Synthesizing Photorealistic Images with Deep Generative Learning

Chuanxia Zheng

Supervisor: Prof. Tat-Jen Cham

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

06 June 2021	Chuanxia Zheng
Date	Chuanxia Zheng

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

.... 06 June 2021 ....

Date

..... 6 June 2021 .....

Prof. Tat-Jen Cham

Authorship Attribution Statement

This thesis contains material from Six paper(s) published in the following peer-reviewed journal(s) / from papers accepted at conferences in which I am listed as the first author.

The work in Chapter 2 is published as C Zheng, TJ Cham, J Cai. T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. Proceedings of the European Conference on Computer Vision (ECCV). 2018.

The contributions of the co-authors are as follows:

- Chuanxia Zheng proposed the initial idea, designed the experiments, and prepared the manuscript.
- Tat-Jen Cham and Jianfei Cai discussed the idea, improved the experiments and revised the manuscript.

The work in Chapter 3 is published as C Zheng, TJ Cham, J Cai. The Spatially-Correlative Loss for Various Image Translation Tasks. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2021.

The contributions of the co-authors are as follows:

- Chuanxia Zheng proposed the initial idea, designed the experiments, and prepared the manuscript.
- Tat-Jen Cham and Jianfei Cai discussed the idea, improved the experiments and revised the manuscript.

The work in Chapter 4 is published as C Zheng, TJ Cham, J Cai. Pluralistic image completion. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019, and published as C Zheng, TJ Cham, J Cai. Pluralistic free-form image completion. International Journal of Computer Vision (IJCV). 2021.

The contributions of the co-authors are as follows:

- Chuanxia Zheng proposed the initial idea, designed the experiments, and prepared the manuscript.
- Tat-Jen Cham and Jianfei Cai discussed the idea, improved the experiments and revised the manuscript.

The work in Chapter 5 is reviewed as C Zheng, TJ Cham, J Cai. TFill: Image Completion via a Transformer-Based Architecture (arXiv). 2021.

The contributions of the co-authors are as follows:

- Chuanxia Zheng proposed the initial idea, designed the experiments, and prepared the manuscript.
- Tat-Jen Cham and Jianfei Cai discussed the idea, improved the experiments and revised the manuscript.

The work in Chapter 6 is published as C Zheng, DS Dao, G Song, TJ Cham, J Cai. Visiting the Invisible: Layer-by-Layer Completed Scene Decomposition. International Journal of Computer Vision (IJCV). 2021.

The contributions of the co-authors are as follows:

- Chuanxia Zheng proposed the initial idea, designed the experiments, and prepared the manuscript.
- Guoxian Song rendered the synthetic data.
- Duy-Son Dao worked for the amodal instance segmentation.
- Tat-Jen Cham and Jianfei Cai discussed the idea, improved the experiments and revised the manuscript.

06 June 2021 ..... Chuanxia Zheng .....

Date Chuanxia Zheng

Acknowledgements

Eleven years ago, for the first time, I was away from my hometown with a dream to at least see a bit of the world. I would like to thank many people who helped me a lot along my path. Without them, I may have no possibility of writing this thesis.

I would like to express my greatest gratitude to my advisor, Tat-Jen Cham, for taking me under his wing. He is not only my academic advisor who guides me with encouraging, constructive and insightful comments on every research topic, but also the life coach that provides suggestions on how to think deeper, how to be a better man, and how to walk toward a humane and inspired life. In the past four years, I enjoyed a fantastic research journey under his supervision.

I would especially like to thank my co-advisor, Jianfei Cai, for his helpful advice and the immense knowledge he has shared with me. Without him, I would not have the opportunity to pursue the research position at NTU, and I can not come to Singapore. His guidance helped me in all the time of research and writing of this thesis.

I would like to thank Nadia Magnenat Thalmann, the director of the institute for media innovation (IMI), for awarding me the Ph.D. Scholarship. In IMI, I always enjoyed the freedom to spend time on my interested research topics.

I would like to thank Irene Goh for providing the powerful AI supercomputer system and helping the deep learning model running smoothly on the corresponding platform. Without these supercomputers in SCSE, I would not have finished many complex experiments.

I would also thank all the members of the IMI lab and MICL who provided tremendous support for my study and research. Special thanks to Teng Deng for guiding me on the study when I first arrived at the lab. I would like to thank Guoxian Song for rendering the high-quality synthetic dataset on our projects. I would like to thank Duy-Son Dao for setting the baseline benchmark on amodal instance segmentation. I would like to thank Xingxing Xia and Frank Guan for providing helpful suggestions on the translation and completion task. I would like

to thank many other folks including Xu Yang, Yuedong Cheng, Bo Hu, Zhonghua Wu, Zhijie Zhang, Junwu Weng, Yujun Cai, Jyothsna Vasudevan, and Ayan Kumar Bhunia for many interesting discussions.

I would like to thank my old friends, *e.g.* Zhongxia Xiong, Yikang Guo, Han Zhang, Shijie Zhang, Boyu Yang, Xiang Wen, Wei Zhao, among others, for their discussing, listening, and sharing. Special thanks to Zhongxia Xiong for providing suggestions and feedback on every project and publicly available code.

Lastly, I am grateful to my parents, my parents-in-law, and my wife for their love and support during this wonderful journey. Special thanks to my wife, Mengping, who always accompanies me and gives me confidence and encouragement in these years. She and our lovely daughter, Keyu, are the most precious treasure in my life.

Chuanxia Zheng
Nanyang Technological University
June 2021

Summary

The goal of this thesis is to present my research contributions towards solving various visual synthesis and generation tasks, comprising image translation, image completion, and completed scene decomposition. This thesis consists of five pieces of work, each of which presents a new learning-based approach for synthesizing images with plausible content as well as visually realistic appearance. Each work demonstrates the superiority of the proposed approach on image synthesis, with some further contributing to other tasks, such as depth estimation.

Part I describes methods for **changing visual appearance**. In particular, in Chapter 2, a *synthetic-to-realistic* translation system is presented to address the real-world *single-image depth estimation*, where only synthetic image-depth pairs and unpaired real images are used for training. This model provides a new perspective on a real-world estimation task by utilizing low-cost, yet high-reusable synthetic data. In Chapter 3, the focus is on general image-to-image (I2I) translation tasks, instead of narrowly synthetic-to-realistic image translation. A novel *spatially-correlative loss* is proposed that is simple, efficient and yet effective for preserving scene structure consistency, while supporting large appearance changes. Spatial patterns of self-similarity are exploited as a means of defining scene structure, with this spatially-correlative loss geared towards only capturing spatial relationships within an image, rather than domain appearance. The extensive experiment results demonstrate significant improvements using this content loss on several I2I tasks, including single-modal, multi-modal, and even single-image translation. Furthermore, this new loss can easily be integrated into existing network architectures and thus allows wide applicability.

Part II presents approaches that **generate semantically reasonable content** for masked regions. Instead of purely modifying the local appearance as in Part I, two approaches are presented to create new content as well as realistic appearance for a given image. In Chapter 4, a new task is introduced, called *pluralistic image completion* — the task of generating *multiple* and *diverse* plausible results, which is as opposed to previous works that attempt to create only a single "guess"

for this highly subjective problem. In this Chapter, a novel probabilistically principled framework is proposed, which achieved state-of-the-art results for this new task and has become the benchmark for later works. However, my subsequent observation is that architectures based on convolutional neural networks (CNN) model long-range dependencies via many stacked layers, where holes are progressively influenced by neighboring pixels, resulting in some artifacts. To mitigate this issue, in Chapter 5, I propose treating image completion as a directionless sequence-to-sequence prediction task, and deploy a *transformer* to directly capture long-range dependencies in the encoder in a first phase. Crucially, a *restrictive CNN* with small and non-overlapping receptive fields (RF) is employed for token representation, which allows the transformer to explicitly model long-range context relations with equal importance in all layers, without implicitly confounding neighboring tokens when larger RFs are used. Extensive experiments demonstrate superior performance compared to previous CNN-based methods on several datasets.

Part III combines recognitive learning and the latest generative modeling into a holistic scene decomposition and completion framework, where a network is trained to *decompose* a scene into individual objects, *infer* their underlying occlusion relationships, and moreover *imagine* what the originally occluded objects may look like, *while using only a single image as input.* In Chapter 6, the aim is to derive a higher-level structural decomposition of a scene, automatically recognizing objects and generating intact shapes as well as photorealistic appearances for occluded regions, without requiring manual masking as in Part II. To achieve this goal, a new pipeline is presented that interleaves the two tasks of instance segmentation and scene completion through multiple iterations, solving for objects in a layer-by-layer manner. The proposed system shows significant improvement over the state-of-the-art methods and enables some interesting applications, such as scene editing and recomposition.

In summary, the thesis introduces a series of works to synthesize photorealistic images by changing the appearance, imagining the semantic content, and inferring the invisible shape and appearance automatically.

Keywords: Image generation, generative adversarial networks, variational auto-encoder, conditional variational auto-encoder, image completion, image translation, multi-modal generative models, depth evaluation, layered scene decomposition, object completion, amodal instance segmentation, instance depth order, scene recomposition, convolutional networks, attention, transformer

Contents

List of Figures

List of Tables

List of Abbreviations

RBM	Restricted Boltzmann Machine
AE	AutoEncoder
CNN	Convolutional Neural Network
MLP	Multi-Layer Perceptron
GAN	Generative Adversarial Network
VAE	Variational AutoEncoder
CVAE	Conditional Variational AutoEncoder
ResNet	Residual Network
RF	Receptive Field
NLP	Natural Language Processing
VQ	Vector Quantization
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural SIMilarity
IS	Inception Score
FID	Fréchet Inception Distance
LPIPS	Learned Perceptual Image Patch Similarity
AMT	Amazon Mechanical Turk
2AFC	2-Alternative-Forced-Choice
D&C	Density and Coverage
DoF	degree of freedom
i.i.d.	independent and identically distributed
w.r.t.	with respect to

Chapter 1

Introduction

This thesis focuses on building intelligent algorithms to synthesize visually realistic images for various computer vision tasks. This chapter provides general background on visual synthesis and realism evaluation. The subsequent chapters will provide more background on each of the specific *visual synthesis* tasks, and present new methods to address them.

1.1 Visual Synthesis and Generation

Visual imagery is one of the most important aspects of the computer world, being part of our modern life via various media applications, such as TikTok, WeChat, Facebook, and YouTube. As a result, people freely create millions of photos and videos per day on the internet, making it possible for researchers to collect astounding amounts of visual content to tame the artificial intelligence model for various computer vision tasks [64].

In the computer vision community, researchers conventionally focus on recognition tasks [110]. Due to the availability of these vast amounts of visual data and the advances of deep learning algorithms, the community has rapidly improved recognition results over a short period of time. For instance, in Figure 1.1 (top), we can now build powerful intelligent systems that accurately recognize the category of a scene [174], localize [61] and segment [148] object instances in it, and even describe the scene in natural language [211]. However, there is also the opposite research direction, *visual synthesis*, which aims to create new visual content based on partial observation of real data. As shown in Figure 1.1 (bottom), we would like to teach machines to learn the capacity of imagination that humans are capable of, and be able to generate visual data with reasonable content and realistic appearance. For

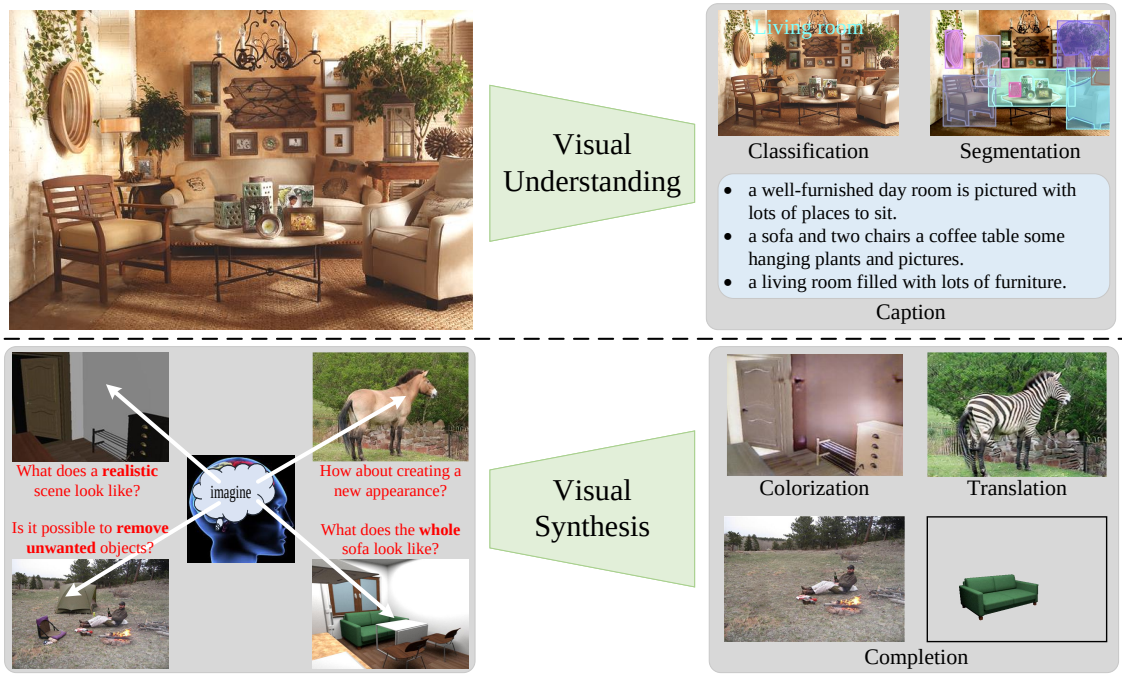


FIGURE 1.1: **The overall exhibition of the goal in this thesis.** In the top row, we first show the general *visual understanding* tasks, which have achieved rapid advances, such as in image classification, instance segmentation, and image captioning, due to vast amounts of visual data along with deep learning networks [64]. In this thesis, we attempt to explore the opposite direction, *visual synthesis*, where we empower the model to imagine and generate new photorealistic images by estimating the data distribution.

example, when a sofa is occluded by other furniture, can machines figure out what does the **whole** sofa look like? How would machines learn to imagine the missing content?

