are highly diverse, often involving multiple distinct semantic regions. Intuitively, providing accurate semantic prior information for these corrupted regions can guide semantically reasonable image inpainting. However, generating high-quality semantic priors from severely corrupted images is non-trivial. Consequently, errors or ambiguities in the semantic prior guidance

can ultimately degrade the inpainting quality. Mutual co-optimization of semantic prior generation and target image texture restoration offers a promising solution to mitigate errors in semantic guidance. This thesis explores this approach to enhance inpainting outcomes.

2. **Adaptively Joining Structure Priors and Semantic Guidance for Complex Image Inpainting**:

   A more challenging scenario arises when corrupted semantic regions involve structurally complex objects (e.g., non-rigid structures like buildings with intricate brick patterns). Semantic-guided inpainting approaches often prove ineffective here, as they leverage category-level semantics but lack intra-class structural priors. While introducing structural guidance (e.g., Canny edges) can alleviate this issue, distinct guidance types (e.g., edges and segmentation maps) focus on complementary aspects of the scene that contribute to the image inpainting process. Hence, a key question is how to jointly and adaptively apply both types of guidance during the progressive texture generation process. Drawing inspiration from how human artists collaboratively utilize structural and semantic information during drawing, this thesis proposes a dynamic collaboration scheme that models structure and semantic priors as complementary modalities for addressing this challenge.

3. **Spatial Reasoning in Diffusion-based Inpainting for Partially Occluded Objects**:

   In text-guided diffusion-style inpainting, external text prompts can describe the expected semantics of corrupted objects in damaged images. However, achieving precise spatial posture control using only textual semantics is difficult due to the inherent gap between high-level language and pixel-level pose information. Although existing visual-guided inpainting methods utilize structural information, they often fail with significantly or partially corrupted objects. This failure stems from their one-way visual control mechanisms, which neglect valuable spatial clues present in the uncorrupted parts of the object – clues essential for posture control. To address this, this thesis introduces a novel dual-path visual control module and proposes two customized datasets to evaluate posture control for partially corrupted objects.

## 1.3 Thesis Organization

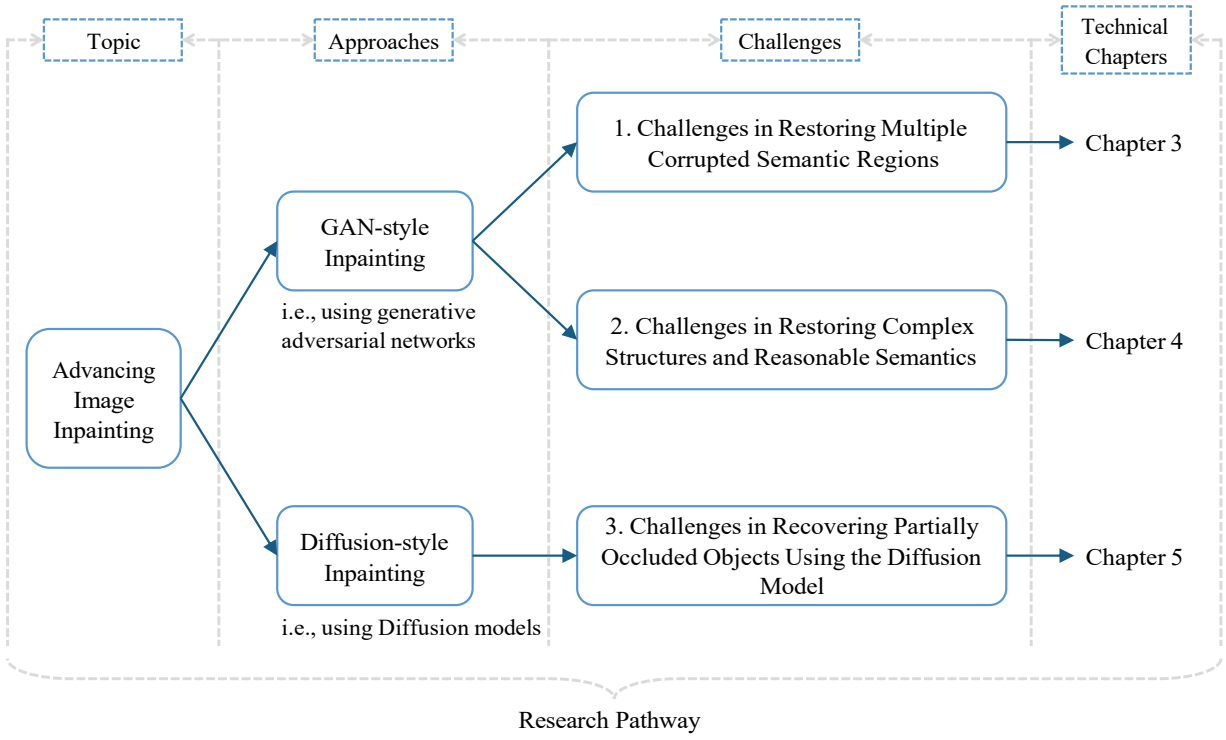



Figure 1.3: The research pathway of this thesis, in which image inpainting approaches are grouped into two representative frameworks: generative adversarial networks and diffusion models, each presenting unique challenges that need to be addressed to advance image inpainting research.

Fig. 1.3 illustrates the research pathway of this thesis, categorizing image inpainting approaches into two types based on the generative model used: GAN-style and Diffusion-style. For each type, distinct technical challenges are identified and addressed to advance the state of the art. This thesis investigates these issues in depth, proposing novel insights and solutions. The structure of the thesis, depicted in Fig. 1.4, is organized as follows:

- **Chapter 2**: This chapter presents a survey of generative image inpainting approaches.
- **Chapter 3**: This chapter presents a dual-task co-optimization framework for guided semantic information and target image texture inpainting, addressing Research Challenge 1. The framework is designed for scenarios with multiple corrupted semantic regions. Specifically, it leverages semantic segmentation maps as guidance and progressively models two intertwined processes: segmentation-guided texture generation and texture-guided segmentation refinement. It iteratively enriches segmentation predictions using the in-

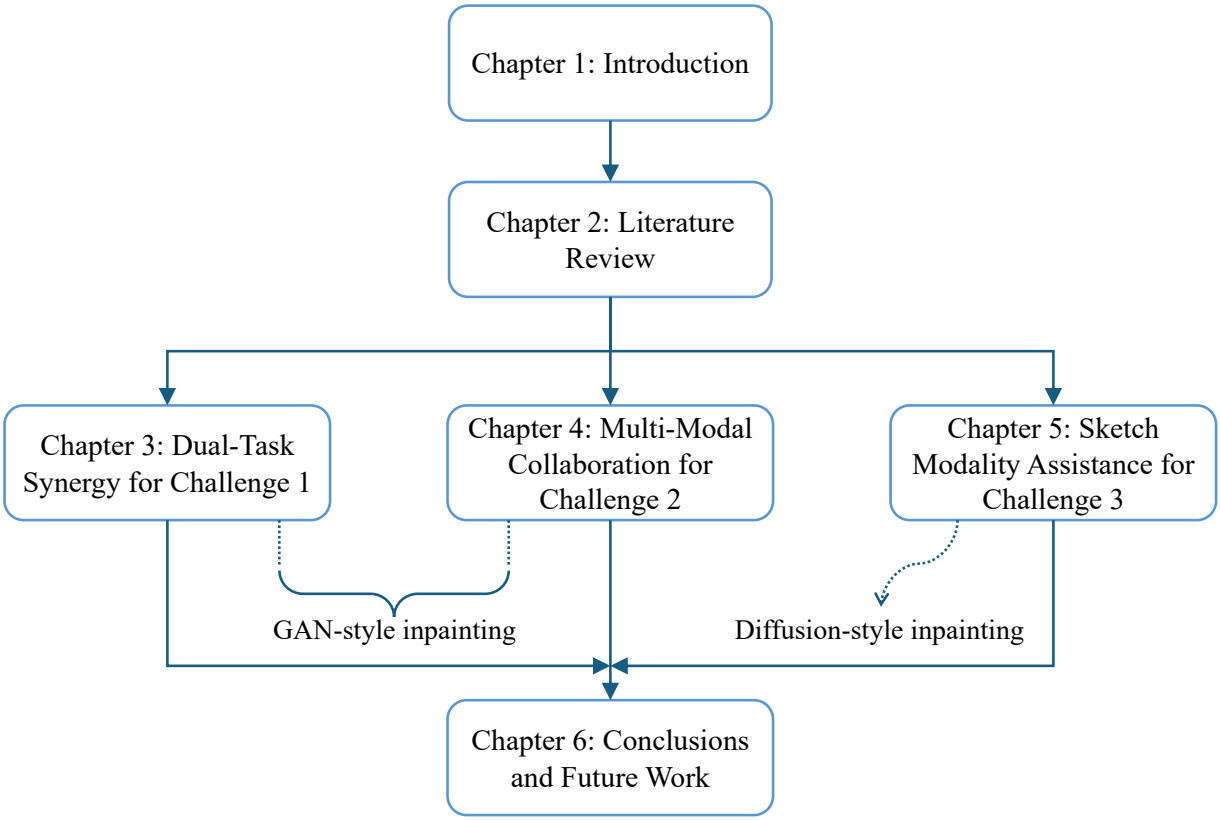



Figure 1.4: The organization of this thesis structure.

painted textures and employs the updated segmentation maps to guide subsequent texture inpainting stages. Comprehensive evaluations demonstrate the framework's effectiveness in leveraging semantic guidance.

- **Chapter 4**: This chapter introduces a method for Research Challenge 2, which targets complex structural and semantic degradation in GAN-based inpainting. Inspired by real-world artistic workflows, the proposed method dynamically integrates structural and semantic cues. It formulates diverse guidance sources as a form of multi-modal collaboration and adopts a progressive inpainting strategy that mimics human drawing processes to guide image texture restoration.

- **Chapter 5**: This chapter proposes a sketch-assisted inpainting method for Research Challenge 3. A dual-path visual control module, utilizing hand-drawn-like sketches, explicitly models the interaction between known object cues and guided sketches before integrating them into corrupted regions for posture control. In addition, two new datasets—CUB-sketch and MSCOCO-sketch—are introduced, providing partial sketches corresponding to severely corrupted object regions to support experimental evaluation.

- **Chapter 6**: This chapter summarizes the thesis contents and contributions, and discusses recommendations for future work.

# Chapter 2

# Literature Review

This chapter presents a comprehensive literature review on image inpainting, focusing on three key areas: semantic learning in GAN-style inpainting, guidance mechanisms in GAN-style inpainting, and diverse guidance prompts in Diffusion-style inpainting. These areas are critical for addressing challenges in achieving semantic coherence, structural alignment, and effective spatial guidance. A detailed summary of these approaches evaluates their strengths, limitations, and research gaps, providing a foundation for this thesis's contributions. Finally, the chapter introduces relevant frameworks used in image inpainting to enhance understanding of the concepts discussed.

## 2.1 Semantic Learning in GAN-style Inpainting

As noted in Section 1.1.1, the primary challenge in GAN-style image inpainting is restoring corrupted regions spanning diverse semantic regions. Since the emergence of Convolutional Neural Networks (CNNs) [79], [80] and Generative Adversarial Networks (GANs) [21], this challenge has attracted considerable research focus. In recent years, two main strategies have emerged to enhance semantic feature learning in image inpainting: **implicit learning of semantic features** and **explicit modeling of semantic guidance**. The former relies on networks to infer semantics indirectly, while the latter incorporates explicit semantic information to ensure coherence, particularly in complex semantic scenes.

### 2.1.1 Implicit Learning of Semantic Features

Implicit learning of semantic features in early GAN-style inpainting employs deep CNNs and GANs to extract latent semantic patterns (semantic cues learned implicitly from data) through large-scale dataset training, facilitating the restoration of corrupted regions. A typical architecture, used in many works, is an encoder–decoder with UNet-shaped convolutional layers [81]. Given an RGB image with missing regions as input, the encoder progressively captures hierarchical representations of textures and semantic patterns using local convolutional kernels. These encoded features are then decoded step by step through upsampling and convolution operations to reconstruct the inpainted image. These approaches excel at capturing multi-level visual details and latent semantic patterns; however, the implicit nature of such features can still lead to semantically inaccurate outputs in complex scenes with multiple object classes, which motivates our development of explicit semantic modeling in inpainting.

For instance, the pioneering work by Pathak et al. [9] proposed the first convolutional neural network-based inpainting model, introducing a context encoder to restore center-located missing regions in images. Building on this, Iizuka et al. [22] enhanced coherence across local and global contexts by introducing dual discriminators—one for global image consistency and another for local texture restoration. Subsequent research proposed various architectural designs to further improve inpainting quality, including progressive generation modules for iterative refinement of predictions [42], recurrent neural networks to sequentially infer boundary details before restoring central regions [26], and convolutional networks with varied receptive field sizes to capture multi-scale contextual information more effectively [30], [43], [82]. Additionally, contextual attention (CA) mechanisms were developed to model long-range dependencies between missing and known image regions. Specifically, CA modules calculate similarity between feature patches from damaged and undamaged regions, enabling the network to "borrow" semantically relevant patches for restoration. This approach was first presented by Yu et al. [10] and later extended in their follow-up work by combining it with gated convolutions [24] to handle arbitrarily shaped holes. Variants of this approach have explored more complex attention schemes [44], [45], [83], such as replacing missing patches with those extracted from high-level feature maps [44] or enforcing contextual similarity through auxiliary supervision [45]. However, these CA-based techniques typically operate on a patch-wise basis and struggle to distinguish semantic class boundaries, often transferring visually similar but semantically inappropriate content. In contrast, this thesis introduces a more robust attention mechanism that models

semantic-level texture associations by focusing on regions belonging to the same semantic class within an image.

Overall, although existing methods of implicit learning can produce visually convincing results, their lack of explicit semantic modeling often leads to content inconsistency in complex scenes containing multiple object categories.

### 2.1.2 Explicit Modeling of Semantic Guidance

The aforementioned implicit learning approaches prioritize general feature extraction through implicit semantic analysis, rendering them simpler but less effective for restoring complex scenes. In contrast, explicit modeling of semantic guidance provides direct semantic information to the inpainting process, enabling effective restoration of coherence across diverse corrupted areas spanning multiple object categories. Within this framework, two major research directions have emerged: (1) **pre-trained semantic constraints**, which leverage high-level representations from pre-trained models to guide inpainting but may encounter instability; and (2) **semantic prior modeling**, which predicts semantic layouts to steer content generation but struggles with effective guidance.

#### *1) Pre-trained Semantic Constraints*

Yang et al. [47] pioneered the use of a pre-trained VGG-19 network [50] to extract high-level features as a supervisory signal during training, ensuring that the inpainting model generates semantically coherent outputs. Known as perceptual loss, this approach has been widely adopted as a flexible loss component integrated into various architectures [33], [46], [84]. Wang et al. [48] extended this concept by employing pre-trained StyleGAN models [51], [52] to search for the closest semantic representation of the corrupted image in the latent space, thereby extracting intermediate features from the StyleGAN's generator to guide the inpainting process. However, reliance on external pre-trained models can compromise the training stability of these methods.

#### *2) Semantic Prior Modeling*

Semantic prior modeling predicts or utilizes semantic layout, such as segmentation maps, to provide direct guidance. Semantic segmentation, a high-level vision task that generates spatial

layouts of object classes [54], [55], enables models to differentiate inter-class variations and relationships. Following the success of semantic segmentation in conditional image generation [85], [86], [87], notably Park et al.'s [86] introduction of the spatially adaptive denormalization module, which integrates one-hot layout maps into the normalization layers of an image generator to enhance fidelity, recent inpainting models [23], [25], [28], [49], [64] have leveraged predicted segmentation priors to guide content generation. For example, Song et al. [23] developed a two-stage network that first predicts a complete segmentation map from the damaged image and then concatenates it to condition RGB image restoration, thereby achieving greater semantic coherence. However, the quality of the predicted segmentation map cannot be guaranteed in the context of image inpainting, as it is non-trivial to generate reliable semantic priors from severely corrupted images. To alleviate this, Liao et al. [25], [28] proposed a joint encoder–decoder framework where the decoder generates the semantic layout and inpainted image in a progressively entangled manner. Ardino et al. [49] introduced a unified one-stage model that simultaneously inpaints textures and generates a corresponding segmentation layout to modulate subsequent texture inpainting. However, coupling semantic prediction with texture generation within a single pipeline poses challenges, as effectively modeling the complementary dependency between these components is non-trivial. In contrast, this thesis proposes a mutually guided dual-task generator framework, where texture-guided segmentation and segmentation-guided texture generation are jointly modeled to enhance each other's performance.

## 2.2 Guidance Mechanisms in GAN-style Inpainting

To effectively reconstruct images with semantically rich corrupted regions and complex structures, many studies have incorporated auxiliary information—such as Canny edges [29], [58], [60], object contours [31], [61], [66], and sketch-like lines [57], [62], [88]—into the GAN-based inpainting pipeline. These auxiliary modalities provide valuable intra-class object structures and complement the semantic prior guidance mechanisms discussed in Section 2.1.2. While semantic priors offer high-level categorical and layout information, auxiliary structures emphasize detailed object edges and geometric coherence. A critical research question is how to effectively integrate these guidance inputs (semantic priors and auxiliary structures) to maximize their impact on the inpainting process. Existing approaches for integrating guided features can be broadly categorized into two paradigms: (1) **direct feature fusion** and (2) **modulated**

**feature fusion**.

### 2.2.1 Direct Feature Fusion

Direct feature fusion integrates auxiliary guidance signals (e.g., edge information or segmentation maps) with corrupted images or intermediate features through simple concatenation, providing a straightforward approach. This method's simplicity ensures broad applicability across inpainting tasks with minimal architectural changes. It has been adopted in numerous works [23], [58], [59], [60], [61], [62], demonstrating its feasibility and versatility. However, simple concatenation treats all channels equally, failing to leverage the complementary strengths of guidance and image features, often resulting in suboptimal outcomes when features are misaligned or at differing abstraction levels.

For instance, Nazeri et al. [58] built upon the two-stage pipeline of SPG-Net [23] by incorporating edge prediction into the first stage. The predicted edge map is concatenated with the corrupted image as input to the second-stage inpainting network, providing cues for more accurate and coherent texture synthesis. Similarly, StructureFlow [59] introduced edge-preserved smooth images [89], fusing them into the restoration process via channel-wise concatenation to enhance recovery of continuous textures and contours. Yang et al. [60] advanced this approach by designing structure embedding layers that learn and encode gradient-based features, which are concatenated back into the inpainting branch to preserve structural information. Despite their simplicity, these direct fusion methods often yield suboptimal performance, as shown in subsequent studies [28], [63], [90], highlighting the need for more adaptive fusion strategies.

### 2.2.2 Modulated Feature Fusion

To overcome the limitations of direct concatenation, modulated feature fusion offers a refined and increasingly adopted approach, prioritizing adaptive integration over simplicity. This strategy employs modulation functions, such as conditional feature normalization or feature gating mechanisms, to integrate semantic or structural priors into the inpainting network. Conditional feature normalization, originally proposed by Park et al. [86], guides image inpainting by learning affine parameters (scaling and shifting) conditioned on guidance inputs. These parameters modulate intermediate features through element-wise multiplication and addition, enabling the model to respond spatially to guidance cues. Feature gating mechanisms, using

activation (e.g., GELU or Sigmoid), selectively activate guided features across the full-pixel space to modulate the inpainting process. However, reliance on local convolutional operations for learned parameters limits global consistency in utilizing guidance. Similarly, feature gating mechanisms neglect channel-level dependencies, reducing performance.

Numerous works have successfully applied this modulation-based paradigm to image inpainting. For instance, Yu et al. [33] introduced an auxiliary denormalization module that integrates Canny edge and segmentation features into the inpainting network. By learning modulation parameters from these inputs, the network could achieve improved alignment and realism in generated content. Similarly, bidirectional cross-domain feature denormalization [91] enables mutual interactions between texture inpainting and layout generation, enhancing coherent joint modeling of textures and semantics. However, modulation methods that rely solely on local convolutions struggle to ensure global consistency, as their limited receptive fields are insufficient to capture long-range dependencies and scene-level relationships. To address this limitation, this thesis introduces a linear multi-head cross-attention mechanism that explicitly models the correlations between auxiliary features and inpainted features, thereby enabling more comprehensive association and guidance.

Recent advancements have incorporated gating mechanisms as dynamic switches to regulate information flow between guidance and restoration branches. For example, Guo et al. [29] proposed edge-gated and texture-gated fusion modules using sigmoid activation, which adaptively regulate the integration of structural and textural features. These modules highlight relevant features while suppressing irrelevant ones, enhancing the model's ability to resolve ambiguities in occluded regions. Feature gating mechanisms, originating from natural language processing [92], [93], [94], [95], [96], [97], play a critical role in controlling information flow and ensuring contextual relevance. In Transformer-based architectures [97], for instance, GELU activations [98] selectively propagate useful information in feedforward layers. However, current gating applications in image inpainting primarily focus on spatial features, neglecting the importance of channel-wise dependencies. This limitation can lead to inconsistencies in coherence, particularly for complex object shapes or overlapping semantic boundaries. Thus, significant potential remains for developing advanced fusion mechanisms that integrate spatial and channel-wise modulations in a context-aware manner, which can be addressed in this thesis.

## 2.3 Diverse Guidance Prompts in Diffusion-style Inpainting

While GAN-style inpainting methods have advanced significantly in incorporating coherent content and structures, Diffusion models has introduced new possibilities and challenges for image inpainting. A key limitation still under-addressed is the accurate inference of object posture and spatial layout in corrupted regions, particularly when the missing content includes partially or significantly occluded objects. This section reviews recent works on Diffusion-style inpainting, focusing on two primary branches: (1) **text-guided image inpainting** and (2) **visual-guided image inpainting**.

### 2.3.1 Text-Guided Image Inpainting

As discussed in earlier sections, traditional CNN- or GAN-based models [22], [24], [32], [45] rely heavily on local contextual cues from unmasked regions to generate plausible content. However, these models struggle when recovering objects with significant corruptions, particularly when the corrupted regions cover the majority of the object's semantically meaningful area. To address this, text-guided inpainting methods [71], [72], [73], [99], [100], [101], [102] use diverse natural language prompts to provide global semantic context, enabling the generation of coherent content for large missing regions. These methods leverage external knowledge from prompts, proving effective when local contextual cues are insufficient. However, text-guided methods lack fine-grained control over object shapes and postures, which require spatially grounded and pixel-aligned information. As a result, these methods often produce semantically accurate but visually misaligned outputs, particularly for partially occluded objects.

Early approaches [99], [100], [101], [102] focused on bridging the modality gap between text and image features. They learned joint embeddings or employed feature fusion strategies to integrate text prompts with corrupted image inputs. Despite their effectiveness, generating results that are both visually realistic and aligned with the text remains challenging, especially when the prompt describes fine-grained spatial layouts or object parts. With the advent of Diffusion models trained on large-scale text–image pairs [36], [103], the ability to generate high-quality images conditioned on text has improved significantly. State-of-the-art text-to-image (T2I) models like Stable Diffusion [13], Imagen [104], Glide [105], and DALLE-2 [106] demonstrate remarkable capability in synthesizing semantically coherent content. Numerous recent inpainting

models [13], [71], [72], [74], [75], [76] leverage these models with inpainting-specific adaptations through fine-tuning. For example, SD-Inpainting [13] feeds a mask, the damaged image, and noise latent into a modified diffusion process, integrating a CLIP-encoded text prompt [70] through cross-attention layers. Similarly, BrushNet [74] integrates unmasked contextual features and masks into the denoising U-Net to enhance prompt consistency. Imagen Editor [72] employs an object-specific masking policy to fine-tune the base Imagen model [104] for image inpainting, achieving high alignment and fidelity. HD-Painter [75] enhances text alignment by adaptively suppressing irrelevant known regions that conflict with the text prompt. Likewise, object-mask-aware models like SmartBrush [38] and PowerPaint [41] use object-shaped masks to enable inpainting of fully corrupted objects with guided prompts.

Despite their ability to provide global context, text-guided models often struggle with ambiguous semantics and lack fine-grained, pixel-level control, as text prompts alone cannot precisely infer shapes, poses, or boundaries.

### 2.3.2 Visual-Guided Image Inpainting

Visual-guided image inpainting employs external visual prompts, such as sketches or edge maps, to ensure precise spatial and structural control, making it ideal for reconstructing corrupted objects with accurate boundaries. Unlike text-guided methods, these approaches prioritize structural accuracy over textual guidance. However, existing methods struggle to seamlessly integrate known and unknown regions and underperform when corrupted objects remain important clues, limiting their effectiveness in partially occluded scenarios.

Recently, visual prompts have emerged as powerful tools for providing intuitive spatial guidance in computer vision tasks, particularly within Diffusion-style frameworks. Models such as SAM-CLIP [107] and Open-Vocabulary SAM [108] combine SAM-generated bounding boxes [109] with CLIP-based text prompts [70] to perform open-set object segmentation. In the broader context of text-to-image (T2I) synthesis, methods like ControlNet [39], T2I-Adapter [110], and Text2Human [111] use structured inputs (e.g., edge maps, semantic maps, depth, or pose) to guide image generation with strong consistency. Unlike general image generation, inpainting requires seamless fusion of generated content with unmasked regions, posing a significant challenge. Recent inpainting models, such as MaGIC [78] and the inpainting version of ControlNet [39], adapt T2I strategies for localized object inpainting. For example, MaGIC [78]

introduces gradient-based sketch supervision using backpropagation to steer the denoising process toward the desired structure, but its training remains unstable and prone to converging to suboptimal solutions. Similarly, ControlNet-based inpainting [39] injects structural features, like sketches, into the U-Net backbone, but lacks a feedback loop to adapt guidance based on corrupted image content. These methods perform well when the main object is fully occluded, allowing for unconstrained content generation. However, they tend to underperform in partially corrupted scenarios due to insufficient modeling of the relationships between known and unknown parts of the object. This highlights the need for robust visual control mechanisms capable of encoding shape, boundary, and structural information from partially visible (uncorrupted) object regions.

Visual guidance, such as sketches, remains a promising solution, yet more effective integration strategies are necessary to achieve controllability, stability, and coherence in partial-object inpainting. To address this issue, this thesis proposes dual-path sketch-aware modules, which fuse sketch-derived features with the contextual information from uncorrupted object regions. This design enables explicit structural control while maintaining consistency between restored and uncorrupted object areas.

## 2.4 Preliminaries of Generative Frameworks in Image Inpainting

This part provides a concise overview of image inpainting technologies, focusing on three foundational generative frameworks: Generative Adversarial Networks (GANs), Transformer-based generative architectures, and Denoising Diffusion Probabilistic Models (DDPMs). These frameworks underpin the concepts discussed in this thesis, enhancing understanding of their applications in image inpainting.

In Section 2.4.1, we introduce the basic architecture of GANs with an encoder–decoder generator, which can be implemented using either CNN or Transformer-based designs. The Transformer-based architecture is described in detail in Section 2.4.2. Section 2.4.3 then presents DDPMs.

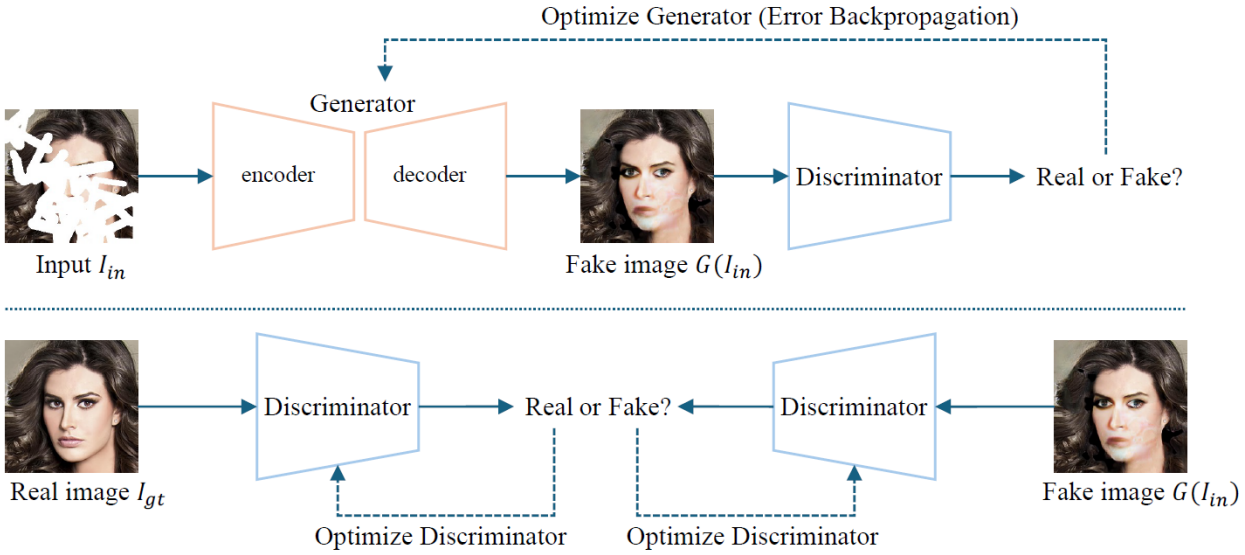



Figure 2.1: Illustration of Generative Adversarial Networks [21] in image inpainting.

### 2.4.1 Generative Adversarial Networks

As shown in Fig. 2.1, a classical Generative Adversarial Network (GAN) architecture consists of two primary components: a generator (G) and a discriminator (D). The generator (G) employs convolutional neural networks (CNNs) or Transformer-based blocks [92], while the discriminator (D) typically consists of multiple convolutional layers [10], [30]. Detailed architectures for G and D are omitted for brevity, and the Transformer-based generator variant is described separately in Section 2.4.2. The generator (G) aims to reconstruct corrupted input images, ensuring that the generated images align with the ground-truth image distributions. The discriminator estimates the likelihood of an input image belonging to the real dataset. During training, the generator and the discriminator compete in a min-max adversarial game, which can be formulated as:

$$\min_G \max_D \mathcal{L}(D,G) = \mathbb{E}_{I_{gt}\sim p(I_{gt})}[\log D(I_{gt})] + \mathbb{E}_{I_{in}\sim p(I_{in})}[\log(1 - D(G(I_{in})))], \tag{2.1}$$

where the generator G seeks to minimize the loss function, while the discriminator D aims to maximize the loss function. $I_{gt}$ denotes the real image from the ground truth dataset, and $I_{in}$ represents a corrupted input image. In the implementation, the objective function to optimize the generator is defined as:

$$\mathcal{L}_{Gen}(G) = \mathbb{E}_{I_{in}\sim p(I_{in})}[\log(1 - D(G(I_{in})))]. \tag{2.2}$$

Meanwhile, to mitigate gradient vanishing, the objective function of the discriminator is formulated as:

$$\mathcal{L}_{Dis}(D) = \frac{1}{2}\mathbb{E}_{I_{in}\sim p(I_{in})}[\log D(G(I_{in}))] + \frac{1}{2}\mathbb{E}_{I_{gt}\sim p(I_{gt})}[\log(1 - D(I_{gt}))]. \tag{2.3}$$

### 2.4.2 Transformer-based Architectures

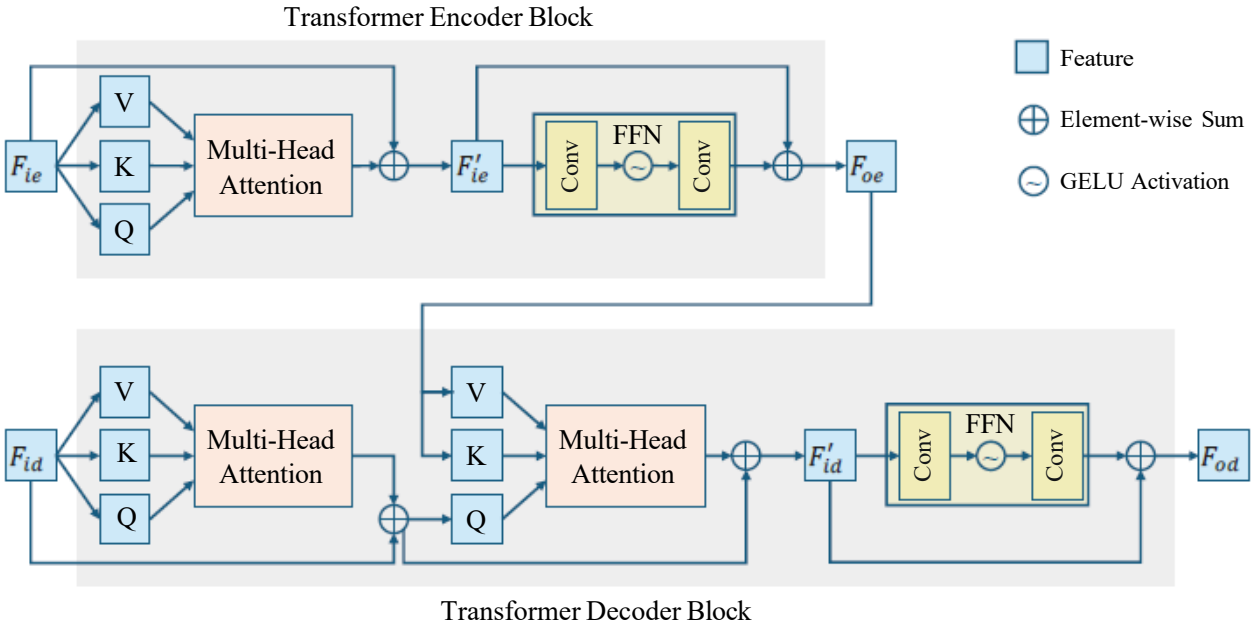



Figure 2.2: The Transformer-model architecture.

The Transformer-based generative framework [92] has proven superior performance compared to CNN-based methods [10], [34], [84] in low-level image inpainting tasks. Its effectiveness stems from capturing long-range dependencies among features, enabling robust contextual learning across entire image regions for reconstructing corrupted areas. As illustrated in Fig. 2.2, the Transformer-model architecture comprises stacked encoder and decoder blocks.

Each encoder block consists of two sub-layers: a multi-head self-attention mechanism and a convolutional feed-forward network (FFN), each surrounded by a residual connection [112]. Given a flattening feature input $F_{ie}$, the encoder block output is formulated as:

$$\begin{aligned} F'_{ie} &= MultiHead(Q, K, V) + F_{ie}, \\ F_{oe} &= FFN(F'_{ie}) + F'_{ie}, \end{aligned} \tag{2.4}$$

where the multi-head self-attention is defined as:

$$\begin{aligned} MultiHead(Q,K,V) &= Concat(head_1,...,head_h), \\ where\ head_i &= Attention(Q_i,K_i,V_i) \\ &= softmax(\frac{Q_iK_i^T}{\sqrt{d_k}})V_i. \end{aligned} \tag{2.5}$$

Typically,( $h = 8$) parallel attention layers or heads are used, enabling the model to focus on diverse representational subspaces across different positions [92].

The transformer decoder block extends the encoder's structure by incorporating an additional sub-layer for cross multi-head attention over the encoder stack's output, alongside the self-attention and FFN sub-layers. Each sub-layer employs a residual connection. The decoder's output formulation mirrors the encoder's and is omitted for brevity.

Stacked Transformer blocks in generative models have achieved state-of-the-art inpainting results. For example, Wan et al. [113] utilized a vanilla Transformer for low-resolution inpainting, while Yu et al. [114] proposed bidirectional autoregressive techniques. Li et al. [32] introduced a dynamic mask-aware mechanism to enhance self-attention in transformers. Further advancements include inter-patch attention [115] and axial self-attention [88], [116]. However, these methods exhibit quadratic computational complexity with respect to the number of image patches $n$ and spatial resolution $h \times w$, limiting their scalability for multi-modal tasks, such as high-resolution structure generation, semantic segmentation prediction, and texture inpainting.

### 2.4.3 Denoising Diffusion Probabilistic Models

Denoising diffusion probabilistic models (DDPMs) are designed to learn the reversal of a parameterized Markovian process that gradually adds noise to images. Starting with samples drawn from isotropic Gaussian noise, DDPMs iteratively refine these samples through denoising to match a target training distribution. The forward and reverse diffusion processes are illustrated in Fig. 2.3. Recent studies [36], [117], [118] have demonstrated that DDPMs are capable of generating high-quality images. This section outlines the core principles of DDPMs, following the formulations and notations presented in [118], [119].

Given an initial data distribution $x_0 \sim q(x_0)$, a forward noising process generates a sequence of

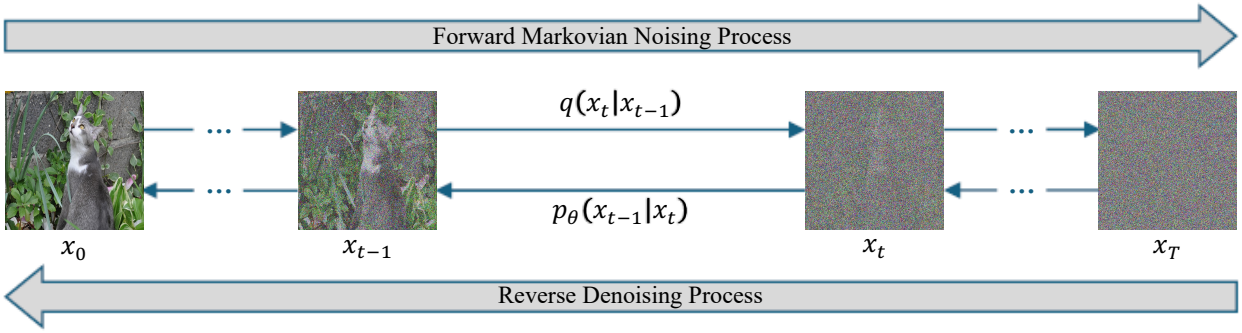



Figure 2.3: The denoising diffusion model, comprising forward and reverse processes. The forward process gradually adds Gaussian noise $q(x_t \mid x_{t-1})$ to a clean image $x_0$, producing increasingly noisy versions until $x_T$ approximates isotropic Gaussian noise. The reverse process then learns to denoise step-by-step, transforming noise $x_T$ back into $x_0$ via a learned posterior $p_\theta(x_{t-1} \mid x_t)$.

latent variables $x_1, ..., x_T$ by adding Gaussian noise with a time-dependent variance $\beta_t \in (0, 1)$:

$$\begin{aligned} q(x_1, \ldots, x_T|x_0) &= \prod_{t=1}^{T} q(x_t|x_{t-1}) \\ q(x_t|x_{t-1}) &= \mathcal{N}(\sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}) \end{aligned} \tag{2.6}$$

As the number of time steps $T$ becomes sufficiently large, the final latent $x_T$ approximates an isotropic Gaussian distribution.

The forward noising process allows direct sampling of any intermediate latent $x_t$ from $x_0$, bypassing preceding steps:

$$\begin{aligned} q(x_t|x_0) &= \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}) \\ x_t &= \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \end{aligned} \tag{2.7}$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=0}^{t} \alpha_s$.

To generate a new sample from the distribution $q(x_0)$, the Markovian process is inverted. Starting from a Gaussian noise sample, $x_T \sim \mathcal{N}(0, \mathbf{I})$, a reverse sequence is constructed by sampling from the posterior distributions $q(x_{t-1}|x_t)$, which were shown to also be Gaussian [120], [121]. Since $q(x_{t-1}|x_t)$ depends on the unknown data distribution $q(x_0)$, a deep neural network $p_\theta$ is trained to predict the mean and covariance of $x_{t-1}$ given $x_t$ as input. Then $x_{t-1}$ may be sampled from the normal distribution defined by these parameters,

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \tag{2.8}$$

Instead of directly predicting $\mu_\theta(x_t, t)$, Ho et al. [36] estimate the noise $\epsilon_\theta(x_t, t)$ added to $x_0$ to produce $x_t$, according to Equation 2.7. Then $\mu_\theta(x_t, t)$ may be derived using Bayes' theorem:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha}_t}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) \tag{2.9}$$

While Ho et al. [36] used a constant $\Sigma_\theta(x_t, t)$, subsequent work [118] has shown that learning it via a neural network, interpolating between the upper and lower bounds, improves performance.

## 2.5 Summary

This chapter outlines the evolution of image inpainting techniques, including GAN-based semantic learning, various guidance mechanisms, and emerging diffusion-based methods that leverage diverse conditioning prompts. It highlights ongoing challenges such as achieving semantic and structural coherence in GAN-based inpainting of complex scenes, and attaining precise spatial control in diffusion models. The chapter then introduces the foundational generative frameworks of GANs and DDPMs, which offer robust mechanisms for image reconstruction through adversarial training and noise reversal, respectively. Detailed solutions to these challenges will be proposed in the subsequent chapters of this thesis.

# Chapter 3

# Dual-task Co-optimization for Semantically Guided GAN-style Inpainting

## 3.1 Motivation

To restore semantically multiple corrupted regions, recent works tend to complete image inpainting using semantic priors (i.e., semantic segmentation) as guidance. Given the semantic prior in damaged areas, the rationale is that it provides semantic layout information for inferring the missing image textures. The restored texture information in damaged areas can best align with the semantic structure as indicated by the priors. However, the challenge to the case of image inpainting is that it is impossible to acquire accurate semantic structure guidance before the image texture information is completed. To tackle this chicken-or-egg problem, existing works follow two approaches: one-off guidance [23] and progressive guidance [25], [28], [49]. As shown in Fig. 3.1 (a), the one-off guidance method predicts the corrupted segmentation layout (e.g., semantic map) in the first stage and uses it to guide image inpainting in the second stage. However, such a prediction process cannot guarantee the quality of the semantic map for image inpainting since predicting reliable semantic layouts from corrupted images with large missing holes is challenging. Moreover, there is no justification or feedback process in the second stage to rectify any possible errors caused by the lower-quality semantic map, so the overall semantic guidance is ineffective. For example, as shown in the second row of Fig. 3.2 (d). we can see

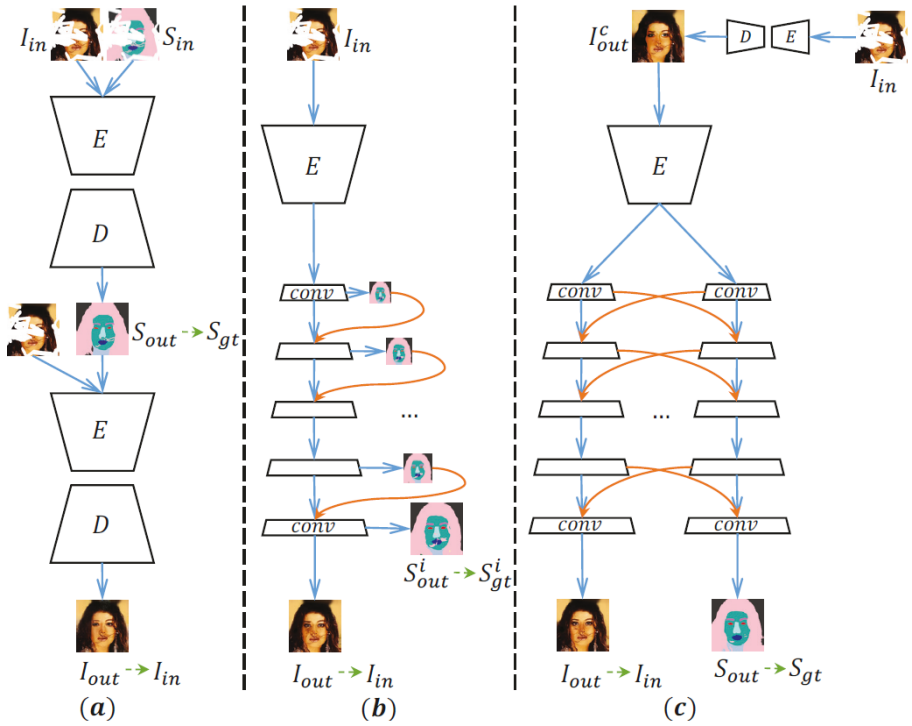



Figure 3.1: Diagram of three types of image inpainting with semantic guidance. (a) One-off guidance method [23], (b) Progressive guidance method [28], (c) Ours. The green dotted arrow represents the training constraint, the blue arrow represents the feature feedforward, and the orange arrow represents the guidance function. $E$ and $D$ represent Encoder and Decoder, respectively.

the asymmetrical eyes on the predicted segmentation map marked by the red box, which also affects the final image inpainting shown in the first row of Fig. 3.2.

Instead of the two-stage framework, the progressive guidance approach [25], [28] adopts a unified framework to exploit texture representations of the given damaged image, as shown in Fig. 3.1 (b). Differently, the guided segmentation map is predicted based on the texture representation at each scale of the decoder. Then, the semantic map is fed to update the next-level texture representation. The quality of the semantic map depends on the quality of the texture generated in each scale due to parameter sharing in the single decoder. Such dependency affects the effectiveness of the semantic guidance provided to texture information generation. (see the last two rows marked by the red box in Fig. 3.2 as examples.)

To tackle the problems above, this thesis proposes a framework of dual branches with a shared encoder. Specifically, each branch has a separate decoder for modeling semantic segmentation and image (texture) inpainting individually, as shown in Fig. 3.1 (c). Moreover, to facilitate mu-

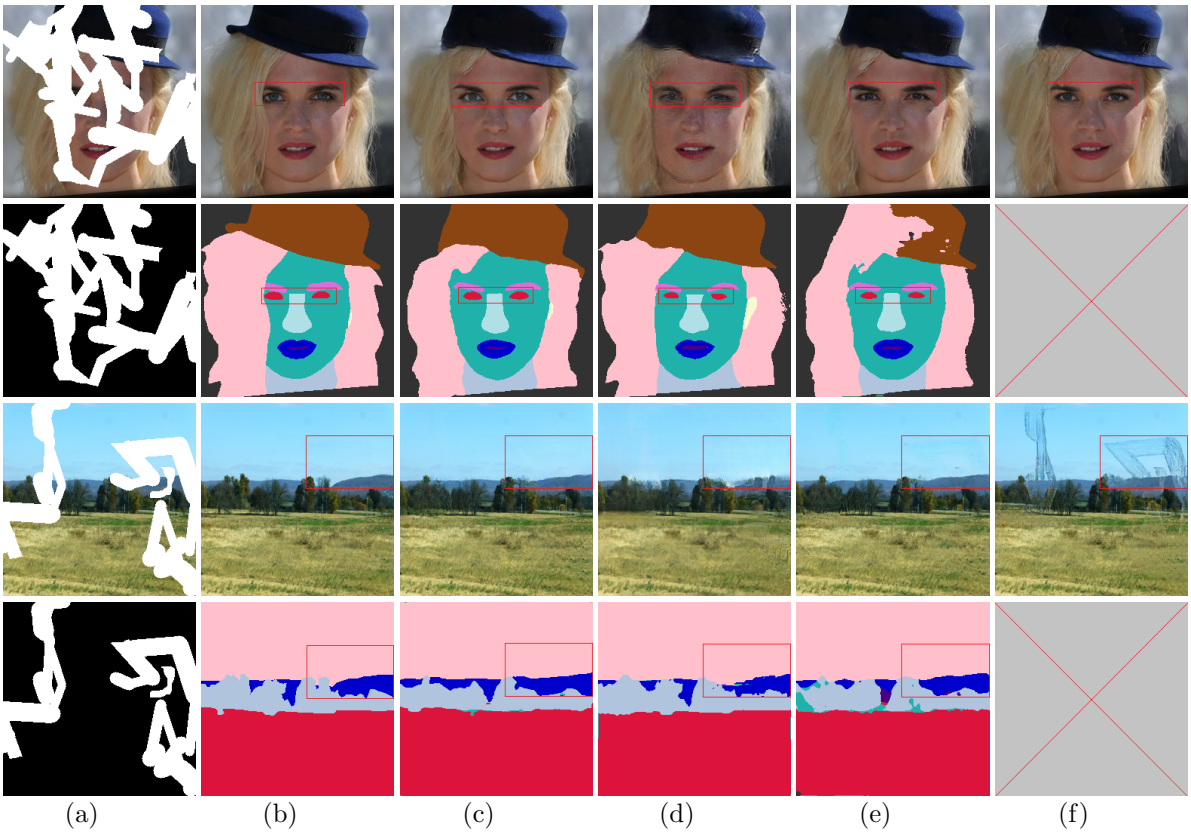
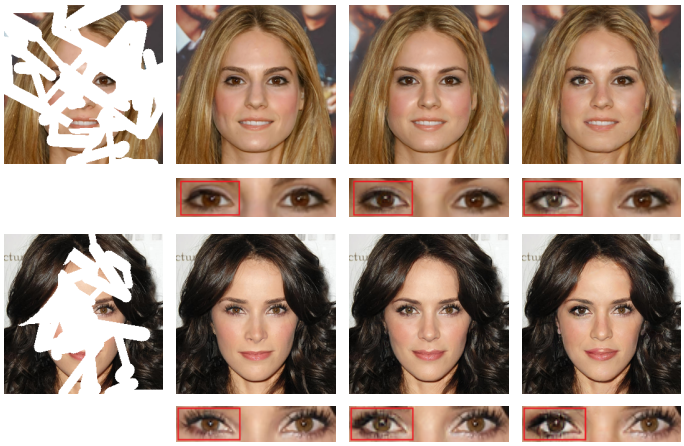



Figure 3.2: Inpainted results of four types of inpainting methods. (a) The damaged image and mask, (b) Ground-truth of the RGB image and segmentation map, (c) Results of ours, (d) Results of one-off guidance method [23], (e) Results of progressive guidance method [28], (f) Results of LGNet [30], a SOTA method without semantic guidance. × means no segmentation guidance for LGNet.

tual and effective dependency of two task branches, this thesis proposes a Cross-domain Feature DeNormalization (CFDN) module that is used at each level of both decoding processes bidirectionally (denoted by orange arrows in Fig. 3.1 (c)), which forms our mutual decoders consisting of segmentation-guided texture (ST) generation and texture-guided segmentation (TS) generation inside. In mutual decoders, texture features are inpainted using semantic segmentation as guidance. At the same time, the segmentation part also receives feedback from the texture inpainting branch, which refines the semantic information guidance for texture generation at subsequent levels. Furthermore, an Adaptive Attention Fusion (AAF) module is introduced to enhance feature consistencies from semantic-affinity and global-context perspectives, which contains three blocks, i.e., the Semantic-affinity cross-Attention (SA), the Global-context self-Attention (GA), and the Gated Feature Fusion (GFF). The SA block considers the texture



(a) Input (b) Gt (c) AAF (d) Wo_AAF

Figure 3.3: The inpainted eye color of methods with or without the proposed AFF module. (a) The damaged input, (b) Ground-truth, (c) Results of the consistent eye color of our method using the AAF module, (d) Results of the inconsistent eye color of our method without the AAF module. [Best view with zoom-in.]

affinity across the inpainted pixels and undamaged pixels on a same semantic level of an image, where the semantic categories are learned from the feature outputs of the TS generation stage. The GA block considers the texture affinity of pixels from the entire image region level as a coordinated supplement to the SA block. Subsequently, the attention feature output from the SA and GA blocks is adaptively fused through the GFF block. As shown in Fig. 3.3 (c), the introduced AAF module achieves more consistent color filling of the left eye with the undamaged right eye than the approach without the AAF module.

The shared encoder and mutual decoders together form the proposed mutual dual-task generator. It is based on a coarse inpainting network to get the preliminary feature in missing areas, which serves the mutual dual-task generator.

The contributions of the proposed dual-task co-optimization framework can be summarized as follows:

- A framework of mutual decoders with bidirectional Cross-domain Feature DeNormalization (CFDN) modules is proposed, which progressively models segmentation-guided texture generation and texture-guided segmentation generation. It demonstrates better effectiveness in terms of semantic guidance for image inpainting.
- An Adaptive Attention Fusion (AAF) module tailed after mutual decoders is developed to

further improve performance by modeling the semantic-affinity and global-context texture dependencies.

- Qualitative and quantitative experiments on multiple datasets verify that the proposed method achieves superiority over the existing state-of-the-art methods.

## 3.2 Dual-task Co-optimization Using Mutual Generator

The proposed end-to-end mutual generator architecture is shown in Fig. 3.4. The full architecture contains three parts and follows two phases. In the coarse inpainting network phase, initial pixel contents for missing holes are generated using an adversarial loss. In the mutual dual-task generator phase, the shared encoder extracts unified features. They are individually decoded into two domain features via dual decoders, i.e., texture and the corresponding semantic segmentation features. Two domain features are collaborated progressively with CFDN modules to assist each other. The adaptive attention fusion module is stacked at the end of mutual decoders for further refining the inpainting quality. The details of each part are described as follows.

### 3.2.1 Coarse Inpainting Network

Given the damaged image $I_{in}$ and the corresponding binary mask $M$ (where 1 indicates the regions where the pixel information is damaged and 0 indicates the undamaged regions), a coarse inpainting network is adopted to produce the coarse inpainted image $I^c_{out}$. As shown in Fig. 3.4 (a), the coarse inpainting network consists of a generator with eight down-sampling convolution layers in the encoder and eight up-sampling convolution layers in the decoder. The adversarial module with a spectral-normalized discriminator [24] follows it to enforce the learning performance. Every down-sampling layer of the encoder uses a $3 \times 3$ convolution with a stride of 2, followed by BatchNorm and LeakyReLU. Every up-sampling layer of the decoder adopts the bilinear up-sampling followed by a 3×3 convolution and BatchNorm-Relu. Transposed convolution is not used as the up-sampling layer, unlike [30], since it often leads to checkerboard artifacts, as reported in [122]. Meanwhile, the skip connection transfers the encoder feature to the corresponding decoder layer for preserving low-level information. As for the discriminator, it consists of four convolutional layers with LeakyReLU activation, followed by a convolutional layer with sigmoid activation.

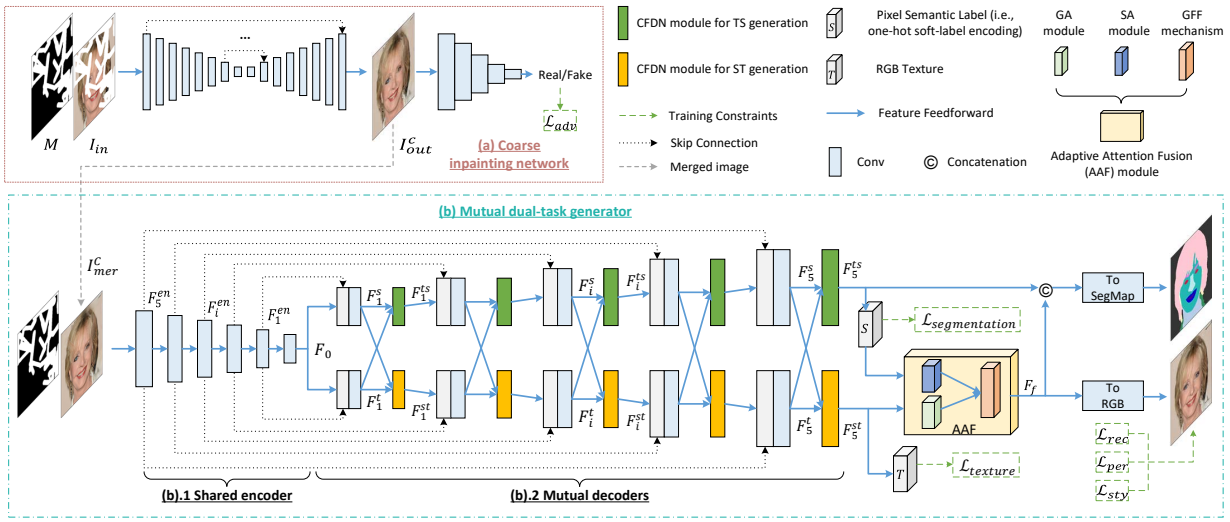



Figure 3.4: The full architecture of the proposed method. It consists of three parts: (a) a coarse network for the preliminary inpainting with adversarial loss, (b).1 Shared encoder for feature extractions, (b).2 Mutual decoders for the texture-guided segmentation generation (TS decoder branch, top green) and segmentation-guided texture generation (ST decoder branch, bottom yellow). The Adaptive Attention Fusion (AAF) module is stacked at the end of mutual decoders to enforce semantic-affinity and global-context texture dependencies for refining the inpainted results.

L1 loss along with adversarial loss is applied on training coarse network, i.e., $\mathcal{L}_c = \mathcal{L}^c_{rec} + \lambda_{adv}\mathcal{L}_{adv}$, where adversarial loss weight $\lambda_{adv}$ is empirically set to 0.1, following prior image inpainting studies [29], [30], [35]. The L1 reconstruction loss and the adversarial loss are formulated as:

$$\mathcal{L}^c_{rec} = \|I^c_{out} - I_{gt}\|_1, \tag{3.1}$$

$$\mathcal{L}_{adv} = \min_G \max_D \Big[\mathbb{E}_{I_{gt}\sim p_{I_{gt}}(I_{gt})}[\log D(I_{gt})] + \mathbb{E}_{I^c_{out}\sim p_{I^c_{out}}(I^c_{out})}[\log(1 - D(I^c_{out}))]\Big], \tag{3.2}$$

where $I_{gt}$ is the ground-truth RGB image, the coarse output of encoder-decoder $G$ is expressed as $I^c_{out} = G(I_{in}, M)$, the damaged image is defined as $I_{in} = I_{gt} \otimes (1 - M) \oplus M$. $\otimes$ stands for element-wise multiplication and $\oplus$ stands for element-wise addition. The parameters of the generator G and discriminator D are updated alternately in a 1:1 ratio.

The purpose of applying a coarse network is to inpaint the initial texture information for the missing pixels, which can improve the quality of the semantic information produced later for guiding the final image inpainting.

### 3.2.2 Mutual Dual-task Generator

***1) Shared Encoder***

After obtaining the coarse inpainted result $I^c_{out}$, we get the merged image $I^c_{mer} = I_{gt} \otimes (1 - M) + I^c_{out} \otimes M$ as the input of the shared encoder. It consists of five convolution layers to extract hierarchical features. These features are also propagated to the corresponding layers of mutual decoders via skips, to strengthen the forward utilization of features from the encoder to decoders and the backward optimization of the gradient from dual decoders to the shared encoder.

***2) Mutual Decoders***

We first describe the processes of the two decoder branches in modeling two separate tasks. Then, introduce the interactive CFDN module to facilitate mutual assistance of both decoders.

The feature $F_0$ extracted in the last layer of the shared encoder goes through two separate decoders (i.e., semantic segmentation generation decoder and image texture generation decoder). They progressively generate the semantic segmentation and texture feature of size $C \times 128 \times 128$. The two decoder branches follow the same architecture with five upsample convolution layers, tailed by a task-specific output layer to produce output for supervised training. It enables the segmentation generation decoder to focus on modeling inter-class semantic layouts according to the input feature $F_0$, and the texture generation decoder to focus on modeling pixel-wise contents. We define $F^s_i$ as the output feature of the $i$-th upsample convolution layer in the segmentation generation decoder, and $F^t_i$ as the one in the texture generation decoder. The modeling processes of two separate decoders are below (where $\Phi_{Con}(*,*)$ stands for the convolution operation after two feature concatenation, and $\Phi_{UP}(*)$ stands for the operation of bilinear upsampling, and $F^{en}_i$ stands for the $i$-th encoding feature):

$$F^s_i = \begin{cases} \Phi_{Con}(\Phi_{UP}(F_0), F^{en}_i), & i = 1, \\ \Phi_{Con}(\Phi_{UP}(F^s_i), F^{en}_i), & 1 < i \leq 5, \end{cases} \tag{3.3}$$

$$F^t_i = \begin{cases} \Phi_{Con}(\Phi_{UP}(F_0), F^{en}_i), & i = 1, \\ \Phi_{Con}(\Phi_{UP}(F^t_i), F^{en}_i), & 1 < i \leq 5. \end{cases} \tag{3.4}$$

As the segmentation feature $F^s_i$ and texture feature $F^t_i$ focus on learning different tasks, concatenating the two features for mutual assistance will disturb the modeling processes of the cor-

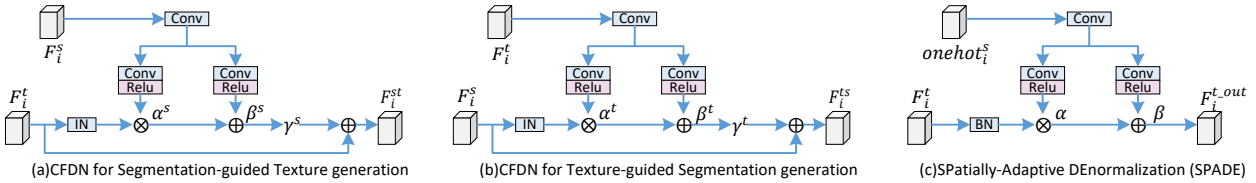



Figure 3.5: Different denormalization modules. (a) Our CFDN module learns the affine transformation parameters $\alpha^s$ and $\beta^s$ from the segmentation feature $F_i^s$ to inject into the process of texture generation (b) Our CFDN module learns the parameters $\alpha^t$ and $\beta^t$ from the texture feature $F_i^t$ to inject into the process of segmentation generation (c) SPADE module [86] learns the parameters $\alpha$ and $\beta$ from one-hot semantic segmentation map $onehot_i^s$ to inject into the image generation with semantic guidance. $\gamma^s$ and $\gamma^t$ are two learnable parameters. IN and BN denote instance and batch normalizations, respectively.

responding decoders. Inspired by SPatially-Adaptive DEnormalization (SPADE) module [86] that learns two spatially-adaptive affine parameters $\alpha$ and $\beta$ from semantic layout to modulate the activations in normalization layers of image texture generation (shown in Fig. 3.5 (c)), we propose a Cross-domain Feature DeNormalization module $\Phi_{CFDN}$ to utilize semantic information to assist the texture inpainting process. As shown in Fig. 3.5 (a), instead of directly using one-hot semantic layout map [1] as inputs, in our module, we learn scale and shift affine parameters $\alpha^s$ and $\beta^s$ from learned semantic features $F_i^s$ which are modeled via a segmentation generation decoder. The parameters $\alpha^s$ and $\beta^s$ stand for learned semantic information with the same spatial size as the texture feature $F_i^t$ and are transformed into $F_i^t$ to produce the updated texture feature $F_i^{st}$ with the segmentation guidance.

Moreover, to facilitate the segmentation generation decoder with more intra-class pixel detail, which in turn will provide better semantic guidance for texture generation in the next scale, we reuse $\Phi_{CFDN}$ to learn the other two parameters $\alpha^t$ and $\beta^t$ from current texture features $F_i^t$. Parameters $\alpha^t$ and $\beta^t$ represent spatial-wise pixel information. They are used to model the updated segmentation feature $F_i^{ts}$ with the texture guidance, as shown in Fig. 3.5 (b). Alternately using CFDN modules on each level of the two decoders promotes mutual dependencies.

[1] It usually requires more sophisticated networks to predict reliable semantic layout maps from segmentation features, as reported in [54]. That is not the core part of our paper.

Our mutual decoders progressively model the features in the following way:

$$\begin{aligned} F_i^{st} &= \Phi_{CFDN}(F_i^s, F_i^t) \\ &= [\Psi_\alpha(F_i^s) \otimes \Phi_{IN}(F_i^t) \oplus \Psi_\beta(F_i^s)] \cdot \gamma^s \oplus F_i^t \\ &= [\alpha^s \otimes \Phi_{IN}(F_i^t) \oplus \beta^s] \cdot \gamma^s \oplus F_i^t, \quad 1 \le i \le 5, \end{aligned} \tag{3.5}$$

$$\begin{aligned} F_i^{ts} &= \Phi_{CFDN}(F_i^t, F_i^s) \\ &= [\Psi_\alpha(F_i^t) \otimes \Phi_{IN}(F_i^s) \oplus \Psi_\beta(F_i^t)] \cdot \gamma^t \oplus F_i^s \\ &= [\alpha^t \otimes \Phi_{IN}(F_i^s) \oplus \beta^t] \cdot \gamma^t \oplus F_i^s, \quad 1 \le i \le 5, \end{aligned} \tag{3.6}$$

where $F_i^{st}$ and $F_i^{ts}$ are the output features of the corresponding $i$-th CFDN module. $\Psi_\alpha(*)$ and $\Psi_\beta(*)$ represent the lightweight block of two convolution layers. $\Phi_{IN}(*)$ denotes instance normalizations. $\gamma^s$ and $\gamma^t$ are two learnable parameters, initialized to zeros, where $\gamma^s$ is used to control what degree of semantic information is integrated, and $\gamma^t$ is used to control what degree of texture information is integrated. $\oplus$ denotes the element-wise addition operation. The added residual connection of the CFDN module retains more information in its own feature domain.

### 3.2.3 Adaptive Attention Fusion (AAF)

The adaptive attention fusion module follows the mutual dual-task generator to refine the output of the mutual dual-task generator further. It consists of three blocks: semantic-affinity cross-attention, global-context self-attention and gated feature fusion to fuse the feature output of two attentions. As shown in Fig. 3.4 (b), it takes the $F_5^{st} \in \mathbb{R}^{C \times H \times W}$ ($C$ stands for the number of channels) and pixel semantic label (i.e., one-hot soft-label encoding) $S \in \mathbb{R}^{N \times H \times W}$ ($N$ stands for the number of category labels) as inputs, where $S$ is predicted according to the output feature $F_5^{ts} \in \mathbb{R}^{C \times H \times W}$ in TS decoder branch by a residual block with three convolution layers. $H \times W$ is $128 \times 128$ in our network. The detailed diagram of the AAF module is shown in Fig. 3.6.

#### *1) Semantic-affinity cross-Attention (SA)*

Following the principle of self-attention [92], we propose a cross-attention between the inpainted feature of damaged areas and the known feature of undamaged areas. For the convenience of description, we define the feature of these two areas as foreground features and background features, respectively. Furthermore, the actual cross-attention is carried out between the foreground and the background' sub-regions, which share the same semantic category

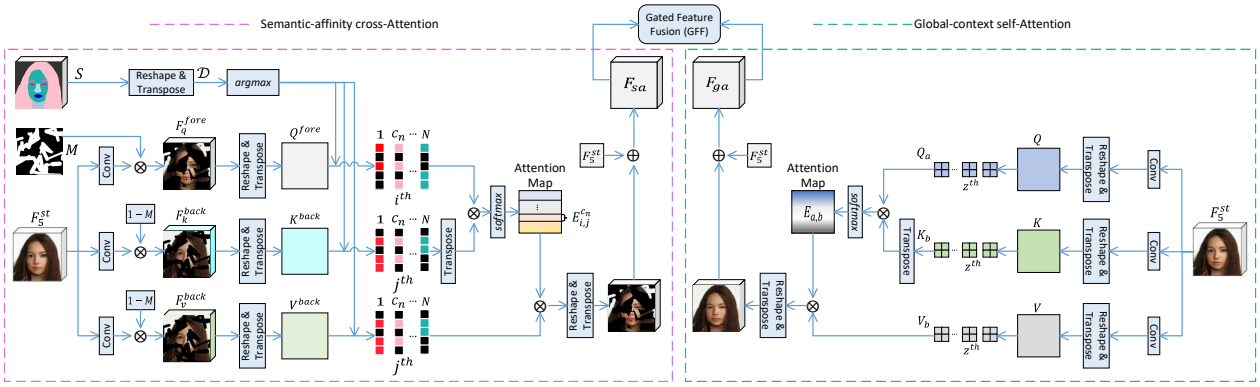



Figure 3.6: Overview of the Adaptive Attention Fusion (AAF) module, which includes three blocks: semantic-affinity cross-attention, global-context self-attention, and gated feature fusion. They focus on constructing texture dependencies from the perspectives of semantic-affinity regions and global-context regions, respectively. Finally, the gated feature fusion block is utilized to fuse them.

label constrained by pixel semantic label $S$. Such cross-attention is called semantic-affinity cross-attention in this work.

As shown on the left side of Fig. 3.6, to produce the $Q^{fore}$ of cross-attention operation, we feed the texture features $F_5^{st}$ into a $1 \times 1$ convolutional layer to generate a new feature. Next, we apply the same mask $M$ used in the mutual dual-task generator to pick the foreground feature $F_q^{fore}$. Then we reshape and transpose $F_q^{fore}$ to get $Q^{fore} \in \mathbb{R}^{Z \times C}$, where $Z \in \mathbb{R}^{HW}$ and $Z$ stands for the number of foreground pixels in $Q^{fore}$. At the same time, to produce $K^{back}$ and $V^{back}$ of cross-attention operation, we put $F_5^{st}$ into the other two $1 \times 1$ convolutional layers and generate two new features. Followed by the same way, we use mask $1 - M$ to get background features $F_k^{back}$ and $F_v^{back}$. We reshape and transpose them to get $K^{back} \in \mathbb{R}^{Z' \times C}$ and $V^{back} \in \mathbb{R}^{Z' \times C}$ , where $Z' \in \mathbb{R}^{HW}$ and $Z'$ stands for the number of background pixels.

We also reshape and transpose $S$ to acquire $\mathcal{D} \in \mathbb{R}^{(HW) \times N}$. According to $\mathcal{D}$, the texture in $Q^{fore}$, $K^{back}$, and $V^{back}$ can be divided into multiple sub-regions, each of which is affiliated under the different semantic category label $c_n$, formulated as:

$$
\begin{aligned}
Q_{c_n}^{fore} &= \{Q_{[i,*]}^{fore} | \, argmax(\mathcal{D}_{[i,*]}) = c_n\}, \\
K_{c_n}^{back} &= \{K_{[j,*]}^{back} | \, argmax(\mathcal{D}_{[j,*]}) = c_n\}, \\
V_{c_n}^{back} &= \{V_{[j,*]}^{back} | \, argmax(\mathcal{D}_{[j,*]}) = c_n\},
\end{aligned}
\tag{3.7}
$$

where $c_n \in [1, N]$ stands for the category label, and function $argmax(\mathcal{D}_{[i,*]}) = c_n$ means all

positions $i$ corresponding to the category label $c_n$. $Q_{c_n}^{fore} \in \mathbb{R}^{Z_{c_n} \times C}, K_{c_n}^{back}, V_{c_n}^{back} \in \mathbb{R}^{Z'_{c_n} \times C}$. $Z_{c_n}$ represents the foreground pixel features belonging to the category $c_n$, and $Z'_{c_n}$ represents the background pixel features belonging to the category $c_n$.