Why would *visual synthesis and generation* be important for the computer vision community? One potential relevance is to *self-supervised representation learning* [12]. As we know, building datasets with extensive labels is a high-effort and high-cost undertaking, while the world is full of unlabeled, free data, particularly on the Internet. The traditional representation learning methods, such as Restricted Boltzmann Machines (RBM) [175] and Auto-Encoders (AE) [15], learn robust features without using labels by attempting to reconstruct the raw input. More recently, some methods [99, 136] further attempted the more challenging label free tasks, such as colorization, completion, solving jigsaws, and rotation prediction, resulting in more robust features for downstream tasks. Alternatively, the synthesized images can be used as augmented data for deep learning [180, 226], especially for 3D-related tasks. For instance, as more and more high-quality 3D CAD models become available online [51, 177], it is possible to render an unlimited

number of photorealistic images to support real-world tasks, *e.g.* depth estimation, object detection and segmentation, and 3D reconstruction.

In addition to promoting machine understanding of the real world, *visual synthesis and generation* can also create visual content that improves human-to-machine and computer-mediated human-to-human interaction. As mentioned above, people upload millions of images and videos per day on the internet, but often they are not entirely satisfied with the quality of such content. Supposing you have taken a photo of camping as shown in Figure 1.1, but you would like to remove the unwanted objects, or create some new elements, or change the color and lighting. We desire to have an intelligent visual synthesis system that can be used to easily improve the picture, *e.g.* removing unwanted objects in Figure 1.1. In this way, we can help users easily synthesize more visually appealing photos to ideally express themselves better.

We investigate a number of data-driven visual synthesis and generation tasks for various applications in this thesis. In the following section, we will briefly define the tasks and give some background on the solutions.

1.1.1 Deep Generative Models

Our methods are mainly built upon *deep generative models*, which estimate complex high-dimensional data distributions using a set of variables in deep layers. Currently, the emerging powerful frameworks, including Generative Adversarial Networks (GANs) [65] and Variational Autoencoders (VAEs) [103], have made impressive advances in many generation tasks. These models try to learn a function that maps the unknown distribution of training data $\mathbf{x}$ from a predefined probability distribution of latent variable $\mathbf{z}$. Formally,

$$\mathbf{x} = g(\mathbf{z}; \theta). \tag{1.1}$$

In the VAE framework, an encoder $q(\mathbf{z}|\mathbf{x})$, acting as an approximate inference network, is used to obtain the latent variable $\mathbf{z}$ from training instances. Once the model is learned, it can generate arbitrary data by resampling from the latent distribution. In a GAN framework, an auxiliary discriminator network is trained adversarially with the generator network, which should ideally lead to minimizing the distribution distance between the generated data and original data.

1.1.2 Image-to-Image Translation

Image-to-Image (I2I) translation involves designing algorithms that can learn to modify an input image $\mathbf{x}$ to fit the *style / appearance* of the target domain, while preserving the original *content*, as shown in Figure 1.1: *horse* $\rightarrow$ *zebra*. The process is to learn such a mapping:

$$f : \mathbf{x} \rightarrow \mathbf{y} \tag{1.2}$$

where the input image $\mathbf{x}$ is translated to another image $\mathbf{y}$ in the target domain. In this thesis, I2I refers to the task of only modifying the appearance, while the content / structure is preserved.

One of the simplest forms is *paired* I2I translation [87]. In this case, the paired training examples $\{x_i, y_i\}_{i=1}^{N}$ are given, where the y_i corresponds to each input x_i. However, obtaining such paired training data is difficult, expensive or even impossible in some situations. Therefore, following [234], we focus on unpaired I2I translation work that learns to translate between domains without paired input-output examples.

1.1.3 Image Completion

Image completion refers to the task of filling alternative reasonable content for missing or deleted parts in images, which can be used for restoring damaged paintings, removing unwanted objects, and generating new content for incomplete scenes. This task is a further development of traditional image "inpainting" [13], which only works for narrow or small holes, due to the lack of deeper semantic understanding. Here, we investigate data-driven visual synthesis approaches to fill in semantic reasonable content with photorealistic appearance into arbitrary missing regions. In particular, given a masked image $\mathbf{I}_m$ that is degraded by a number of missing pixels, the goal is to learn a model Φ to infer the content, conditioned on partially visible information:

$$\mathbf{I}_g = \Phi(\mathbf{I}_m; \theta) \tag{1.3}$$

where the input masked image $\mathbf{I}_m$ is combined with newly synthesized content to become a completed image $\mathbf{I}_g$.

The earlier learning-based approaches use the conventional convolutional operation to train a model in a deterministic way. In this thesis, I introduce a new direction, pluralistic image completion, that aims to generate multiple and diverse results for this highly subjective task. As it is important to explore the global

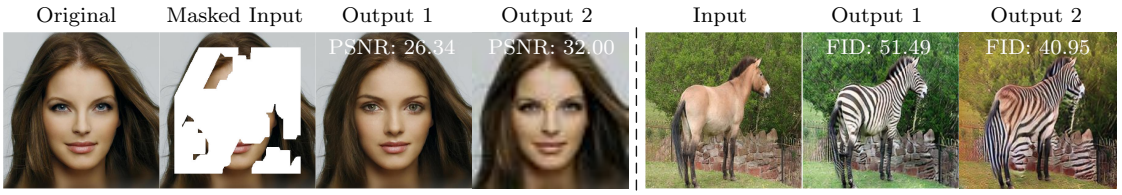


FIGURE 1.2: **Which output is the "better" result for each input in these examples?** In each case, the existing metrics disagree with human judgments. The traditional metrics, including ℓ_1, PSNR, and SSIM, support the blurry "Output 2" in the first case because the latter is optimized only by ℓ_1 reconstruction loss that encourages the same content to the original image. In the unpaired I2I task, the learning-based metrics, such as IS and FID, agree with the "Output 2" due to many results in this setting have repeated zebra's texture.

visible information for missing content inference, a transformer-based image completion network is also later investigated in this thesis.

1.1.4 Completed Scene Decomposition

The goal of *completed scene decomposition* is to build an intelligent system that automatically *decomposes* a scene into individual objects, *infers* their underlying occlusion relationships and moreover *imagines* what occluded objects may look like. This means that the learning algorithm must be able to *understand* the scene to predict the geometry and categories of all objects in it (as shown in Figure 1.1 (top)), and also *synthesize* invisible parts of objects and backgrounds (as shown in Figure 1.1 (bottom)).

To do so, we aim at deriving a higher-level structure decomposition of a scene. As humans, we are highly aware of the shapes of individual objects and their ordering relationships, and we can generally imagine what occluded objects may look like. For instance, as shown in Figure 1.1 (bottom), humans can easily recognize the sofa and the table, and deduce the former is occluded by the latter, and even guess what the whole sofa looks like, based on global visible information and prior knowledge.

1.2 Evaluation of Image Visual Realism

Image quality evaluation is a difficult research problem in computer vision. As laypeople, we may not be able to create realistic images just like artists, but we can easily judge whether a given image is "realistic", and we are able to correctly recognize which parts make a "fake" image appear unreal.

Classic Metrics However, "what makes a real image look realistic?" has no clear answer in computer vision. In traditional works, researchers investigated a lot of factors, *e.g.* color, texture, boundary, structure, and illumination, yet it is still hard to precisely define the impact of these factors mathematically. While various classic metrics, such as ℓ_1 loss, Peak Signal-to-Noise Ratio (PSNR), and Structural SIMilarity (SSIM) [196], are proposed to assess image quality via unambiguous formulae, they are poorly related to human judgment due to independent pixel- and patch-level evaluation [222].

A well-known example is shown in Figure 1.2 (left). Compared to the high-quality completed "Output 1" [229], the blurry completed "Output 2" has smaller ℓ_1 reconstruction error (0.0144 *vs* 0.0255), and larger PSNR (32.00 *vs* 26.34) and SSIM score (0.9153 *vs* 0.8248), with respect to the original image. This is because "Output 2" is trained using only ℓ_1 reconstruction loss to the original unmasked image, for which blurry content can be smaller than content that is almost identical but slightly misaligned. Therefore, designing a "perceptual metric" that measures image quality similar to human judgment has been a longstanding goal.

Learned Metrics In more recent work, researchers have started focusing on learning-based feature-level distances, *e.g.* Learned Perceptual Image Patch Similarity (LPIPS) metric [222], Inception Score (IS) [163] and Fréchet Inception Distance (FID) [77]. These learned metrics mitigate the above-mentioned issue by evaluating the image quality in a deep neural network layer with large receptive field, instead of assuming pixel-wise independence in traditional metrics. For example, the LPIPS strongly agrees with human judgment that "Output 1" is more perceptual similar to the original image than the blurry "Output 2". However, these learned metrics are also not perfectly matched the human judgment as the currently pretrained networks tend to base their decisions much more on texture than shape [58], while humans are more strongly focused on image structure [109] and related-context [222].

As an example depicted in Figure 1.2 (right), a horse is translated to the zebra domain, where there is no ground truth for evaluation. As humans, we can easily judge that "Output 1" is more realistic than "Output 2". However, the FID score, which compares the distance between distributions of translated and real images in a deep feature domain, evaluates "Output 2" as having a lower distribution distance, because all results in the second scenario have more obvious zebra stripes.

The currently learned metrics are therefore not yet the perfect solutions for image quality evaluation.

Human Perceptual Metrics Finally, we briefly introduce the human perceptual metrics, as proposed in [221] and widely adopted for image generation [87, 139, 144, 234, 235]. These are online metrics that the authors developed based on user studies, in which they provided some generated results and ground truth real images on Amazon Mechanical Turk (AMT)[1], and asked the participants to manually distinguish "real" and "fake" images.

In this thesis, following existing state-of-the-art approaches, we report the corresponding evaluation metrics for different tasks. However, we would like to remind the reader that none of these are perfect for assessing generated image quality.

1.3 Dissertation Overview

The main research objective in this dissertation is to create new intelligent systems that can imagine and generate visually realistic natural photographs, which can be used in artistic creation, image editing, and further help real-world tasks (*e.g.* depth estimation [226] and semantic segmentation [230]). As introduced in Section 1.1, in this dissertation, we explore three kinds of synthesis tasks: image translation [226, 228], image completion [227, 229], and completed scene decomposition [230].

- **Part I. Changing Visual Appearance** Chapters 2 and 3 describe methods for unpaired I2I translation that converts the visual appearance of the input image. In this task, deep learning is applied to learn a function $f : \mathbf{x} \rightarrow \mathbf{y}$, where $\mathbf{x}$ from a particular image domain $\mathcal{X}$, and $\mathbf{y}$ is the corresponding output that should belong to the target image domain $\mathcal{Y}$. Chapter 2 focuses on *synthetic-to-realistic* translation, in which I aim to bridge the gap between virtual and real scenes. This method is further integrated with a single-image depth estimation task to circumvent the challenges of obtaining accurate and sufficient 3D data of real scenes. In Chapter 3, a new content loss is introduced for arbitrary unpaired I2I translation tasks, in which I use self-similar correlations to better separate scene structure and appearance.

- **Part II. Generating Semantic Reasonable Content** Chapters 4 and 5 present approaches that learn to fill alternative reasonable content into

[1] https://www.mturk.com/

missing regions of degraded images. In Chapter 4, a new goal is introduced, **pluralistic image completion**, in which *multiple* and *diverse* plausible results are generated in a mathematically principled manner for this highly subjective process problem. Chapter 5, further presents a newer framework that aims to generate single "best" result by directly modeling long-range dependencies in the masked image via a Transformer-based architecture.

- **Part III. Modeling shape and appearance** Chapter 6 combines the classical recognition task and the latest generation task into an end-to-end scene decomposition network, where a network is trained to *decompose* a scene into individual objects, *infer* their underlying occlusion relationships, and moreover *imagine* what occluded objects may look like. In this task, a layer-by-layer algorithm is presented that can predict the geometry and categories of all objects in a scene, as well as generate their realistic appearance for originally occluded parts.

- **Discussion** In Chapter 7, we summarize the contributions of this thesis and discuss several future directions in image synthesis using deep learning.

Part I

Changing Visual Appearance: Image-to-Image Translation

Chapter 2

Synthetic-to-Realistic Translation

The main research goal presented in this chapter is to generate photorealistic images, which can be used to contribute the real-world single depth estimation task. The depth estimation is a classic research topic in computer vision, which has many different applications, such as autonomous driving, augmented reality, and scene reconstruction. As we live in a 3D world, humans are able to judge relative distances well even when only a single photograph is provided. However, it is still a challenge for a machine to accurately evaluate the depth from a *single* RGB image. A main limitation is that 3D data is difficult to collect compared to the 2D images, due to requiring more specialized equipment. To address this issue, we aim to provide an alternative perspective by utilizing *synthetic* image-depth pairs instead of real paired data. As more and more 3D CAD indoor scene models, such as in the SUNCG [177][1] and 3D-FRONT [51] datasets, become publicly available on the internet, researchers can cheaply and effectively render a vast number of paired datasets. To bridge the gap between synthetic and real images, a *wide-spectrum* translation network is proposed to convert synthetic-looking images with different levels of realism into realistic ones, such that after we train the depth estimation network on the translated images, the model can be directly applied to real images.