Subsequently, we carry out a matrix multiplication followed by a Softmax layer to compute the attention map $E_{i,j}^{c_n}$ under the same semantic category $c_n$. It is expressed as:

$$\begin{aligned} S_{i,j}^{c_n} &= Q_{c_{n_{[i,*]}}}^{fore} \cdot (K_{c_{n_{[j,*]}}}^{back})^{\mathrm{T}}, \\ E_{i,j}^{c_n} &= \frac{\exp(S_{i,j}^{c_n})}{\sum_{j=1}^{N'_{c_n}} \exp(S_{i,j}^{c_n})}, \end{aligned} \tag{3.8}$$

where $E_{i,j}^{c_n}$ describes the attention relation between the $i^{th}$ feature vector of foreground and the $j^{th}$ feature vector of background under the semantic class $c_n$.

Then, we carry out a matrix multiplication between $E_{i,j}^{c_n}$ and $V_{c_n}^{back}$. We reshape and transpose the result back to $\mathbb{R}^{C \times H \times W}$, and rebuild foreground features. The rebuilt features are merged into the input feature map $F_5^{st}$ via element-wise addition. The output $F_{sa}$ of the SA module is obtained as below ($\delta$ is a learnable parameter initialized to zero):

$$F_{sa} = \delta \cdot [(E_{i,j}^{c_n} \cdot V_{c_{k_{[j,*]}}}^{back})] \oplus F_5^{st}. \tag{3.9}$$

### *2) Global-context self-Attention (GA)*

In the global-context self-attention block, we regard all positions of $F_5^{st}$ as the global-context regions where we calculate the self-attention of texture features between the different positions. This block further augments the self-correlation of features across all regions in the image.

The GA block is a good complement to the SA block. In SA, the pixels in the undamaged regions can contribute to the inpainting process only if the semantic labels of the pixels in the undamaged regions are the same as the labels of the pixels in the damaged region. However, if the pixels in the damaged region cannot find any pixels in the undamaged region, the SA cannot provide the referable information for the inpainting process. Since the GA calculates the attention relation based on the texture information only without considering the semantic labels of the pixels. Thus, the GA complements the SA to provide other referable information from undamaged to damaged regions.

As mentioned in [123], patch-wise contextual attention improves the efficiency of the attention mechanism. As shown on the right side of Fig. 3.6, we follow the similar process as the SA

block for the GA block to encode the texture feature $F_5^{st}$ into $Q$, $K$ and $V$ but split them into $z$ patches (i.e., $P_a \in \mathbb{R}^{h\times w\times C}(a = 1, ..., z)$) and compute patch-wise self-attention. Specifically, we compute attention correlation $E_{a,b}$, rebuild each patch $\hat{P}_a$ of $F_5^{st}$, and reshape and transpose $\hat{P}_a$ to get the final updated feature $F_{ga}$ as below (Where $\delta'$ is a learnable parameter initialized to zero, $\Phi_{R\&T}$ denotes the reshaping and transposing):

$$\begin{aligned} E_{a,b} &= softmax(\frac{Q_a \cdot K_b^{\mathrm{T}}}{\sqrt{h \cdot w \cdot C}}), \quad a, b \in 1, ..., z \\ \hat{P}_a &= \textstyle\sum_{b=1}^{z} E_{a,b} V_b, \\ F_{ga} &= \delta' \cdot \Phi_{R\&T}(\hat{P}_a) \oplus F_5^{st}. \end{aligned} \tag{3.10}$$

***3) Gated Feature Fusion (GFF)***

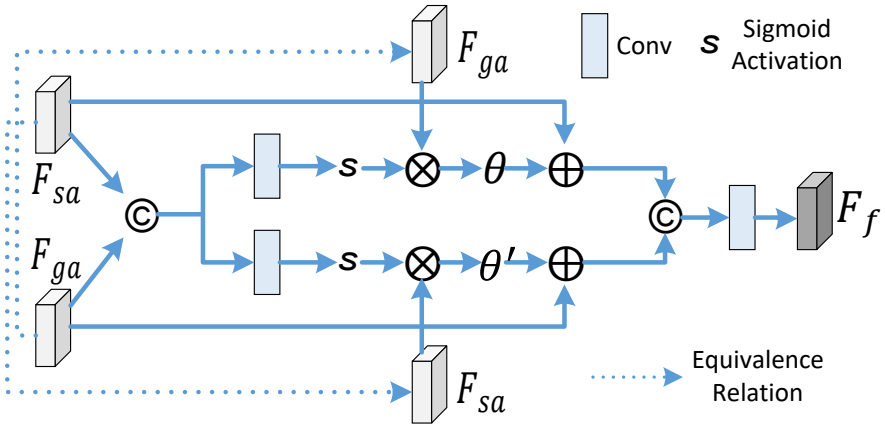



Figure 3.7: The detailed illustration of the Gated Feature Fusion (GFF) block.

We were inspired by bi-directional gated feature fusion [29], which is used to adaptively fuse edge structures and texture features. We apply it to fuse two texture feature outputs in our SA and GA blocks. Taking $F_{sa}$ and $F_{ga}$ as inputs, the GFF module generates the fused attention features through the diagram shown in Fig. 3.7. We formulate the diagram as follows:

$$\begin{aligned} A &= F_{sa} \oplus \theta \cdot [F_{ga} \otimes (s(\Phi_{conv}(Cat(F_{sa}, F_{ga}))))], \\ B &= F_{ga} \oplus \theta^{'} \cdot [F_{sa} \otimes (s(\Phi_{conv}(Cat(F_{sa}, F_{ga}))))], \\ F_f &= \Phi_{conv}(Cat(A, B)), \end{aligned} \tag{3.11}$$

where $Cat(*, *)$ stands for channel concatenation, $\Phi_{conv}(*)$ is a convolution layer with $3 \times 3$ kernels to halve the number of channels, $s(*)$ is Sigmoid activation to merge $F_{ga}$ into $F_{sa}$ or $F_{sa}$ into $F_{ga}$ adaptively. $\theta$ and $\theta^{'}$ are two learnable parameters initialized to zero. Finally, we generate fused feature $F_f$ by another convolution layer with half the number of channels.

### 3.2.4 Loss Functions

The feature $F_f$ goes through four groups of dilated convolutions with different rates, followed by a ToRGB convolution layer to generate the inpainted results $I_{out}$. Meanwhile, concatenating the $F_f$ and previous segmentation feature $F_5^{ts}$, we use a residual block with two convolution layers inside, tailed by a TosegMap convolution layer to predict the final segmentation map $S_{out}$.

We train the proposed mutual dual-task generator to recover the segmentation and texture information by a joint loss, containing pixel reconstruction loss, semantic segmentation loss, perceptual loss, and style loss. We formulate these losses as:

$$\begin{aligned}
\mathcal{L}_{inter} &= \mathcal{L}_{segmentation} + \mathcal{L}_{texture} \\
&= CE(softmax(S), \xi(F_{2\times}(S_{gt}))) + \ell_1(T, F_{2\times}(I_{gt})), \\
\mathcal{L}_{rec} &= \mathcal{L}_{vaild} + \lambda_h \cdot \mathcal{L}_{hole} \\
&= \|(I_{out} - I_{gt}) \otimes (1 - M)\|_1 + \lambda_h \cdot \|(I_{out} - I_{gt}) \otimes M\|_1 , \\
\mathcal{L}_{per} &= \sum_i^3 \|\phi_i(I_{out}) - \phi_i(I_{gt})\|_1 , \\
\mathcal{L}_{sty} &= \sum_i^3 \|\psi_i(I_{out}) - \psi_i(I_{gt})\|_1 ,
\end{aligned} \tag{3.12}$$

where $\mathcal{L}_{inter}$ is the intermediate loss consisting of semantic segmentation and pixel reconstruction losses, which encourage our mutual decoders to focus on learning semantic segmentation and texture inpainting, respectively. $\mathcal{L}_{segmentation}$ is the *cross entropy* loss, and $\xi(*)$ stands for transferring the category label of the ground truth segmentation map into the one-hot format. $F_{2\times}(*)$ means the operation of $Downsample_{2\times}$. $T$ denotes the output RGB image of the segmentation-guided texture generation decoder with size $128 \times 128$.

$\mathcal{L}_{rec}$ represents the reconstruction loss between $I_{out}$ and $I_{gt}$ in the pixel space, with different weight values in hole areas and non-hole valid areas.

$\mathcal{L}_{per}$ and $\mathcal{L}_{sty}$ stand for perceptual loss and style loss, respectively. They are conducted in the feature space. $\phi_i(*)$ stands for the activation map of the $i$-th pooling layer in pre-trained VGG-16. $\psi_i(*) = \phi_i(*)^{\mathrm{T}}\phi_i(*)$ is the Gram matrix.

We formulate the joint loss as:

$$\mathcal{L}_{joint} = \lambda_{inter}\mathcal{L}_{inter} + \mathcal{L}_{rec} + \lambda_{per}\mathcal{L}_{per} + \lambda_{sty}\mathcal{L}_{sty}. \tag{3.13}$$

We empirically set the values $\lambda_{inter} = 1$, $\lambda_h = 6$, $\lambda_{per} = 0.05$ and $\lambda_{sty} = 250$ in our experiments.

In summary, our full framework of proposed methods is trained in an end-to-end way, and the total training loss is a combination of losses of coarse inpainting network and mutual dual-task generator, i.e., $\mathcal{L}_c + \mathcal{L}_{joint}$.

## 3.3 Experiments

### 3.3.1 Experimental Setup

***1) Datasets***

We train the proposed mutual generator in three public datasets, including CelebA-HQ [124], Cityscapes [125], and Outdoor Scenes [126]. CelebA-HQ has 30K celebrity face images with corresponding semantic segmentation annotations for 19 facial categories. We use the first 29K images for training and the last 1K for testing. Cityscapes have 5K street-view images belonging to 20 annotation categories. We use 2,975 images from the original training set and 1,525 images from the original test set to construct our training dataset. The 500 images from the original validation are set as our test datasets. Outdoor Scenes has 9,900 training images and 300 testing images with fine annotations for 8 semantic categories. All datasets are randomly cropped and resized to $256 \times 256$ as input to our network. Binary masks set the damaged area of images, and we follow the same setup of irregular masks as [28]. Meanwhile, the irregular mask is created for our three training datasets according to [24].

***2) Parameter setting in training***

We set the batch size to 4 and use the Adam optimizer with $\beta_1 = 0.5$, $\beta_2 = 0.999$ to train our network in an end-to-end way. The learning rate of the generator and discriminator of the coarse inpainting network is set to 0.0002 and 0.00002, respectively. The mutual dual-task generator' learning rate is 0.0002.

### 3.3.2 Quantitative and Qualitative Results

We compare the proposed method with seven state-of-the-art inpainting methods, including general learning-based methods like GC [24], RFR [26], LGNet [30]; segmentation-guided methods like SPGNet [23], SGE [25], SWAP [28]; and edge-structure-guided methods like

CTSDG [29], WNet [31]. To pursue a fair comparison, all these methods are retrained according to our training data settings and their own training configurations. Regarding SGE and SWAP methods, we obtain comparative results from their published papers.

***1) Quantitative Evaluations***

We follow commonly used metrics in previous works [26], [30] to conduct our quantitative evaluations. They include SSIM (the structural similarity) [127], PSNR (peak signal-to-noise ratio), $\ell_1$ error, mean $\ell_1$ error (MAE), FID (Frechet inception distance) [128] and LPIPS (learned perceptual image patch similarity) [129]. The first four metrics evaluate the low-level pixel values between the ground-truth image and the inpainted image. The last two are based on the perception-level visual metrics for evaluation. We randomly generate irregular masks for each test image to form test mask-image pairs in the same way as the SWAP method. The mask-to-image ratio is distributed at 1∼20%, 20∼40%, and 40∼60%. All compared methods share the same mask-image pairs for testing.

Table 3.1 shows the quantitative results of the CelebA-HQ test dataset. Our method achieves the best scores in all the compared inpainting methods. Only with the PSNR metric, our method outperforms the other five comparison methods and is comparable to LGNet. Table 3.2 shows the quantitative results of Outdoor Scenes and Cityscapes datasets. The method of SGE and SWAP only provides quantitative scores of randomly sampling mask-to-image ratio in the 0∼60% interval on the three metrics of SSIM, PSNR, and FID. We take the average scores of the three sets of mask-to-image ratios as 'average' to compare. In the 'average' column, our method has the best scores in all metrics among all the inpainting methods, except for the marginal SSIM score on Outdoor Scenes test sets.

***2) Qualitative Evaluations***

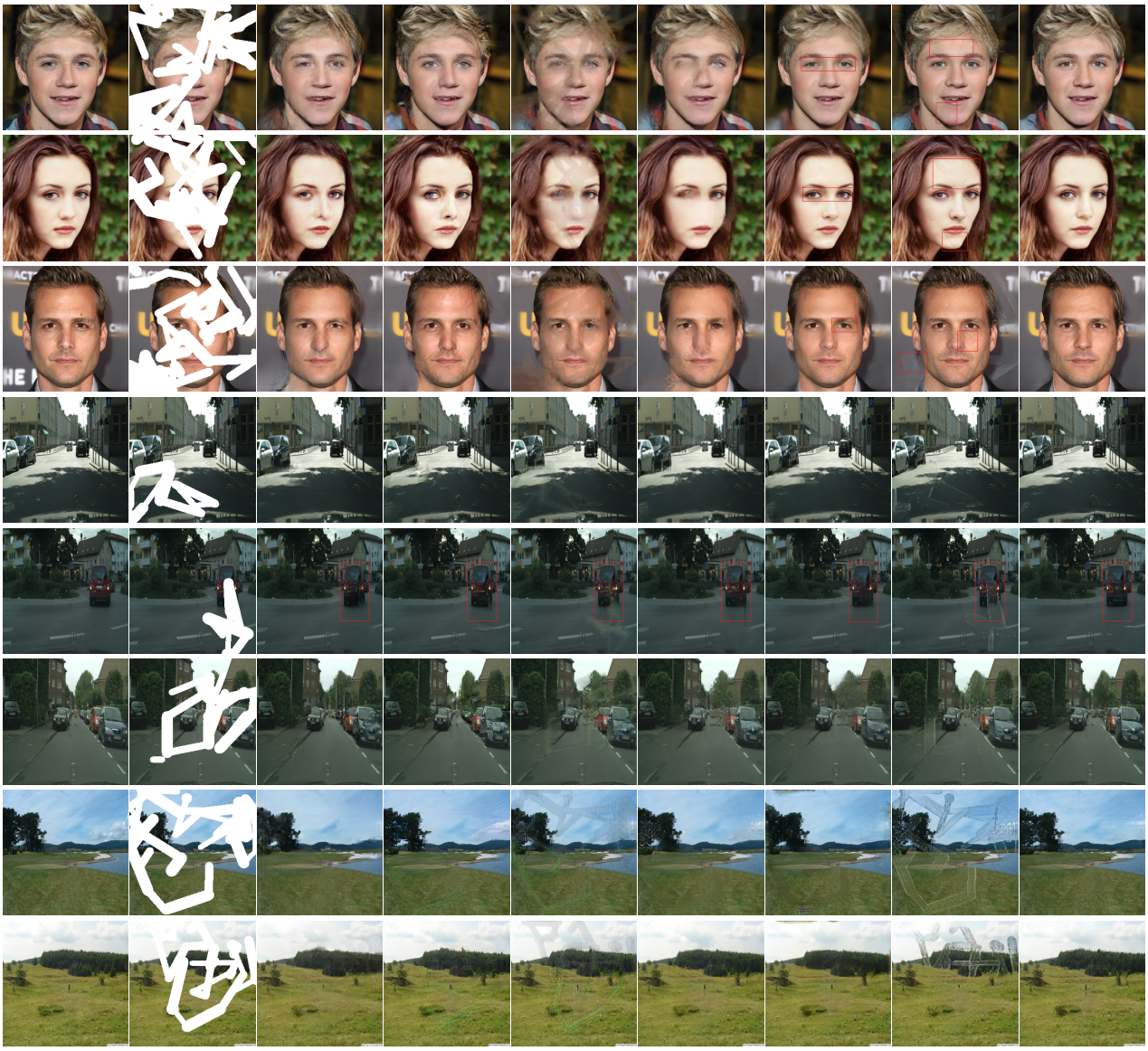
Qualitative visual results compared with six different inpainting methods are shown in Fig. 3.8. For the visual effect on CelebA-HQ images, we observe that our method achieves better consistency in color saturation between the inpainted areas and undamaged image areas. The WNet that uses deep features as structure information to interact with texture features still generates vague texture results in eye areas (see the region marked by the red box in Fig. 3.8 (g)). Although the LGNet method generates the second-best visual results than our method and the other five methods, we can see sight artifacts in inpainted parts of damaged areas (marked

Table 3.1: Quantitative results of our methods with six state-of-the-art inpainting methods on the CelebA-HQ test set. ↑ Higher is better. ↓ Lower is better. The red font indicates the best score, and the blue font indicates the second-best score.

| | Masks | 1∼20% | 20∼40% | 40∼60% |
|---|---|---|---|---|
| SSIM↑ | SPGNet [23] | 0.825 | 0.780 | 0.735 |
| | GC [24] | 0.956 | 0.892 | 0.833 |
| | RFR [26] | 0.947 | 0.870 | 0.804 |
| | CTSDG [29] | 0.956 | 0.907 | 0.859 |
| | WNet [31] | 0.962 | 0.903 | 0.842 |
| | LGNet [30] | 0.962 | 0.907 | 0.857 |
| | Ours | 0.966 | 0.913 | 0.863 |
| PSNR↑ | SPGNet [23] | 25.35 | 23.19 | 21.80 |
| | GC [24] | 32.01 | 26.47 | 24.16 |
| | RFR [26] | 30.89 | 25.91 | 23.77 |
| | CTSDG [29] | 31.92 | 27.49 | 25.15 |
| | WNet [31] | 32.32 | 27.00 | 24.43 |
| | LGNet [30] | 33.22 | 27.87 | 25.62 |
| | Ours | 33.61 | 27.88 | 25.46 |
| FID↓ | SPGNet [23] | 32.67 | 33.40 | 35.47 |
| | GC [24] | 3.80 | 8.83 | 13.15 |
| | RFR [26] | 6.05 | 15.68 | 26.81 |
| | CTSDG [29] | 8.76 | 12.92 | 17.49 |
| | WNet [31] | 4.71 | 12.28 | 22.78 |
| | LGNet [30] | 3.10 | 7.14 | 10.36 |
| | Ours | 2.66 | 6.16 | 8.97 |
| LPIPS↓ | SPGNet [23] | 0.127 | 0.148 | 0.169 |
| | GC [24] | 0.022 | 0.056 | 0.087 |
| | RFR [26] | 0.031 | 0.079 | 0.122 |
| | CTSDG [29] | 0.041 | 0.068 | 0.096 |
| | WNet [31] | 0.023 | 0.060 | 0.097 |
| | LGNet [30] | 0.016 | 0.040 | 0.061 |
| | Ours | 0.013 | 0.034 | 0.054 |
| $\ell_1$(%)↓ | SPGNet [23] | 4.38 | 4.81 | 5.34 |
| | GC [24] | 0.69 | 1.61 | 2.48 |
| | RFR [26] | 0.81 | 1.99 | 3.09 |
| | CTSDG [29] | 1.04 | 1.70 | 2.37 |
| | WNet [31] | 0.59 | 1.50 | 2.48 |
| | LGNet [30] | 0.66 | 1.46 | 2.20 |
| | Ours | 0.50 | 1.28 | 2.04 |
| MAE(%)↓ | SPGNet [23] | 6.55 | 6.88 | 7.42 |
| | GC [24] | 0.88 | 1.98 | 3.05 |
| | RFR [26] | 0.99 | 2.45 | 3.80 |
| | CTSDG [29] | 1.27 | 2.07 | 2.89 |
| | WNet [31] | 0.72 | 1.81 | 3.02 |
| | LGNet [30] | 0.84 | 1.81 | 2.72 |
| | Ours | 0.61 | 1.56 | 2.49 |

Table 3.2: Quantitative results of our methods with eight state-of-the-art inpainting methods on Outdoor Scenes and Cityscapes test sets. ↑ Higher is better. ↓ Lower is better. The red font indicates the best score, and the blue font indicates the second one.

| | Test Datasets | Outdoor Scenes | | | | Cityscapes | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Masks | 1∼20% | 20∼40% | 40∼60% | average | 1∼20% | 20∼40% | 40∼60% | average |
| SSIM↑ | SPGNet [23] | 0.817 | 0.730 | 0.649 | 0.732 | 0.896 | 0.817 | 0.737 | 0.817 |
| | SGE [25] | - | - | - | 0.760 | - | - | - | 0.740 |
| | SWAP [28] | - | - | - | 0.800 | - | - | - | 0.760 |
| | GC [24] | 0.935 | 0.840 | 0.750 | 0.842 | 0.943 | 0.867 | 0.793 | 0.868 |
| | RFR [26] | 0.928 | 0.822 | 0.727 | 0.826 | 0.942 | 0.860 | 0.780 | 0.861 |
| | CTSDG [29] | 0.918 | 0.843 | 0.770 | 0.844 | 0.941 | 0.882 | 0.822 | 0.882 |
| | WNet [31] | 0.938 | 0.847 | 0.758 | 0.848 | 0.949 | 0.875 | 0.791 | 0.872 |
| | LGNet [30] | 0.924 | 0.813 | 0.713 | 0.817 | 0.946 | 0.867 | 0.791 | 0.868 |
| | Ours | 0.937 | 0.845 | 0.756 | 0.846 | 0.950 | 0.884 | 0.817 | 0.884 |
| PSNR↑ | SPGNet [23] | 24.78 | 22.08 | 20.23 | 22.36 | 26.93 | 23.21 | 20.49 | 23.54 |
| | SGE [25] | - | - | - | 19.46 | - | - | - | 17.78 |
| | SWAP [28] | - | - | - | 20.31 | - | - | - | 17.86 |
| | GC [24] | 28.74 | 23.41 | 21.14 | 24.43 | 30.76 | 25.35 | 22.80 | 26.30 |
| | RFR [26] | 28.73 | 23.64 | 21.51 | 24.63 | 31.11 | 25.79 | 23.33 | 26.74 |
| | CTSDG [29] | 28.34 | 23.97 | 22.71 | 25.01 | 30.34 | 26.07 | 24.28 | 26.90 |
| | WNet [31] | 28.59 | 23.79 | 21.41 | 24.60 | 31.04 | 25.49 | 22.45 | 26.33 |
| | LGNet [30] | 28.11 | 23.02 | 20.85 | 23.99 | 31.38 | 26.17 | 23.69 | 27.08 |
| | Ours | 29.25 | 24.76 | 21.64 | 25.22 | 31.68 | 26.61 | 23.44 | 27.24 |
| FID↓ | SPGNet [23] | 30.85 | 52.74 | 70.89 | 51.49 | 22.21 | 38.11 | 56.34 | 38.89 |
| | SGE [25] | - | - | - | 39.14 | - | - | - | 41.45 |
| | SWAP [28] | - | - | - | 36.74 | - | - | - | 38.18 |
| | GC [24] | 13.50 | 32.72 | 46.38 | 30.87 | 13.97 | 28.01 | 41.52 | 27.83 |
| | RFR [26] | 15.64 | 39.29 | 58.20 | 37.71 | 15.58 | 34.06 | 54.27 | 34.64 |
| | CTSDG [29] | 24.68 | 39.88 | 52.53 | 39.03 | 26.86 | 39.19 | 54.00 | 40.02 |
| | WNet [31] | 15.44 | 35.98 | 53.32 | 34.91 | 16.98 | 39.72 | 70.38 | 42.36 |
| | LGNet [30] | 23.17 | 47.80 | 65.49 | 45.49 | 14.83 | 30.05 | 44.12 | 29.67 |
| | Ours | 12.67 | 31.81 | 46.78 | 30.42 | 11.85 | 26.74 | 35.83 | 24.81 |
| LPIPS↓ | SPGNet [23] | 0.093 | 0.148 | 0.204 | 0.148 | 0.065 | 0.121 | 0.178 | 0.121 |
| | GC [24] | 0.035 | 0.088 | 0.138 | 0.087 | 0.035 | 0.081 | 0.125 | 0.080 |
| | RFR [26] | 0.044 | 0.109 | 0.167 | 0.107 | 0.038 | 0.090 | 0.140 | 0.089 |
| | CTSDG [29] | 0.085 | 0.121 | 0.155 | 0.120 | 0.069 | 0.104 | 0.140 | 0.104 |
| | WNet [31] | 0.037 | 0.089 | 0.142 | 0.090 | 0.042 | 0.099 | 0.158 | 0.099 |
| | LGNet [30] | 0.047 | 0.111 | 0.166 | 0.108 | 0.037 | 0.087 | 0.131 | 0.085 |
| | Ours | 0.029 | 0.073 | 0.114 | 0.072 | 0.027 | 0.066 | 0.103 | 0.065 |
| $\ell_1$(%)↓ | SPGNet [23] | 3.76 | 4.89 | 6.04 | 4.90 | 2.53 | 3.65 | 5.02 | 3.73 |
| | GC [24] | 1.00 | 2.42 | 3.75 | 2.39 | 0.82 | 1.77 | 2.76 | 1.78 |
| | RFR [26] | 1.02 | 2.57 | 3.99 | 2.52 | 0.75 | 1.78 | 2.83 | 1.78 |
| | CTSDG [29] | 2.03 | 2.94 | 3.82 | 2.93 | 1.32 | 1.97 | 2.66 | 1.98 |
| | WNet [31] | 0.92 | 2.30 | 3.66 | 2.29 | 0.71 | 1.75 | 3.05 | 1.83 |
| | LGNet [30] | 1.17 | 2.78 | 4.28 | 2.74 | 0.84 | 1.74 | 2.66 | 1.74 |
| | Ours | 0.90 | 2.26 | 3.55 | 2.24 | 0.66 | 1.54 | 2.48 | 1.56 |
| MAE(%)↓ | SPGNet [23] | 4.26 | 5.52 | 6.76 | 5.51 | 4.13 | 5.98 | 8.23 | 6.11 |
| | GC [24] | 1.14 | 2.76 | 4.26 | 2.72 | 1.35 | 2.89 | 4.52 | 2.92 |
| | RFR [26] | 1.14 | 2.90 | 4.51 | 2.85 | 1.21 | 2.89 | 4.61 | 2.90 |
| | CTSDG [29] | 2.31 | 3.34 | 4.34 | 3.33 | 2.14 | 3.19 | 4.33 | 3.22 |
| | WNet [31] | 1.04 | 2.61 | 4.14 | 2.60 | 1.15 | 2.89 | 5.11 | 3.05 |
| | LGNet [30] | 1.33 | 3.15 | 4.82 | 3.10 | 1.39 | 2.84 | 4.32 | 2.85 |
| | Ours | 1.02 | 2.58 | 4.04 | 2.55 | 1.07 | 2.50 | 4.05 | 2.54 |



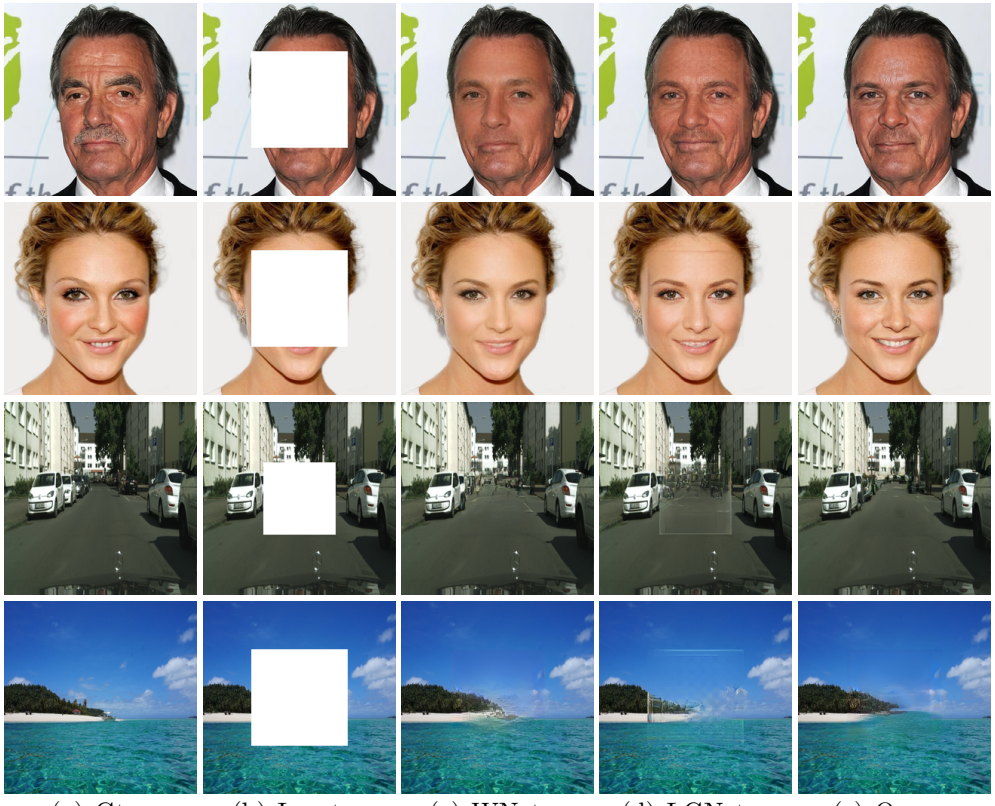
(a) Gt (b) Input (c) SPGNet (d) GC (e) RFR (f) CTSDG (g) WNet (h) LGNet (i) Ours

Figure 3.8: Qualitative results of our methods with SPGNet, GC, RFR, CTSDG, WNet, and LGNet on three datasets with irregular masks. From left to right: (a) GT, (b) Masked Input, (c) Inpainted results of SPGNet, (d) Inpainted results of GC, (e) Inpainted results of RFR, (f) Inpainted results of CTSDG, (g) Inpainted results of WNet, (h) Inpainted results of LGNet, (i) Inpainted results of Ours. From top to bottom: Three groups (each group contains three or two rows) separately correspond to CelebA-HQ [124], Cityscapes [125], and Outdoor Scenes [126] test images. [Best view with zoom-in.]

by the red box, best observation with zoom-in), which is not consistent with the undamaged image feature. The artifacts become more evident in Cityscapes and Outdoor Scenes images, as these two datasets have more complex features among different semantic classes for models

to learn. Unfortunately, the state-of-the-art method of LGNet is challenging to learn these features without anticipating the semantic information of images.

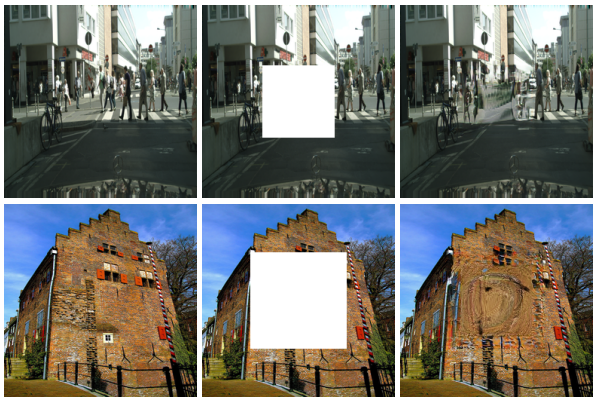
Meanwhile, we can also observe the results of the fifth row that our method can inpaint the whole red car with an intact semantic layout and consistent intra-class textures, which other comparison methods cannot achieve. For example, the one-off guidance image inpainting method, SPGNet, generates the semantically unreasonable layout for the damaged red car. Moreover, the inpainted texture inside the red car is turbid. It adequately reflects the superiority of our semantic-guided inpainting method by integrating the mutual dual-task generator and adaptive attention fusion.



(a) Gt (b) Input (c) WNet (d) LGNet (e) Ours

Figure 3.9: Qualitative results of our method with WNet and LGNet on three datasets with regular masks. From top to bottom: CelebA-HQ (first two rows), Cityscapes, and Outdoor Scenes test images, respectively.

Furthermore, as shown in Fig. 3.9, we select the latest two inpainting models (i.e., WNet and LGNet) to evaluate the qualitative results on regular-masked test images. The proposed method delivers more realistic visual effects compared with WNet and LGNet. Nevertheless, when the masked area covers multiple small objects (e.g., the case of the first row in Fig. 3.10



(a) Gt (b) Input (c) Ours

Figure 3.10: Negative cases of our method.

(a) 1∼20% (b) 20∼40% (c) 40∼60% (d) regular-mask

Figure 3.11: Inpainting examples with different mask-to-image ratios. [The first row has four damaged images. The second row shows the inpainting results. From left to right, it shows the results with different mask-to-image ratios. The last column is about the result using the regular shape mask.]

(b)) or contains rich textures (e.g., the case of the 2nd row in Fig. 3.10 (b)), our method has the difficulties to generate satisfying inpainting results (see the results in Fig. 3.10 (c)). The main reason causing the problem is that the proposed method relies on stable semantic information to guide the inpainting process progressively. However, it is hard to recover adequate semantic segmentation information when the masked area has many small objects (i.e., the thing is wholly masked out) or rich texture information. Consequently, it causes the problem of inpainting, which relies on semantic information as the guidelines.

Additionally, we give visual inpainting examples of our method for an image with different

mask-to-image ratios (e.g., 1∼20%, 20∼40%, 40∼60%, and the regular mask). As shown below in Fig. 3.11, as the mask ratio increases with more occluded areas, the information on the inpainted regions has more artifacts or inconsistencies.

### 3.3.3 Analysis of Model Complexity and Run-Time

To compare the model complexity and the run-time performance of the proposed method against its main competitors, we use FLOPs, the number of parameters, and inference time as the criteria. Inference time indicates the time for a forward pass of models. All methods are evaluated on a single NVIDIA QUADRO RTX 6000 GPU (24GB). The results are reported in Table 3.3. Because our model adopts the mutual dual-task generator with a coarse inpainting network, the number of parameters is relatively larger. However, our model has comparable computational complexity (in FLOPs) and reasonable inference time to other methods.

Table 3.3: Model complexity and run-time statistics. The best and second-best values are marked in bold and underlined.

| Model | FLOPs | Params | Infer. time |
|---|---|---|---|
| SPGNet [23] | 63.03 G | 96.08 M | <u>17.59 ms</u> |
| GC [24] | 55.52 G | **4.05 M** | **15.12 ms** |
| RFR [26] | 206.17 G | <u>30.59 M</u> | 58.98 ms |
| CTSDG [29] | **17.67 G** | 52.15 M | 37.95 ms |
| WNet [31] | <u>25.19 G</u> | 48.75 M | 39.86 ms |
| LGNet [30] | 69.67 G | 115.00 M | 25.32 ms |
| Ours | 43.82 G | 66.17 M | 36.38 ms |

### 3.3.4 Real-world Applications

We demonstrate the application of the proposed method to several real-world scenarios. As shown in Fig. 3.12, from the top left to bottom right, the examples are attribute editing of face, watermark removal, visual photo manipulation, and unwanted object removal. Users draw a mask in pictures to point out the editing area or unwanted objects and obtain the results by

using the proposed method. Our method produces visually pleasing results, which are not tuned by any post-processing.

Figure 3.12: Examples of real-world applications adopting our proposed mutual generator.

### 3.3.5 Ablation Studies

In this section, we use the CelebA-HQ test set to verify the role of different modules of our method. Meanwhile, we report the quantitative and qualitative results.

Table 3.4: Quantitative results of our method with or without coarse inpainting network on the CelebA-HQ test set. ↑ Higher is better. ↓ lower is better. Notes: PSNR and FID are used for measuring the quality of image inpainting. mIoU is used for measuring the quality of semantic segmentation.

| | Masks | 1∼20% | 20∼40% | 40∼60% |
|---|---|---|---|---|
| PSNR↑ | *Ours* | 33.61 | 27.88 | 25.46 |
| | *Ours_w/o_cn* | 33.35 | 27.63 | 25.19 |
| FID↓ | *Ours* | 2.66 | 6.16 | 8.97 |
| | *Ours_w/o_cn* | 2.85 | 6.51 | 9.51 |
| mIoU(%)↑ | *Ours* | 71.98 | 70.22 | 68.59 |
| | *Ours_w/o_cn* | 71.31 | 68.94 | 66.69 |


Figure 3.13: Qualitative results of our method with or without coarse inpainting network on the CelebA-HQ test image. From left to right: (a) Masked input, (b) Gt image, (c) Inpainted image of *Ours*, (d) Inpainted image of *Ours_w/o_cn*, (e) Gt segmentation map, (f) Semantic segmentation map of *Ours*, (g) Semantic segmentation map of *Ours_w/o_cn*. [Best view with zoom-in.]

#### *1) Coarse Inpainting Network*

We consider whether the coarse inpainting network impacts the learning of the mutual dual-task generator and the image segmentation and inpainting performance. We retrain the network after removing the coarse inpainting network, as shown in Fig. 3.4 (a). The new network, denoted as *Ours_w/o_cn*, retains the adversarial mechanism that is the same with coarse inpainting networks. We follow the same training setup as the full network denoted as *Ours* to train *Ours_w/o_cn*. Our full network achieves better numerical metrics in quantitative results shown in Table 3.4. It also reflects in the visual effects shown in Fig. 3.13 (c) and (f), as we can see that the network of *Ours_w/o_cn* can not generate the accurate facial jawline and mouth shape marked by the red box in Fig. 3.13 (g) and (d).

It validates the role of a coarse inpainting network, i.e., generating initial texture features in damaged regions of images. The initial features intuitively benefit the modeling of the mutual dual-task generator on the semantic segmentation task, which leads to better image inpainting guided by semantic segmentation. We adopt mean intersection-over-union (mIoU) as segmentation metrics to evaluate the semantic segmentation performance of networks. As shown in Table 3.4, it proves that the coarse inpainting network can prompt relatively better segmentation performance to benefit image inpainting.

Table 3.5: Quantitative results of our method with different variants in bidirectional CFDN modules on the CelebA-HQ test set. ↑ Higher is better. ↓ Lower is better. Notes: PSNR and FID are used for measuring the quality of image inpainting. mIoU is used for measuring the quality of semantic segmentation.

| | Masks | 1∼20% | 20∼40% | 40∼60% |
|---|---|---|---|---|
| PSNR↑ | *Ours* | 33.61 | 27.88 | 25.46 |
| | *Ours_cfc* | 33.46 | 27.77 | 25.35 |
| | *Ours_s_cfdn* | 33.44 | 27.72 | 25.29 |
| FID↓ | *Ours* | 2.66 | 6.16 | 8.97 |
| | *Ours_cfc* | 2.73 | 6.18 | 8.99 |
| | *Ours_s_cfdn* | 2.72 | 6.28 | 9.02 |
| mIoU(%)↑ | *Ours* | 71.98 | 70.22 | 68.59 |
| | *Ours_cfc* | 71.96 | 69.91 | 68.15 |
| | *Ours_s_cfdn* | 70.66 | 69.07 | 67.19 |

(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 3.14: Qualitative results of our method with different variants in bidirectional CFDN modules on the CelebA-HQ test image. From left to right: (a) Masked input, (b) Gt image, (c) Inpainted image of *Ours_cfc*, (d) Inpainted image of *Ours_s_cfdn*, (e) Inpainted image of *Ours*, (f) Gt segmentation map, (g) Semantic segmentation map of *Ours_cfc*, (h) Semantic segmentation map of *Ours_s_cfdn*, (i) Semantic segmentation map of *Ours*. [Best view with zoom-in.]

### *2) Cross-domain Feature DeNormalization (CFDN)*

The bidirectional CFDN module is proposed to facilitate mutual dependencies between the segmentation feature and the texture feature in mutual decoders, i.e., segmentation-guided texture (ST) generation and texture-guided segmentation (TS) generation. The TS branch receives texture feedback from the texture generation branch for refining segmentation information, which

will help texture generation. We explore two variants in bidirectional CFDN modules to verify the impact on segmentation and inpainting performances. One directly concatenates the two features for mutual dependencies. We denote this variant as bidirectional Cross-domain Feature Concatenation, i.e., CFC. The other removes the texture feedback and keeps a CFDN operation for the segmentation-guided texture generation. We denote this variant as Single CFDN, i.e., S_CFDN. We replace the bidirectional CFDN modules in our method with the two variants and construct two new networks, identified as $Ours\_cfc$ and $Ours\_s\_cfdn$. We retrain the two networks with the same training setup as our full framework. We report quantitatively and qualitatively image segmentation performances with corresponding image inpainting performances in Table 3.5 and Fig. 3.14, respectively.

We observe that $Ours$ reports better quantitative results than $Ours\_cfc$ and $Ours\_s\_cfdn$ while the numerical results of $Ours\_cfc$ are slightly better than those of $Ours\_s\_cfdn$. In the inpainted areas marked by the red box in Fig. 3.14, $Ours$ achieves better semantic segmentation results, which, in turn, provides better guidance for the following image inpainting in generating preferable visual details. On the one hand, it proves that bidirectional CFDN modules are more effective than bidirectional CFC modules in modeling mutual feature dependencies. On the other hand, it also reflects that the bidirectional CFDN modules can further transfer texture feedback to conduct the texture-guided segmentation task, which helps semantic segmentation and improves the quality of image inpainting guided by segmentation. However, the single CFDN module without the texture feedback cannot produce high-quality segmentation and inpainting results.

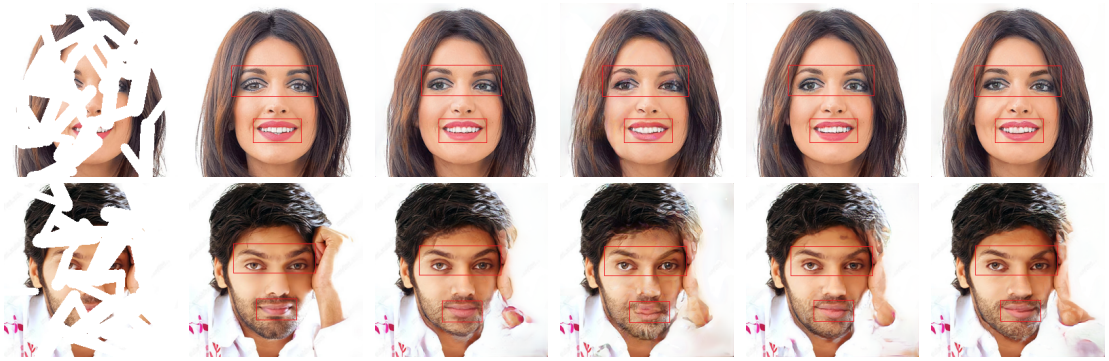
#### *3) Adaptive Attention Fusion (AAF)*

The adaptive attention fusion module is stacked at the end of mutual decoders to improve the inpainting results further. The module contains three blocks: Semantic-affinity cross-Attention (SA), Global-context self-Attention (GA) and Gated Feature Fusion (GFF). In the mutual dual-task generator, the segmentation-guided texture generation decoder and the texture-guided segmentation generation decoder can provide mutually reinforcing outputs into this module.

We explore three variants in the AAF module and report the impact on inpainting performance (i.e., $Ours\_w/o\_aaf$, $Ours\_w/o\_gff$ and $Ours\_w/o\_ga$). The $Ours\_w/o\_aaf$ represents the network that removes the AAF module from our proposed full network. The $Ours\_w/o\_gff$ represents the network that replaces the GFF operation with the concatenation operation. The

Table 3.6: Quantitative results of our method with different variants in the AAF module on the CelebA-HQ test set. ↑ Higher is better. ↓ lower is better.

| | Masks | 1∼20% | 20∼40% | 40∼60% |
|---|---|---|---|---|
| PSNR↑ | *Ours* | 33.61 | 27.88 | 25.46 |
| | *Ours_w/o_aaf* | 33.19 | 27.48 | 25.07 |
| | *Ours_w/o_gff* | 33.47 | 27.76 | 25.33 |
| | *Ours_w/o_ga* | 33.43 | 27.69 | 25.28 |
| FID↓ | *Ours* | 2.66 | 6.16 | 8.97 |
| | *Ours_w/o_aaf* | 2.90 | 6.72 | 9.72 |
| | *Ours_w/o_gff* | 2.72 | 6.16 | 9.04 |
| | *Ours_w/o_ga* | 2.73 | 6.32 | 9.10 |



(a) (b) (c) (d) (e) (f)

Figure 3.15: Qualitative results of our method with different variants in the AAF module on the CelebA-HQ test image. From left to right: (a) Masked input, (b) Gt image, (c) Inpainted image of *Ours*, (d) Inpainted image of *Ours_w/o_aaf*, (e) Inpainted image of *Ours_w/o_gff*, (f) Inpainted image of *Ours_w/o_ga*. [Best view with zoom-in.]

*Ours_w/o_ga* represents the network that removes the GA operation of the AAF module. It means that the network *Ours_w/o_ga* contains an SA block and has no GFF block to fuse the outputs of two attentions. We retrain the three new networks following the same training setup as the proposed full method, denoted as *Ours*.

The numerical comparisons and visual effects are shown in Table 3.6 and Fig. 3.15, respectively. From the quantitative results, $Ours > Ours\_w/o\_gff > Ours\_w/o\_ga > Ours\_w/o\_aaf$ (the symbol '>' represents 'is better than'). It verifies that the whole AAF module is an essential part of the proposed methods (e.g., $Ours > Ours\_w/o\_aaf$) and refines the inpainting results

of mutual decoders. The GFF block can better achieve feature fusion than the channel concatenation (e.g., $Ours > Ours\_w/o\_gff$). The GA block integrated into the SA block can act as a complement to model global texture dependency to improve the inpainting quality (e.g., '$Ours > Ours\_w/o\_ga$' or '$Ours\_w/o\_gff > Ours\_w/o\_ga$'). The SA block can establish cross-attention restrained by semantic labels to promote the performance (e.g., $Ours\_w/o\_ga > Ours\_w/o\_aaf$). We can also observe from Fig. 3.15 (c) that the whole AAF module of our proposed methods produces seamless eyelid lines and clear textures, which other variant counterparts cannot achieve.

## 3.4 Summary

In this chapter, we propose the Mutual Dual-task Generator, consisting of a shared encoder and mutual decoders to model the mutual dependencies between image texture and semantic segmentation, which has yet to be considered by existing inpainting methods with the guidance of semantic segmentation tasks. The bidirectional Cross-domain Feature DeNormalization module is introduced in the two decoders to construct mutual decoder branches. They progressively and hierarchically model the ST and TS generations, i.e., segmentation-guided texture generation (ST) and texture-guided segmentation generation (TS). It achieves better semantic guidance for image inpainting. Furthermore, following the designed Adaptive Attention Fusion module, the Mutual Dual-task Generator further improves the inpainting performances through learning semantic-affinity and global-context texture consistencies for inpainted textures of missing regions. Extensive experiments demonstrate the superiority of the proposed methods.

# Chapter 4

# Multi-Modal Collaboration for Consistent GAN-style Inpainting

## 4.1 Motivation

When the corrupted regions in an image involve semantic category diversity and structural complexity, the primary challenge in image inpainting is to ensure that the restored content maintains visual consistency with the non-missing areas in terms of texture authenticity, structural continuity, and semantic plausibility. To tackle this challenge, we draw inspiration from artistic drawing processes [67], [68], [69]. Humans approach drawing through a *pre-perception process* [67], [68], which relies on the global understanding of the whole image to decompose the complicated drawing tasks into three pre-perception primitive tasks, progressively drawing out high and low-frequency information. These three primitive tasks include sketching spatial shapes or boundaries like semantic layouts to represent the inter-class placement of image objects, sketching spatial strokes like edges to detail intra-class structures of objects, and then applying colors to render the textures of these objects [67], [68]. The information generated in these three primitive tasks can be regarded as three modalities. Another crucial process is *cross-perception collaboration* [69], which establishes mutual interaction among these three primitives to maintain consistency and refine details collectively. Sketched semantic layouts and edges define the boundaries and structures of objects while providing overall prior guidance for rendering textures, ensuring the restored textures align with these semantic and structural details. Simultaneously, artists continuously inspect partially rendered texture states and provide

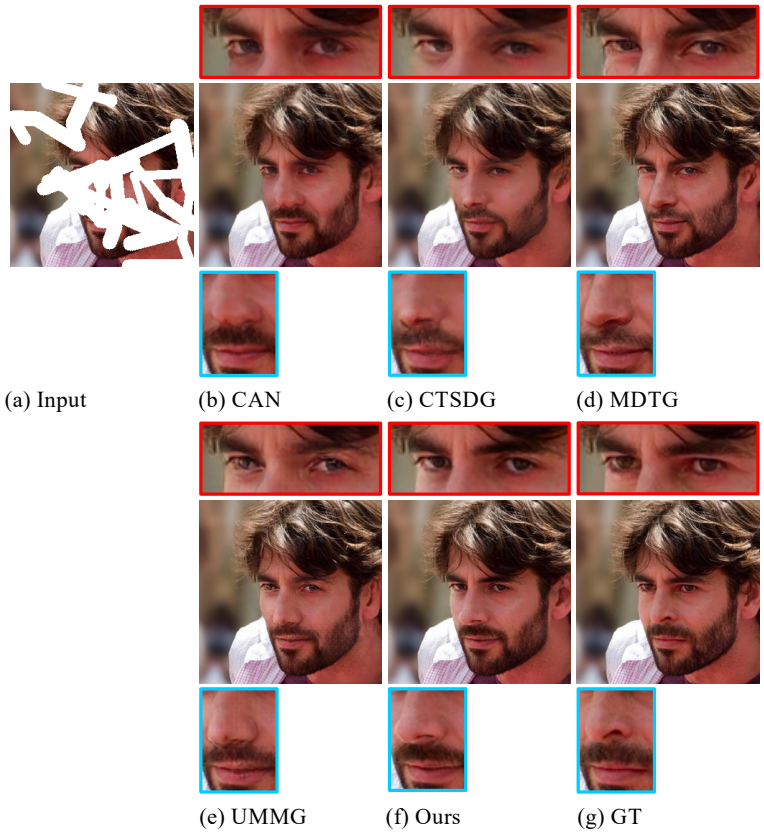



Figure 4.1: Inpainted results from different methods: (a) the corrupted input, (b) the result from the general SOTA method CAN [35], (c) the result from the edge-guided method CTSDG [29], (d) the result from the segmentation-guided method MDTG [91], (e) the result from the edge-and-segmentation-guided method UMMG [33], (f) our result, which exhibits more authentic textures, coherent structures, and reasonable semantics in the eyes and nose areas highlighted by the colored box, and (g) the ground-truth image.

feedback to refine semantic or edge placements, adjusting guidance based on evolving texture patterns and distributions [69]. This iterative process is vital for accurately refining guidance information for subsequent texture rendering stages.

The rationale behind the pre-perception process lies in decomposing a complex inpainting task into relatively simple sub-tasks, enabling individual optimization and easy control of their generation processes. However, previous deep learning inpainting methods [9], [10], [22], [24], [30], [34], [35], [43], [45], [84] uniformly model the relationship between corrupted and ground truth images through extensive supervised training. Such generic modeling does not explicitly consider image structure and semantic layout information, resulting in structural and semantic artifacts in restored results (see the distorted nose in Fig. 4.1 (b) as an example). Recently,

some studies have incorporated additional guidance knowledge, such as the image structure information [29], [60], [61], [130] (i.e., edges) or the semantic layout information [23], [25], [28], [49], [64], [91] (i.e., semantic segmentation maps), to improve inpainting quality. Despite the effectiveness of using image structure and semantic information as guidance, the individual use of one type of guidance presents limitations and makes it hard to obtain satisfactory inpainting results (see Fig. 4.1 (c) and (d)).

Recent advancements [33] consider integrating both types of information to guide image inpainting but struggle to model the mutual collaboration between image structure, semantic segmentation, and texture information to jointly optimize their generation, as adopted in the cross-perception collaborative process by human artists. Specifically, Yu *et al.* [33] used convolution denormalization to integrate image structure and semantic segmentation information to guide texture inpainting. However, convolution outputs lack a global interaction field, leading to potential inconsistencies due to insufficient integration of holistic structure and semantic information into the recovered texture (see the structurally blurred eyes and incongruous nose in the inpainting result shown in Fig. 4.1 (e)). Additionally, they ignore the effective feedback loop from recovered textures to relevant image structure and semantic information for further adjustment. In the current method, image structures and semantic layouts only guide texture inpainting without considering information updates caused by the progressively improved texture information, potentially disrupting the subsequent guidance process [63] and affecting the consistency of image inpainting.

To address these issues, we propose a new scheme (see Fig. 4.2), which mimics the human pre-perception and cross-perception collaborative process iteratively for inpainting tasks. The proposed scheme comprises a shared encoder and three-modality decoding stages that explicitly decouple the image inpainting task into three pre-perception primitive tasks to generate edges (i.e., image structures), semantic segmentation (i.e., image semantic layouts), and textures (i.e., image color details). Such a pre-perception process is achieved through the *proposed Pre-Perceptual Transformer Block (Pre-P TB, shown in Fig. 4.3)*, closely followed by the *proposed Cyclic Cross-Perceptual Interaction (CCPI, shown in Fig. 4.4)* to mimic the cross-perception collaborative process.

The proposed components (i.e., *Pre-P TB and CCPI*) are alternately deployed in decoding stages from rough to fine-grained levels. The *Pre-P TB* module captures global contextual de-

pendencies to mimic the artist's holistic understanding, followed by a dual-gated self-perceptron module that regulates information transmission across feature channels and spatial dimensions, aligning with the human pre-perception process of filtering vital patterns and distributions. This design enables the three information modalities to be optimized individually, effectively controlling their respective generations. Meanwhile, at each level, the three modalities mutually collaborate for joint optimization until all fine-grained details are fully restored. Mutual collaboration is realized through the *CCPI* module, which establishes globally relevant guidance via linear cross-modality attention and refines the two guidance processes through effective texture feedback enabled by dual-gated feedback interaction mechanisms. Together, these processes provide a functional simulation of human cross-perception, ensuring consistent inpainting quality across entire image structures, semantic layouts, and textures (as shown in Fig. 4.1 (f)). The contributions of the proposed multi-modal collaboration scheme are as follows:

- This thesis proposes a new inpainting scheme aligned with the pre- and cross-perception collaborative processes of human drawing, proving significant superiority.
- The proposed *Pre-P TB* models the pre-perception process to reconstruct image structure, segmentation, and texture individually, explicitly optimizing each generation process for the following mutual collaboration.
- The proposed *CCPI* follows the cross-perception collaborative process to deliver global guidance and effective feedback loops, gradually enhancing the details of all three modalities and ensuring consistent image inpainting.

## 4.2 Multi-Modal Collaboration via Pre-Perception and Cross-Perception Processes

As shown in Fig. 4.2, given a damaged image $I_{in} \in \mathbb{R}^{H \times W \times C}$, where $I_{in} = I_{gt} \odot (1 - M) \oplus M$, with $M$ being a binary mask (ones for corrupted regions and zeros for uncorrupted regions), and $I_{gt}$ the ground truth image, we utilize an embedding layer [32] to generate feature embeddings for tokens. These embeddings are input into a shared encoder to produce the latent feature $F_{enc}$. The $F_{enc}$ undergoes the three-modality decoder, which comprises three levels of pre-perception and cross-perception collaborative processes mimicking human drawing behaviors, to yield the final image components: edges $E_{out}$, RGB textures $I_{out}$, and semantic segmenta-

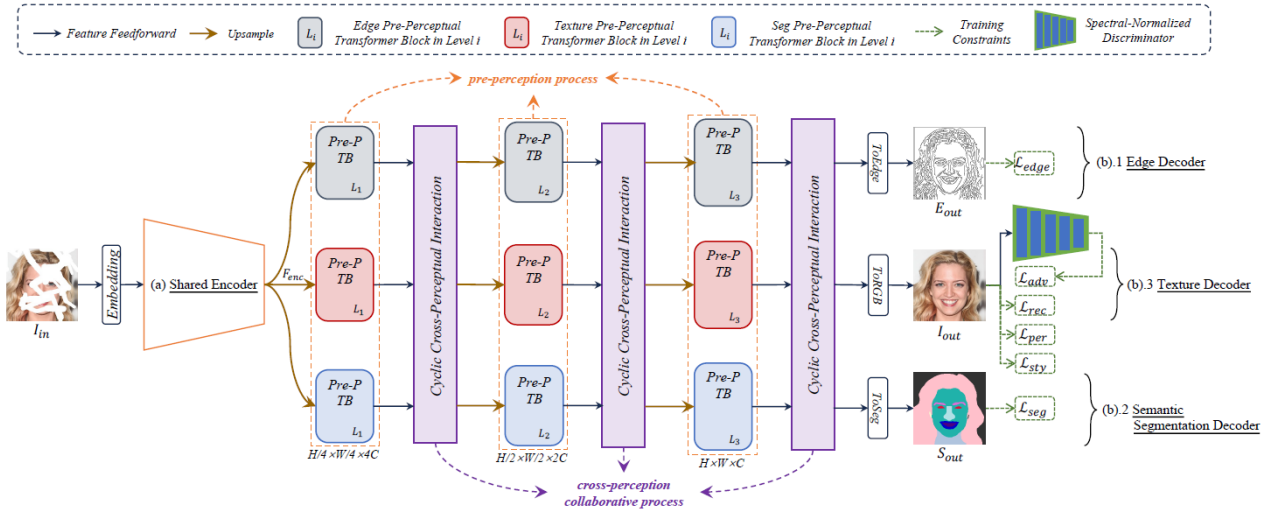



Figure 4.2: Full framework of our multi-modal collaboration method. It consists of four parts: (a) a shared encoder for context encoding, (b).1 Edge decoder, (b).2 Semantic segmentation decoder and (b).3 Texture decoder. The three-modality decoder progressively reconstructs $E_{out}$, $S_{out}$ and $I_{out}$ by modeling the pre-perception process and cross-perception collaborative processes.

tion $S_{out}$. *The proposed pre-perceptual transformer block* facilitates the pre-perception process, enabling hierarchical modeling of contextual dependencies and transmissions for edge, texture, and segmentation features individually. *The cyclic cross-perceptual interaction* emulates the cross-perception collaborative process to comprehensively model the interrelation among the three-modality features and progressively enhance their details for guiding consistent image inpainting. The interrelation can be described in two steps: providing globally reliable guidance for texture inpainting via *cross-task feedforward interaction* and incorporating effective texture feedback via *dual-gated feedback interaction* to refine the guidance process. In the framework, pixel-unshuffle [131] is applied within each encoder level for downsampling the spatial resolutions of features, and each encoder level also uses the pre-perceptual transformer block for context encoding. Pixel-shuffle [131] is used at the decoder level for upsampling feature spatial resolutions.

### 4.2.1 Pre-Perceptual Transformer Block (Pre-P TB)

Fig. 4.3 depicts the overview of Pre-P TB, designed to model the pre-perception process in human drawing. It includes a quasi-chunked linear self-attention (QCLSA) to capture global contextual dependencies across the entire image, simulating the artist's holistic understanding.

This ensures each modality (edges, semantics, textures) is aware of the global context, enabling context-informed feature extraction. Complementing QCLSA, a dual-gated self-perceptron (DGSP) filters vital patterns and distributions by processing differences between initial and refined features from QCLSA across channel (emphasizing salient feature patterns) and spatial (highlighting critical pixel-level distributions) dimensions. This DGSP mechanism mirrors how artists consciously suppress irrelevant details while amplifying essential ones during pre-perception, distinguishing Pre-PTB from generic multi-task learning networks (e.g., [132]) that focus solely on intra-task dependencies without explicitly capturing intra-modal contextual discrepancies.

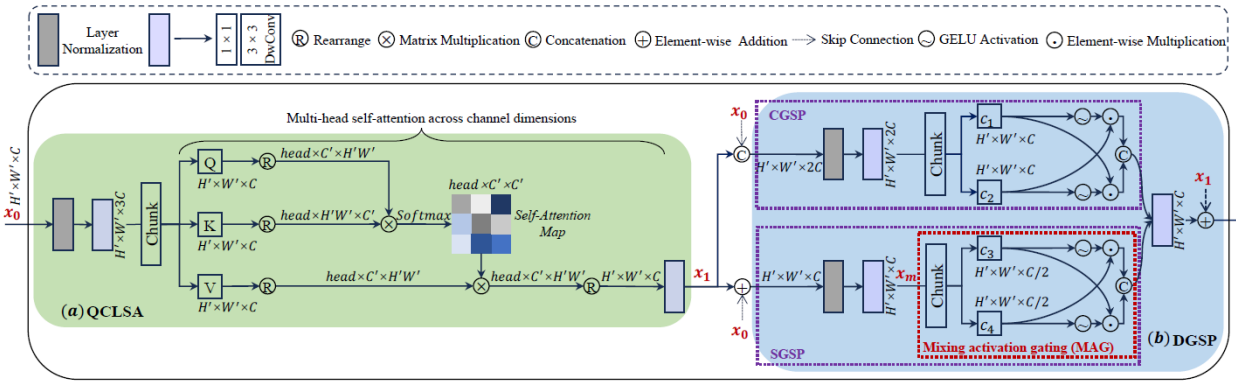



Figure 4.3: Diagram of the proposed Pre-Perceptual Transformer Block (Pre-PTB), comprising (a) Quasi-Chunked Linear Self-Attention (QCLSA), which models contextual dependencies to emulate an artist's holistic understanding, and (b) Dual-Gated Self-Perceptron (DGSP), which facilitates information transmission at spatial and channel levels, mirroring an artist's conscious filtering of critical patterns and distributions during pre-perception.

### *1) Quasi-Chunked Linear Self-Attention (QCLSA)*

We use the *chunk* operation [133] and implement self-attention across channel dimensions to construct the QCLSA module, depicted in Fig. 4.3 (a). Given the input feature $x_0 \in \mathbb{R}^{H' \times W' \times C}$ into QCLSA, the query $Q$, key $K$, and value $V$ are produced via the *chunk* function, which divides features across feature channels after $x_0$ undergoes Layer Normalization and convolution operations. The convolution comprises a $1 \times 1$ point-wise convolution and a $3 \times 3$ depth-wise convolution for learning cross-channel and channel-wise local-spatial contexts, respectively. Subsequently, the rearrangement function (denoted as $R$ in Fig. 4.3 (a)) reshapes $Q$, $K$ and $V$ as tokens and splits the total channel number $C$ into 'heads' (i.e., $head \times C' = C$) for multi-

head self-attention learning. Finally, $x_1$ is produced based on the learned self-attention map and another convolution. We express QCLSA as $x_1 = f_{qclsa}(x_0)$.

### *2) Dual-Gated Self-Perceptron (DGSP)*

The QCLSA conducts self-attention on the initial $x_0$ and produces the refined feature $x_1$. In the human pre-perception process, the differences between these two features are consciously analyzed, and the vital information (i.e., the salient feature pattern and critical pixel distribution) is selectively propagated to the next stage. To model this objective, we introduce Dual-Gated Self-Perceptron (DGSP), which applies mixing activation gating across both the channel feature dimension (deciding which feature patterns to activate and retain) and the spatial feature dimension (deciding which spatial distributions to activate and retain). In this way, the differences between $x_0$ and $x_1$ are fully activated and modulated, enhancing feature representation and enabling subsequent layers to focus on meaningful features in corrupted regions.

Illustrated in Fig. 4.3 (b), DGSP comprises a channel-gated self-perceptron (CGSP), which takes $x_0 \copyright x_1$ as input with augmented information in the channel dimension; and a spatial-gated self-perceptron (SGSP), which accepts $x_0 \oplus x_1$ as input with enhancing pixel-level information in the spatial dimension. The processing flows of CGSP and DGSP are similar; however, they differ in their focus on differences between features $x_0$ and $x_1$ in distinct dimensions, including feature and spatial dimensions, followed by embedded mixing activation gating (MAG) to implement feature activation and mutual modulation. We use the MAG schematic in SGSP (marked by the red dotted box in Fig. 4.3 (b)) to describe its mechanism and then formulate the processes of CGSP and SGSP.

Given the input feature $x_m \in \mathbb{R}^{H' \times W' \times C}$, MAG initially divides it into two local features $c_3$ and $c_4$ with half the channel numbers, expressed as $c_3, c_4 = f_{ck}(x_m)$ ($c_3, c_4 \in \mathbb{R}^{H' \times W' \times C/2}$) via the *chunk* function. The two local features are simultaneously activated through GELU functions operated at spatial pixel positions, and the resulting activated feature maps are used to adaptively gate (or modulate) the transformation of the other side feature in an element-wise multiplication manner (i.e., $\Theta_g(c_3) \odot c_4$ and $\Theta_g(c_4) \odot c_3$, $\Theta_g(*)$ is the GELU function). Finally, we aggregate the two transformed local features by concatenating their feature dimensions. The overall process

of MAG is formulated as follows:

$$\begin{aligned} f_{mag}(x_m) = [\Theta_g(c_3) \odot c_4] \textcircled{C} [\Theta_g(c_4) \odot c_3], \\ c_3, c_4 = f_{ck}(x_m), \end{aligned} \tag{4.1}$$

Through this MAG process for information mutual modulation across local features, the initial input $x_m$ is fully modulated to enhance its representation capacity. Based on MAG, we formulate CGSP and SGSP as follows:

$$\begin{aligned} f_{cgsp}(x_0 \textcircled{C} x_1) = f_{mag}(f_{lc}(x_0 \textcircled{C} x_1)), \\ f_{sgsp}(x_0 \oplus x_1) = f_{mag}(f'_{lc}(x_0 \oplus x_1)), \end{aligned} \tag{4.2}$$

where $f_{lc}(*)$ and $f'_{lc}(*)$ denotes layer normalization, followed by $1 \times 1$ and $3 \times 3$ depth-wise convolution.

Finally, by combining the results of $f_{cgsp}(x_0 \textcircled{C} x_1)$ and $f_{sgsp}(x_0 \oplus x_1)$ via convolutional weighting, along with an information residual $x_1$, we obtain the final result $x_2$ of DGSP, as depicted in Fig. 4.3 (b). The quantitative and qualitative effects of CGSP, SGSP, and the MAG used in them are analyzed in Chapter 4.3.5.

### 4.2.2 Cyclic Cross-Perceptual Interaction (CCPI)

As shown in Fig. 4.4 (a), we propose CCPI to simulate the cross-perception collaborative process, aiming to model the mutual collaboration among the three-modality features for consistent image inpainting. Specifically, CCPI includes the Cross-Task Feedback Interaction (CTFI) that uses linear cross-attention to model the relevance between texture features (queries) and guidance features (keys/values from edges/semantics), simulating how artists leverage structural and semantic priors to guide texture rendering. Moreover, CCPI introduces the Dual-Gated Feedback Interaction (DGFI), which explicitly models feedback from rendered textures back to guidance features. This DGFI captures the essential iterative feedback characteristic of human cross-perception.

The CCPI is performed after constructing edge, semantic, and texture features separately (denoted as $e_0$, $s_0$ and $t_0$ respectively) using the Pre-P TB in the pre-perception process. The diagram of CCPI, which consists of two CTFI and two DGFI components, can be formulated

as follows ($\alpha$ and $\gamma$ are two learnable parameters):

$$t_0^{ud} = \alpha \cdot f_{ctfi}(e_0, t_0) \oplus \gamma \cdot f_{ctfi}(s_0, t_0) \oplus t_0, \tag{4.3}$$

$$e_0^{ud} = f_{dgfi}(e_0, t_0^{ud}), \quad s_0^{ud} = f_{dgfi}(s_0, t_0^{ud}), \tag{4.4}$$

where Formula 4.3 aims to fuse edge and semantic auxiliary information to guide the update of texture features via CTFI, while Formula 4.4 represents feeding the updated texture back to refine the guidance feature via DGFI. The concepts of CTFI and DGFI will be introduced below.

#### *1) Cross-Task Feedforward Interaction (CTFI)*

As depicted in Fig. 4.4 (b), CTFI consists of a linear cross-attention (LCA) and a spatial-gated self-perceptron (SGSP). LCA accepts a specified guidance feature $g_0$ (i.e., $e_0$ or $s_0$, generalized as $g_0$ for simplicity) and texture feature $t_0$ as inputs. The guidance feature undergoes layer normalization and $1 \times 1$ convolution to generate the key K and value V, while the texture query Q is obtained from $t_0$. Subsequently, the output of LCA is obtained primarily through cross-attention computation across channel dimensions between Q, K and V (i.e., $softmax(R(Q) \otimes R(K)) \otimes R(V)$, where $R$ represents the Rearrange operation). We denote the process of LCA as $f_{lca}(g_0, t_0)$ to acquire the global correlative guidance $tg$. Then, we employ $f_{sgsp}(g_0 \oplus tg)$ from spatial dimensions to convey guidance information to texture features. The overall process of CTFI can be expressed as $f_{ctfi}(g_0, t_0) = f_{sgsp}(g_0 \oplus f_{lca}(g_0, t_0))$.

LCA explicitly selects appropriate guidance information based on cross-modality attention between texture features and guidance information, while SGSP utilizes the proposed mixing activation gating to modulate the selected guidance into the texture inpainting process. All of this ensures that CTFI achieves comprehensive guidance for image texture restoration. We validate the effectiveness of the proposed CTFI in Chapter 4.3.5 compared to the commonly used guidance module of feature normalization [33], [86], which comprises convolution components for implicit and local guidance.

#### *2) Dual-Gated Feedback Interaction (DGFI)*

After obtaining the updated texture $t_0^{ud}$, we carry out texture feedback to auxiliary guidance information through the proposed DGFI. DGFI, akin to DGSP in Chapter 4.2.1, applies its mechanism to cross-modality features. Illustrated in Fig. 4.4 (c), DGFI comprises both a

channel-gated cross-perceptron (CGCP) and a spatial-gated cross-perceptron (SGCP). We express their processes as follows:

$$\begin{aligned} f_{cgcp}(g_0 \textcircled{C} t_0^{ud}) &= f_{mag}(f_{lc}(g_0 \textcircled{C} t_0^{ud})), \\ f_{sgcp}(g_0 \oplus t_0^{ud}) &= f_{mag}(f'_{lc}(g_0 \oplus t_0^{ud})), \end{aligned} \tag{4.5}$$

Our DGFI effectively identifies discrepancies between the updated textures and the initial guidance features, such as edge structures and semantic segmentation maps. These discrepancies are selectively activated and modulated by the mixing activation gating (MAG) mechanism across both the channel dimension (selecting influential texture patterns) and the spatial dimension (selecting key texture distributions). Consequently, DGFI transmits meaningful texture patterns and distributions back to refine the guidance branches, thereby enhancing the quality of the guidance features. This process, akin to the feedback in the cross-perception collaborative process used by human artists, mitigates potential information disruptions caused by simple channel-concatenation operations. The effects of DGFI are analyzed in Chapter 4.3.5.