The rest of this chapter is structured as follows: Sections 2.1 and 2.2 describe the motivation and related works. Next, I explain the proposed framework in Section 2.4. Section 2.5 introduces the synthetic datasets used. I then describe and discuss the experiments in Section 2.6 and conclude in Section 2.7.

[1]This work was published as *T^2Net: Synthetic-to-Realistic Translation for Solving Single-Image Depth Estimation Tasks* in ECCV, 2018 [226]. At the time of publication, the SUNCG is still publicly available online.

2.1 Introduction

Single-image depth estimation is a challenging ill-posed problem for which good progress has been made in recent years, using supervised deep learning techniques [45, 46, 108, 122] that learn the mapping between image features and depth maps from large training datasets comprising image-depth pairs. An obvious limitation, however, is the need for vast amounts of paired training data for each scene type. Building such extensive datasets for specific scene types is a high-effort, high-cost undertaking [57, 165, 173] due to the need for specialized depth-sensing equipment. The limitation is compounded by the difficulty that traditional supervised learning models face in generalizing to new datasets and environments [122].

To mitigate the cost of acquiring large paired datasets, a few unsupervised learning methods [55, 63, 106] have been proposed, focused on estimating accurate disparity maps from easier-to-obtain binocular stereo images. Nonetheless, stereo imagery are still not as readily available as individual images, and systems trained on one dataset will find difficulty in generalizing well to other datasets (observed in [63]), unless camera parameters and rigs are identical in the datasets.

A recent trend that has emerged from the challenge of real data acquisition is the approach of training on synthetic data for use on real data [79, 152, 172], particularly for scenarios in which synthetic data can be easily generated. Inspired by these methods, we have researched a single-image depth estimation method that utilizes synthetic image-depth pairs instead of real paired data, but which also exploits the wide availability of unpaired real images. In short, our scenario is thus: we have a large set of real imagery, but these do not have any corresponding ground-truth depth maps. We also have access to a large set of synthetic 3D scenes[2], from which we can render multiple synthetic images from different viewpoints and their corresponding depth maps. The main goal then is to learn a depth map estimator when presented with a real image. Consider two of the more obvious approaches:

1. Train an estimator using only synthetic image and depth maps, and hope that the estimator applies well to real imagery (**Naive** in Figure 2.1).

2. Use a two-stage framework in which synthetic imagery is first translated into the real-image domain using a GAN, and then train the estimator as before (**Vanilla version** in Figure 2.1).

[2]One 3D CAD model can be rendered to a vast number of paired data by setting different camera parameters.

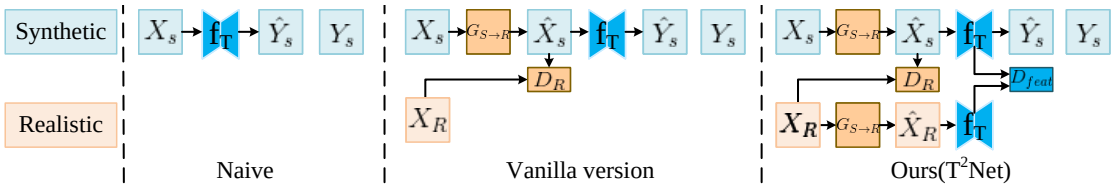


FIGURE 2.1: **Depth prediction strategies given synthetic image-depth pairs (x_s, y_s).** (Naive) structure directly trains an estimator using only synthetic image and depth maps. (Vanilla version) translates the synthetic image to the real domain and then trains the depth estimator. (Ours T^2Net) introduces a *wide-spectrum* translation network that simultaneously maps the arbitrary images to target domains.

The problem with 1) is that it is unlikely the estimator is oblivious to the differences between synthetic and real imagery. In 2), while a GAN may encourage synthetic images to map to the distribution of real images, it does not explicitly require the translated realistic image to have any physically-correct relationship to its corresponding depth map, meaning that the learned estimator will not apply well to actual real input. This may be somewhat mediated by introducing some regularization loss to try and keep the translated image "similar" in content to the original synthetic image (as in SimGAN [172]), but we cannot identify any principled regularization loss functions, only heuristic ones.

In this chapter, an interesting perspective is introduced on the approach of 2). We propose to have the entire inference pipeline be agnostic as to whether the input image is real or synthetic, *i.e.* it should work equally well regardless. To do so, we want the synthetic-to-realistic translation network to also behave as an identity transform when presented with real images, which is effected by including a reconstruction loss when training with real images.

The broad idea here is that, in a whole spectrum of synthetic images with differing levels of realism, *the network should modify a realistic image less than a more obviously synthetic image*. This is not true of original GANs, which may transform a realistic image into a different realistic image. In short, for the synthetic-to-real translation portion, real training images are challenged with a reconstruction loss, while synthetic images are challenged with a GAN-based adversarial loss [65]. This real-synthetic agnosticism is the principled formulation that allows us to dispense with an ad hoc regularization loss for synthetic imagery. When coupled with a task loss for the image-to-depth estimation portion, it leads to an end-to-end trainable pipeline that works well, and does not require the use of any real image-depth pairs nor stereo pairs (**Ours(T^2Net)** in Figure 2.1).

2.2 Background

This task is related to two sets of work: *single image depth estimation* and *unpaired I2I translation*. Here, we briefly review these approaches.

Single Image Depth Estimation After classical learning techniques were earlier applied to single-image depth estimation [80, 98, 107, 164, 165], deep learning approaches took hold. In [46], a two-scale CNN architecture was proposed to learn the depth map from raw pixel values. This was followed by several CNN-based methods, which included combining deep CNN with continuous CRFs for estimating depth values [122], simultaneously predicting semantic labels and depth maps [192], and treating the depth estimation as a classification task [17]. One common drawback of these methods is that they rely on large quantities of paired images and depths in various scenes for training. Unlike RGB images, real RGB-depth pairs are much scarcer.

To overcome the above-mentioned problems, some unsupervised and semi-supervised learning methods have recently been proposed that do not require image-depth pairs during training. In [55], the autoencoder network structure is translated to predict depths by minimizing the image reconstruction loss of image stereo pairs. More recently, this approach has been extended in [63, 106], where left-right consistency was used to ensure both good quality image reconstruction and depth estimation. While the data availability for these cases was perhaps not as challenging since special capture devices were not needed, nevertheless they depend on the availability or collection of stereo pairs with highly accurate rigs for consistent camera baselines and relative poses. This dependency makes it particularly difficult to cross datasets (*i.e.* training on one dataset and testing on another), as evidenced by the results presented in [63]. To alleviate this problem, an unsupervised adaption method [183] was proposed to fine-tune a stereo network to a different dataset from which it was pre-trained on. This was achieved by running conventional stereo algorithms and confidence measures on the new dataset, but on much fewer images and at sparser locations.

Unpaired I2I Translation Separately, several other works have explored image-to-image translation without using paired data. The earlier style-translation networks [56, 93] would synthesize a new image by combining the "content" of one image with the "style" of another image. In [125], the weight-sharing strategy was

introduced to learn a joint representation across domains. This framework was extended in [124] by integrating variational autoencoders and generative adversarial networks. Other concurrent works [101, 209, 234] utilized cycle consistency to encourage a more meaningful translation. However, these methods were focused on generating visually pleasing images, whereas for us image translation is an intermediate goal, with the primary objective being depth estimation, and thus the fidelity of 3D shape semantics in the translation has overriding importance.

In [172], a SimGAN was proposed to render realistic images from synthetic images for gaze estimation as well as human hand pose estimation. A self-regularization loss is used to force the generated target images to hold the similar content to the original source images. However, we consider this loss to be somewhat ad hoc and runs counter to the translation effort; it may work well in small domain shifts, but is too limiting for large style translation in our problem. As such, we use a more principled reconstruction loss as detailed in Section 2.4. More recently, a cycle-consistent adversarial domain adaption method was proposed [79] to generate target domain training images for digit classification and semantic segmentation. However this method is too complex for end-to-end training, which we consider to be an important requirement to achieve good results.

2.3 Overview

The main research goal is to train an image-to-depth network f_T, such that when presented with a single RGB image, it predicts the corresponding depth map accurately.

In terms of data availability for training, we assume that we have access to a collection of individual real-world images x_r, *without* stereo pairing nor corresponding ground truth depth maps. Instead, we assume that we have access to a collection of synthetic 3D models, from which it is possible to render numerous synthetic images and corresponding depth maps, denoted in pairs of (x_s, y_s).

Instead of directly training f_T on the synthetic (x_s, y_s) data, we expect that the synthetic images are insufficiently similar to the real images, to require a prior image translation network $G_{S \to R}$ for domain adaptation to make the synthetic images more realistic. However, as discussed previously, existing image translation methods do not adequately preserve the geometric content for accurate depth prediction, or require heuristic regularization loss functions.

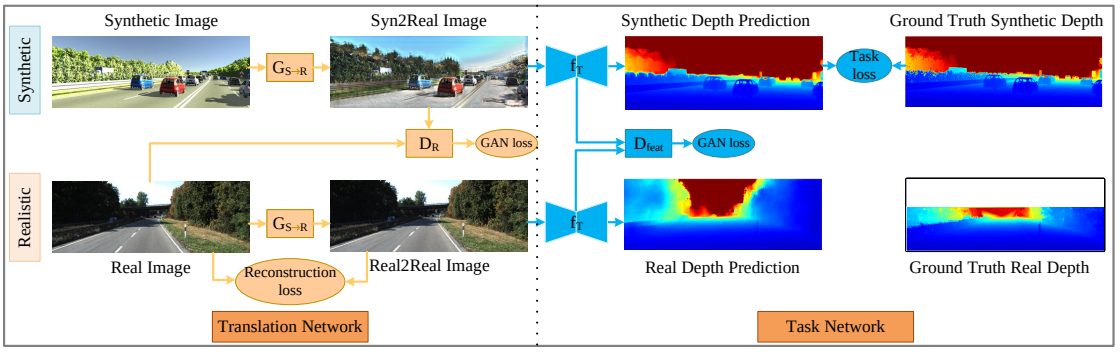


FIGURE 2.2: **The overall pipeline of the proposed method.** The proposed T^2Net consists of the Translation part (left, orange) and Task prediction part (right, blue). The $G_{S\to R}$ is a generator to translate images from synthetic domain to real domain, and D_R is the corresponding discriminator to judge whether the translated image is real or fake. f_T is depth task estimation network and D_{feat} is a discriminator on feature domain.

The *key novel insight* is this: instead of training $G_{S\to R}$ to be a narrow-spectrum translation network that translates one specific domain to another, we will train it as a *wide-spectrum* translation network, to which we can feed a range of input domains, *i.e.* synthetic imagery as well as actual real images. The intention is to have $G_{S\to R}$ implicitly learn to apply the minimum change needed to make an image realistic, and consider this the most principled way to regularize a network for preserving shape semantics needed for depth prediction.

Next, I introduce the proposed approach (Section 2.4) and the data collection (Section 2.5) in details.

2.4 Approach

To achieve the above-mentioned goal, a twin pipeline training framework is proposed (shown in Figure 2.2), which is named as T^{2}Net to highlight the combination of an image ***t***ranslation network and a ***t***ask prediction network. The upper portion shows the training pipeline with synthetic (x_s, y_s) pairs, while the lower portion shows the training pipeline with real images x_r. Note that both pipelines share identical weights for the $G_{S\to R}$ network, and likewise for the f_T network. More specifically:

- For real images, we want $G_{S\to R}$ to behave as an AE [15] and apply minimal change to the images, and thus use a *reconstruction loss*.
- For synthetic data, we want $G_{S\to R}$ to translate synthetic images into the real-image domain, and use a *GAN loss* via discriminator D_R on the output.

The translated images are next passed through f_T for depth prediction, and then compared to the synthetic ground truth depths y_s via a *task loss*.

- In addition, we also propose that the inner feature representations of f_T should share similar distributions for both real and translated images, which can be implemented through a feature-based GAN via D_{feat}.

Note that one key benefit of this framework is that it can and should be trained end-to-end, with the weights of $G_{S\to R}$ and f_T simultaneously optimized.

2.4.1 Synthesis Loss

Intuitively, the gap between synthetic and realistic imagery comes from low-level differences such as color and texture (*e.g.* of trees, roads), rather than high-level geometric and semantic differences. To bridge this gap between the two domains, an ideal translator network, for use within an image-to-depth framework, needs to output images that are impossible to be distinguished from real images and yet retain the original scene geometry present in the synthetic input images. The distribution of real-world images can be replicated using adversarial learning, where a generator $G_{S\to R}$ tries to transform a synthetic image x_s to be indistinguishable from real images of x_r, while a discriminator D_R aims to differentiate between the generated image $\hat{x}_s$ and real images x_r. Following the typical GAN approach [65], we model this minimax game using an *adversarial loss* given by

$$\mathcal{L}_{\text{GAN}}(G_{S\to R}, D_R) = \mathbb{E}_{x_r\sim X_R}[\log D_R(x_r)] + \mathbb{E}_{x_s\sim X_S}[\log(1 - D_R(G_{S\to R}(x_s)))] \tag{2.1}$$

where generator and discriminator parameters are updated alternately.