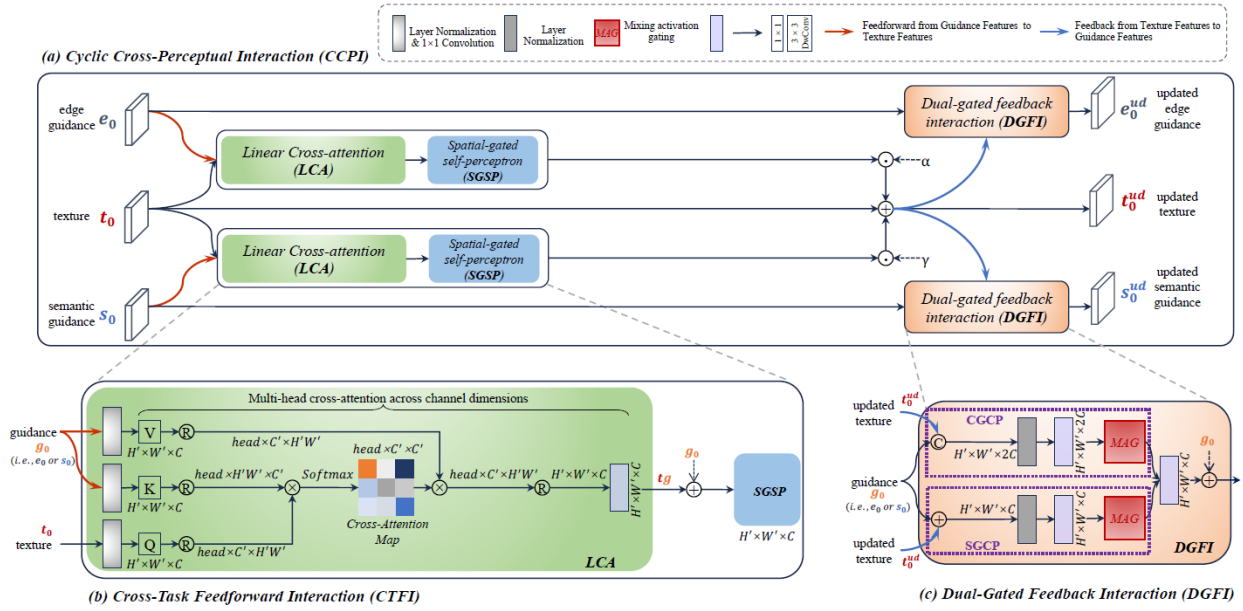



Figure 4.4: Diagram of the proposed Cyclic Cross-Perceptual Interaction (CCPI) in (a), comprising (b) Cross-Task Feedforward Interaction (CTFI) components, which fuse reliable guidance into texture features to emulate how artists use structural and semantic priors for texture rendering, and (c) Dual-Gated Feedback Interaction (DGFI) components, which enable effective texture feedback to refine guidance information, mirroring the iterative feedback process of human cross-perception.

### 4.2.3 Loss Functions

We train our network end-to-end via a set of joint multi-task loss terms as follows:

$$\begin{aligned}\mathcal{L}_{joint} = \lambda_{rec}\mathcal{L}_{rec} + \mathcal{L}_{per} + \lambda_{sty}\mathcal{L}_{sty} \\ + \lambda_{adv}\mathcal{L}_{adv} + \mathcal{L}_{edge} + \mathcal{L}_{seg},\end{aligned} \tag{4.6}$$

where the reconstruction loss $\mathcal{L}_{rec}$=$\|I_{out} - I_{gt}\|_1$, the perceptual loss $\mathcal{L}_{per}$=$\sum_i^3 \|\phi_i(I_{out}) - \phi_i(I_{gt})\|_1$, the style loss $\mathcal{L}_{sty}$=$\sum_i^3 \|\psi_i(I_{out}) - \psi_i(I_{gt})\|_1$ and adversarial loss $\mathcal{L}_{adv}$= $-\mathbb{E}[D(I_{out})]$ are used to train texture inpainting decoder. $\phi_i(*)$ stands for the activation map of the $i$-th pooling layer in the pre-trained $VGG$-16 [134], and $\psi_i(*) = \phi_i(*)^{\mathrm{T}}\phi_i(*)$ is the Gram matrix. The discriminator $D$, which uses spectral normalization [24], consists of five convolution layers with LeakyReLU activation functions (as depicted in Fig. 4.2). The weighting values $\lambda_{rec}$, $\lambda_{sty}$ and $\lambda_{adv}$ are empirically set to 2, 250, and 0.1, respectively, following [33], [84], [91].

$\mathcal{L}_{edge}$=$\mathcal{BCE}(E_{out}, E_{gt})$ and $\mathcal{L}_{seg}$=$\mathcal{CE}(\xi(S_{out}), \xi(S_{gt}))$ denote the binary cross-entropy and cross-entropy loss terms for training the edge decoder and semantic segmentation decoder, similar in [33]. $\xi(*)$ is the function used to convert class labels into the one-hot format. $I_{gt}$, $E_{gt}$ and $S_{gt}$ denote the ground-truth RGB image, edge map and semantic segmentation map, respectively.

## 4.3 Experiments

### 4.3.1 Experimental Setup

#### *1) Datasets*

Three publicly available datasets with diverse semantic contents and edge structures are used to train our network. These datasets, as introduced in Chatper 3.3.1, include CelebA-HQ [124], consisting of 30K celebrity face images across 15 semantic categories; Cityscapes [125], containing 5K street-view images categorized into 20 semantic annotations; and Outdoor Scenes [126], including 9,900 training images and 300 testing images across eight semantic categories. In CelebA-HQ, the first 29K images are designated for training, and the last 1K images are used for testing. For Cityscapes, 500 images from the original validation set are used for testing, while the remaining images are used for training. All datasets are randomly cropped and resized to $256 \times 256$ for network input during training and testing. Binary masks for the specified missing regions are randomly generated following previous works [24], [91]. The Canny edge

algorithm [56] extracts ground-truth edge maps from RGB images, with the sigma parameter set to 1 for CelebA-HQ and Outdoor Scenes and 3 for Cityscapes.

#### *2) Comparison Methods*

To evaluate the efficacy of our approaches, we consider eight state-of-the-art image inpainting methods: CTSDG [29], an edge-guided method; MDTG [91], a segmentation-guided method; UMMG [33], a method guided by both edges and segmentation; Magic [135], a latent-diffusion-based method guided by multiple external references as guidance (e.g., text, canny edge, segmentation); and four general learning-based methods: LGNet [30], KBII [34], MAT [32] and CAN [35]. For fairness, all comparison methods are re-trained using our training data settings and their respective training parameter configurations, except for Magic, a training-free model based on pre-trained stable diffusion [13] at $512 \times 512$ image resolutions. We interpolate test images to a resolution of 512 and set the guided text as 'Null' while using ground-truth edge and segmentation maps corresponding to the corrupted input as references for testing Magic. Subsequently, the output results of Magic are interpolated back to a resolution of 256.

#### *3) Parameter Setting*

We use the Adam optimizer and a batch size of 4 to train our network. The learning rate for the multi-modality generator is set to $2 \times 10^{-4}$, while the discriminator's learning rate is $1 \times 10^{-5}$.

### 4.3.2 Quantitative and Qualitative Results

#### *1) Quantitative Results*

We follow widely accepted quantitative metrics in previous studies [29], [32], [91], including FID [128], LPIPS [129], SSIM [127], PSNR, and MAE (mean $\ell_1$ error). The initial two metrics rely on high-level visual perception to evaluate the distribution disparity between ground truth and inpainted images, while the latter three assess low-level pixel similarity. Randomly generated irregular masks are applied to each test image, with mask-to-image ratios distributed at 1∼20%, 20∼40%, and 40∼60%. All comparison methods share the same mask-image pairs for testing.

Table 4.1 presents the quantitative results. Our method obtains the best scores for FID and

Table 4.1: Quantitative results with the state-of-the-art inpainting techniques on three test datasets. ↑ higher is better. ↓ lower is better. The best and second-best scores are marked in bold and underlined.

| Test Datasets | | CelebA-HQ | | | Cityscapes | | | OST | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Mask Ratios | | 1 ∼ 20% | 20 ∼ 40% | 40 ∼ 60% | 1 ∼ 20% | 20 ∼ 40% | 40 ∼ 60% | 1 ∼ 20% | 20 ∼ 40% | 40 ∼ 60% |
| FID↓ | LGNet [30] | 3.10 | 7.14 | 10.36 | 14.83 | 30.05 | 44.12 | 23.17 | 47.80 | 65.49 |
| | KBII [34] | 2.71 | 5.78 | 8.34 | 10.77 | 21.54 | 29.12 | 12.96 | 32.20 | 45.20 |
| | MAT [32] | 2.83 | 6.44 | 9.05 | 10.94 | 21.39 | 28.99 | 11.50 | 28.49 | 40.50 |
| | CTSDG [29] | 8.76 | 12.92 | 17.49 | 26.86 | 39.19 | 54.00 | 24.68 | 39.88 | 52.53 |
| | MDTG [91] | 2.66 | 6.16 | 8.98 | 11.85 | 26.74 | 35.83 | 12.67 | 31.81 | 46.78 |
| | UMMG [33] | 2.41 | 5.53 | 8.04 | 9.98 | 20.21 | 28.24 | 11.47 | 27.91 | 40.37 |
| | CAN [35] | 2.18 | 5.19 | 8.55 | 8.97 | 18.44 | 28.74 | 9.82 | 23.40 | 35.88 |
| | Magic [135] | 3.48 | 5.98 | 8.03 | 10.35 | 19.17 | 26.02 | 10.51 | 23.27 | **30.45** |
| | Ours | **1.92** | **4.65** | **6.93** | **8.69** | **17.74** | **24.80** | **8.95** | **21.94** | 31.26 |
| LPIPS↓ | LGNet [30] | 0.016 | 0.040 | 0.061 | 0.037 | 0.087 | 0.131 | 0.047 | 0.111 | 0.166 |
| | KBII [34] | 0.015 | 0.035 | 0.055 | 0.026 | 0.061 | 0.097 | 0.027 | 0.068 | 0.106 |
| | MAT [32] | 0.015 | 0.039 | 0.061 | 0.028 | 0.066 | 0.103 | 0.031 | 0.078 | 0.123 |
| | CTSDG [29] | 0.041 | 0.068 | 0.096 | 0.069 | 0.104 | 0.140 | 0.085 | 0.121 | 0.155 |
| | MDTG[91] | 0.014 | 0.034 | 0.055 | 0.027 | 0.066 | 0.103 | 0.029 | 0.073 | 0.114 |
| | UMMG [33] | 0.014 | 0.034 | 0.053 | 0.024 | 0.057 | 0.091 | 0.028 | 0.069 | 0.107 |
| | CAN [35] | 0.011 | 0.028 | 0.051 | 0.022 | 0.050 | 0.083 | 0.024 | 0.057 | 0.093 |
| | Magic [135] | 0.019 | 0.041 | 0.057 | 0.031 | 0.063 | 0.096 | 0.032 | 0.071 | 0.106 |
| | Ours | **0.009** | **0.025** | **0.040** | **0.021** | **0.049** | **0.078** | **0.023** | **0.055** | **0.088** |
| PSNR↑ | LGNet [30] | 33.22 | 27.87 | 25.62 | 31.38 | 26.17 | 23.69 | 28.11 | 23.02 | 20.85 |
| | KBII [34] | 33.35 | 27.91 | 25.54 | 32.08 | 26.25 | 23.54 | 29.81 | 24.40 | 22.03 |
| | MAT [32] | 32.93 | 27.23 | 24.76 | 31.42 | 25.59 | 23.00 | 29.03 | 23.56 | 21.18 |
| | CTSDG [29] | 31.92 | 27.87 | 25.62 | 30.34 | 26.07 | 24.28 | 28.34 | 23.97 | **22.71** |
| | MDTG [91] | 33.61 | 27.88 | 25.46 | 31.68 | 26.61 | 23.44 | 29.25 | 24.76 | 21.64 |
| | UMMG [33] | 33.76 | 28.18 | 25.85 | 32.39 | 26.72 | 24.10 | 29.65 | 24.44 | 22.15 |
| | CAN [35] | 34.38 | 29.07 | 26.35 | 32.47 | 27.01 | 24.06 | 29.66 | 24.85 | 22.41 |
| | Magic [135] | 33.15 | 28.12 | 25.77 | 31.01 | 26.26 | 23.84 | 28.19 | 23.88 | 21.76 |
| | Ours | **35.16** | **29.19** | **26.72** | **33.05** | **27.14** | **24.38** | **30.46** | **24.99** | 22.54 |
| SSIM↑ | LGNet [30] | 0.962 | 0.907 | 0.857 | 0.946 | 0.867 | 0.791 | 0.924 | 0.813 | 0.713 |
| | KBII [34] | 0.966 | 0.917 | 0.870 | 0.955 | 0.890 | 0.824 | 0.943 | 0.857 | 0.775 |
| | MAT [32] | 0.964 | 0.910 | 0.859 | 0.950 | 0.881 | 0.811 | 0.938 | 0.844 | 0.755 |
| | CTSDG [29] | 0.956 | 0.907 | 0.859 | 0.941 | 0.882 | 0.822 | 0.918 | 0.843 | 0.770 |
| | MDTG [91] | 0.966 | 0.913 | 0.863 | 0.950 | 0.884 | 0.817 | 0.937 | 0.845 | 0.756 |
| | UMMG [33] | 0.966 | 0.914 | 0.866 | 0.956 | 0.894 | 0.831 | 0.941 | 0.854 | 0.771 |
| | CAN [35] | 0.972 | **0.932** | 0.880 | 0.957 | 0.900 | 0.838 | 0.944 | 0.867 | 0.783 |
| | Magic [135] | 0.951 | 0.903 | 0.863 | 0.926 | 0.868 | 0.805 | 0.921 | 0.834 | 0.754 |
| | Ours | **0.974** | **0.932** | **0.891** | **0.961** | **0.905** | **0.846** | **0.947** | **0.868** | **0.792** |
| MAE(%)↓ | LGNet [30] | 0.84 | 1.81 | 2.72 | 1.39 | 2.84 | 4.32 | 1.33 | 3.15 | 4.82 |
| | KBII [34] | 0.61 | 1.50 | 2.39 | 0.10 | 2.35 | 3.84 | 0.94 | 2.37 | 3.74 |
| | MAT [32] | 0.65 | 1.64 | 2.63 | 1.10 | 2.60 | 4.16 | 1.05 | 2.65 | 4.22 |
| | CTSDG [29] | 1.27 | 2.07 | 2.89 | 2.14 | 3.19 | 4.33 | 2.31 | 3.34 | 4.34 |
| | MDTG [91] | 0.61 | 1.56 | 2.49 | 1.07 | 2.50 | 4.05 | 1.02 | 2.58 | 4.04 |
| | UMMG [33] | 0.66 | 1.62 | 2.52 | 0.95 | 2.22 | 3.59 | 0.96 | 2.39 | 3.74 |
| | CAN [35] | 0.53 | 1.30 | 2.32 | 0.96 | 2.22 | 3.73 | 0.91 | 2.22 | 3.69 |
| | Magic [135] | 1.44 | 2.59 | 3.08 | 1.96 | 3.15 | 4.51 | 1.94 | 3.29 | 4.58 |
| | Ours | **0.49** | **1.27** | **2.04** | **0.89** | **2.10** | **3.43** | **0.87** | **2.21** | **3.51** |

LPIPS across all compared approaches on three datasets, except for a slightly lower FID score than Magic at 40∼60% mask ratios on the OST dataset, validating that our framework, which follows the pre-perception and cross-perception collaborative processes of human painting behaviors, can attain new state-of-the-art quantitative results. Additionally, outcomes from low-level pixel-to-pixel metrics such as PSNR, SSIM and MAE indicate that our method yields evaluations close to raw images.

### *2) Qualitative Results*

To further verify the performance of the proposed method, we present the qualitative results to show visual discrepancies. Fig. 4.5 displays the visual outcomes of our method compared to eight other methods on CelebA-HQ test images. Our method outlines clear eye structures (especially in double-fold eyelids) and accurately reproduces the mouth layout, resulting in more coherent image inpainting, as highlighted by the red box in the first row of Fig. 4.5.

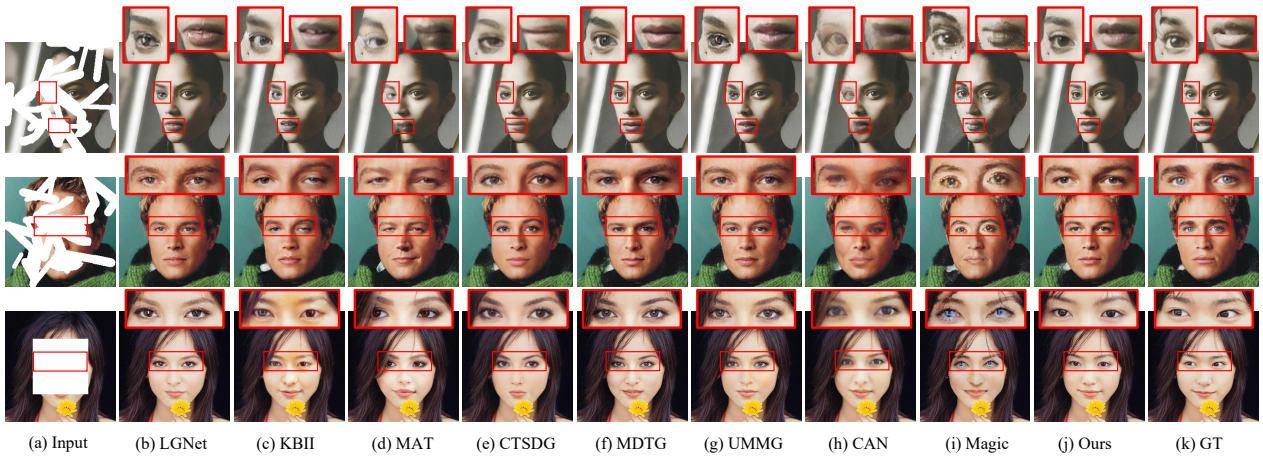



Figure 4.5: Qualitative results of our method with LGNet, KBII, MAT, CTSDG, MDTG, UMMG, CAN and Magic on CelebA-HQ [124] dataset. GT indicates the ground-truth image. [Best view with zoom-in.]

Regarding the inpainting results on Cityscapes test images shown in Fig. 4.6, our approach restores corrupted cars with more reasonable semantic layouts and consistent textures compared to other methods, as indicated by the red box. In Fig. 4.7, which depicts the inpainting results of Outdoor Scenes, our method effectively reconstructs continuous structures and realistic textures. Noteworthy examples include restoring the funnel-shaped roof, marked by the red box in the second row of Fig. 4.7, demonstrating superior performance compared to other methods, particularly the UMMG method of the same type. These observations validate the effectiveness

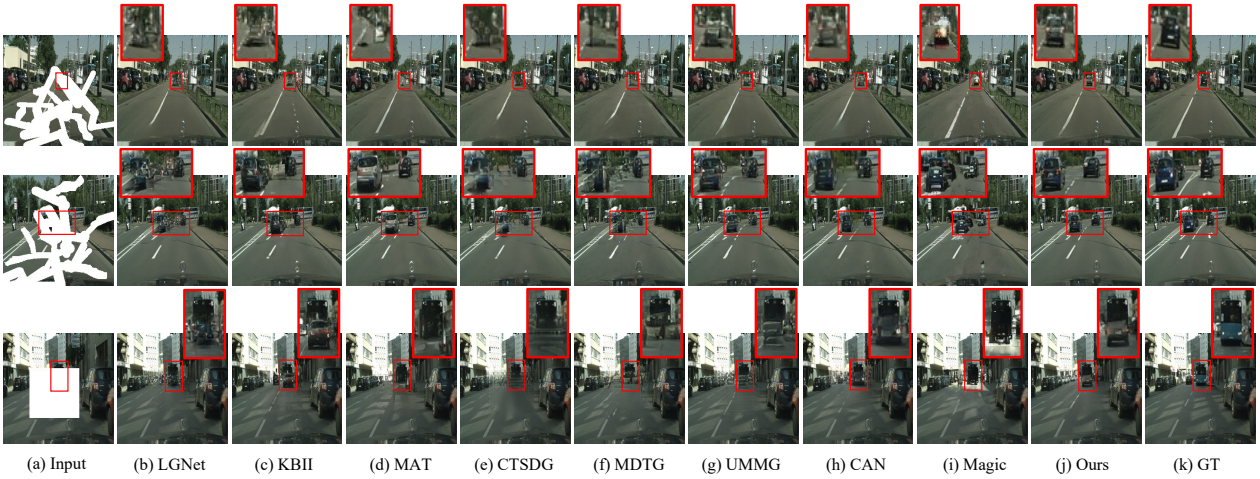



Figure 4.6: Qualitative results of our method with LGNet, KBII, MAT, CTSDG, MDTG, UMMG, CAN and Magic on Cityscapes [125] dataset. GT indicates the ground-truth image. [Best view with zoom-in.]

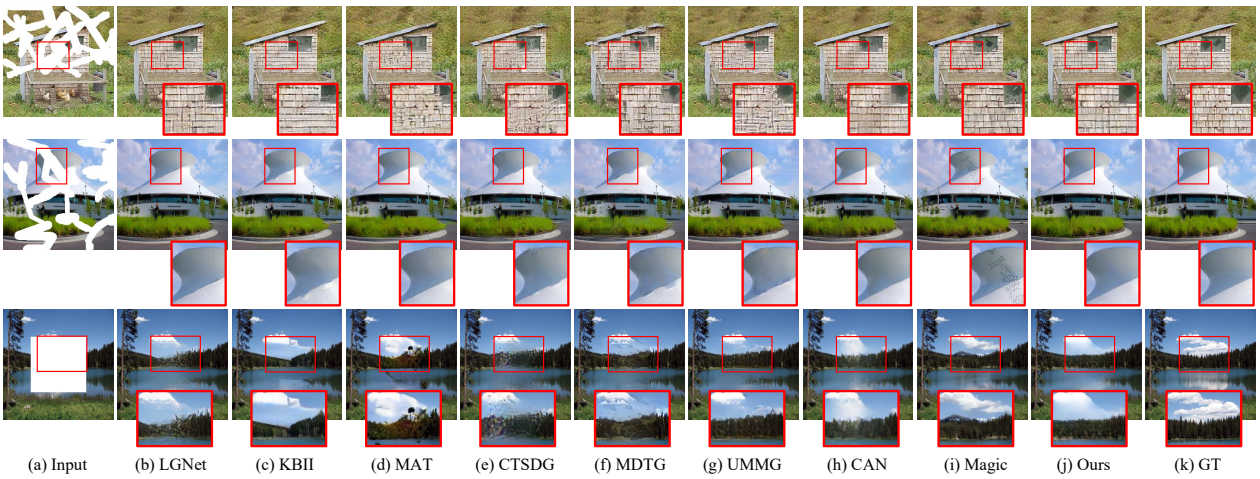



Figure 4.7: Qualitative results of our method with LGNet, KBII, MAT, CTSDG, MDTG, UMMG, CAN and Magic on Outdoor Scenes [126] dataset. GT indicates the ground-truth image. [Best view with zoom-in.]

of our approach in achieving consistent image texture inpainting with comprehensive guidance from auxiliary edge structures and semantic layouts, attributable to the proposed modules that mimic two human perceptual processes in drawing.

Fig. 4.8 presents the visual results of our method compared to other auxiliary-guided approaches. The final generated guidance maps, including reconstructed edges and predicted semantic segmentation maps, are shown below the inpainted RGB images. As marked by the red box in Fig. 4.8, our method generates more authentic textures, continuous and rich edge structures, and reasonable semantic layout maps. For instance, our method produces consecutive earring hoops and bangs as seen in the restored RGB image (highlighted by the red box in the first row of Fig. 4.8) and in the reconstructed edges and predicted semantic segmentation maps (highlighted by the red box in the second row of Fig. 4.8). This observation reaffirms the efficacy of our method in individually optimizing image textures, edges, and semantic layouts while fully modeling their collaboration for consistent image inpainting. This modeling process closely aligns with human drawing behaviors.


Figure 4.8: Qualitative results of our method and those of auxiliary-guided approaches on CelebA-HQ (rows 1∼2), Cityscapes (rows 3∼4), and Outdoor Scenes test images (rows 5∼6). The results include the inpainted images obtained with auxiliary guidance alongside the final generated guidance maps, which are displayed below the inpainted images. × denotes that no such guidance information is available for the corresponding methods. [Best view with zoom-in.]

### 4.3.3 Analysis of Model Complexity and Run-Time

Table 4.2 reports the model complexity and runtime of the proposed framework compared to its competitors. Evaluation criteria include FLOPs (floating point operations), the number of parameters, and inference time (the time of a forward pass through the network). All statistical assessments of the various model methods are conducted on a single NVIDIA QUADRO RTX 6000 GPU with 24GB of memory. Notably, our model's parameter count is 38.31M, significantly lower than that of the seven main competitors except for CAN (16.91M parameters). However, the CAN model does not incorporate explicit semantic layouts and edge-structure guidance to enhance image inpainting, as these guidance elements inevitably increase model parameters.

The hierarchical deployment of the pre-perceptual transformer block and cyclic cross-perceptual interaction within our multi-modality decoding framework leads to significant consumption of FLOPs and inference time compared to competitors. Nevertheless, these values are comparable to the standard transformer-based network MAT, which does not involve additional explicit guidance costs. However, our multi-modality transformer features three-branch decoders designed to execute multi-task generation, specifically addressing primary image inpainting guided by two auxiliary tasks consisting of edge reconstruction and segmentation prediction. As for the latent-diffusion method, Magic consumes considerable computational resources due to the iterative backward inference process inherent in the diffusion generation architecture.

Table 4.2: Model complexity and run-time statistics. The best and second-best values are marked in bold and underlined.

| Model | FLOPs | Params | Infer. time |
|---|---|---|---|
| LGNet [30] | 69.67 G | 115.00 M | **25.32 ms** |
| KBII [34] | <u>41.78 G</u> | 70.34 M | 82.74 ms |
| MAT [32] | 140.12 G | 59.77 M | 108.60 ms |
| CTSDG [29] | **17.67 G** | 52.15 M | 37.95 ms |
| MDTG [91] | 43.82 G | 66.17 M | 36.38 ms |
| UMMG [33] | 125.97 G | 51.25 M | <u>31.56 ms</u> |
| CAN [35] | 127.21 G | **16.91 M** | 76.19 ms |
| Magic [135] | - | 1103.95 M | 1007.68 ms |
| Ours | 150.85 G | <u>38.31 M</u> | 100.77 ms |

### 4.3.4 Various Applications with the Proposed Method

We also demonstrate several real-world applications of the proposed network. As shown in Fig. 4.9, from top to bottom, the examples include image editing, watermark removal, and unwanted object removal. Users delineate a mask within the images to specify the editing area or identify unwanted objects. Subsequently, our network processes the masked image to generate the desired output. As depicted in Fig. 4.9, our method consistently produces visually appealing results.

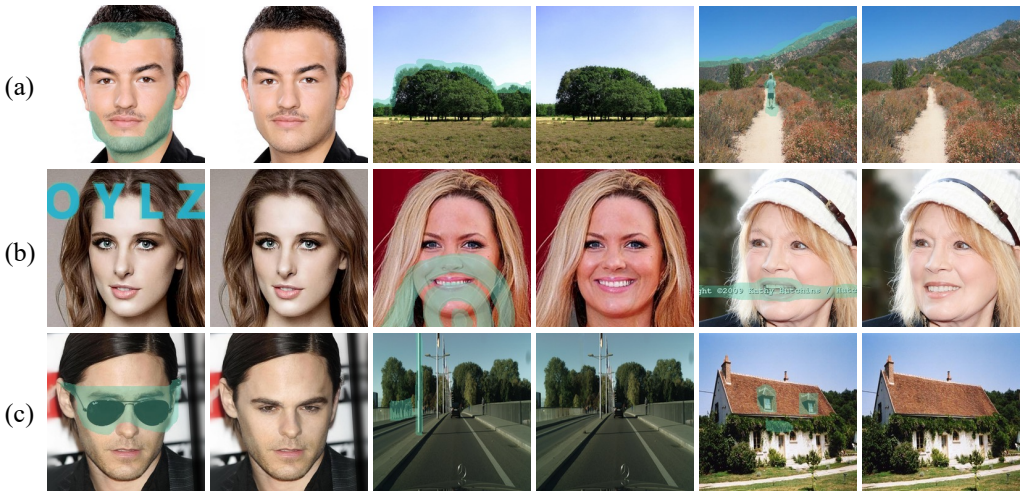



Figure 4.9: Examples of real-world applications adopting our method include (a) image editing, (b) watermark removal, and (c) unwanted object removal.

### 4.3.5 Ablation Studies

In this section, we use the CelebA-HQ test dataset to evaluate the impact of the proposed modules within our full framework. We construct nine sub-frameworks to investigate their effects. All sub-frameworks adhere to the same training configuration as our full framework *Ours*.

#### *1) Effect of Cross-Task Feedforward Interaction (CTFI)*

To assess the efficacy of CTFI, we replace all CTFI modules with commonly used feature normalization modules, such as Auxiliary DeNormalization (ADN) [33] introduced in Chapter 2.2.2, for incorporating edge and segmentation guidance into image inpainting. The resulting sub-framework is denoted as $f\_w\_adn$.

Table 4.3: Quantitative outcomes from ablation studies of evaluating the effect of different sub-modules. ↑ higher is better. ↓ lower is better.

| Framework | Description | FID↓ | | | PSNR↑ | | |
|---|---|---|---|---|---|---|---|
| | | 1 ∼ 20% | 20 ∼ 40% | 40 ∼ 60% | 1 ∼ 20% | 20 ∼ 40% | 40 ∼ 60% |
| *f_w_adn* | Abla.A) replace CTFI with ADN | 1.95 | 4.68 | 6.95 | 35.10 | 29.13 | 26.66 |
| *f_w_cwc* | Abla.B) replace DGFI with CWC | 1.96 | 4.71 | 7.01 | 35.08 | 29.07 | 26.62 |
| *f_wo_e* | Abla.C) remove edge guidance | 1.99 | 4.82 | 7.16 | 34.93 | 28.94 | 26.50 |
| *f_wo_s* | Abla.C) remove segmentation guidance | 1.99 | 4.83 | 7.17 | 34.99 | 29.02 | 26.56 |
| *f_wo_es* | Abla.C) remove all guidance | 2.01 | 4.90 | 7.20 | 34.89 | 28.93 | 26.49 |
| *f_wo_es_1* | Abla.D) remove all DGSP | 3.15 | 7.35 | 10.90 | 33.08 | 27.29 | 24.80 |
| *f_wo_es_2* | Abla.D) use direct activation in DGSP | 2.47 | 5.76 | 8.56 | 34.03 | 28.14 | 25.68 |
| *f_wo_es_3* | Abla.D) remove CGSP in DGSP | 2.10 | 5.03 | 7.35 | 34.74 | 28.81 | 26.34 |
| *f_wo_es_4* | Abla.D) remove SGSP in DGSP | 2.10 | 5.05 | 7.40 | 34.72 | 28.75 | 26.30 |
| Ours | Full framework | **1.92** | **4.65** | **6.93** | **35.16** | **29.19** | **26.72** |

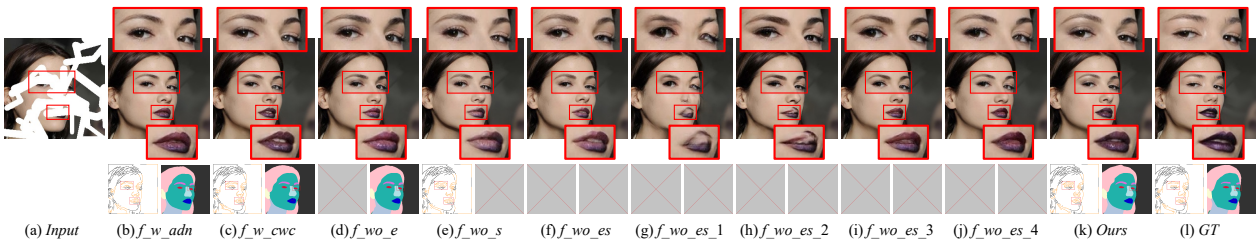



Figure 4.10: Qualitative visual results from our ablation studies on the CelebA-HQ test dataset. From left to right: (a) the corrupted input, (b-k) output results generated by various frameworks (refer to Table 4.3 for detailed descriptions of the different frameworks), and (l) the ground truth. The results include the inpainted images obtained with auxiliary guidance, alongside the final generated guidance maps shown below the inpainted images. × denotes that no such guidance information is available for the corresponding methods. [Best viewed with zoom-in.]

Quantitative results in the first and last rows of Table 4.3 reveal that $f\_w\_adn$ performs worse than our method in terms of FID and PSNR. The visual differences in the first row of Fig. 4.10 (b) and (k) show that our full method produces clearer double eyelid contours compared to $f\_w\_adn$.

This verifies that our method, employing CTFI, effectively selects suitable edge structure and semantic layout information from global views via the proposed cross-attention mechanism to enhance image inpainting quality, aligning with how humans use such information for detailed texture rendering.

#### *2) Effect of Dual-Gated Feedback Interaction (DGFI)*

Due to progressively improving texture information, some extent of information back-donation can refine guidance branches. Therefore, we replace all DGFI modules within our full framework with channel-wise concatenation (CWC) operations to verify the feedback effect from texture to edge and segmentation features. The resulting sub-framework is labeled as $f\_w\_cwc$.

The numerical results for FID and PSNR in the second and last rows of Table 4.3 indicate that $Ours$ outperforms $f\_w\_cwc$. Visual discrepancies, such as mouth artifacts and non-similarity between the two restored eyeballs, appear in the results of $f\_w\_cwc$ but not in $Ours$, as highlighted in the first row of Fig. 4.10 (c) and (k), respectively. The generated edges and semantic segmentation maps are shown below the inpainted RGB image for reference. For instance, the predicted right eye layout from $f\_w\_cwc$ overlaps the semantic layout of the nose; this discrepancy is best viewed with zoom-in.

These results suggest that DGFI can achieve more effective texture feedback through dual-gated feature transmission. This process mimics the feedback in the cross-perception collaborative process of human drawing by activating and transmitting the restored texture state, thereby enhancing the modeling of auxiliary edges and segmentation and refining their subsequent guidance processes.

#### *3) Effect of Different Guidance*

We create three sub-frameworks to confirm the contributions of edge structure and semantic segmentation to image inpainting: $f\_wo\_e$, $f\_wo\_s$, and $f\_wo\_es$. The sub-framework $f\_wo\_e$ signifies the removal of the edge decoder, thereby retaining only the segmentation branch. In

this case, the segmentation features alone guide texture inpainting through the segmentation-based CTFI, while the restored texture features provide feedback to the segmentation branch through the segmentation-based DGFI, as illustrated in Fig. 4.4. The sub-framework *f_wo_s* denotes the removal of the semantic segmentation decoder. In this configuration, the edge features independently guide texture inpainting via the edge-based CTFI, and the texture feedback is delivered to the edge branch through the edge-based DGFI. The sub-framework *f_wo_es* represents the removal of both edge and segmentation decoders, resulting in the absence of any guidance or fusion mechanism.