However, a vanilla GAN is insufficiently constrained to preserve scene geometry [87]. To regularize this in a principled manner, we want generator $G_{S\to R}$ to behave as a *wide-spectrum* translator, able to take in both real and synthetic imagery, and in both cases produce real imagery. When the input is a real image, we would want the image to remain as much unchanged perceptually, and a *reconstruction loss*

$$\mathcal{L}_r(G_{S\to R}) = ||G_{S\to R}(x_r) - x_r||_1 \tag{2.2}$$

is applied when the input to $G_{S\to R}$ is a real image x_r. Note that while this may bear some resemblance to the use of reconstruction losses in CycleGAN [234] and α-GAN [161], ours is a unidirectional forward loss, and not a cyclical loss.

2.4.2 Task Loss

After a synthetic image x_s is translated, we obtain a generated realistic image $\hat{x}_s$, which can still be paired to the corresponding synthetic depth map y_s. This paired translated data $(\hat{x}_s, y_s)$ can be used to train the task network f_T. Following convention, we directly measure per-pixel difference between the predicted depth map and the synthetic (ground truth) depth map as a task loss:

$$\mathcal{L}_t(f_T) = ||f_T(\hat{x}_s) - y_s||_1. \tag{2.3}$$

We also regularize f_T for real training images. Since real ground truth depth maps are not available during training, a locally smooth loss is introduced to guide a more reasonable depth estimation, in keeping with [55, 63, 75, 106]. As depth discontinuities often occur at object boundaries, we use a robust penalty with an edge-aware term to optimize the depths, similar to [63]:

$$\mathcal{L}_s(f_T) = |\partial_x f_T(x_r)|e^{-|\partial_x x_r|} + |\partial_y f_T(x_r)|e^{-|\partial_y x_r|} \tag{2.4}$$

where x_r is the real-world image, and noting that f_T share identical weights in both real and synthetic input pipelines.

In addition, we also want the internal feature representations of real and translated synthetic images in the encoder-decoder network of f_T to have similar distributions [54]. In theory, the decoder portion of f_T should generate similar prediction results from the two domains when their feature distributions are similar. Thus we further define a feature-level GAN loss as follows:

$$\mathcal{L}_{\text{GAN}_f}(f_T, D_{\text{feat}}) = \mathbb{E}_{f_{\hat{x}_s} \sim f_{\hat{X}_s}}[\log D_{\text{feat}}(f_{\hat{x}_s})] + \mathbb{E}_{f_{x_r} \sim f_{X_r}}[\log(1 - D_{\text{feat}}(f_{x_r}))] \tag{2.5}$$

where $f_{\hat{x}_s}$ and f_{x_r} are features obtained by the encoder portion of f_T for translated-synthetic images and real images respectively. As noted in [65], the optimal solution measures the Jensen-Shannon divergence between the two distributions.

2.4.3 Full Objective

Taken together, our full objective is:

$$\begin{aligned}\mathcal{L}_{\text{T}^2\text{Net}}(G_{S\to R}, f_T, D_R, D_{\text{feat}}) =& \mathcal{L}_{\text{GAN}}(G_{S\to R}, D_R) + \alpha_f \mathcal{L}_{\text{GAN}_f}(f_T, D_{\text{feat}}) \\ &+ \alpha_r \mathcal{L}_r(G_{S\to R}) + \alpha_t \mathcal{L}_t(f_T) + \alpha_s \mathcal{L}_s(f_T)\end{aligned} \tag{2.6}$$

where $\mathcal{L}_{\text{GAN}}$ encourages translated synthetic images to appear realistic, $\mathcal{L}_r$ spurs translated real images to appear identical, $\mathcal{L}_{\text{GAN}_f}$ enforces closer internal feature distributions, $\mathcal{L}_t$ promotes accurate depth prediction for synthetic pairs, and $\mathcal{L}_s$ prefers an appropriate local depth variation for real predictions. In our end-to-end training, this objective is used in solving for optimal f_T parameters:

$$f_T^* = \arg\min_{f_T} \min_{G_{S\to R}} \max_{D_R, D_{\text{feat}}} \mathcal{L}_{\text{T}^2\text{Net}}(G_{S\to R}, f_T, D_R, D_{\text{feat}}). \tag{2.7}$$

2.4.4 Network Architecture

The transform network, $G_{S\to R}$, is a residual network (ResNet) [74] similar to SimGAN [172]. Limited by memory constraints and the large size of scene images, one down-sampling layer is used in our model and the output is only passed through 6 blocks. For the image discriminator networks, we use PatchGANs [172, 234], which have produced impressive results by discriminating locally whether image patches are real or fake.

The task prediction network is inspired by [63], which outputs four predicted depth maps of different scales. Instead of encoding input images into very small dimensions to extract global information, we instead use multiple dilation convolutions [212] with a large feature size to preserve fine-grained details. In addition, we employ different weights for the paths with skip connections [160], which can simultaneously process larger-scale semantic information in the scene and yet also predict detailed depth maps. The use of these techniques allows our task prediction network f_T to achieve state-of-the-art performance in our own real-supervised benchmark method (training f_T on pairs of real images and depth), even when the encoder portion of f_T is primarily based on VGG, as opposed to a more typical ResNet50-type network used in other methods [63, 106].

2.5 Data Collection

As mentioned above, in this work, we assume the synthetic images and the corresponding depth maps are easily obtained due to more and more 3D CAD models are publicly available. In the experiments, we collected both indoor and outdoor synthetic datasets.

Synthetic Indoor Dataset To generate the paired synthetic training data, we rendered RGB images and depth maps from the SUNCG dataset [177], which contains 45,622 3D houses with various room types and all 3D CAD models are publicly available at the time of publication. We chose the camera locations, poses, and parameters based on the distribution of real NYUDv2 dataset [173]. We retained valid depth maps using the criteria presented in [177]: a) valid depth area (depth values in the range of 1m to 10m) larger than 70% of the image area, and b) more than two object categories in the scene. The RGB images are rendered using the default OpenGL-rendered method, resulting in low realism[3]. Therefore, our synthetic to realistic translation approach is applied to fit the synthetic images to real. Theoretically, we can render infinite numbers of paired images for training with the whole 3D CAD models of scenes. In this work, we generated 130,190 valid views from 4,562 different houses.

Synthetic Outdoor Dataset We used Virtual KITTI (vKITTI) [53], a photo-realistic synthetic dataset that contains 21,260 image-depth paired frames generated from different virtual urban worlds. The scenes and camera viewpoints are similar to the real KITTI dataset [133]. However, the ground truth depths in vKITTI and KITTI are quite different. The maximum sensed depth in a real KITTI image is typically on the order of 80m, whereas vKITTI has precise depths to a maximum of 655.3m because it is rendered from Unity game engine without equipment limitation. To reduce the effect of ground truth differences, the vKITTI depth maps were clipped to 80m.

2.6 Experiment

We evaluated our model on the outdoor KITTI dataset [57] and the indoor NYU Depth v2 dataset [173]. During the training process, we only used unpaired real images from these datasets in conjunction with synthetic image-depth pairs, obtained via SUNCG [177] and vKITTI [53] datasets, in our proposed framework.

2.6.1 Implementation Details

Training Details In order to control the effect of GAN loss, we substituted the vanilla negative log likelihood objective with a least-squares loss [129], which has

[3]In Chapter 6, a high-quality image rendering pipeline is introduced using Maya [3], yet it still can not fully match a given real dataset.

proven to be more stable during adversarial learning [234]. Hence, for GAN loss $\mathcal{L}_{\text{GAN}}(G_{S\to R}, D_R)$ in (2.1), we trained $G_{S\to R}$ by minimizing

$$\mathbb{E}_{x_s\sim X_s}[(D_R(G_{S\to R}(x_s)) - 1)^2]$$

and trained D_R by minimizing

$$\mathbb{E}_{x_r\sim X_r}[(D_R(x_r) - 1)^2] + \mathbb{E}_{x_s\sim X_s}[D_R^2(G_{S\to R}(x_s))].$$

A similar procedure was also applied for the GAN loss in (2.5).

Our f_T-only Benchmark Models Besides our full T^{2}Net model, we also tested our partial model, which comprised solely the f_T task prediction network. We evaluated this in two scenarios: (1) an **"all-real"** scenario, in which we used real image and depth map pairs for training, for which we would expect to *upper bound* our full model performance, and (2) an **"all-synthetic" (naive version)** scenario, in which we used only synthetic image-depth pairs and eschewed even unpaired real images, for which we would expect to *lower bound* our full model performance.

Evaluation Metrics We evaluated the performance of our approach using the depth evaluation metrics reported in [46]:

$$\begin{aligned}
&\textbf{RMSE(log)}: \sqrt{\tfrac{1}{|T|}\textstyle\sum_{i=1}^{T} ||\log \hat{y}_{r,i} - \log y_{r,i}||^2} && \textbf{RMSE}: \sqrt{\tfrac{1}{|T|}\textstyle\sum_{i=1}^{T} ||\hat{y}_{r,i} - y_{r,i}||^2} \\
&\textbf{Sq. relative}: \tfrac{1}{|T|}\textstyle\sum_{i=1}^{T} ||\hat{y}_{r,i} - y_{r,i}||^2 / y_{r,i} && \textbf{Abs relative}: \tfrac{1}{|T|}\textstyle\sum_{i=1}^{T} |\hat{y}_{r,i} - y_{r,i}| / y_{r,i} \\
&\textbf{Accuracy}: \ \%\ \textbf{of}\ \mathbf{y_{r,i}}\ \textbf{s.t.}\ max(\tfrac{\hat{y}_{r,i}}{y_{r,i}}, \tfrac{y_{r,i}}{\hat{y}_{r.i}}) = \delta < thr
\end{aligned} \tag{2.8}$$

2.6.2 NYUDv2 Dataset

Translated Results Figure 2.3 shows sample output from translation through $G_{S\to R}$. We observe that the visual differences between synthetic and real images are obvious: colors, textures, illumination and shadows in real scenes are more complex than in synthetic ones. Compared to synthetic images, the translated images are visually more similar to real images in terms of low-level appearance.

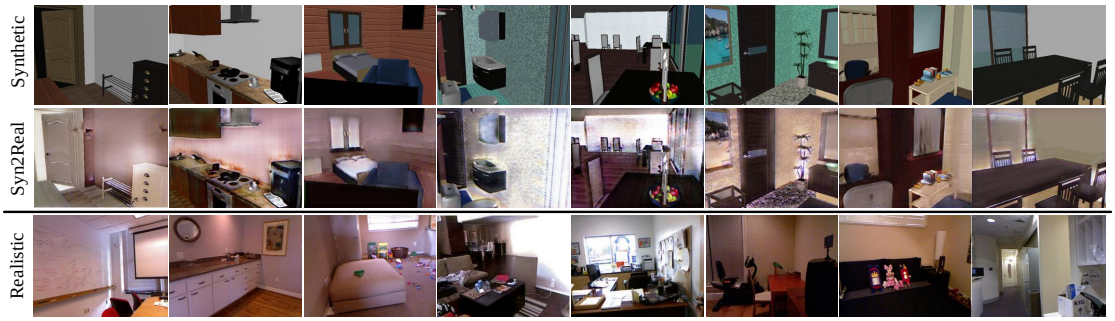


FIGURE 2.3: **Example outputs of our translation network for indoor scene.** Top: synthetic images rendered from SUNCG. Middle: corresponding images after $G_{S\to R}$ translation. Bottom: real images from NYUDv2 [173] (no correspondence to above rows).

Method	↓				↑		
	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Ladicky et al. [107]	-	-	-	-	0.542	0.829	0.940
Eigen et al. [46] Fine	0.215	0.212	0.907	0.285	0.611	0.887	0.971
Liu et al. [122]	0.213	-	0.759	-	0.650	0.906	0.976
Eigen et al. [45] (VGG)	0.158	0.121*	0.641	0.214	0.769	0.950*	0.988*
Baseline, train set mean	0.439	0.641	1.148	0.415	0.412	0.692	0.856
Our f_T, all-real	0.157*	0.125	0.556*	0.199*	0.779*	0.943	0.983
Our f_T, all-synthetic	0.304	0.394	1.024	0.369	0.458	0.771	0.916
Our T^{2}Net, D_{feat} only	0.320	0.405	0.991	0.343	0.480	0.792	0.933
Our T^{2}Net, D_{image} only	0.274	0.336	1.001	0.325	0.496	0.814	0.938
Our full T^{2}Net	**0.257**	**0.281**	**0.915**	**0.305**	**0.540**	**0.832**	**0.948**

TABLE 2.1: **Depth estimation results on NYUDv2 dataset [173].** *Gray rows indicate methods in which training is conducted **without** real image-depth pairs. Best supervised results are marked with *, while best unsupervised results are in bold.* ↓ = lower is better. ↑ = higher is better.

Depth Estimation Results In Table 2.1, we report the performance of our models (varying different applications of the two GANs) as compared to latest state-of-the-art methods on the public NYUDv2 dataset. In the indoor dataset, these previous works were all based on supervised learning with real image-depth pairs. The gray rows highlight methods in which real image-depth pairs were *not* used in training. The **train-set-mean** baseline used the mean synthetic depth map in the training dataset as prediction, with the results providing an indication of the correlation between depth maps in the synthetic and real datasets. We also present results from our f_T-only benchmark models in the "all-real" and "all-synthetic" setups, which we expect to provide the upper bound and lower bound of our model respectively.

Our proposed models produced a clear gap to the train-set-mean baseline and the synthetic-only benchmark. While our models were unable to outperform the

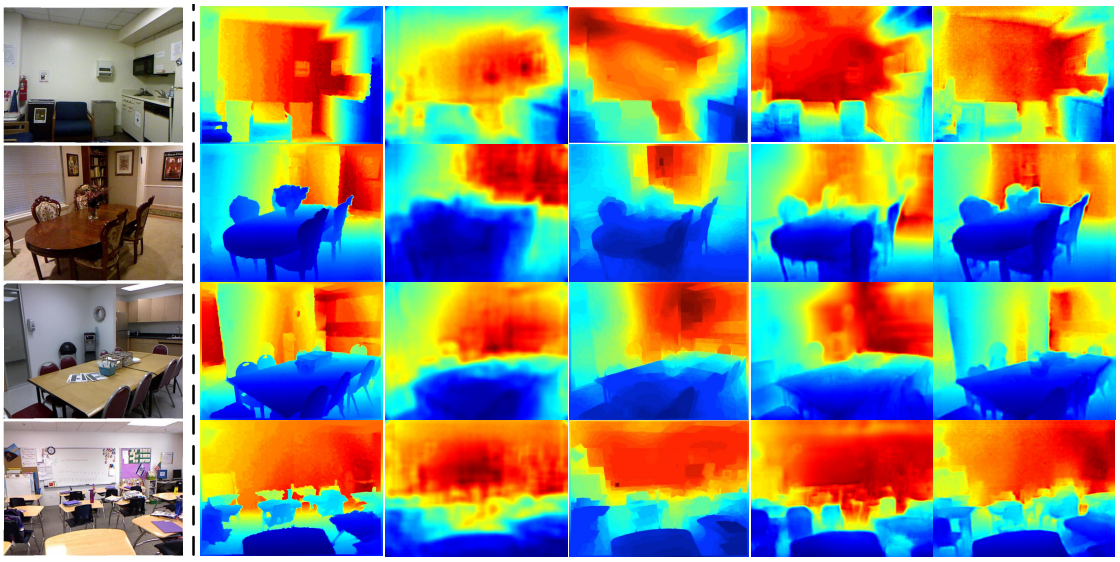


FIGURE 2.4: **Qualitative results on NYUDv2.** All results are shown as relative depth maps (red = far, blue = close).

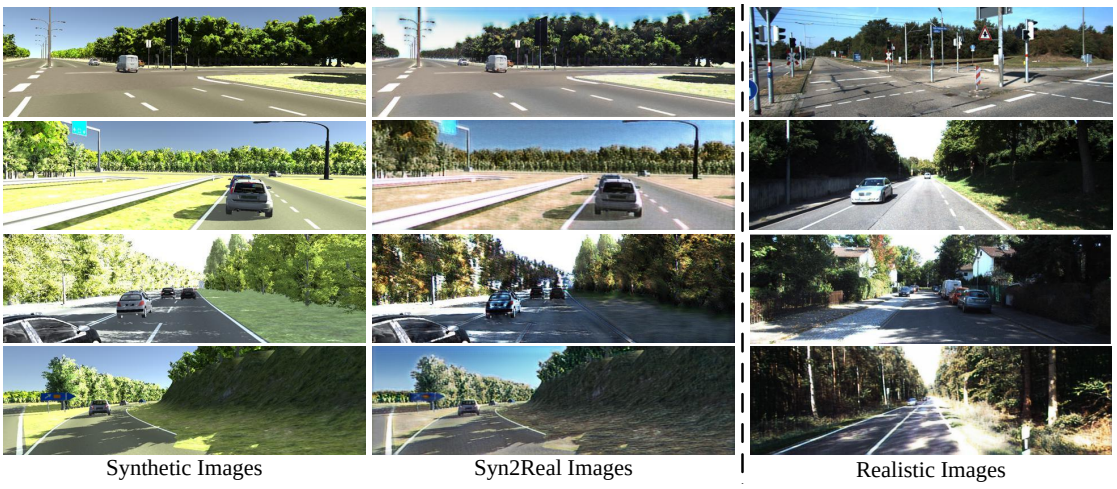


FIGURE 2.5: **Example translated images for the outdoor vKITTI dataset [53].** (Left) synthetic images from vKITTI and translated images. (Right) images in real KITTI.

latest fully-supervised methods trained on real paired data, the full T^{2}Net model was even able to outperform the earlier supervised learning method of [107] on two of the three metrics, despite not using real paired data.

We also show qualitative results in Figure 2.4. Although the absolute values of our predicted depths were not as accurate as the latest supervised learning methods, we observe that our T^{2}Net model generates reasonably good relative depths with distinct furniture shapes, even without using real paired training data.

Method	Dataset	Cap	↓				↑		
			Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen et al.[46] Fine	K(I+D)	0-80m	0.190	1.515	7.156	0.270	0.692	0.899	0.967
Garg et al.[55] L12 Aug.8x	K(L+R)	1-50m	0.169	1.080	5.104	0.273	0.740	0.904	0.962
Godard et al. [63]	CS+K(L+R)	1-50m	0.117	0.762	3.972	0.206	0.860	0.948	0.976
Kuznietsov et al. [106]	K(D+L+R)	1-50m	0.108*	0.595*	3.518*	0.179	0.875*	0.964*	0.988*
Baseline, train set mean	vK(I+D)	1-50m	0.521	11.024	10.598	0.473	0.638	0.755	0.835
Our f_T, all-real	K(I+D)	1-50m	0.114	0.627	3.549	0.178*	0.867	0.960	0.986
Our f_T, all-synthetic	vK(I+D)	1-50m	0.278	3.216	6.268	0.322	0.681	0.854	0.929
Our T^{2}Net, D_{feat} only	vK(I+D) + K(I)	1-50m	0.233	2.902	6.285	0.300	0.743	0.880	0.938
Our T^{2}Net, D_{image} only	vK(I+D) + K(I)	1-50m	**0.168**	**1.199**	**4.674**	**0.243**	**0.772**	**0.912**	**0.966**
Our full T^{2}Net	vK(I+D) + K(I)	1-50m	0.169	1.230	4.717	0.245	0.769	**0.912**	0.965

TABLE 2.2: **Results on KITTI 2015 [133]** using the split of Eigen *et al.* [46]. For dataset, K is the real KITTI dataset [133], CS is Cityscapes [31] and vK is the synthetic KITTI dataset [53]. L, R are the left and right stereo images, and I, D are the images and depths. *The gray rows highlight methods that did not use real image-depth pairs nor stereo pairs for training. Best real-supervised or stereo-based results are marked with *, while best unsupervised results are in bold.* ↓ = lower is better. ↑ = higher is better.

2.6.3 KITTI Dataset

Translated Results Figure 2.5 shows examples of synthetic, translated, and real images from the outdoor datasets. As shown, the translated images have substantially greater resemblance to the real images than the synthetic images. Our translation network can visually replicate the distributions of colors, textures, shadows and other low-level features present in the real images, and meanwhile preserve the scene geometry of the original synthetic images.

Depth Estimation Results In order to compare with previous work, we used the test split of 697 images proposed in [46]. Following [63], we chose 22,600 RGB images from the remaining 32 scenes for training the translation network. As before, we did not use real depths nor stereo pairs in our T^{2}Net models. The ground truth depth maps in KITTI were obtained by aligning laser scans with color images, which produced less than 5% depth values and introduced sensor errors. For fair comparison with state-of-the-art single view depth estimation methods, we evaluated our results based on the cropping given in [55] and clamping the predicted depth values within the range of 1–50m.

Table 2.2 shows quantitative results of testing with real images of the KITTI dataset. We can observe that the performance of T^{2}Net has a substantial 9.1% absolute improvement compared to our all-synthetic trained model. Unlike the indoor results, the best performance comes from without D_{feat}. This is likely due to the translated images much closer to real KITTI, which does not need to match the feature distribution using D_{feat} adversarial learning. We also observe that our

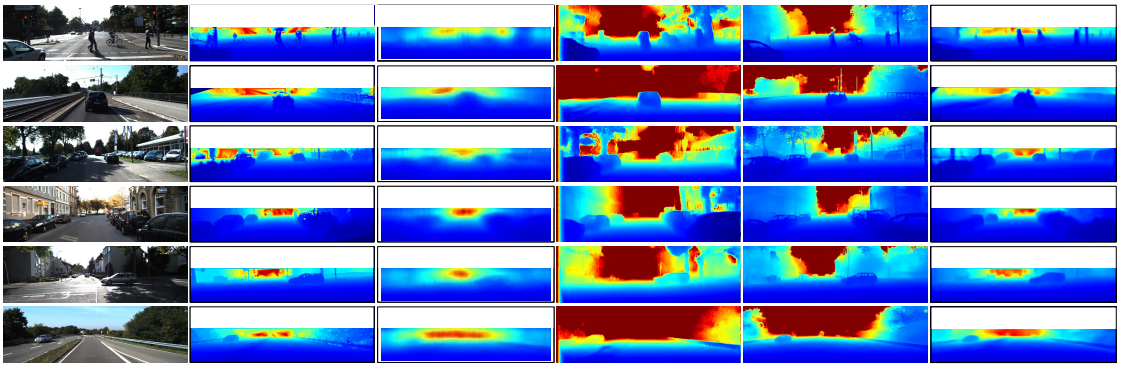


FIGURE 2.6: **Qualitative results on KITTI** with Eigen split [46]. The ground truth depths in the original dataset were very sparse and have been interpolated for visualization. We converted the disparity maps provided in [63] to depth maps.

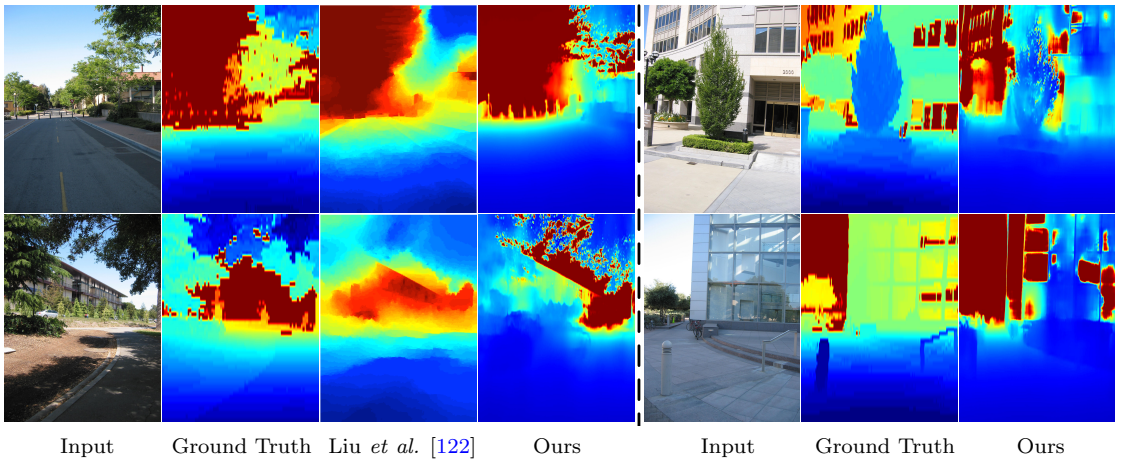


FIGURE 2.7: **Qualitative results on Make3D [165].** For most cases the model generated reasonable depths except scenes with new object types not present in the synthetic data.

model. despite training without real paired data, is able to outperform the method of [46] trained on real paired image-depth data, as well as the method of [55] trained on real left-right stereo data.

We also qualitatively compared the performance of the proposed model with the state-of-the-art in Figure 2.6. We only chose two representatives that either used real paired color-depth images [46], or real left-right stereo images [63]. Compared to [46], our model can generate full dense depth maps of input image size. Our method is also able to detect more detail at object boundaries than [63], with a likely reason being that the synthetic training depth maps preserved object details better. Another interesting observation is the predicted depth maps were treating glass windows as permeable based on synthetic data, while they were mostly sensed as opaque in the laser-based ground truth.

Input SimGAN [172] CycleGAN [235] Ours (no rec.) Ours (with rec.)

FIGURE 2.8: **Ablation study for different translation networks**. The qualitative results of different unpaired image-to-image translation methods trained using vKITTI and real KITTI dataset.

Method	↓				↑		
	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
baseline, synthetic only	0.278	3.216	6.268	0.322	0.681	0.854	0.929
vanilla task network, synthetic only	0.295	3.793	8.403	0.363	0.600	0.817	0.912
vanilla task network, full approach	0.259	2.891	6.380	0.324	0.694	0.853	0.927
separated training	0.234	2.706	6.068	0.293	0.747	0.882	0.942
separated training with CycleGAN	0.212	1.973	5.340	0.269	0.750	0.895	0.952
self-domain reconstruction	0.199	1.517	5.349	0.298	0.695	0.866	0.9420
No reconstruction loss(epoch 3)	0.201	1.941	5.619	0.286	0.741	0.882	0.945
No feature loss	**0.168**	**1.199**	**4.674**	**0.243**	**0.772**	**0.912**	**0.966**
No image GAN loss	0.233	2.902	6.285	0.300	0.743	0.880	0.938
our full approach	0.169	1.230	4.717	0.245	0.769	0.912	0.965

TABLE 2.3: **Quantitative results of different variants of our T^{2}Net** on KITTI using the split of [46]. All methods are trained without the real-world ground truth depth map. ↓ = lower is better. ↑ = higher is better.