Quantitative evaluations concerning FID and PSNR, displayed in the third-to-fifth and last rows of Table 4.3, demonstrate the superiority of *Ours* over *f_wo_e*, *f_wo_s*, and *f_wo_es*. Additionally, the quantitative performance of *f_wo_e* and *f_wo_s* surpasses that of *f_wo_es*. This is also reflected in the visual effects shown in the first row of Fig. 4.10 (d), (e), (f), and (k). *f_wo_e* produces unsharp contours of the right eyelid. At the same time, *f_wo_s* predicts discordant texture in the right eyeball compared to the left eyeball due to the removal of semantic segmentation guidance as the same-class layout constraint. These phenomena worsen in the inpainted results produced by *f_wo_es*. In contrast, our full method outputs a more harmonious and consistent effect, as presented in the first row of Fig. 4.10 (k).

This confirms that combining edge structure and semantic segmentation in our method enhances image inpainting quality more effectively. It not only verifies the rationale for decoupling image inpainting tasks into three primitives but also demonstrates the importance of considering their mutual collaboration to jointly refine their details for consistent image inpainting, as practiced by human artists.

#### *4) Effect of Dual-Gated Self-Perceptron (DGSP)*

As sub-framework *f_wo_es* retains only the texture decoder with the pre-perceptual transformer block as the internal decoding unit, it is intuitive to assess the effect of embedding the dual-gated self-perceptron (DGSP). DGSP includes three crucial components: the channel-gated self-perceptron (CGSP), which regulates information transmission from feature dimensions; the spatial-gated self-perceptron (SGSP), which regulates information transmission from spatial-pixel dimensions; and the mixing activation gating (MAG) mechanism embedded in CGSP and SGSP to learn activated representations. Therefore, we devise four additional frameworks based on *f_wo_es* to explore these effects:

- Sub-framework $f\_wo\_es\_1$: removal of all DGSP modules from the pre-perceptual transformer block. In this configuration, both the channel-gated self-perceptron (CGSP) and spatial-gated self-perceptron (SGSP) are removed, and consequently, the mixing activation gating (MAG) embedded in CGSP and SGSP is also disabled.
- Sub-framework $f\_wo\_es\_2$: replacement of MAG in DGSP with Direct GELU activation used in the feed-forward perceptron layer [92], [136].
- Sub-framework $f\_wo\_es\_3$: removal of the CGSP sub-module within DGSP.
- Sub-framework $f\_wo\_es\_4$: removal of the SGSP sub-module within DGSP.

As depicted in the fifth-to-ninth rows of Table 4.3, the overall quantitative results suggest that $f\_wo\_es > f\_wo\_es\_3 > f\_wo\_es\_4 > f\_wo\_es\_2 > f\_wo\_es\_1$ in terms of FID and PSNR metrics (where '>' denotes 'is better than'). The visual differences produced by the four sub-frameworks are shown in Fig. 4.10 (g) $\sim$ (j), respectively. These results emphasize that (1) the CGSP and SGSP sub-modules within DGSP play significant roles in meaningful information propagation from both channel and spatial perspectives (as evidenced by $f\_wo\_es > f\_wo\_es\_3 > f\_wo\_es\_4$); (2) the mixing activation gating efficiently activates features and modulates their assistance, posing a challenging task for the Direct GELU activation to achieve (as indicated by $f\_wo\_es > f\_wo\_es\_2$); (3) the DGSP module complements QCLSA within the pre-perceptual transformer block, enhancing feature representation and propagation (as demonstrated by $f\_wo\_es > f\_wo\_es\_1$).

## 4.4 Summary

In this chapter, we design a new image inpainting framework inspired by the pre-perception and cross-perception collaborative processes in human drawing. Our approach incorporates a pre-perceptual transformer block to mimic the pre-perception process, tailored to learn contextual dependencies for modeling image edges, semantic segmentation, and texture information separately. Additionally, we embed a proposed mixing activation gating mechanism to enhance feature representation capacity. Furthermore, we introduce cyclic cross-perceptual interaction to simulate the cross-perception collaborative process. This mechanism aims to model the joint optimization among edge, semantic, and texture representations, and progressively refine their details to enhance the consistency of image inpainting. Experiments demonstrate that this

collaborative framework achieves higher structural consistency and visual realism compared to baseline models.

# Chapter 5

# Spatial Reasoning for Partially Occluded Objects in Diffusion-style Inpainting

## 5.1 Motivation

Inpainting becomes particularly challenging when dealing with partially corrupted objects, where substantial parts of an object are missing while others remain visible (e.g., the masked image in Fig. 5.2). In such cases, critical semantic details of the object and its surrounding background may be obscured, yet the remaining uncorrupted object areas offer valuable visual cues. However, current diffusion-based inpainting methods often fall short in leveraging these cues. Recent advances in diffusion-based text-to-image models [13], [36], [104], [105], [106], [138], [139] have shown impressive capabilities for generating high-quality images conditioning on text prompts, and advanced breakthroughs of text-guided diffusion pipelines for object inpainting. Such dedicated pipelines typically extend the input channels of existing diffusion models to incorporate the corrupted image and its mask, enabling text-guided object inpainting [13], [38], [72], [74], [75], [76], [137], [140], as illustrated in Fig. 5.1 (a). Although effective at generating semantically novel content within masked regions based on a given text prompt, these methods often lack precise pixel-level control over object structures or spatial postures, even when provided with detailed text prompts. For instance, in Fig. 5.2 (a), although the text specifies "two hind legs apart, not pressed together," the reconstructed zebra's legs remain

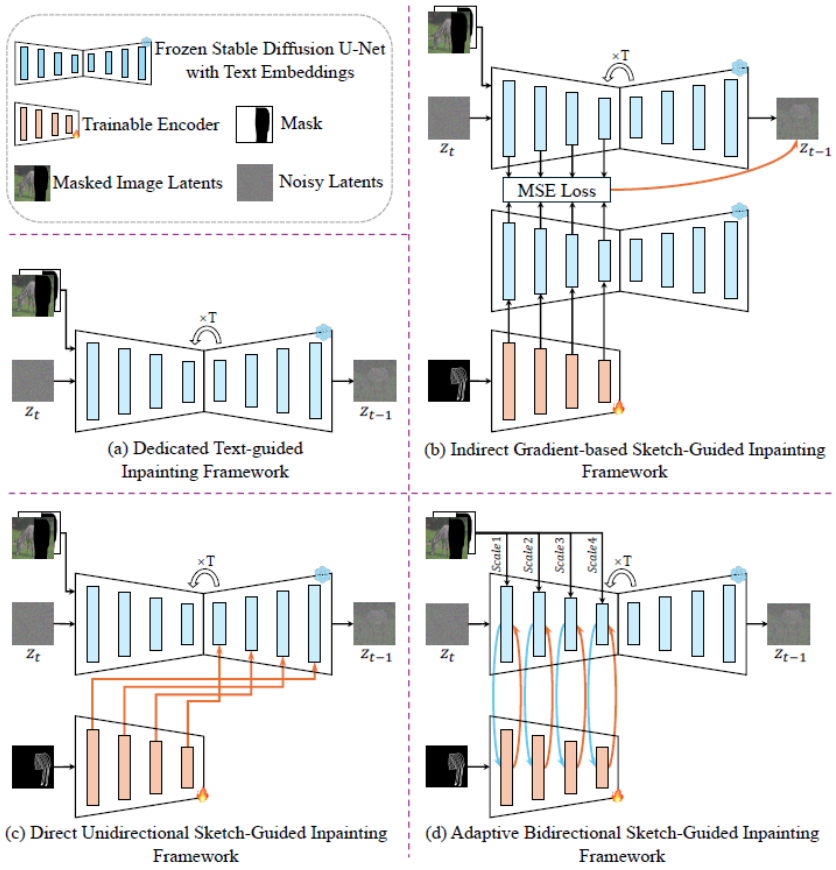



Figure 5.1: Comparison of diffusion-based object inpainting frameworks: (a) text-guided [13], [38], [137]; (b) sketch-guided with indirect gradient-based guidance [78]; (c) sketch-guided with direct unidirectional guidance [39]; (d) our adaptive bidirectional sketch-guided approach.

close together. This reveals a key limitation: while text prompts convey high-level semantics, they often struggle to translate abstract concepts into spatially pixel-wise reconstructions such as exact postures or orientations. To address this, recent shape-aware inpainting methods [38], [40], [137], [141] incorporate the object-specific mask and object label prompt to steer diffusion-based object inpainting. However, when the corrupted object possesses distinctive structural traits such as an animal's characteristic posture, these approaches often fail to recover object details that align with the visible parts of the object (see Fig. 5.2 (b)).

As the saying goes, "a picture is worth a thousand words." Accordingly, some studies propose using visual prompts such as sketches [142], [143] to offer fine-grained spatial control in text-guided diffusion inpainting. These sketches provide guidance on the object's spatial orientation and are integrated into the diffusion process to direct the reconstruction. Existing sketch-guided inpainting methods can be grouped into two categories: indirect gradient-based guidance [78], [144] and direct integration [39], [145].


Figure 5.2: Inpainted results from different diffusion-based inpainting methods: (a,b) Text-guided results using Stable Diffusion [13] and PowerPaint [137]; (c–e) Results of our sketch-guided method with corresponding sketch prompts; (f–h) ControlNet inpainting guided by sketch prompts [39]; (i–k) MaGIC inpainting guided by sketch prompts [78].

Gradient-based methods guide the denoising process by gradients between the latent features of the inpainting model and a target sketch map. As shown in Fig. 5.1 (b), a trainable encoder extracts sketch features that are compared via MSE loss to the inpainting latents, with the resulting gradients back-propagated to update the denoised latent to agree with the sketch. Although this strategy allows the model to follow the sketch, the indirect nature of gradient guidance often leads to unstable training and inaccurate alignment with the sketch (see Fig. 5.2 (i)∼(k)). Moreover, when the object is only partially corrupted, these methods can only push the sketch to the masked area, ignoring potential associations with unmasked object regions, which results in inconsistent reconstructions.

Direct integration methods mitigate this issue by inserting sketch features into the inpainting model via operations such as element-wise addition. ControlNet [39], shown in Fig. 5.1 (c), uses a trainable encoder to embed the sketches, which are then passed to a pre-trained text-

guided diffusion model. While this approach avoids unstable gradient flows, it still struggles in partially corrupted scenarios. Specifically, ControlNet unilaterally injects partial sketches without adapting to the uncorrupted object regions during the denoising inpainting process, resulting in ambiguous and inconsistent guidance (see Fig. 5.2 (f)∼(h)).

To address these challenges, we propose an **adaptive bi-directional sketch-guided inpainting framework** specifically designed for partially corrupted object inpainting. Our method also builds upon a frozen pre-trained text-guided Stable Diffusion model, but introduces three major innovations over direct-integration approaches like ControlNet (see Fig. 5.1 (d)): (1) **Bi-directional feature interaction**. Before incorporating sketch features, we first integrate multi-scale latents from the masked object image and mask into the denoised latents of the diffusion model. These latent features are then fused and passed to the sketch branch (blue arrows in Fig. 5.1 (d)). This enables the sketch features to adapt to uncorrupted object contexts and ongoing denoising progress. To this end, we propose a context-aware feature fusion module that learns a *visual mask* from the fused object latent features and the guided sketch. (2) **Sketch feature modulation**. Instead of direct element-wise addition, we propose a sketch-conditional affine transformation based on the learned *visual mask*. This modulation weights the degree of sketch integration, encouraging fine-grained guidance and consistency with visible object parts. (3) **Early-stage integration**. We insert the bi-directional interaction module in the encoder stage of the diffusion U-Net, enabling early control over object structure fidelity. Guided sketch features are further propagated to the decoder via skip connections.

Lastly, while most existing sketch-guided diffusion inpainting methods overlook the unique challenges of partially corrupted object restoration, we focus explicitly on this scenario. To support this work and promote future research, we contribute two novel datasets—CUB-sketch and MSCOCO-sketch—that provide four-tuple annotations (text, partial mask, partially masked image, and partial sketch) for each image.

In summary, our contributions are as follows:

- We propose a novel diffusion-based sketch-guided framework with bi-directional feature interaction tailored for partially corrupted object inpainting.
- We introduce context-aware feature fusion and sketch-conditional affine transformation to adaptively integrate sketch information in accordance with uncorrupted object regions.

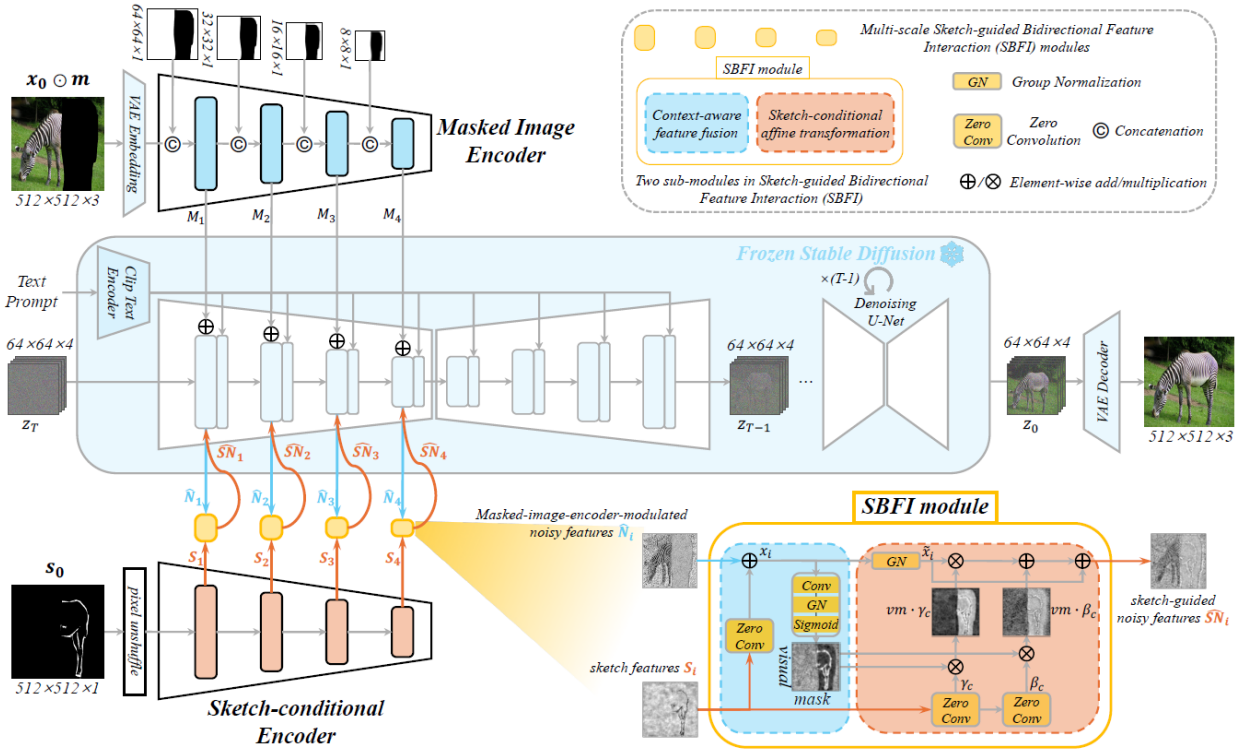



Figure 5.3: The proposed pipeline builds on the frozen T2I Stable Diffusion model and incorporates the following key components: a masked image encoder, which integrates binary mask localization and object contextual information from the corrupted image into the noisy features; and a sketch-conditional encoder followed by multi-scale Sketch-guided Bidirectional Feature Interaction (SBFI) modules, which enable fine-grained sketch integration while adapting to the uncorrupted object context during the integration process.

- We release CUB-sketch and MSCOCO-sketch, two benchmark datasets with paired text, mask, and sketch annotations to facilitate evaluation and future research.

## 5.2 Achieving Spatial Reasoning via Sketch-Guided Bidirectional Feature Interaction

As shown in Fig. 5.3, to enable a frozen text-embedded Stable Diffusion model for inpainting partially corrupted objects using partial sketch guidance, we employ a *Masked Image Encoder* to extract multiscale features from the corrupted image. These features provide both mask localization and rich contextual information from the visible, uncorrupted regions, which are then injected into the denoised latents of the Stable Diffusion model. This results in multiscale, masked-image-encoder-modulated noisy features.

Simultaneously, a *Sketch-Conditional Encoder* extracts multiscale features from the partial sketch. At the core of our framework lies the *Sketch-guided Bidirectional Feature Interaction (SBFI)* module, which first fuses the sketch-derived and masked-image-encoder-modulated features at each encoder level through a *context-aware feature fusion sub-module*. This fusion enables the guided sketch to adapt to the uncorrupted object context and the ongoing denoising process. Following this, a *sketch-conditional affine transformation sub-module* modulates the sketch integration based on the predicted visual mask generated by the context-aware feature fusion. The final output, sketch-guided noisy features, is then reintegrated into the frozen Stable Diffusion model.

The SBFI module operates across multiple scales in the encoder phase of the diffusion model, promoting fine-grained structural control and sketch-guided consistency from the earliest stages of generation. Meanwhile, the embedded text prompt, inherent to the frozen diffusion model, ensures that the restored object aligns with the specified semantic description. The framework is described in detail in the following subsections.

### 5.2.1 Preliminaries of the Training Objective for Diffusion Models

The proposed pipeline builds on the pre-trained text-to-image diffusion model, Stable Diffusion [13], which comprises a variational autoencoder (VAE) and a UNet denoiser. The VAE encodes a clean image $x_0$ into the latent space $z_0$ and decodes it for reconstruction. The UNet denoiser conducts diffusion in the latent space through a forward and reverse process.

In the forward process, Gaussian noise $\epsilon$ is added to the clean latent image $z_0$ to generate a noisy sample $z_t$ at timestep $t$ as:

$$z_t = \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \tag{5.1}$$

where $\bar{\alpha}_t$ represents the noise level. In the reverse process, the learnable UNet denoiser $\epsilon_\theta$ predicts the added noise $\epsilon_t$ at each timestep $t$, conditioned on text, enabling step-by-step denoising from Gaussian noise. The training objective for the diffusion model is:

$$\mathcal{L} = \mathbb{E}_{z_0,t,\epsilon_t} \left\| \epsilon_t - \epsilon_\theta \left( z_t, \tau_\theta(text), t \right) \right\|_2^2, \tag{5.2}$$

where $\tau_\theta$ is the CLIP text encoder. Based on this pre-trained model, our fine-tuning objective of incorporating trainable masked image encoder $enc_\theta(m)$ and sketch-conditional encoder $enc_\theta(s)$

is:

$$\mathcal{L} = \mathbb{E}_{z_0,t,\epsilon_t,m,s} \left\| \epsilon_t - \epsilon_\theta \left( z_t, \tau_\theta(text), enc_\theta(m), enc_\theta(s), t \right) \right\|_2^2 . \tag{5.3}$$

### 5.2.2 Masked Image Encoder

To fine-tune Stable Diffusion for sketch-guided image inpainting, we introduce a masked image encoder, as shown in the top left of Fig. 5.3. This encoder conveys the masked image information, including uncorrupted object details and partially occluded mask location, to the multi-level noisy features of the denoiser UNet encoder. It provides object contextual positioning for the subsequent insertion of partial sketches into the corrupted object area.

The architecture of the masked image encoder replicates the UNet denoiser encoder but removes the text cross-attention layer. The well-trained weights of the denoiser encoder serve as a strong prior for extracting masked image features. Given an input masked image $x_0 \odot m \in \mathbb{R}^{h \times w \times 3}$, with $h = w = 512$, where the binary mask $m$ with value 0 indicates corrupted object regions, we first embed it into a latent space using the VAE encoder. The spatial resolution of the latent representation is determined by the VAE downsampling factor (e.g., $\downarrow_8$), ensuring alignment with the latent data distribution of the denoiser UNet.

The masked image encoder then extracts multi-scale masked features $M = \{M_1, ..., M_L\}$, where $L$ corresponds to the number of scales in the UNet encoder. At each scale, the corresponding downsampled binary mask is concatenated channel-wise to emphasize the corrupted spatial regions. The dimensions of $M$ match the intermediate noisy features $N = \{N_1, ..., N_L\}$ in the denoiser UNet encoder. $M$ is added to $N$ at each scale, and the masked feature extraction and insertion process is formulated as follows:

$$M = enc_\theta\Big( vae_{emb}(x_0 \odot m), \{\downarrow_{f_i} (m)\}_{i=1}^{L} \Big), \tag{5.4}$$

$$\hat{N}_i = N_i + M_i, \quad i = 1, ..., L, \quad L = 4, \tag{5.5}$$

where $\downarrow_{f_i}$ denotes the downsampling operation at scale $i$ (in our experiments, $f_1 = 8, f_2 = 16, f_3 = 32, f_4 = 64$). The features $\hat{N}_i$ represent the noisy features modulated by the masked image encoder, as illustrated by the blue arrows in Fig. 5.3. These features incorporate both the uncorrupted object information and the mask position information.

### 5.2.3 Sketch-Conditional Encoder with Sketch-guided Bidirectional Feature Interaction

This thesis proposes a sketch-conditional encoder with multi-scale Sketch-guided Bidirectional Feature Interaction (SBFI) modules to inject encoded partial sketch features into the corrupted regions of noisy features. The architecture of the sketch-conditional encoder is identical to that of the masked image encoder, as shown in the bottom left of Fig. 5.3.

#### *1) Sketch-Conditional Encoder*

Given an input partial sketch $s_0 \in \mathbb{R}^{h \times w \times 1}$, which contains sketch details within the partially corrupted area, we first downsample it to the same latent resolution of the pre-trained UNet using pixel unshuffle [131]. The downsampled features are then fed into the sketch-conditional encoder to extract multi-scale sketch features $S = \{S_1, ..., S_L\}$, whose dimensions match those of the masked-image-encoder-modulated noisy features $\hat{N} = \left\{\hat{N}_1, ..., \hat{N}_L\right\}$. The sketch features are fused with the modulated noisy features at each scale using the proposed SBFI module, represented as $f_{sbfi}$. The process is formulated as:

$$S = enc_\theta\Big(\downarrow_{f_{pu}}(s_0)\Big), \tag{5.6}$$

$$\hat{SN}_i = f_{sbfi}(\hat{N}_i, S_i), \quad i = 1, ..., L, \quad L = 4, \tag{5.7}$$

where $\downarrow_{f_{pu}}$ denotes the pixel unshuffle operation with downsampling scale $\downarrow_8$. The features $\hat{SN}_i$ integrate both the sketch guidance and the masked-image-modulated noisy features at each scale.

#### *2) Sketch-guided Bidirectional Feature Interaction*

The SBFI module consists of two directional operations, each corresponding to a sub-module: context-aware feature fusion and sketch-guided affine transformation, as illustrated in the bottom right of Fig. 5.3.

In the context-aware feature fusion sub-module, the sketch feature $S_i$ is first processed through a zero convolution layer [39], and then added pixel-wise to the masked-image-encoder-modulated noisy feature $\hat{N}_i$. The resulting feature map, denoted as $x_i$, contains a rough object contour and incorporates visual information from the uncorrupted object regions. This feature is then passed through a $Conv{\rightarrow}GroupNorm{\rightarrow}Sigmoid$ module to predict a visual mask $vm \in \mathbb{R}^{h_j \times w_j}$,

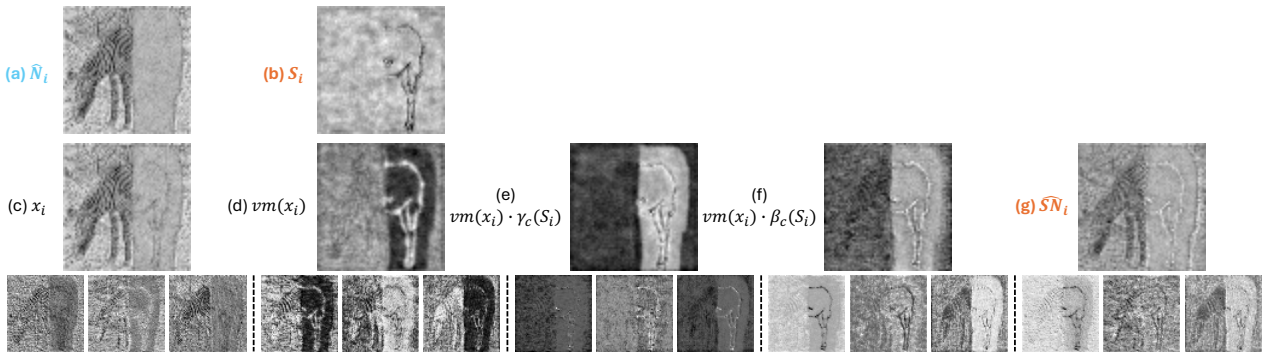



Figure 5.4: The visualization of features generated by the Sketch-guided Bidirectional Feature Interaction (SBFI) module, including: (a) masked-image-encoder-modulated noisy features $\hat{N}_i$, (b) sketch features $S_i$, (c, d) features from the context-aware feature fusion sub-module, and (e∼g) features from the sketch-guided affine transformation sub-module. Here, we use the scale factor $i = 1$ in the multi-scale SBFI module as an example to visualize the overall feature distribution by averaging all channels, as shown in the first two rows. Additionally, in the third row, we visualize the local distribution patterns by displaying the first three individual channels of the corresponding features in (c∼g).

where each pixel value ranges between $[0, 1]$ and indicates the degree to which the subsequent sketch-conditional affine transformation should be applied. The intermediate features $x_i$ and $vm$ produced by this sub-module are visualized in Fig. 5.4 (c) and (d), respectively. The visual mask highlights both the contours of uncorrupted object regions and the partial sketch details within the corrupted area to combine as a whole, guiding the subsequent affine transformation to refine feature interactions with sketch information. This step lays the groundwork for accurate sketch-based manipulation while preserving consistency with the uncorrupted visual context.

For an input batch $x_i \in \mathbb{R}^{b\times c\times h\times w}$, the sketch-conditional affine transformation is applied in a channel-wise manner after Group Normalization (GN), as follows:

$$\hat{x}_i^{bchw} = vm_{(h_j,w_j)}(\gamma_c(S_i)\tilde{x}_i^{bchw} + \beta_c(S_i)), \tag{5.8}$$

where $\tilde{x}_i^{bchw}$ is the GN-normalized features, and $\gamma_c(S_i)$ and $\beta_c(S_i)$ are two affine parameters learned from the guided sketches $S_i$ through the zero convolution layer to mitigate noise interference during the early training stages. The intermediate outputs $vm \cdot \gamma_c(S_i)$, $vm \cdot \beta_c(S_i)$, and the final sketch-guided feature $\hat{SN}_i$ are visualized in Fig. 5.4 (e), (f), and (g), respectively.

This sketch-conditional affine transformation modulates the influence of the partial sketch via two learnable affine parameters, while preserving sketch-guided consistency with uncorrupted

object cues through additional conditioning on the visual mask predicted by the preceding fusion module. The resulting sketch-guided noisy features $\hat{SN}_i$ are reintegrated into the frozen text-embedded Stable Diffusion model, as shown in the bottom right of Fig. 5.3. This integration guides the model to inpaint partially corrupted objects under sketch guidance, while also maintaining high-level semantic coherence through the frozen text-conditioned generative prior.

### 5.2.4 Dataset Preparation for Model Training

As existing datasets mainly focus on complete object occlusion and overlook scenarios involving partial degradation of objects, the proposed method targets the inpainting of partially corrupted objects, where the occlusion mask covers a significant portion of the object's semantic region and may also include background areas. This scenario presents a non-trivial challenge, as existing random masking strategies [24], [84] are inadequate for such partial occlusions. Specifically, random masks often cover background regions or only obscure small portions of the object, making it difficult to control the occlusion ratio between object and background areas. As a result, these strategies struggle to support object-level inpainting guided by text and sketches. To overcome this limitation, we propose a data preparation approach for partially occluded object scenes. Each generated sample includes a partial mask, partial sketch, partially occluded image, and corresponding textual description. As illustrated in Fig. 5.5, we use a simple object image to illustrate the data preparation process. This process consists of three steps: mask generation, partial masking, and partial sketch generation, which are described in detail below.

#### *Step 1: Mask Generation*

In the first step, enlarged instance masks are generated to cover the background regions. This process uses a mask dilation indicator $d \sim [0, D]$, which controls the degree of dilation applied to the object instance mask $m_0$ from the annotations. The dilation process is defined as:

$$m_d = Dilation(m_0, k_d), \tag{5.9}$$

where $k_d$ denotes the dilation kernel size. When $d = 0$, the mask remains unchanged as $m_0$. As $d$ increases, the mask $m_d$ gradually expands outward. At $d = D$, the mask $m_d$ becomes the bounding box of the instance object, losing specific shape information.

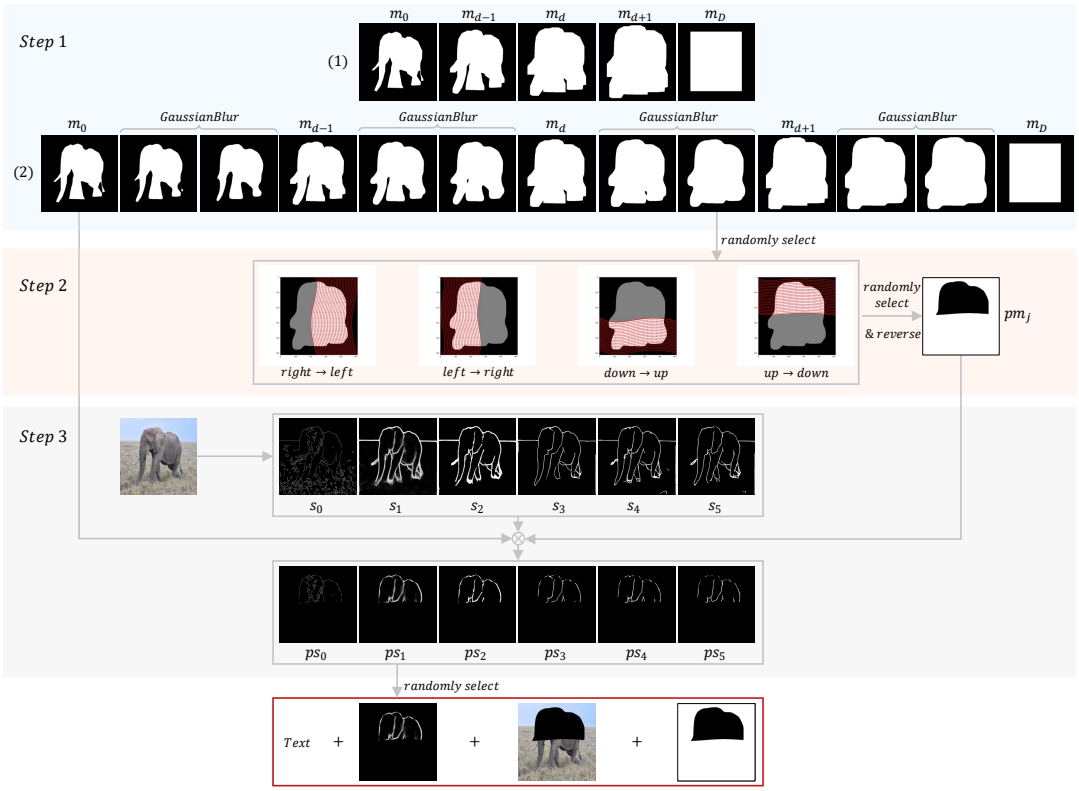



Figure 5.5: Data preparation process for constructing partially occluded object masks and corresponding partial sketches in three steps: Step 1 (Mask Generation), where Step 1 (1) performs mask dilation to expand the object boundaries and include non-object background, and Step 1 (2) smooths the edges between adjacent dilated masks; Step 2 (Partial Masking); and Step 3 (Partial Sketch Generation). The outputs from these steps are used to construct 4-tuple data samples for each image, where the accompanying text is directly taken from the original dataset annotations.

To alleviate the excessive expansion of originally sharp boundaries in $m_0$ caused by the dilation process, as shown in Fig. 5.5 Step 1 (1), Gaussian blur is applied between adjacent dilation masks $m_d$ and $m_{d+1}$ $(0 \leq d, d+1 \leq D)$, smoothing the edges and producing masks with varying levels of precision. This process is controlled by a mask blur indicator $s \sim [0, S]$ and is defined as:

$$m_s = GaussianBlur(m_d, m_{d+1}, k_s), \tag{5.10}$$

where $k_s$ is the Gaussian kernel size. When $s = 0$, $m_s$ corresponds to $m_d$. As $s$ increases, the mask becomes progressively smoother until $s = S$, where $m_s$ is equal to $m_{d+1}$. Examples of masks generated with varying $d$ and $s$ values are shown in Fig. 5.5, Step 1 (2). A mask is randomly selected from this set for the next step.

***Step 2: Partial Masking***

In the second step, a Bézier curve is randomly generated and used to scan the selected mask from Step 1 in one of four directions: right to left, left to right, down to up, or up to down. The scanning continues until the covered area meets a predefined percentage ($50\% \sim 60\%$ in our experiments) of the input mask's area. This process produces four partially occluded object masks. One of these masks is randomly selected and reversed to generate the partial mask $pm_j$ ($j \in \mathbb{Z}, 0 \leq j \leq 3$), as shown in Fig. 5.5, Step 2.

***Step 3: Partial Sketch Generation***

In the final step, six types of sketches are generated from the clean RGB image to support sketch manipulation with varying freehand styles during inference. As illustrated in Fig. 5.5, Step 3, the sketches are created as follows:

Sketch $s_0$: Extracted using the Canny algorithm [56].

Sketch $s_1$: Generated using PidiNet [146].

Sketch $s_2$: Created by applying hard threshold filtering to $s_1$.

Simplified Sketches $s_3, s_4, s_5$: Produced using the rough sketch simplification (RSS) algorithm [147], based on $s_1$ as input with three different initialization strokes.

The partial sketch $ps_i$ is calculated using the formula $(1 - pm_j) \otimes m_0 \otimes s_i$, where $i \in \mathbb{Z}$ and $0 \leq i \leq 5$, and $m_0$ represents the instance mask. This computation generates the partial sketch corresponding to the object within the partial mask $pm_j$. The goal of this process is to increase the diversity of sketch types during training, thus enhancing the model's generalizability to user-drawn sketch types during inference. Finally, a partial sketch is randomly selected and incorporated into the 4-tuple.

Using the above steps and annotations from CUB [148] and MSCOCO [149], we construct two customized datasets: CUB-sketch and MSCOCO-sketch, providing benchmarks for sketch-based partial object inpainting.

## 5.3 Experiments

We evaluate the effectiveness of the proposed method against state-of-the-art diffusion-based approaches. Extensive experimental results demonstrate the superiority of our pipeline in achieving fine-grained posture guidance using partial sketches for inpainting partially corrupted objects, while preserving sketch-guided consistency with uncorrupted object cues. Finally, we perform ablation studies to assess the contribution of each component within the proposed framework to the overall performance.

### 5.3.1 Experimental Setup

***1) Datasets***

We train the proposed pipeline using four-tuple data consisting of partially masked images, partial masks, partial sketches, and corresponding text descriptions. Existing diffusion-based inpainting methods guided by text or sketches do not specifically target the restoration of partially occluded objects. Moreover, the datasets they commonly use, such as CUB [148] and MSCOCO [149], lack the necessary partial masks and partial sketches required for this task. To generate the four-tuple data, we leverage the instance masks and text captions from the original datasets to construct two new datasets: CUB-sketch and MSCOCO-sketch, as described in Chapter 5.2.4.