2.6.4 Performance on Make3D

To compare the generalization ability of our T^{2}Net to a different test dataset, we used our full T^{2}Net model, trained only on vKITTI paired data and (unpaired) real KITTI images, for testing on the Make3D dataset [165]. We evaluated our model quantitatively on Make3D using the standard C1 metric. The RMSE(m) accuracy is 8.935, Log-10 is 0.574, Abs Rel is 0.508 and Sqr Rel is 6.589. The qualitative results presented in Figure 2.7 show that our model can generate reasonable depth maps in most situations. The right part of Figure 2.7 displays some failure cases, likely due to large building windows not being widely observed in the vKITTI datasets.

2.6.5 Ablation Study

We evaluated the contribution of different design choices in the proposed T^{2}Net. Table 2.3 shows the quantitative results and Figure 2.8 shows some example outputs of different methods for unpaired image translation.

Method	↓				↑		
	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
baseline, synthetic only	0.304	0.394	1.024	0.369	0.458	0.771	0.916
separated training	0.288	0.364	1.095	0.352	0.463	0.768	0.902
separated training with CycleGAN	0.280	0.362	0.971	0.355	0.478	0.777	0.921
self-domain reconstruction	0.287	0.352	0.968	0.351	0.491	0.782	0.934
No reconstruction loss(epoch 3)	0.278	0.341	0.942	0.345	0.514	0.808	0.929
No feature loss	0.274	0.336	1.001	0.325	0.496	0.814	0.938
No image GAN loss	0.320	0.405	0.991	0.343	0.480	0.792	0.933
our full approach	**0.257**	**0.281**	**0.915**	**0.305**	**0540**	**0.832**	**0.948**

TABLE 2.4: **Quantitative results of different variants of our T^{2}Net** on NYUv2 dataset [173]. All methods are trained without the real-world ground truth depth map. ↓ = lower is better. ↑ = higher is better.

End-to-End *vs* Separated We began by evaluating the effect of end-to-end learning. We found that end-to-end training outperformed separated training of the translation network and task prediction network. One reasonable explanation is that task loss is a form of supervised loss for synthetic-to-realistic translation. This incentivizes the translation network to preserve geometric content present in a synthetic image.

We also experimented with the unpaired image translation network CycleGAN [234]. This model has two encoder-decoder translation networks and two discriminators, but we were limited by machine memory and trained the CycleGAN and task network separately. From Figure 2.8, we found that while this model generated very visually realistic images, it also created some realistic-looking details that significantly distorted scene geometry. The quantitative performance is close to our separated training results.

No Image Reconstruction We studied what happens when training without real-image reconstruction loss. In Figure 2.8, we may surmise that the task loss in the depth domain is able to encourage reasonable depiction of scene geometry in the translation network. However, the lack of a real image reconstruction loss appears to make it harder to generate high-resolution images. In addition, we noticed that while the removal of reconstruction loss still led to relatively good results as seen in Table 2.3 and 2.4, this was only true in early training with best results in epoch 3, with accuracy dropping after more training epochs.

Target Reconstruction *vs* Self-Regularization Since the self-regularization component of SimGAN is closest to our target-domain reconstruction concept, we also trained our full model with L1 reconstruction loss for synthetic imagery, which

forces the generated target images to be similar to original input images. From Figure 2.8, we observe that this is unable to work well for large domain shifts, for the GAN loss and self-domain reconstruction loss play opposite roles in the translation.

2.7 Limitations and Discussion

A novel, end-to-end trainable T^{2}Net deep neural network is presented for single-image depth estimation, that requires only synthetic image-depth pairs and unpaired real images for training. The overall system comprises an image translation network and a depth prediction network. It is able to generate realistic images via a learning framework that combines adversarial loss for synthetic input and target-domain reconstruction loss for real input in the translation network, and a further combination of a task loss and feature GAN loss in the depth prediction network. The T^{2}Net can be trained end-to-end, and does not require real image-depth pairs nor stereo pairs for training. It is able to produce good results on the NYUDv2 and KITTI datasets despite the lack of access to real paired training data, and even outperformed early deep learning methods that were trained on real paired data. Many recent works [25, 156, 225] have also begun to explore the single-image depth estimation on different datasets. In particular, Zhao *et al.* [225] and Chen *et al.* [25] follow our experiment setting to address the gap between synthetic and real domain, and consistently consider our method as a state-of-the-art benchmark for single image depth estimation using only synthetic ground truth depth. In the future, we intend to explore mechanisms that provide greater generalization capability across different datasets.

While the proposed *wide-spectrum* translation network works well on this synthetic-to-realistic translation task, it requires joint training with the task network, which ensures *depth / structure* consistency during the end-to-end training. However, it is not always the case that a complementary task is available to support an I2I problem. For example, it remains challenging to explicitly model the *content* and *style* for I2I translation. In Chapter 3, we will introduce a spatially-correlative loss, which can explicitly extract the structure representation to allow preservation of scene structure consistency during the translation when appearance may dramatically change.

Chapter 3

Spatially-Correlative Loss for Various Image Translation Tasks

The previous *wide-spectrum* translation network works well only when translation and task networks are jointly optimized, in which the task network can provide a geometry loss to support synthesis with depth consistency. However, it is not scalable to many I2I translation scenarios, where only unpaired images in the two domains are available. As the goal in I2I translation is to modify the input image to fit the *style / appearance* of the target domain, while preserving the original *content / structure*, learning to assess the *content* and *style* correctly is thus of central importance. In this chapter, a novel spatially-correlative loss is proposed that is simple, efficient, and yet effective for preserving scene structure consistency. Previous methods attempt this by using pixel-level cycle-consistency or feature-level matching losses, but the domain-specific nature of these losses hinder translation across large domain gaps. To address this, we exploit the spatial patterns of self-similarity as a means of defining scene structure. The spatially-correlative loss is geared towards only capturing spatial relationships within an image, rather than domain appearance. A new self-supervised learning method is also introduced to explicitly learn spatially-correlative maps for each specific translation task. We show distinct improvement over baseline models in all three modes of unpaired I2I translation: single-modal, multi-modal, and even single-image translation.

We first introduce the motivation in Section 3.1 and review previous works in Section 3.2. Section 3.3 explains how to calculate the spatially-correlative loss for I2I translation tasks and Section 3.4 demonstrates the superiority of the proposed loss. We discuss the loss in Section 3.5.

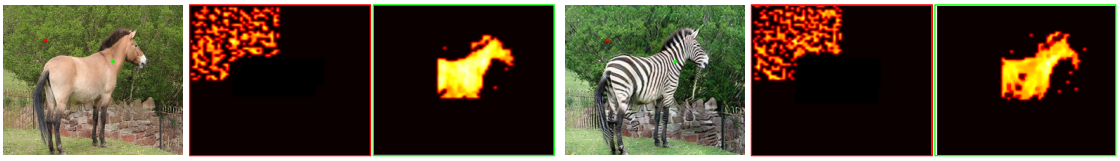


FIGURE 3.1: **Our learned spatially-correlative representation** encodes local scene structure based on self-similarities. Despite vast appearance differences between the *horse* and *zebra*, when the scene structures are identical (*i.e.* same poses), the spatial patterns of self-similarities are as well.

3.1 Introduction

I2I translation refers to the task of modifying an input image to fit the *style / appearance* of the target domain, while preserving the original *content / structure* (as shown in Figure 3.1: *horse* $\rightarrow$ *zebra*); learning to assess the *content* and *style* correctly is thus of central importance. While GANs [65] have the ability to generate images that adhere to the overall dataset distribution, it is still difficult to preserve scene structure during translation when image-conditional GANs are optimized with purely adversarial loss.

To mitigate the issue of scene structure discrepancies, a few loss functions for comparing the content between input and output images have been proposed, including (a) *pixel-level* image reconstruction loss [23, 87, 172] and cycle-consistency loss [101, 209, 234]; (b) *feature-level* perceptual loss [43, 93] and PatchNCE loss [143]. However, these losses still have several limitations. First, pixel-level losses do *not* explicitly decouple structure and appearance. Second, feature-level losses help but continue to conflate domain-specific structure and appearance attributes. Finally, most feature-level losses are calculated using a fixed ImageNet [35] pre-trained network (*e.g.* VGG16 [174]), which will not correctly adapt to arbitrary domains.

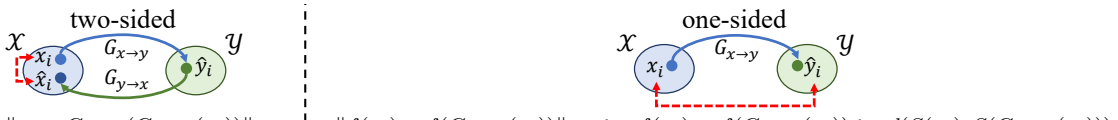
In this chapter, we aim to design a *domain-invariant* representation to precisely express scene structure, rather than using original pixels or features that couple both appearance and structure. To achieve this, we propose to revisit the idea of *self-similarity*. Classically, low-level self-similarity has been used for matching [169] and image segmentation [170], while feature-level self-similarity in deep learning manifests as self-attention maps [200]. We propose to go further, to advance an assumption that *all* regions within same categories exhibit some form of self-similarity. For instance, while the horse and zebra in Figure 3.1 appear very different, there is obvious visual self-similarity in their own regions. We believe a network can learn deeper representations of self-similarities (beyond just visual

ones) that can encode intact object shapes, even when there are variations in appearances within an object. Then through estimating such co-occurrence signals in self-similarity, we can explicitly represent the structure as multiple *spatially-correlative* maps, visualized as heat maps in Figure 3.1. Based on this within-shape self-similarity, we propose then that *a structure-preserving image translation will retain the patterns of self-similarity in both the source and translated images, even if appearances themselves change dramatically.*

Our basic spatially-correlative map, called *FSeSim*, is obtained by computing the **F**ixed **Se**lf-**Sim**ilarity of features extracted from a pre-trained network. While this basic version achieved comparative or even better results than state-of-the-art methods [52, 143, 234] on some tasks, the generality is limited because features extracted from an ImageNet pre-trained network are biased towards photorealistic imagery. Hence, this will not optimally work with images in non-realistic styles.

To obtain a more general spatially-correlative map, the **L**earned **Se**lf-**Sim**ilarity, called *LSeSim*, is presented by using a form of contrastive loss, in which we explicitly encourage homologous structures to be closer, regardless of their appearances, and reciprocally dissociate dissimilar structures even they have similar appearances. To do this, the model learns a domain-invariant spatially-correlative map, where having the same scene structure leads to similar maps, even if the images are from different domains.

There are several advantages of using the proposed F/LSeSim loss: (a) In contrast to the existing losses that directly compare the loss on pixels [234] or features [93], F/LSeSim captures the domain-invariant structure representation, regardless of the absolute pixel values; (b) Through contrastive learning, the LSeSim learns a spatially-correlative map for a specific image translation task, rather than features extracted from a fixed pre-trained network, as in *e.g.* perceptual loss [93], contextual loss [132]; (c) The translation model is more efficient and faster than the widely used cycle-consistency architectures, because our F/LSeSim explicitly encodes the structure, bypassing the expensive multi-cycle looping; (d) As we show in Figure 3.5, our F/LSeSim correctly measures the structural distance even when the two images are in completely different domains; (e) Finally, our F/LSeSim can easily be integrated into various frameworks. In our experiments, we directly used the generator and discriminator architectures of CycleGAN [234], MUNIT [84] and StyleGAN [96, 97] for extensive I2I translation tasks. The experimental results show that our model outperformed the existing both one-sided translation methods [2, 11, 52, 143] and two-sided translation methods [84, 209, 234].



$\|x_i - G_{y\to x}(G_{x\to y}(x_i))\|_p$ (a) DiscoGAN [101], CycleGAN [234], DualGAN [209], MUNIT [84], StarGAN [28], DRIT++ [112] ...

$\|f(x_i) - f(G_{x\to y}(x_i))\|_p$ (b) DeePSiM [43], StyleNet [185], PerceptualLoss [93], SimGAN [172], ContextualLoss [132], TSIT [90] ... CUT [143] ...

$f(x_i) \propto f(G_{x\to y}(x_i))$ (c) DistanceGAN [11], TraVelGAN [2], GcGAN [52], ...

$d(S(x_i), S(G_{x\to y}(x_i)))$ (d) Ours

FIGURE 3.2: **Comparison of unpaired I2I translation methods with various content losses**. (a) The cycle-consistency loss [101, 209, 234] in a two-sided framework. (b) Pixel-level image reconstruction loss [172] and feature-level matching loss [93]. (c) Various indirect relationships [11, 52] between the input and output. (d) Our spatially-correlative loss based on a learned spatially-correlative map.

3.2 Background

Existing unpaired I2I translation either use cycle-consistency loss in a two-sided framework [101, 209, 234], or other forms of pixel-level and feature-level losses in a one-sided framework [2, 11, 52] for preserving content (Figure 3.2).

Two-Sided Unsupervised Image Translation *Cycle-consistency* has become a de facto loss in most works, whether the cycles occur in the image domain [28, 79, 101, 111, 209, 234], or in latent space [84, 112, 235]. However, without explicit constraints, the content in a translated image can be easily distorted [111]. Furthermore, the cycle-based methods require auxiliary generators and discriminators for the reverse mapping.

One-Sided Unsupervised Image Translation To avoid cycle-consistency artifacts, DistanceGAN [11] and GcGAN [52] pre-define an implicit distance in a one-sided framework. In contrast, the feature-level losses [93, 132] evaluate the content distance in a deep feature space, which have been applied in both style transfer [56, 93, 132, 222] and image translation [23, 90, 144, 193]. However, the underlying assumption that high-level semantic information is solely determined in feature space does not always hold. Furthermore, these features are extracted from a fixed pre-trained network (*e.g.* VGG16 [174]). While the latest CUT [143] learns a PatchNCE loss for a specific task, the distance used is directly computed from extracted features, and will still be affected by domain-specific peculiarities.

Contrastive Representation Learning Driven by the potential of discriminative thought, a series of self-supervised methods [4, 24, 71, 76, 78, 141, 199] have emerged in recent years. These self-supervised methods learn robust features by

associating "positive" pairs and dissociating "negative" pairs. CUT [143] first introduced contrastive learning for unsupervised I2I translation. While we utilize a patch-wise contrastive loss within an image in a similar manner to CUT, we propose a systematic way to learn a structure map that excludes appearance attributes. As described below, our LSeSim method learns a domain-independent structure representation.

3.3 Approach

As shown in Figure 3.2, given a collection of images $\mathcal{X} \subset \mathbb{R}^{H \times W \times C}$ from a particular domain (*e.g.* horse), our main goal is to learn a model Φ that receives the image $x \in \mathcal{X}$ as input and transfers it into the target domain $\mathcal{Y} \subset \mathbb{R}^{H \times W \times C}$ (*e.g.* zebra), in a manner that retains the original scene structure but converts the appearance appropriately.

Here, we focus on designing a loss function that measures the structural similarity between the input image x and the translated image $\hat{y} = \Phi(x)$. However, unlike most existing approaches that directly attempt to evaluate the structural similarity between input and translated images at some deep feature level, we will instead compute the *self-similarity* of deep features *within each image*, and then *compare the self-similarity patterns* between the images.

In subsequent sections, we investigate two losses, *fixed self-similarity (FSeSim)* and *learned self-similarity (LSeSim)*. In the first instance, we directly compare the self-similarity patterns of features extracted from a fixed pre-trained network (*e.g.* VGG16 [174]). In the second instance, we additionally introduce a structure representation model that learns to correctly compare the self-similarity patterns, in which we use the contrastive infoNCE loss [141] to learn such a network without label supervision.

3.3.1 Fixed Self-Similarity (FSeSim)

We first describe our fixed spatially-correlative loss. Given an image x in one domain and its corresponding translated image $\hat{y}$ in another, we extract the features f_x and $f_{\hat{y}}$ using a simple network (*e.g.* VGG16 [174]). Instead of directly computing the feature distance $\|f_x - f_{\hat{y}}\|_p$, we compute the self-similarity in the form of a map.

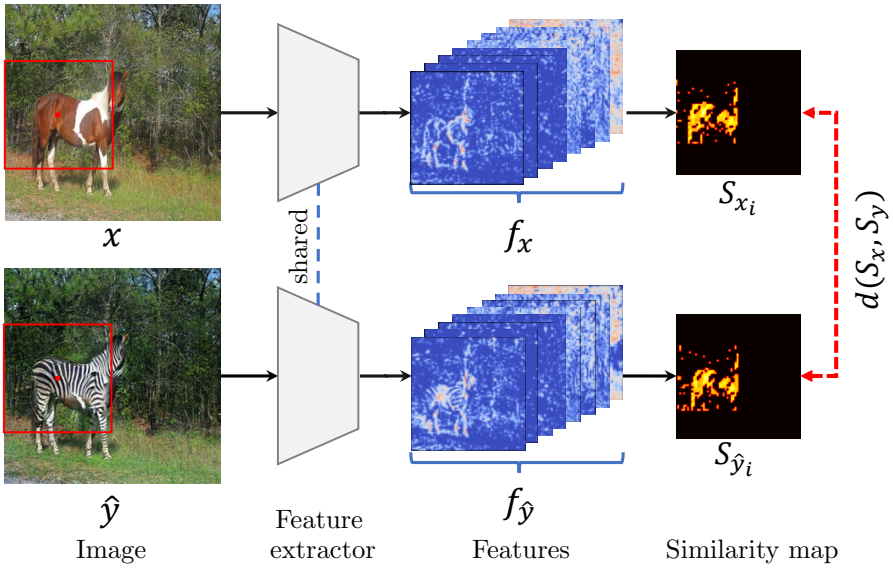


FIGURE 3.3: **An example of computing spatially-correlative loss from self-similarity maps.** The image x and corresponding translated image $\hat{y}$ are first fed into the feature extractor. We then compute the local self-similarity for each query point. Here, we show one example for the red query point.

We call this a *spatially-correlative map*, formally:

$$S_{x_i} = (f_{x_i})^T (f_{x_*}) \tag{3.1}$$

where $f_{x_i}^T \in \mathbb{R}^{1\times C}$ is the feature of a query point x_i, $f_{x_*} \in \mathbb{R}^{C\times N_p}$ contains corresponding features in a patch of N_p points, and $S_{x_i} \in \mathbb{R}^{1\times N_p}$ captures the feature spatial correlation between the query point and other points in the patch. We show one query example in Figure 3.3, where the spatially-correlative map for the query patch is visualized as a heat map. Note that unlike the original features that would still encode domain-specific attributes such as color, lighting and texture, the self-similarity map only captures the spatially-correlative relationships.

Next, we represent the structure of the whole image as a collection of multiple spatially-correlative maps $S_x = [S_{x_1}; S_{x_2}; \ldots; S_{x_s}] \in \mathbb{R}^{N_s\times N_p}$, where N_s is the numbers of sampled patches. This is a semi-sparse representation, but is more computationally efficient. We then compare the multiple structure similarity maps between the input x and the translated image $\hat{y}$, as follows:

$$\mathcal{L}_s = d(S_x, S_{\hat{y}}) \tag{3.2}$$

where $S_{\hat{y}}$ are corresponding spatially-correlative maps in the target domain. Here,

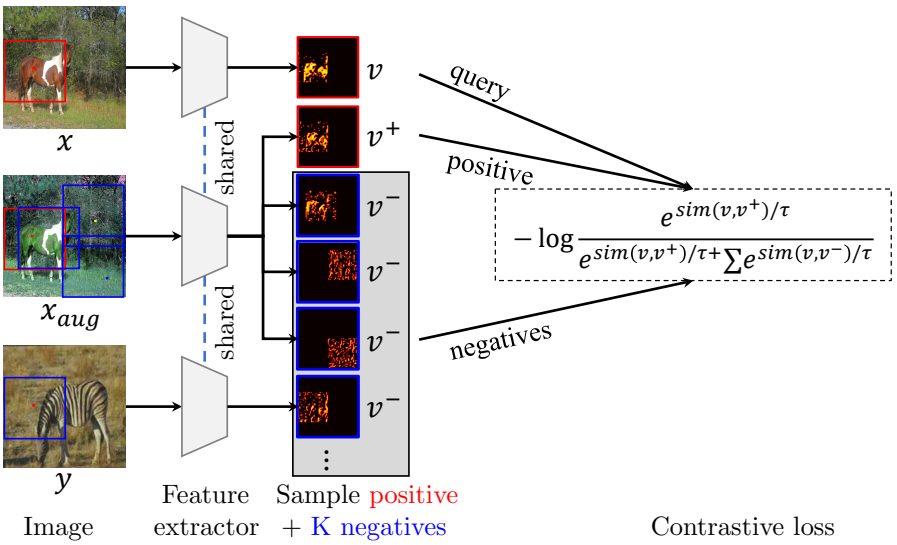


FIGURE 3.4: **Patchwise contrastive learning for the learned self-similarity.** Three images are fed into the feature extractor, in which two images, x and x_{aug}, are homologous with the same structure but varied appearances, and y is another randomly sampled image. For each query patch in x, the "positive" sample is the corresponding patch in x_{aug}, and all other patches are considered as "negative" samples.

we consider two forms for $d(\cdot)$, the L_1 distance $\|S_x - S_{\hat{y}}\|_1$ and the cosine distance $\|1 - \cos(S_x, S_{\hat{y}})\|$. The former term strongly encourages the spatial similarity to be consistent at all points in a patch, while the latter term supports pattern correlation without concern for differences in magnitude.

3.3.2 Learned Self-Similarity (LSeSim)

Although our FSeSim provides strong supervision for structure consistency, it does *not* explicitly learn a structure representation for a specific translation task. As opposed to existing feature-level losses [93, 132] that only utilize the features from a fixed pre-trained network, we propose to additionally learn a structure representation network for each task that expresses the learned self-similarity, or LSeSim.

In order to learn such a model *without supervision*, we consider the self-supervised contrastive learning that associates similar features, while simultaneously dissociates different features. Following PatchNCE [143], we build our contrastive loss at patch level, except here the pairs for comparison are our spatially-correlative maps, rather than the original features in existing works [24, 71, 78, 143]. To help generate pairs of similar patch features for self-supervised learning, we create augmented images by applying structure-preserving transformations.

Formally, let $\boldsymbol{v} = S_{x_i} \in \mathbb{R}^{1\times N_p}$ denotes the spatially-correlative map of the "query" patch. Let $\boldsymbol{v}^+ = S_{\hat{x}_i} \in \mathbb{R}^{1\times N_p}$ and $\boldsymbol{v}^- \in \mathbb{R}^{K\times N_p}$ be "positive" and "negative" patch samples, respectively. The query patch is positively paired with a patch in the same position i within an augmented image x_{aug}, and negatively paired to patches sampled from other positions in x_{aug}, or patches from other images y. The number of negative patches used is $K = 255$.

Our LSeSim design is illustrated in Figure 3.4. The contrastive loss is given by:

$$\mathcal{L}_c = -\log \frac{e^{sim(\boldsymbol{v},\boldsymbol{v}^+)/\tau}}{e^{sim(\boldsymbol{v},\boldsymbol{v}^+)/\tau} + \sum_{k=1}^{K} e^{sim(\boldsymbol{v},\boldsymbol{v}_k^-)/\tau}} \tag{3.3}$$

where $sim(\boldsymbol{v}, \boldsymbol{v}^+) = \boldsymbol{v}^T\boldsymbol{v}^+/\|\boldsymbol{v}\|\|\boldsymbol{v}^+\|$ is the cosine similarity between two spatially-correlative maps, and τ is a temperature parameter. To minimize this loss, our network encourages the corresponding patches with the same structure to be close even they have very different visual appearances, which fits in with the goal of image translation. Note that, this contrastive loss is only used for optimizing the structure representation network. The spatially-correlative loss for the generator is always the loss in equation (3.2).

3.3.3 Full Objective

Overall, we train the networks by jointly minimizing the following losses:

$$\begin{aligned}
\mathcal{L}_D &= -\mathbb{E}_{y\sim p_d}[\log D(y)] - \mathbb{E}_{\hat{y}\sim p_g}[\log(1 - D(\hat{y}))] \\
\mathcal{L}_S &= \mathcal{L}_c \\
\mathcal{L}_G &= \mathbb{E}_{\hat{y}\sim p_g}[\log(1 - D(\hat{y}))] + \lambda d(S_x, S_{\hat{y}})
\end{aligned} \tag{3.4}$$

where $\mathcal{L}_D$ is the adversarial loss for the discriminator $D(\cdot)$, $\hat{y}$ is the translated image, and $\mathcal{L}_S$ is the contrastive loss for the structure representation network $f(\cdot)$. $\mathcal{L}_G$ is the loss for the generation (translation) network $G(\cdot)$, which consists of the style loss term and the structure loss term. λ is a hyper-parameter to trade off between style and content.

3.3.4 Analysis

Readers may wonder why the proposed F/LSeSim losses would perform better than existing feature-level losses [93, 132, 143]. An intuitive interpretation is that *self-similarity deals only with spatial relationships of co-occurring signals, rather than*

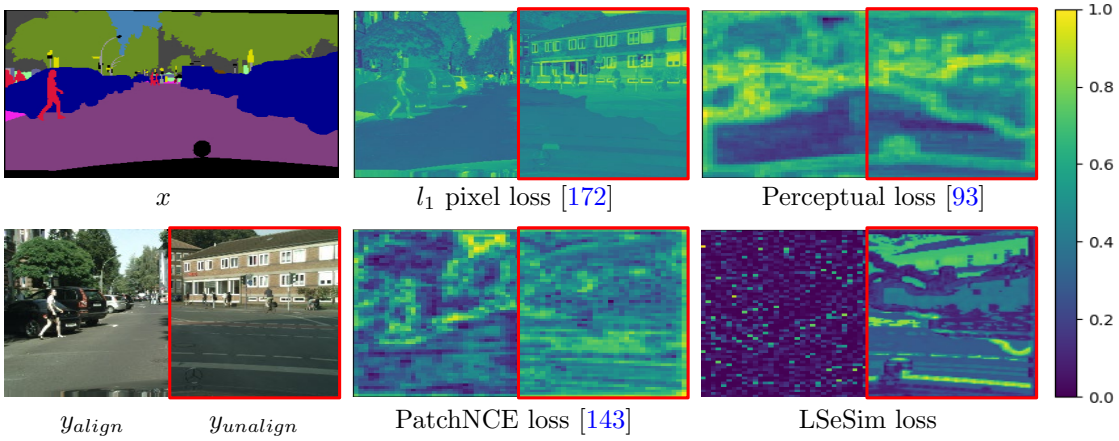


FIGURE 3.5: **Error map visualization.** Our LSeSim has small errors on the left where ground truth paired data is provided, while having large errors on the right for obviously unpaired data.

their original absolute values.