The resulting CUB-sketch and MSCOCO-sketch datasets contain 11,788 and 30,809 four-tuple samples, respectively. These datasets, along with the associated code, are available at `https://github.com/yonglezhang95/PartiallyObjectInpainting`. Specifically, CUB-sketch includes 8,855 samples for training and 2,933 for testing, while MSCOCO-sketch comprises 20,526 training samples and 10,283 testing samples. The MSCOCO-sketch training set is derived from the MSCOCO training set, and its testing set is sourced from the MSCOCO validation set.

***2) Compared Methods***

We compare our pipeline with six state-of-the-art diffusion-based methods: SD-Inpainting [13], BrushNet [74], PowerPaint [137], MaGIC [78], ControlNet [39], and PowerPaint-ControlNet [137], the latter of which integrates the ControlNet adapter. The first three methods are text-guided inpainting approaches, while the latter three are sketch-guided inpainting methods based

on the pretrained text-to-image Stable Diffusion model.

To ensure fair comparison, we adopt the recommended hyperparameters provided in each method's official implementation and apply them consistently to our training and testing datasets. For SD-Inpainting, BrushNet, and PowerPaint, we use 3-tuple data samples comprising partial masks, partially masked images, and text prompts. For MaGIC, ControlNet, PowerPaint-ControlNet, and our proposed pipeline, we use 4-tuple data samples consisting of partial masks, partially masked images, text prompts, and partial sketches. All methods perform inference using 50 diffusion steps and a Classifier-Free Guidance (CFG) scale of 7.5, which is a standard setting in prior works [13], [137], [150].

#### *3) Evaluation Metrics*

We evaluate all methods using four widely adopted metrics: Aesthetic Score (AS) [151], [152], Fréchet Inception Distance (FID) [128], CLIP Score [153], and Learned Perceptual Image Patch Similarity (LPIPS) [129]. AS quantifies image quality based on human perception using a linear regression model trained on real image rating pairs. FID assesses the realism and visual consistency of the recovered object within the overall image distribution. CLIP Score evaluates the semantic alignment between the generated content and the input text prompt. LPIPS measures perceptual similarity by comparing deep features extracted from pretrained neural networks, providing a reliable estimate of how close the restored image is to the ground truth.

In addition to these quantitative metrics, we conduct a human study to assess the alignment between the guided sketch and the restored object within the damaged region. This evaluation reflects how well the model respects the sketch guidance and maintains visual consistency with the uncorrupted object regions during the inpainting process, which is difficult to capture through automated metrics alone.

#### *4) Parameter Setting*

We use the Adam optimizer with a learning rate of 0.00001 and a batch size of 16 to train our pipeline, based on the pre-trained Stable Diffusion v1.5. During training, to avoid overfitting and ensure exposure to different occlusion levels, each image is randomly assigned one of three mask types using a 6:3:1 ratio: partial masks covering 50% $\sim$ 60% of the target object, segmentation masks fully covering the object, or bounding-box masks covering the object and

its surrounding context. The same masking protocol is applied during evaluation to keep all methods comparable, providing diverse corruption conditions for assessing model robustness.

### 5.3.2 Quantitative and Qualitative Comparisons

#### *1) Quantitative Results*

Table 5.1 presents the numerical results of all methods evaluated on the CUB-sketch and MSCOCO-sketch test sets. The results across four metrics, AS, CLIP Score, FID, and LPIPS, consistently demonstrate the effectiveness of our pipeline in restoring partially corrupted objects. On the CUB-sketch test set, PowerPaint and ControlNet exhibit inferior performance

Table 5.1: Quantitative comparisons with state-of-the-art diffusion-based methods for partially corrupted object inpainting on the test sets of CUB-sketch and MSCOCO-sketch. ↑ indicates higher is better, and ↓ indicates lower is better. The best scores are marked in bold.

| Method | CUB-sketch | | | | MSCOCO-sketch | | | |
|---|---|---|---|---|---|---|---|---|
| | AS ↑ | CLIP Score ↑ | FID ↓ | LPIPS ($\times 10^2$) ↓ | AS ↑ | CLIP Score ↑ | FID ↓ | LPIPS ($\times 10^2$) ↓ |
| SD-Inpainting [13] | 5.77 | 29.01 | 8.21 | 10.01 | 5.64 | 25.29 | 4.87 | 12.06 |
| BrushNet [74] | 5.76 | 29.02 | 9.79 | 11.22 | 5.68 | 25.35 | 5.56 | 13.99 |
| PowerPaint [137] | 5.71 | 28.52 | 10.05 | 12.09 | 5.65 | 25.09 | 5.06 | 13.92 |
| MaGIC [78] | 5.79 | 28.67 | 8.83 | 8.72 | 5.59 | 25.78 | 4.90 | 11.52 |
| ControlNet [39] | 5.73 | 28.53 | 10.77 | 12.01 | 5.61 | 25.03 | 5.22 | 13.84 |
| PowerPaint-ControlNet [137] | 5.73 | 29.03 | 8.78 | 12.79 | 5.63 | 25.46 | 4.89 | 15.72 |
| Ours | **5.81** | **29.08** | **8.17** | **8.48** | **5.71** | **25.90** | **4.83** | **10.95** |

Table 5.2: Quantitative comparisons with state-of-the-art diffusion-based methods for fully corrupted object inpainting on the test sets of CUB-sketch and MSCOCO-sketch. ↑ indicates higher is better, and ↓ indicates lower is better. The best scores are marked in bold.

| Method | CUB-sketch | | | | | MSCOCO-sketch | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | AS ↑ | CLIP Score ↑ | FID ↓ | L-LPIPS ($\times 10^2$) ↓ | LPIPS ($\times 10^2$) ↓ | AS ↑ | CLIP Score ↑ | FID ↓ | L-LPIPS ($\times 10^2$) ↓ | LPIPS ($\times 10^2$) ↓ |
| SD-Inpainting [13] | 5.65 | 28.13 | 17.50 | 7.68 | 13.50 | 5.70 | 25.81 | 7.53 | 10.66 | 15.51 |
| BrushNet [74] | 5.90 | 28.64 | 19.21 | 7.90 | 14.18 | 5.68 | 25.85 | 9.24 | 11.54 | 17.48 |
| PowerPaint [137] | 5.82 | 28.24 | 18.38 | 7.44 | 12.85 | 5.66 | 25.72 | 8.38 | 11.82 | 17.24 |
| MaGIC [78] | **5.94** | 28.88 | 17.96 | 6.54 | **10.85** | 5.78 | 25.83 | **7.21** | 10.19 | **14.93** |
| ControlNet [39] | 5.78 | 28.51 | 16.95 | 6.36 | 13.32 | 5.71 | 25.78 | 7.46 | 11.09 | 16.72 |
| PowerPaint-ControlNet [137] | 5.81 | 28.27 | 18.02 | 7.23 | 14.06 | 5.65 | 25.73 | 8.01 | 11.36 | 18.59 |
| Ours | **5.94** | **29.01** | **16.74** | **6.33** | 10.89 | **5.80** | **25.99** | 7.48 | **9.97** | 15.35 |

across all four metrics. This can be attributed to their limited ability to model contextual

relationships between uncorrupted objects and corrupted regions. Specifically, ControlNet's inherent controllability, derived from Text-to-Image (T2I) tasks, makes it less effective for object restoration under partial occlusions, as it injects sketches independently, disregarding the ongoing inpainting process. Similarly, MaGIC's performance is surpassed by our method, which leverages multi-scale bidirectional feature interaction mechanisms. These mechanisms dynamically weight sketch guidance while adapting to uncorrupted object regions, mitigating the instability of MaGIC's indirect, gradient-based guidance. As a result, our approach achieves superior restoration outcomes.

On the MSCOCO-sketch test set, methods such as SD-Inpainting and BrushNet, which rely solely on text prompts without spatial sketch control, underperform compared to our pipeline. Our approach utilizes four-tuple inputs (corrupted image, text, partial sketch, and mask) integrated with a frozen text-embedded Stable Diffusion model, enabling finer object pose details through sketch guidance and pretrained textual semantic priors. In contrast, MaGIC, ControlNet, and PowerPaint-ControlNet produce suboptimal results, likely due to their lack of specialized modules for perceiving visible object contexts and facilitating tailored restoration of partially corrupted scenes.

Additionally, Table 5.2 compares the performance of our method with existing diffusion-based approaches for fully corrupted object restoration on the CUB-sketch and MSCOCO-sketch test sets. In these cases, objects are entirely masked using either instance segmentation masks or bounding box masks, with a ratio of 8:2 in the test set. We introduce a Local-LPIPS (L-LPIPS) metric to measure perceptual similarity between the fully inpainted object and the ground truth instance. Among the compared methods, our approach achieves competitive qualitative metrics, further validating that bidirectional interaction between guidance information and inpainting features during the denoising process enhances restoration performance compared to independent integration of guidance information.

#### *2) Qualitative Results*

Fig. 5.6 presents qualitative results for six comparison methods. Our pipeline generates high-fidelity visual details in restored objects, achieving strong alignment with both the sketch prompt and textual semantics.

Methods relying solely on text prompts, such as SD-Inpainting, BrushNet, and PowerPaint,

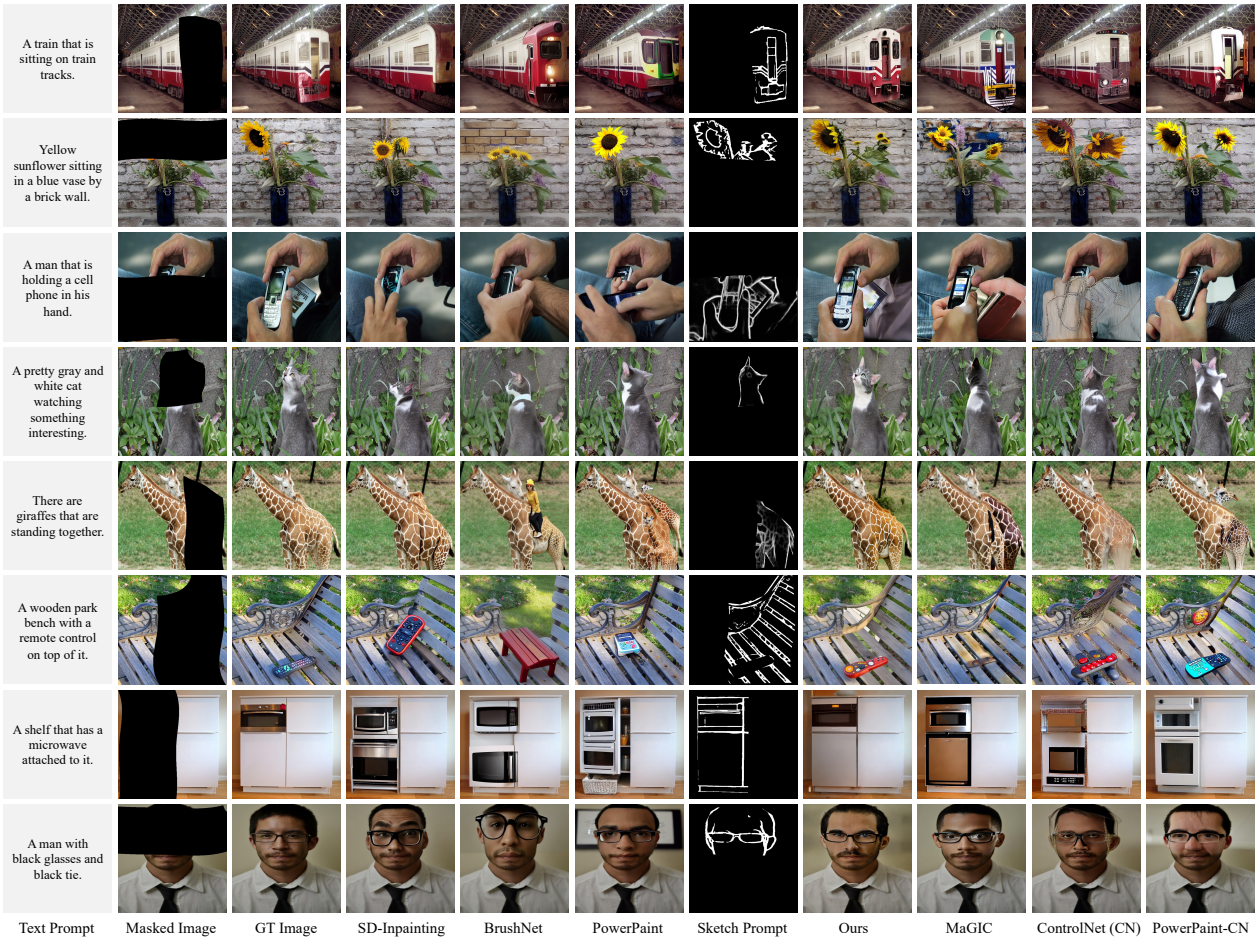



Figure 5.6: Qualitative comparison of our method with SD-Inpainting [13], BrushNet [74], PowerPaint [137], MaGIC [78], ControlNet [39], and PowerPaint-ControlNet [137] for partially corrupted object inpainting on the CUB-sketch and MSCOCO-sketch test images. Among these methods, MaGIC, ControlNet, and PowerPaint-ControlNet utilize both text and sketch prompts, while the other three rely solely on the text prompt.

produce semantically meaningful object restorations but struggle with visual consistency and exhibit arbitrary spatial postures. For instance, in Fig. 5.6, the restored train heads (first row), hands (third row), and cat heads (fourth row) display inconsistent spatial structures relative to uncorrupted object regions. These methods often fail to control object poses in inpainted regions, highlighting the limitations of text-based guidance in achieving fine-grained posture accuracy.

In contrast, MaGIC, ControlNet, and PowerPaint-CN incorporate the same sketch prompt as our method but produce inpainted results with inconsistent object postures relative to the sketch. For example, the sunflower shape (second row) and the man's hand (third row) in

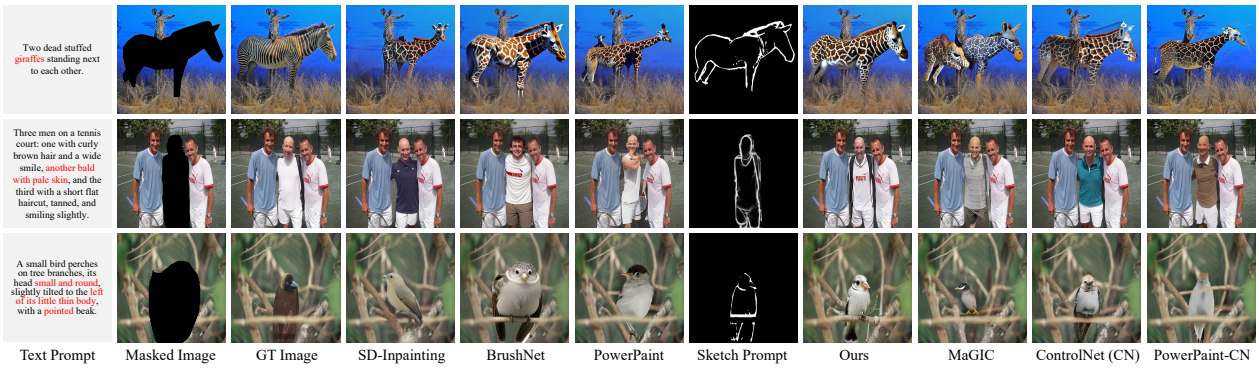



Figure 5.7: Qualitative comparison of our method with SD-Inpainting [13], BrushNet [74], PowerPaint [137], MaGIC [78], ControlNet [39], and PowerPaint-ControlNet [137] for fully corrupted object inpainting on the CUB-sketch and MSCOCO-sketch test images. Among these methods, MaGIC, ControlNet, and PowerPaint-ControlNet utilize both text and sketch prompts, while the other three rely solely on the text prompt.

Fig. 5.6 reveal difficulties in controlling the completion of partially corrupted objects using sketch prompts. Our method, however, integrates a masked image encoder to identify corrupted regions and uncorrupted object contexts, a sketch-conditional encoder with multi-scale bidirectional feature fusion modules to ensure precise sketch integration and consistency with uncorrupted regions, and a pretrained Stable Diffusion model for robust text-guided priors. This results in high-fidelity object completion with both visual and semantic coherence.

Fig. 5.7 provides additional qualitative comparisons for fully corrupted object inpainting. Our method consistently outperforms existing approaches, demonstrating superior visual alignment with the guiding sketch prompt (e.g., the restored bird in the third row) and improved consistency with undamaged contexts. Text-guided methods, such as SD-Inpainting, BrushNet, and PowerPaint, fail to achieve pixel-level accuracy in object postures, even when provided with detailed text prompts, such as "its head slightly tilted to the left of its little thin body" for the bird inpainting example (third row). For instance, the inpainted bird's head in SD-Inpainting and PowerPaint tilts to the right, contrary to the prompt's description.

### 5.3.3 Comparison Between Text-Only and Text+Sketch Guidance

In Chapter 5.3.2, we demonstrated that text prompts alone are insufficient for guiding diffusion-based models to achieve pixel-level accuracy in object pose inpainting. Here, we further compare the visual outcomes of inpainting guided solely by text prompts with those guided by both text

and sketch prompts, focusing on our method, MaGIC, and ControlNet.

In the text-only setting, the sketch prompt is replaced with a completely black image. The left side of Fig. 5.8 displays the inpainting results for each method using only text prompts. For example, in the first row, ControlNet fails to capture the structural description "both ears upright" from the text prompt. Similarly, in the second row, both MaGIC and ControlNet fail to depict the "upright stems" specified in the text. Although our method also struggles to fully reconstruct the spatial structure implied by the text, these results highlight the inherent limitations of text prompts, which provide high-level semantic cues but lack pixel-level spatial specificity.

When sketch prompts are incorporated, as shown on the right side of Fig. 5.8, our method achieves significantly more accurate pixel-level pose reconstructions, closely aligning with the guided sketch structure compared to MaGIC and ControlNet. This improvement arises from our sketch-guided bidirectional feature interaction mechanism, which ensures consistency between the sketch guidance and the inpainting process by adapting to the surrounding object context—a capability absent in MaGIC and ControlNet.

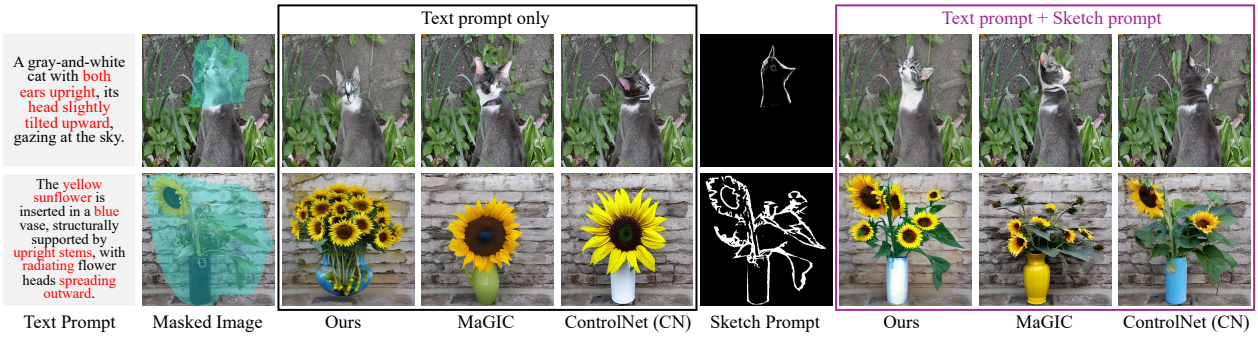



Figure 5.8: Qualitative comparison of our method with MaGIC [78] and ControlNet [39] for object inpainting under text-only guidance and combined text and sketch prompts.

### 5.3.4 Subjective Assessment via User Study

We conducted two user studies to evaluate user preference and sketch alignment score metrics. For the user preference assessment, we randomly selected 35 partially corrupted object images from the MSCOCO-sketch test set and inpainted them using six competing methods and our proposed pipeline. Twenty participants were recruited, half of whom were image processing majors, while the other half were PhD students from other disciplines. Each participant was

Table 5.3: User Study Results for User Preference and Sketch Alignment Score. The study involved 20 participants. User Preference assesses the naturalness and textual semantic fidelity of inpainted object images, while Sketch Alignment Score evaluates the alignment of inpainted object regions with the guiding sketch and their consistency with uncorrupted regions. ↑ indicates higher is better.

| Method | User Preference (%) ↑ | Sketch Alignment Score ↑ (1 = poor, 5 = best) |
|---|---|---|
| SD-Inpainting [13] | 9.83 | – |
| BrushNet [74] | 23.33 | – |
| PowerPaint [137] | 10.33 | – |
| MaGIC [78] | 14.66 | 3.7366 |
| ControlNet [39] | 9.83 | 3.7033 |
| PowerPaint-ControlNet [137] | 10.66 | 2.7133 |
| Ours | **59.33** | **4.0700** |

presented with the corrupted images, their corresponding inpainted results, and text prompts, and asked to select one or more inpainted images they perceived as natural with high semantic fidelity. As shown in the User Preference column of Table 5.3, our method's inpainted results were preferred in 59.33% of the selections. For the sketch alignment score assessment, we randomly selected 15 partially corrupted object images from the MSCOCO-sketch test set, each accompanied by four partial sketches at different scales as guides. Twenty participants were presented with inpainting results from competing sketch-guided methods and our pipeline, along with the guiding sketches, and asked to rate how well the inpainted object regions aligned with the sketch and remained consistent with the uncorrupted regions, using a score from 1 (poor alignment) to 5 (best alignment). The Sketch Alignment Score column in Table 5.3 indicates that participants favored the inpainting results of our method.

### 5.3.5 Model Flexibility with Diverse Text Prompts and Various Sketches

Fig. 5.2 (c)∼(e) present the results generated by our pipeline using the same text prompt but different sketch prompts. Fig. 5.9 showcases more controllable object inpainting, conditioned on

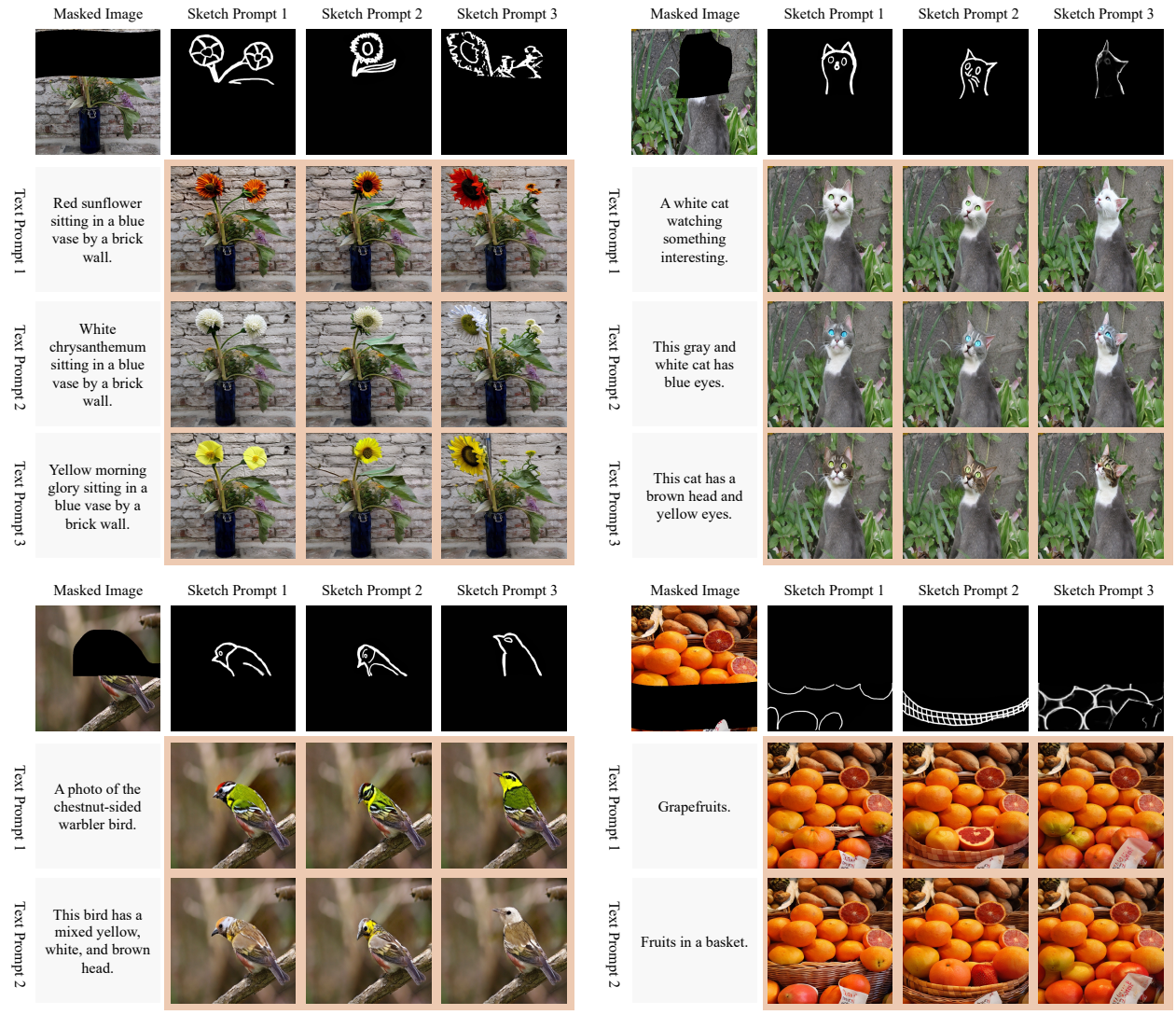



Figure 5.9: Controllable object inpainting results generated by our pipeline, conditioned on different combinations of sketch and text prompts, demonstrating high-fidelity outcomes.

various combinations of sketch and text prompts. Our pipeline produces high-fidelity inpainted results that maintain visual consistency with the sketch prompt and semantic consistency with the text prompt.

### 5.3.6 Ablation Studies

We conducted ablation studies to assess the effect of different component designs in our pipeline, which comprises three main components: the Masked Image Encoder (MIE), the Sketch-Conditional Encoder (SCE), and the Sketch-guided Bidirectional Feature Interaction (SBFI) module. We verify corresponding effects by examining combinations of these components at each stage. We evaluated their effects by testing different combinations of these components

at each stage. Table 5.4 presents the numerical results of these models on the CUB-sketch test set for partially corrupted object inpainting.

In Table 5.4, models that include the SCE but exclude the SBFI module integrate sketch information into the Frozen Stable Diffusion model via element-wise addition, replacing the SBFI module. Similarly, for models with the SCE but without the MIE, we expanded the input channel dimension of the Stable Diffusion model to incorporate the masked image and mask. All ablation models were trained under the same training configuration as our full pipeline.

Table 5.4: Quantitative comparisons from ablation studies on different component designs in our pipeline. ↑ indicates higher is better, and ↓ indicates lower is better.

| Model | CUB-sketch | | |
|---|---|---|---|
| | AS ↑ | CLIP Score ↑ | FID ↓ |
| Frozen SD+SCE | 5.46 | 27.65 | 13.94 |
| Frozen SD+SCE+SBFI | 5.54 | 27.85 | 10.88 |
| Frozen SD+MIE | 5.76 | 28.92 | 9.91 |
| Frozen SD+MIE+SCE | 5.78 | 29.03 | 8.48 |
| Frozen SD+MIE+SCE+SBFI(Ours) | **5.81** | **29.08** | **8.17** |

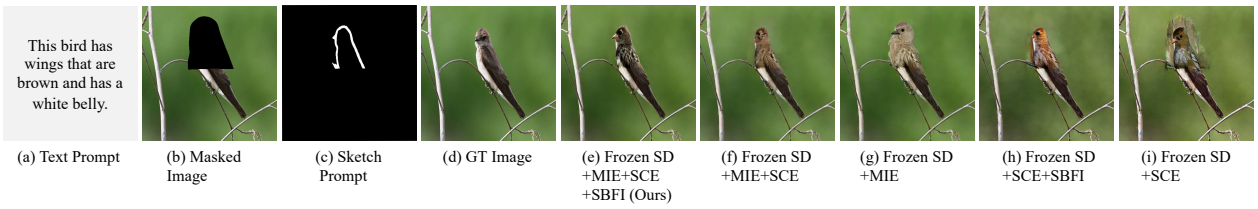



Figure 5.10: Qualitative comparisons from ablation studies on different component configurations in our pipeline. Frozen SD denotes the pre-trained text-to-image Stable Diffusion model with parameters frozen. Each panel shows the generated image under different model variants to illustrate the effect of each component.

### *1) Effect of Sketch-guided Bidirectional Feature Interaction (SBFI)*

The quantitative results in Table 5.4 demonstrate that incorporating the SBFI module significantly improves model performance. Specifically, the model "Frozen SD + SCE" underperforms

compared to "Frozen SD + SCE + SBFI" in terms of AS, CLIP Score, and FID. Similarly, the model "Frozen SD + MIE + SCE" shows inferior performance than "Frozen SD + MIE + SCE + SBFI (Ours)." The qualitative comparisons in Fig. 5.10 further support these findings: the model "Frozen SD + MIE + SCE + SBFI (Ours)" generates a sharper and more accurate outline of the bird's head than "Frozen SD + MIE + SCE," and more visual artifacts appear in the corrupted object region produced by the model "Frozen SD + SCE" compared to "Frozen SD + SCE + SBFI."

These quantitative and qualitative results indicate that the SBFI module effectively incorporates sketch prompt information into the object restoration process while preserving visual consistency with the uncorrupted parts of the object. The performance gain is attributed to the module's ability to fuse multi-scale sketch-derived and corrupted-object-modulated features, ensuring spatial alignment for subsequent sketch integration based on affine transformation.

#### *2) Effect of Masked Image Encoder (MIE)*

As shown in Table 5.4, the model "Frozen SD + MIE + SCE" achieves better results than "Frozen SD + SCE" across all three evaluation metrics. Likewise, "Frozen SD + MIE + SCE + SBFI" outperforms "Frozen SD + SCE + SBFI." These findings suggest that the MIE plays a crucial role in generating an initial visual representation that clearly distinguishes between corrupted and uncorrupted regions. This region-aware information provides multi-scale spatial object context and mask localization, enabling the sketch prompt to be better adapted to the uncorrupted object content. The differences in visual quality are further illustrated in Fig. 5.10, panels (f) and (i), and (e) and (h), respectively.

#### *3) Effect of Sketch-Conditional Encoder (SCE)*

The fourth and fifth columns of Table 5.4 show that adding the SCE component to the model "Frozen SD + MIE" leads to performance improvements. The SCE extracts multi-scale sketch features, learning different levels of representative information from the input sketch. Furthermore, incorporating the SBFI module into "Frozen SD + MIE + SCE" yields even better results, as seen in the final (sixth) row of Table 5.4. This underscores the importance of effectively integrating sketch information into the corrupted object regions while maintaining consistency with the uncorrupted parts—both of which are well achieved by the proposed SBFI module.

Qualitative results in Fig. 5.10, panels (f) and (g), show that without the SCE module, the model "Frozen SD + MIE" generates a larger bird head with an unstable posture. In contrast, the full model "Frozen SD + MIE + SCE + SBFI" demonstrates finer control over the object's shape and posture than "Frozen SD + MIE + SCE", even when guided by a simplified hand-drawn sketch, as shown in Fig. 5.10, panels (e) and (f).

### *4) Inpainting Guided by Prompts from Abstract to Detailed*

The object inpainting results in Fig. 5.11 reveal a key trend: as the sketch prompt transitions from abstract to clear and the text prompt shifts from a broad to a more detailed description of the corrupted area, the inpainted objects exhibit increased visual and semantic consistency.

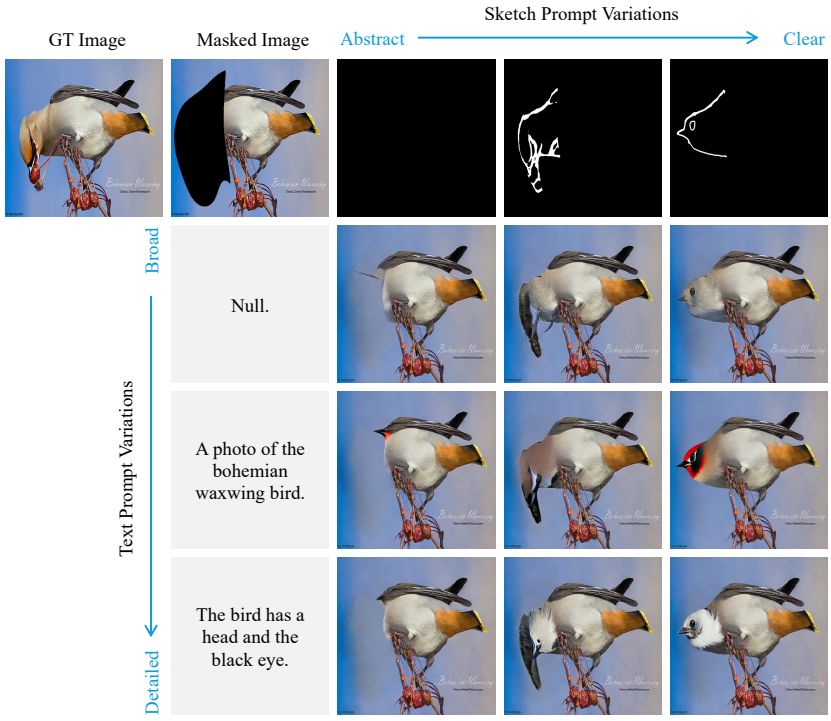



Figure 5.11: Object inpainting results from our pipeline under varying levels of sketch and text prompt specificity. Sketch prompts range from abstract (a completely black input with no sketch) to clear (a precise outline of a bird's head). Text prompts range from broad (an empty prompt) to detailed (a well-defined text prompt describing the bird's head).

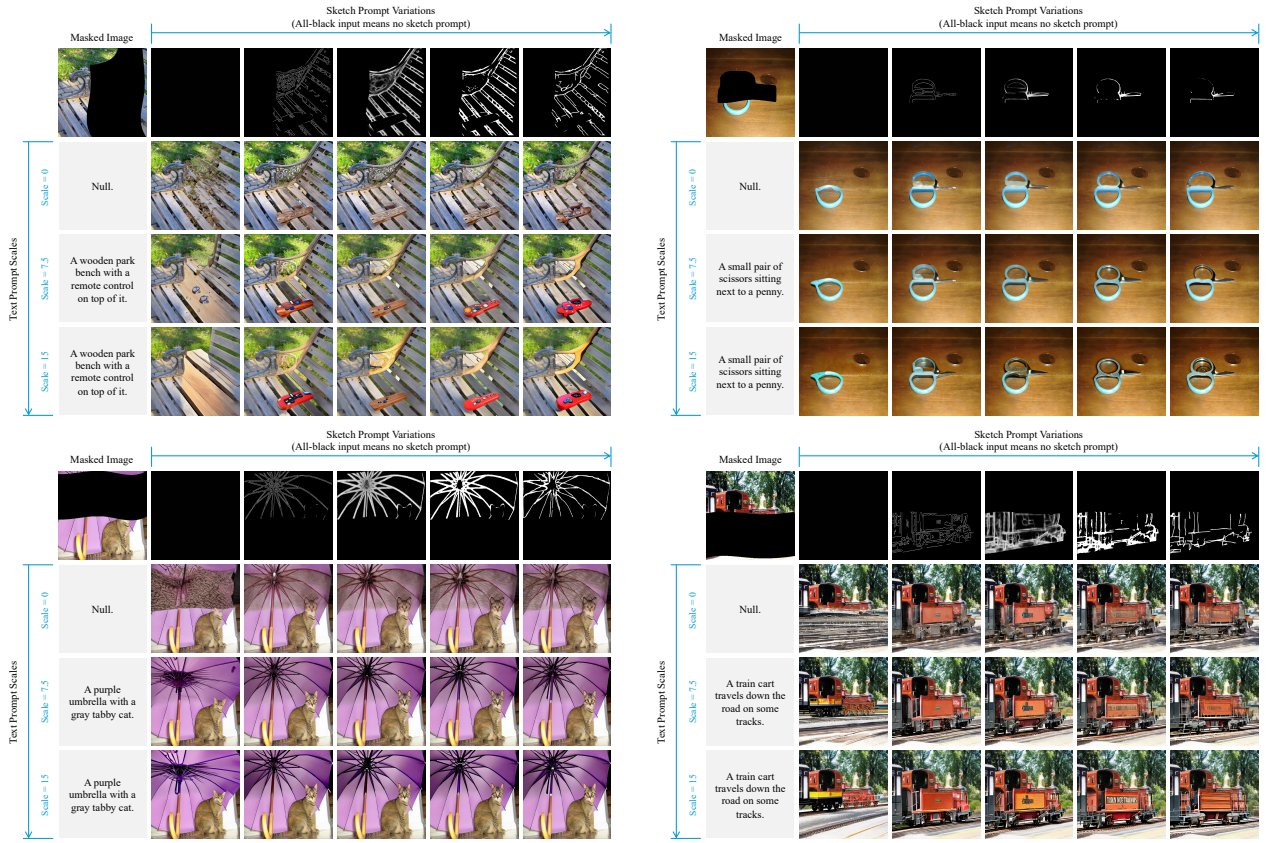



Figure 5.12: Object inpainting results generated by our pipeline under various combinations of sketch prompt variations and text prompt scales in partially masked images. [Note: The sketch prompt is an all-black input, meaning no sketch prompt is used. The text prompt scale is 0, meaning the pipeline ignores the text prompt (i.e., with null text guidance) and generates completions solely based on its unconditional prior.]

### *5) Object Inpainting under Variations in Sketch Prompts and Scaling of Text Prompts*

We experiment with five different sketch prompts, including an all-black input image (indicating the absence of a sketch) and four types of sketches generated in Step 3 of Chapter 5.2.4 (after removing two simplified sketches). These variations define a series of sketch prompt conditions. Simultaneously, we regulate the influence of the text prompt using the Classifier-Free Guidance (CFG) scale [150], a technique in pre-trained Stable Diffusion models [13] that adjusts text guidance strength and image fidelity. It operates by blending noise predictions conditioned and unconditioned on text, as defined by the equation:

$$\epsilon_{guided} = \epsilon_{unconditional} + s \cdot (\epsilon_{conditional} - \epsilon_{unconditional}), \tag{5.11}$$

where $s$ represents the CFG scale. We employ three settings:

CFG scale = 0: The model disregards the text prompt, generating images solely based on its learned unconditional prior, resulting in outputs uninfluenced by textual descriptions.

CFG scale = 7.5: A commonly adopted setting in our pipeline and comparison methods, balancing diversity and text adherence. The generated image aligns well with the prompt while maintaining naturalness.

CFG scale = 15: The model strictly follows the text prompt, which can introduce artifacts or unnatural details due to excessive reliance on textual guidance at the expense of diversity and realism.

By controlling both the text prompt scales and sketch prompt variations, we present object inpainting results in Fig. 5.12. These results demonstrate that the proposed model maintains robustness across different sketches, particularly at a text prompt scale of 7.5, effectively guiding spatial posture generation in partially corrupted object regions. Additionally, when no sketch is provided, the model relies solely on the text prompt scale, which leads to arbitrary spatial structures and unrealistic completions that fail to seamlessly integrate with the uncorrupted object regions.

***6) Limitations***

We examine scenarios where the content of the guiding sketch is inconsistent with or contradicts the description provided by the text prompt. For example, if the sketch depicts a cow's head while the text prompt describes a different object, such as a dog, our pipeline struggles to produce a coherent completion, as illustrated in Fig. 5.13. Such inconsistencies lead the model to generate unrealistic and incoherent results.

In addition, when the sketch prompt contains a complex structure with noise or ambiguity to guide object inpainting, as in the sketch shown in Fig. 5.14, our method fails to reconstruct a clear structural layout of the viaduct, even though the result still outperforms MaGIC, ControlNet, and PowerPaint-CN to some extent. This limitation arises because our method, like MaGIC, ControlNet, and PowerPaint-CN, does not have the ability to correct poor-quality sketch inputs.

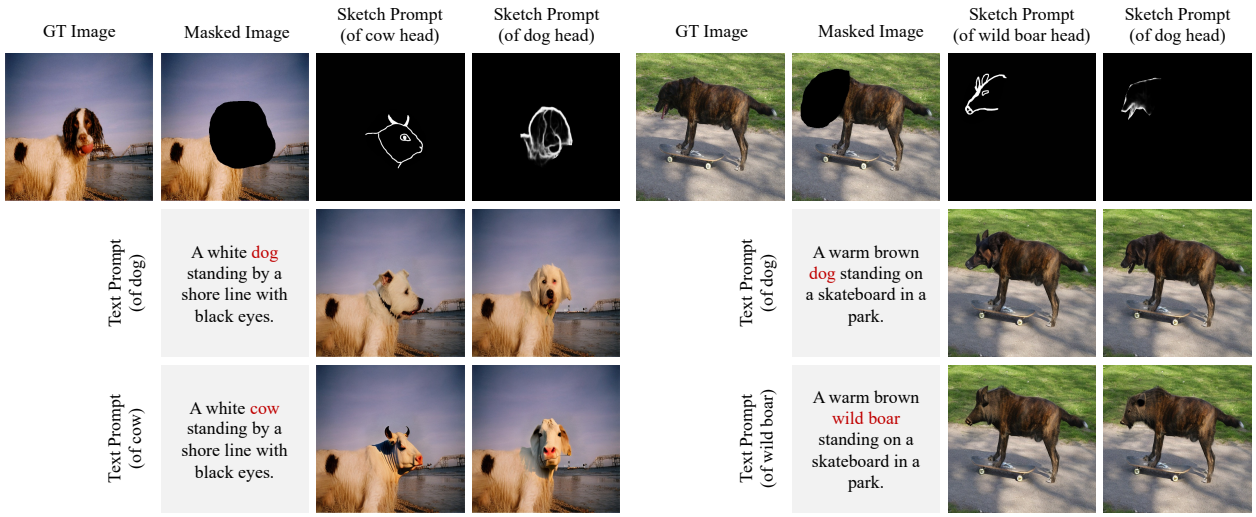



Figure 5.13: Object inpainting results generated by our pipeline in scenarios where the sketch prompt is inconsistent with the text prompt. For instance, the sketch may depict a cow's head, while the text prompt instead describes a different object, such as a dog, leading to incoherent completions.

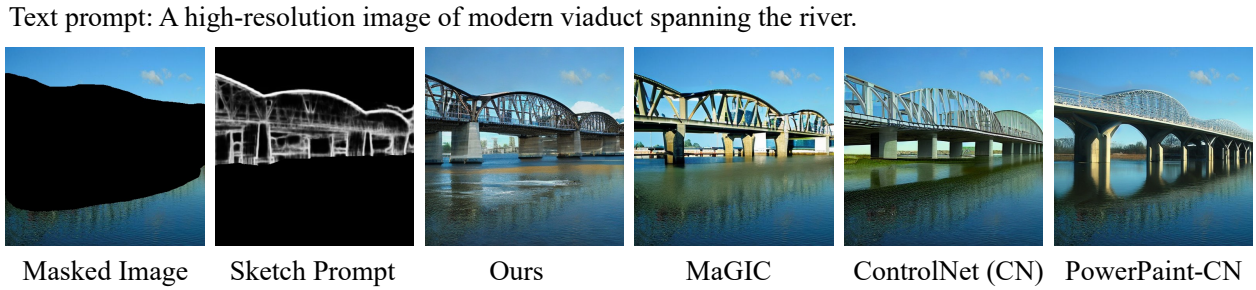



Figure 5.14: Inpainting result using a complex and noisy sketch prompt to guide the reconstruction of a viaduct, where existing methods fail to recover a clear structure.

## 5.4 Summary

In this chapter, we propose a novel pipeline that utilizes partial sketches as visual control for inpainting partially corrupted objects within a frozen text-guided Stable Diffusion model. Our pipeline integrates three key components: a Masked Image Encoder that incorporates masks and uncorrupted contexts into the denoising latent process, a Sketch-Conditional Encoder that extracts multi-scale sketch features, and a core multi-scale Sketch-guided Bidirectional Feature Interaction (SBFI) module that fuses sketch-derived features with noisy features modulated by uncorrupted contexts. This design ensures consistent sketch-based control and enhances visual-semantic consistency with uncorrupted regions during denoising. Extensive experiments

on the CUB-sketch and MSCOCO-sketch datasets demonstrate the superior performance of our approach through both quantitative and qualitative results.

# Chapter 6

# Conclusions and Future Work

## 6.1 Conclusion

Image inpainting has been widely studied during the past decade, evolving from early GAN-style (i.e., using generative adversarial networks) methods to the current Diffusion-style (i.e., using Diffusion Models) approaches. An advanced image inpainting algorithm must be capable of addressing arbitrary corruptions in images. To develop such robust inpainting architectures, this thesis tackles three key challenges: 1) restoring multiple corrupted semantic regions in GAN-style inpainting to achieve semantic fidelity; 2) reconstructing complex structures and reasonable semantics in GAN-style inpainting, guided by structural and semantic information (e.g., edges and semantic segmentation maps) for consistent results; and 3) recovering partially occluded objects in Diffusion-style inpainting, overcoming limitations in spatial reasoning for such cases. This thesis discusses the shortcomings of existing methods in addressing these challenges and proposes novel approaches from three perspectives: semantically guided GAN-style inpainting, consistent GAN-style inpainting with multiple guidance sources, and spatially reasonable Diffusion-style inpainting of partially occluded objects.

Chapter 3 explores semantically guided GAN-style inpainting through dual-task co-optimization using the proposed mutual generator. This generator comprises a shared encoder and mutual decoders designed to capture the interdependencies between image texture restoration and semantic segmentation guidance. Bidirectional Cross-domain Feature DeNormalization modules are introduced within the two decoders to hierarchically model segmentation-guided texture (ST) generation and texture-guided segmentation (TS) generation, enhancing semantic guid-

ance for inpainting. Additionally, the Mutual Dual-task Generator, supported by an Adaptive Attention Fusion module, improves inpainting performance by learning semantic-affinity and global-context texture consistencies for inpainted textures of missing regions. Extensive experiments validate the effectiveness of these proposed methods.

Chapter 4 focuses on achieving consistent GAN-style image inpainting through multi-modal collaboration, drawing inspiration from pre- and cross-perception processes in human drawing. A pre-perceptual transformer block is proposed to emulate the pre-perception process, learning contextual dependencies across three modalities—image edges, semantic segmentation, and texture information—independently. A mixing activation gating mechanism is embedded to strengthen feature representations for each modality. Furthermore, a cyclic cross-perceptual interaction simulates the collaborative cross-perception process, modeling the interplay among the three modalities and refining their details progressively to ensure inpainting consistency.

Chapter 5 addresses the spatial reasoning limitations in Diffusion-style inpainting for partially occluded objects. A novel pipeline is proposed, leveraging partial sketches as spatial and visual aids within a frozen text-guided Stable Diffusion model. This pipeline integrates three components: a Masked Image Encoder that incorporates masks and uncorrupted contexts into the denoising latent process, a Sketch-Conditional Encoder that extracts multi-scale sketch features, and a multi-scale Sketch-guided Bidirectional Feature Interaction (SBFI) module that fuses sketch-derived features with noisy features modulated by uncorrupted contexts. This approach ensures precise sketch-based control and maintains visual-semantic consistency with uncorrupted regions during denoising. Extensive experiments on the newly introduced CUB-sketch and MSCOCO-sketch datasets demonstrate superior performance through quantitative and qualitative results.

The research presented in Chapters 3, 4, and 5 is supported by peer-reviewed publications in leading journals. All related publications are listed in the List of Publications section of this thesis.

## 6.2 Future Work

The advent of diffusion models and large language models (LLMs) has transformed the field of image generation, providing powerful tools for a wide range of image editing tasks. These

advancements have also opened up promising new research directions and challenges in the domain of image inpainting. While current methods have achieved significant success, several critical issues remain to be explored to further enhance the effectiveness, efficiency, and controllability of image inpainting systems. The following directions outline some of the most pressing and promising areas for future work:

***1) Improving Semantic Alignment Between Textual Guidance and Visual Context***

In current diffusion-style image inpainting pipelines, text prompts are often encoded using models such as CLIP text encoders, which generate high-level semantic representations that guide the inpainting of missing or occluded object regions. While such guidance is effective for hallucinating semantically plausible content in severely damaged regions, a major limitation lies in the lack of interaction between the encoded textual features and the uncorrupted visual context. This disjointedness may lead to semantic inconsistencies, where the synthesized content does not harmonize well with the surrounding regions. Future work could explore cross-modal feature fusion techniques that tightly integrate textual semantics with visual cues from the unoccluded areas, leading to more coherent and context-aware inpainting results.

***2) Reducing Inference Cost for Lightweight Inpainting Applications***

Despite their remarkable performance, diffusion models typically require numerous inference steps (e.g., 50 or more) to produce high-quality inpainted images. This poses a significant challenge for real-time or resource-constrained applications, such as mobile devices or interactive tools. Future research could focus on accelerating inference through techniques such as distillation, early stopping, progressive refinement, or hybrid generation strategies that combine the strengths of deterministic and probabilistic methods. Developing lightweight diffusion architectures specifically optimized for inpainting could also help strike a better balance between quality and efficiency.

***3) Balancing Multi-Modal Conditioning in Diffusion-Based Inpainting***

As diffusion models evolve toward multi-modal conditional generation, researchers have begun integrating diverse forms of guidance—such as text prompts, sketches, reference images, and color hints—to control the inpainting process. However, determining how to balance the influence of each modality remains an open problem. Overemphasis on one modality can suppress

valuable signals from others, potentially leading to suboptimal or biased results. Future studies should investigate adaptive guidance weighting mechanisms, such as attention-based gating, learnable fusion strategies, or user-controllable sliders, to dynamically regulate the impact of each input modality during generation.

***4) Enabling Real-Time Interactive and Iterative Inpainting***

Although diffusion-based models now support highly customizable inpainting through flexible condition inputs, the lack of real-time interaction limits their practical utility. In many scenarios, the initial inpainting result may not align with user intent due to ambiguous prompts or imperfect guidance information. Therefore, enabling interactive, iterative inpainting workflows—where users can adjust prompts, sketch over areas, or provide feedback to refine results in real time—is an important direction for future development. This will require advances in user intent modeling, interactive UI design, and fast feedback loops within diffusion pipelines to support dynamic guidance refinement and on-the-fly correction of generation errors.

In summary, the future of image inpainting lies in improving generation quality while enhancing semantic alignment, computational efficiency, multi-modal integration, and user interactivity. By exploring these future directions, researchers can build inpainting systems that are more intelligent, controllable, and responsive to real-world needs.

# Bibliography


[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 417–424.

[2] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," *IEEE transactions on image processing*, vol. 10, no. 8, pp. 1200–1211, 2001.

[3] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on image processing*, vol. 13, no. 9, pp. 1200–1212, 2004.

[4] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra, "Texture optimization for example-based synthesis," in *ACM Siggraph 2005 Papers*, 2005, pp. 795–802.

[5] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, p. 24, 2009.

[6] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[7] C. Guillemot and O. Le Meur, "Image inpainting: Overview and recent advances," *IEEE signal processing magazine*, vol. 31, no. 1, pp. 127–144, 2013.

[8] T. Ružić and A. Pižurica, "Context-aware patch-based image inpainting using markov random field modeling," *IEEE transactions on image processing*, vol. 24, no. 1, pp. 444–456, 2014.

[9] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2536–2544.

[10] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5505–5514.

[11] D. Ding, S. Ram, and J. J. Rodríguez, "Image inpainting using nonlocal texture matching and nonlinear filtering," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1705–1719, 2018.

[12] M. Zhu et al., "Image inpainting by end-to-end cascaded refinement with mask awareness," *IEEE Transactions on Image Processing*, vol. 30, pp. 4855–4866, 2021.

[13] 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*, 2022, pp. 10 684–10 695.

[14] H. Wang, Q. Li, and Q. Zou, "Inpainting of dunhuang murals by sparsely modeling the texture similarity and structure continuity," *Journal on Computing and Cultural Heritage (JOCCH)*, vol. 12, no. 3, pp. 1–21, 2019.

[15] X. Yang and S. Wang, "Dunhuang mural inpainting in intricate disrepaired region based on improvement of priority algorithm," *Journal of Computer-Aided Design & Computer Graphics*, vol. 23, no. 2, pp. 284–289, 2011.

[16] C. Qin, Z. He, H. Yao, F. Cao, and L. Gao, "Visible watermark removal scheme based on reversible data hiding and image inpainting," *Signal Processing: Image Communication*, vol. 60, pp. 160–172, 2018.

[17] A. Mosleh, N. Bouguila, and A. B. Hamza, "Automatic inpainting scheme for video text detection and removal," *IEEE Transactions on Image processing*, vol. 22, no. 11, pp. 4460–4472, 2013.

[18] D. Winter, M. Cohen, S. Fruchter, Y. Pritch, A. Rav-Acha, and Y. Hoshen, "Objectdrop: Bootstrapping counterfactuals for photorealistic object removal and insertion," in *European Conference on Computer Vision*, Springer, 2024, pp. 112–129.

[19] 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.

[20] W. Sun, X.-M. Dong, B. Cui, and J. Tang, "Attentive eraser: Unleashing diffusion model's object removal potential via self-attention redirection guidance," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, 2025, pp. 20 734–20 742.

[21] I. J. Goodfellow et al., "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[22] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–14, 2017.

[23] Y. Song, C. Yang, Y. Shen, P. Wang, Q. Huang, and C.-C. J. Kuo, "Spg-net: Segmentation prediction and guidance network for image inpainting," *arXiv preprint arXiv:1805.03356*, 2018.

[24] 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*, 2019, pp. 4471–4480.

[25] L. Liao, J. Xiao, Z. Wang, C.-W. Lin, and S. Satoh, "Guidance and evaluation: Semantic-aware image inpainting for mixed scenes," in *European conference on computer vision*, Springer, 2020, pp. 683–700.

[26] J. Li, N. Wang, L. Zhang, B. Du, and D. Tao, "Recurrent feature reasoning for image inpainting," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 7760–7768.

[27] H. Liu et al., "Deflocnet: Deep image editing via flexible low-level controls," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 765–10 774.

[28] L. Liao, J. Xiao, Z. Wang, C.-W. Lin, and S. Satoh, "Image inpainting guided by coherence priors of semantics and textures," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 6539–6548.

[29] X. Guo, H. Yang, and D. Huang, "Image inpainting via conditional texture and structure dual generation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 14 134–14 143.

[30] W. Quan, R. Zhang, Y. Zhang, Z. Li, J. Wang, and D.-M. Yan, "Image inpainting with local and global refinement," *IEEE Transactions on Image Processing*, vol. 31, pp. 2405–2420, 2022.

[31] R. Zhang, W. Quan, Y. Zhang, J. Wang, and D.-M. Yan, "W-net: Structure and texture interaction for image inpainting," *IEEE Transactions on Multimedia*, vol. 25, pp. 7299–7310, 2022.

[32] 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*, 2022, pp. 10 758–10 768.

[33] Y. Yu, D. Du, L. Zhang, and T. Luo, "Unbiased multi-modality guidance for image inpainting," in *European Conference on Computer Vision*, Springer, 2022, pp. 668–684.

[34] J. Jain, Y. Zhou, N. Yu, and H. Shi, "Keys to better image inpainting: Structure and texture go hand in hand," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2023, pp. 208–217.

[35] Y. Deng, S. Hui, S. Zhou, W. Huang, and J. Wang, "Context adaptive network for image inpainting," *IEEE Transactions on Image Processing*, vol. 32, pp. 6332–6345, 2023.

[36] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[37] 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*, 2022, pp. 11 461–11 471.

[38] S. Xie, Z. Zhang, Z. Lin, T. Hinz, and K. Zhang, "Smartbrush: Text and shape guided object inpainting with diffusion model," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 22 428–22 437.

[39] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 3836–3847.

[40] M. T. Chiu et al., "Brush2prompt: Contextual prompt generator for object inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 12 636–12 645.

[41] J. Zhuang, Y. Zeng, W. Liu, C. Yuan, and K. Chen, "A task is worth one word: Learning with task prompts for high-quality versatile image inpainting," in *European Conference on Computer Vision*, Springer, 2024, pp. 195–211.

[42] J. Li, F. He, L. Zhang, B. Du, and D. Tao, "Progressive reconstruction of visual structure for image inpainting," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 5962–5971.

[43] Y. Zhang et al., "A joint guidance-enhanced perceptual encoder and atrous separable pyramid-convolutions for image inpainting," *Neurocomputing*, vol. 396, pp. 1–12, 2020.

[44] Y. Zeng, J. Fu, H. Chao, and B. Guo, "Learning pyramid-context encoder network for high-quality image inpainting," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1486–1494.

[45] Y. Zeng, Z. Lin, H. Lu, and V. M. Patel, "Cr-fill: Generative image inpainting with auxiliary contextual reconstruction," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 14 164–14 173.

[46] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, Springer, 2016, pp. 694–711.

[47] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6721–6729.

[48] 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*, 2022, pp. 11 421–11 430.

[49] P. Ardino, Y. Liu, E. Ricci, B. Lepri, and M. De Nadai, "Semantic-guided inpainting network for complex urban scenes manipulation," in *2020 25th International Conference on Pattern Recognition (ICPR)*, IEEE, 2021, pp. 9280–9287.

[50] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[51] 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*, 2019, pp. 4401–4410.

[52] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8110–8119.

[53] Z. Yi, Q. Tang, S. Azizi, D. Jang, and Z. Xu, "Contextual residual aggregation for ultra high-resolution image inpainting," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 7508–7517.

[54] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, Springer, 2020, pp. 173–190.

[55] T. Chen, Y. Yao, L. Zhang, Q. Wang, G.-S. Xie, and F. Shen, "Saliency guided inter- and intra-class relation constraints for weakly supervised semantic segmentation," *IEEE Transactions on Multimedia*, vol. 25, pp. 1727–1737, 2022.

[56] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[57] J. Liu, S. Yang, Y. Fang, and Z. Guo, "Structure-guided image inpainting using homography transformation," *IEEE Transactions on Multimedia*, vol. 20, no. 12, pp. 3252–3265, 2018.

[58] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, and M. Ebrahimi, "Edgeconnect: Structure guided image inpainting using edge prediction," in *Proceedings of the IEEE/CVF international conference on computer vision workshops*, 2019, pp. 0–0.

[59] Y. Ren, X. Yu, R. Zhang, T. H. Li, S. Liu, and G. Li, "Structureflow: Image inpainting via structure-aware appearance flow," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 181–190.

[60] J. Yang, Z. Qi, and Y. Shi, "Learning to incorporate structure knowledge for image inpainting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, 2020, pp. 12 605–12 612.

[61] W. Xiong et al., "Foreground-aware image inpainting," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5840–5848.

[62] C. Cao and Y. Fu, "Learning a sketch tensor space for image inpainting of man-made scenes," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 14 509–14 518.

[63] W. Zhang et al., "Context-aware image inpainting with learned semantic priors," in *IJCAI*, 2021, pp. 1323–1329.

[64] W. Yu, J. Du, R. Liu, Y. Li, and Y. Zhu, "Interactive image inpainting using semantic guidance," in *2022 26th international conference on pattern recognition (ICPR)*, IEEE, 2022, pp. 168–174.

[65] W. Zhang, Y. Wang, B. Ni, and X. Yang, "Fully context-aware image inpainting with a learned semantic pyramid," *Pattern Recognition*, vol. 143, p. 109 741, 2023.

[66] C. Liu, S. Xu, J. Peng, K. Zhang, and D. Liu, "Towards interactive image inpainting via robust sketch refinement," *IEEE Transactions on Multimedia*, 2024.

[67] B. Dodson, *Keys to drawing*. Penguin, 1990.

[68] B. Edwards, *Drawing on the artist within*. Simon and Schuster, 2008.

[69] A. Hertzmann, "Toward modeling creative processes for algorithmic painting," *arXiv preprint arXiv:2205.01605*, 2022.

[70] A. Radford et al., "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, PmLR, 2021, pp. 8748–8763.

[71] O. Avrahami, O. Fried, and D. Lischinski, "Blended latent diffusion," *ACM transactions on graphics (TOG)*, vol. 42, no. 4, pp. 1–11, 2023.

[72] S. Wang et al., "Imagen editor and editbench: Advancing and evaluating text-guided image inpainting," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 18 359–18 369.

[73] M. Ni, X. Li, and W. Zuo, "Nuwa-lip: Language-guided image inpainting with defect-free vqgan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14 183–14 192.

[74] X. Ju, X. Liu, X. Wang, Y. Bian, Y. Shan, and Q. Xu, "Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion," in *European Conference on Computer Vision*, Springer, 2024, pp. 150–168.

[75] H. Manukyan, A. Sargsyan, B. Atanyan, Z. Wang, S. Navasardyan, and H. Shi, "Hd-painter: High-resolution and prompt-faithful text-guided image inpainting with diffusion models," in *The Thirteenth International Conference on Learning Representations*, 2023.

[76] C. Saharia et al., "Palette: Image-to-image diffusion models," in *ACM SIGGRAPH 2022 conference proceedings*, 2022, pp. 1–10.

[77] N. Sharma, A. Tripathi, A. Chakraborty, and A. Mishra, "Sketch-guided image inpainting with partial discrete diffusion process," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2024, pp. 6024–6034.

[78] H. Wang, Y. Yu, T. Luo, H. Fan, and L. Zhang, "Magic: Multi-modality guided image completion," in *The Twelfth International Conference on Learning Representations*, 2024.

[79] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[80] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, Springer, 2014, pp. 818–833.

[81] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, Springer, 2015, pp. 234–241.

[82] Y. Wang, X. Tao, X. Qi, X. Shen, and J. Jia, "Image inpainting via generative multi-column convolutional neural networks," *Advances in neural information processing systems*, vol. 31, 2018.

[83] H. Sun, W. Li, Y. Duan, J. Zhou, and J. Lu, "Learning adaptive patch generators for mask-robust image inpainting," *IEEE Transactions on Multimedia*, vol. 25, pp. 4240–4252, 2022.

[84] 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)*, 2018, pp. 85–100.

[85] 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*, 2018, pp. 8798–8807.

[86] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2337–2346.

[87] Y. Wang, L. Qi, Y.-C. Chen, X. Zhang, and J. Jia, "Image synthesis via semantic composition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 749–13 758.

[88] Q. Dong, C. Cao, and Y. Fu, "Incremental transformer structure enhanced image inpainting with masking positional encoding," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 358–11 368.

[89] L. Xu, Q. Yan, Y. Xia, and J. Jia, "Structure extraction from texture via relative total variation," *ACM transactions on graphics (TOG)*, vol. 31, no. 6, pp. 1–10, 2012.

[90] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness," in *International conference on learning representations*, 2018.

[91] Y. Zhang, Y. Liu, R. Hu, Q. Wu, and J. Zhang, "Mutual dual-task generator with adaptive attention fusion for image inpainting," *IEEE Transactions on Multimedia*, vol. 26, pp. 1539–1550, 2023.

[92] A. Vaswani et al., "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[93] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[94] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[95] W. Hua, Z. Dai, H. Liu, and Q. Le, "Transformer quality in linear time," in *International conference on machine learning*, PMLR, 2022, pp. 9099–9117.

[96] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International conference on machine learning*, PMLR, 2017, pp. 933–941.

[97] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[98] D. Hendrycks and K. Gimpel, "Gaussian Error Linear Units (GELUs)," *arXiv preprint arXiv:1606.08415*, 2016.

[99] Q. Lin, B. Yan, J. Li, and W. Tan, "Mmfl: Multimodal fusion learning for text-guided image inpainting," in *Proceedings of the 28th ACM international conference on multimedia*, 2020, pp. 1094–1102.

[100] X. Wu et al., "Adversarial learning with mask reconstruction for text-guided image inpainting," in *Proceedings of the 29th ACM international conference on multimedia*, 2021, pp. 3464–3472.

[101] L. Zhang, Q. Chen, B. Hu, and S. Jiang, "Text-guided neural image inpainting," in *Proceedings of the 28th ACM international conference on multimedia*, 2020, pp. 1302–1310.

[102] Z. Zhang, Z. Zhao, Z. Zhang, B. Huai, and J. Yuan, "Text-guided image inpainting," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 4079–4087.

[103] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2021.

[104] C. Saharia et al., "Photorealistic text-to-image diffusion models with deep language understanding," *Advances in neural information processing systems*, vol. 35, pp. 36 479–36 494, 2022.

[105] A. Q. Nichol et al., "Glide: Towards photorealistic image generation and editing with text-guided diffusion models," in *International Conference on Machine Learning*, PMLR, 2022, pp. 16 784–16 804.

[106] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, vol. 1, no. 2, p. 3, 2022.

[107] H. Wang et al., "Sam-clip: Merging vision foundation models towards semantic and spatial understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 3635–3647.

[108] H. Yuan, X. Li, C. Zhou, Y. Li, K. Chen, and C. C. Loy, "Open-vocabulary sam: Segment and recognize twenty-thousand classes interactively," in *European Conference on Computer Vision*, Springer, 2024, pp. 419–437.

[109] A. Kirillov et al., "Segment anything," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4015–4026.

[110] C. Mou et al., "T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 38, 2024, pp. 4296–4304.

[111] Y. Jiang, S. Yang, H. Qiu, W. Wu, C. C. Loy, and Z. Liu, "Text2human: Text-driven controllable human image generation," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 1–11, 2022.

[112] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[113] Z. Wan, J. Zhang, D. Chen, and J. Liao, "High-fidelity pluralistic image completion with transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 4692–4701.

[114] Y. Yu et al., "Diverse image inpainting with bidirectional and autoregressive transformers," in *Proceedings of the 29th ACM international conference on multimedia*, 2021, pp. 69–78.

[115] Y. Deng, S. Hui, S. Zhou, D. Meng, and J. Wang, "Learning contextual transformer network for image inpainting," in *Proceedings of the 29th ACM international conference on multimedia*, 2021, pp. 2529–2538.

[116] J. Ho, N. Kalchbrenner, D. Weissenborn, and T. Salimans, "Axial attention in multidimensional transformers," *arXiv preprint arXiv:1912.12180*, 2019.

[117] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in neural information processing systems*, vol. 34, pp. 8780–8794, 2021.

[118] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *International conference on machine learning*, PMLR, 2021, pp. 8162–8171.

[119] O. Avrahami, D. Lischinski, and O. Fried, "Blended diffusion for text-driven editing of natural images," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 18 208–18 218.

[120] W. Feller, "Retracted chapter: On the theory of stochastic processes, with particular reference to applications," in *Selected Papers I*, Springer, 2015, pp. 769–798.

[121] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*, pmlr, 2015, pp. 2256–2265.

[122] S. W. Zamir et al., "Multi-stage progressive image restoration," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 14 821–14 831.

[123] N. Wang, J. Li, L. Zhang, and B. Du, "Musical: Multi-scale image contextual attention learning for inpainting.," in *IJCAI*, 2019, pp. 3748–3754.

[124] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," in *International Conference on Learning Representations*, 2018.

[125] M. Cordts et al., "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[126] X. Wang, K. Yu, C. Dong, and C. C. Loy, "Recovering realistic texture in image super-resolution by deep spatial feature transform," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 606–615.

[127] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[128] 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.

[129] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 586–595.

[130] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, and M. Ebrahimi, "Edgeconnect: Structure guided image inpainting using edge prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct. 2019.

[131] W. Shi et al., "Real-time single image and video super-resolution using an efficient subpixel convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874–1883.

[132] B. Lin, W. Jiang, P. Chen, Y. Zhang, S. Liu, and Y.-C. Chen, "Mtmamba: Enhancing multi-task dense scene understanding by mamba-based decoders," in *European Conference on Computer Vision*, Springer, 2024, pp. 314–330.

[133] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 116–131.

[134] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations (ICLR 2015)*, 2015.

[135] Y. Yu, H. Wang, T. Luo, H. Fan, and L. Zhang, "Magic: Multi-modality guided image completion," in *The Twelfth International Conference on Learning Representations*, 2024.

[136] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[137] J. Zhuang, Y. Zeng, W. Liu, C. Yuan, and K. Chen, "A task is worth one word: Learning with task prompts for high-quality versatile image inpainting," in *European Conference on Computer Vision*, Springer, 2024, pp. 195–211.

[138] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *International conference on machine learning*, PMLR, 2022, pp. 12 888–12 900.

[139] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*, PMLR, 2023, pp. 19 730–19 742.

[140] S. Yang, X. Chen, and J. Liao, "Uni-paint: A unified framework for multimodal image inpainting with pretrained diffusion model," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 3190–3199.

[141] Y. Chen et al., "Improving text-guided object inpainting with semantic pre-inpainting," in *European Conference on Computer Vision*, Springer, 2024, pp. 110–126.

[142] X. Xing, C. Wang, H. Zhou, J. Zhang, Q. Yu, and D. Xu, "Diffsketcher: Text guided vector sketch synthesis through latent diffusion models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 15 869–15 889, 2023.

[143] Y. Vinker, Y. Alaluf, D. Cohen-Or, and A. Shamir, "Clipascene: Scene sketching with different types and levels of abstraction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4146–4156.

[144] A. Voynov, K. Aberman, and D. Cohen-Or, "Sketch-guided text-to-image diffusion models," in *ACM SIGGRAPH 2023 Conference Proceedings*, 2023, pp. 1–11.

[145] K. Kim, S. Park, J. Lee, and J. Choo, "Reference-based image composition with sketch via structure-aware diffusion model," *arXiv preprint arXiv:2304.09748*, 2023.

[146] Z. Su et al., "Pixel difference networks for efficient edge detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 5117–5127.

[147] H. Mo, E. Simo-Serra, C. Gao, C. Zou, and R. Wang, "General virtual sketching framework for vector line art," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–14, 2021.

[148] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.

[149] T.-Y. Lin et al., "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, Springer, 2014, pp. 740–755.

[150] J. Ho and T. Salimans, "Classifier-free diffusion guidance," in *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[151] Y. Kao, C. Wang, and K. Huang, "Visual aesthetic quality assessment with a regression model," in *2015 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2015, pp. 1583–1587.

[152] C. Schuhmann et al., "Laion-5b: An open large-scale dataset for training next generation image-text models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 25 278–25 294, 2022.

[153] J. Hessel, A. Holtzman, M. Forbes, R. L. Bras, and Y. Choi, "Clipscore: A reference-free evaluation metric for image captioning," *arXiv preprint arXiv:2104.08718*, 2021.