To provide further clarity, we consider a scenario where given a semantic map x (Figure 3.5), the task is to translate it to a photorealistic image y. We consider an ideal result (the paired ground truth y_{align} in the dataset) and a wrong result (another image $y_{unalign}$), respectively. Under such a setting, a good structure loss should penalize the wrong result, while supporting the ideal result. To visualize the error maps, for each corresponding pair of query patches in x and y we computed the error at that patch location for different losses. As can be seen, pixel-level loss [172] is naturally unsuitable when there are large domain gaps, and while Perceptual loss [93] will report significant errors for both aligned and unaligned results. PatchNCE [143] mitigates the problem by calculating the cosine distance of features, but it can be seen the loss map still retains high errors in many regions within the aligned result, due to extracted features consisting of appearance attributes, such as color and texture.

In contrast, appearance attributes are ignored in LSeSim by representing scene structure as a spatially-correlative map. Figure 3.5 shows that our LSeSim leads to low errors for the aligned image (left), even when they are in quite different domains, but large errors for the non-aligned image (right). Even for $y_{unalign}$, LSeSim differentiates between related structures (*e.g.* roads) and unrelated structures (trees vs windows), with lower errors for the former. Hence LSeSim can better help preserve scene structure even across large domain gaps.

In Figure 3.6, we report a qualitative comparison of various losses that be applied to a same translation network architecture. All methods following the setting

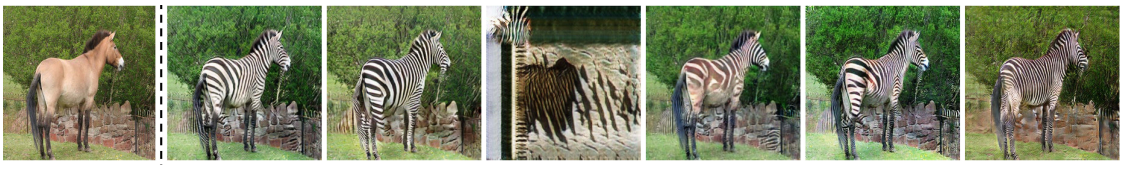


FIGURE 3.6: **Comparing results under different content losses**. All results are reported following the same setting of CycleGAN [234], except using different content losses. Our model generates much better visual results with only loss modification.

in CycleGAN [235], except that the content loss is changed. Cycle-consistency is achieved using the auxiliary generator and discriminator, and all other methods are one-sided translation. We find that our method produces results with much better visual quality.

Discussion Similar to conventional feature-level losses [93, 132], our F/LSeSim is computed in a deep feature space. However, we represent the structure as multiple spatially-correlative maps. So rather than directly at feature level which is not free from domain-specific attributes, our comparison is done at a more abstract level that is intended to transcend domain specificity.

While attention maps have been used in previous image translation works [1, 26], it is fundamentally different from our F/LSeSim in concept — their attention maps effectively function as saliency maps to guide the translation, but content preservation is primarily still dependent on cycle-consistency loss. In our case, the multiple spatially-correlative maps are used to encode and determine invariance in scene structure. Our F/LSeSim also differs from the content loss used in [105], in which the self-similarity was calculated at random positions without a clear purpose. Our F/LSeSim is on the other hand organized at a local patch level to explicitly represent the scene structure. As shown in Section 3.4.2, our local structure representation is better than just using random spatial relationships. Furthermore, our LSeSim is a metric learned from the infoNCE loss, which generalizes well robustly on various tasks. While PatchNCE loss [143] can also learn feature similarity using contrastive loss, it directly compares features in two domains.

3.4 Experiment

To demonstrate the generality of our method, we instantiated F/LSeSim in multiple frameworks on various I2I translation tasks, including *single-modal*, *multi-modal*,

and even *single-image* translation. For each task, we used a suitable baseline architecture, but replaced their content losses with our F/LSeSim loss. In addition, we are only interested in scenarios where scene structure is preserved during the translation [87, 234, 235], rather investigating translations incorporating shape modification [6, 28, 29, 100, 140].

3.4.1 *Single-Modal* Unpaired Image Translation

We first evaluated our loss on the classical single-modal unpaired I2I translation.

Implementation details In this task, we chose CycleGAN [234] as the reference architecture, but only used half of their pipeline and replaced the cycle-consistency loss with our F/LSeSim loss. Specifically, we used the ResNet-based generator with PatchGAN discriminator [87]. Details can be found on their website.

Our FSeSim is based on the ImageNet-pretrained VGG16 [174], where we used features from layers `relu3_1` and `relu4_1`. While the LSeSim employs the same structure as FSeSim, the weights are not fixed and additionally two convolution layers, implemented as 1×1 kernels, are included to select better features. As for the selection of patches to build the contrastive loss, we found that random sampling the patch locations performed much better than uniform sampling on a grid, leading to better convergence when training the structure representation network. We set $\lambda = 10$ in FSeSim and $\tau = 0.07$ in LSeSim.

Metrics Our evaluation protocols are adopted from previous work [77, 143, 144]. We first used the popular Fréchet Inception Distance (FID) [77][1] to assess the visual quality of generated images by comparing the distance between distributions of generated and real images in a deep feature domain. For *semantic image synthesis*, we further applied semantic segmentation to the generated images to estimate how well the predicted masks match the ground truth segmentation masks as in [23, 143, 144, 193]. Following [90, 143], we used the pre-trained DRN [213].

Results In Table 3.1, we reported either published results or our reproductions with publicly-available code, choosing the better. Our simple, inexpensive losses

[1] As claimed in StyleGANv2 [97], ImageNet-pretrained classifiers tend to evaluate the distribution on texture than shape, while humans focus on shape. The best FID score does *not* ensure the best image quality for translated images. As such, for a fair comparison, we reported the best FID score from all trained epochs for all methods, rather than the score in the last epoch.

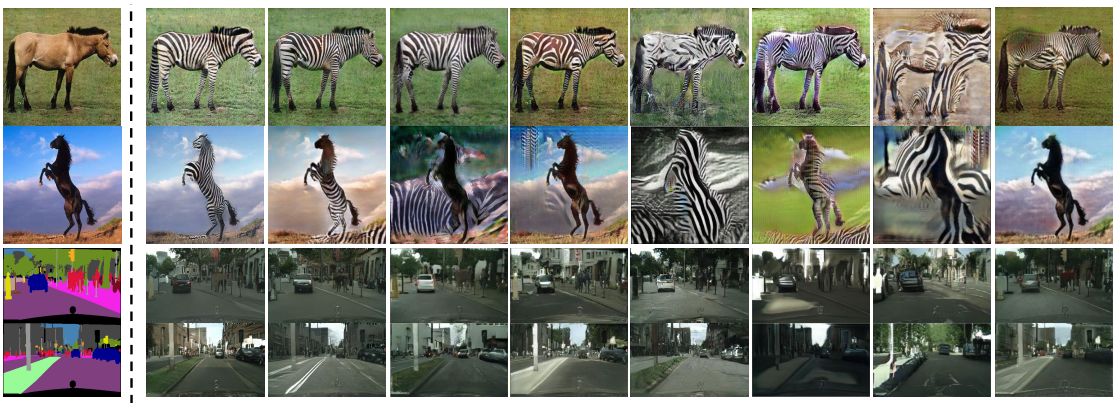


FIGURE 3.7: **Qualitative comparison on single-modal image translation.** Here, we show results for *horse→zebra* and *label→ image*.

Method	Cityscapes		Horse→Zebra	
	pixAcc↑	FID↓	FID↓	Mem↓
CycleGAN [234]	57.2	76.3	77.2	4.81
MUNIT [84]	58.4	91.4	98.0	9.43
DRIT++ [112]	60.3	96.2	88.5	11.2
Distance [11]	47.2	75.9	67.2	2.72
GcGAN [52]	65.5	57.4	86.7	4.68
CUT [143]	68.8	56.4	45.5	3.33
FSeSim	69.4	53.6	40.4	**2.65**
LSeSim	**73.2**	**49.7**	**38.0**	2.92

TABLE 3.1: **Quantitative comparison on single-modal image translation**. FID [77] measures the distance between distributions of generated images and real images. "Mem" denotes the memory cost during training.

substantially outperformed state-of-the-art methods, including two-sided frameworks with multiple cycle-consistency losses [84, 112, 235], and one-sided frameworks using self-distance [11], geometry consistency [52] and contrastive loss [143].

When compared to CycleGAN [234] and CUT [143], although we used the same settings for the generator and discriminator, our method led to significant improvement. Unlike CUT [143] that depends on an identity pass for good performance, our results were achieved by training with only one pass using F/LSeSim and GAN losses. This suggests that once we explicitly decouple scene structure and appearance, it is easier for the model to modify the visual appearance correctly. As our model belongs to one-sided image translation that does *not* require additional generators and discriminators, our model is also memory-efficient.

Qualitative results are shown in Figures 3.6 and 3.7. In Figure 3.6, despite keeping the same settings and only comparing different content losses, our method translated the zebra appearance more cleanly. We also compared results using the same examples as [143] in Figure 3.7, where our method achieved better visual results, even for some failure cases of [143].

3.4.2 *Multi-Modal* Unpaired Image Translation

Our F/LSeSim is also naturally suited for multi-modal image translation, since the use of our spatial-correlative maps imposes only structural consistency and not appearance constraints. We performed a thorough comparison of F/LSeSim to state-of-the-art methods, along with comprehensive ablation experiments.

Implementation details Our multi-modal setting is based on MUNIT [84, 112], except our model uses only one generator and one discriminator of MUNIT [84] without requiring the auxiliary generators and discriminators for multiple cycle training. Specifically, we used the ResNet-based generator with Instance Normalization (IN) [185] in the encoder and Adaptive Instance Normalization (AdaIN) [83, 96] in the decoder, plus multi-scale discriminators [193]. The details of the architecture can be found on their website. The F/LSeSim used is identical to that used in Section 3.4.1.

Metrics Besides using FID to measure quality, we also used the average LPIPS distance [222] to evaluate the diversity of generated results. The LPIPS distance is calculated by comparing the features of two images. Following [84, 235], we computed the distances between 1900 pairs, sampling 100 images 19 times. We also report the latest metrics of Density and Coverage (D&C) [137], which separately evaluate the diversity and fidelity of generated results. Likewise, we used the 1900 sampled pairs to compute D&C scores. Higher scores here indicate larger diversity and better coverage to the ground-truth domain, respectively.

Results We compared our F/LSeSim to state-of-the-art methods in multi-modal image translation in Table 3.2. Our method outperformed the two baselines, MUNIT [84] and DRIT++ [112], although we deployed the same network architecture. In Table 3.2, our method achieved larger diversity with higher LPIPS score and better image quality with lower FID score. Besides, BicycleGAN [235] achieved

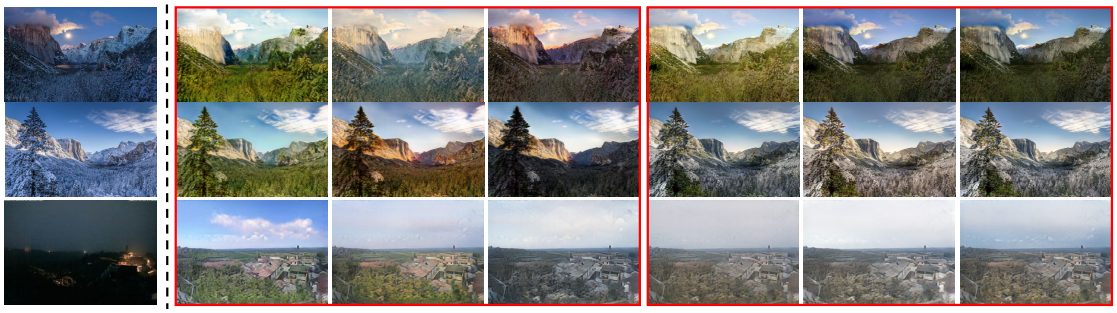


FIGURE 3.8: **Qualitative comparison on multi-modal image translation**. Here, we show the examples of *winter→summer* and *night→day*. Our model provides not only better visual results, but also produces larger diversity.

Method	Winter→Summer			Night→Day		
	LPIPS ↑	FID ↓	D & C ↑	LPIPS ↑	FID ↓	D & C ↑
Real images	0.770	44.1	0.997 / 0.986	0.684	146.1	0.977 / 0.962
BicycleGAN [235]	**0.285**	99.2 ± 3.2	**0.536** / 0.667	**0.349**	290.9 ± 6.5	**0.375** / 0.515
MUNIT [84]	0.160	97.4 ± 2.2	0.439 / 0.707	0.152	267.1 ± 2.7	0.271 / 0.548
DRIT++ [112]	0.186	93.1 ± 2.0	0.494 / 0.753	0.167	258.5 ± 2.3	0.298 / 0.631
FSeSim	0.216	90.5 ± 1.9	0.501 / 0.779	0.203	234.3 ± 2.8	0.332 / 0.638
LSeSim	0.232	**89.4 ± 1.9**	0.516 / **0.793**	0.215	**224.9 ± 2.0**	0.347 / **0.652**

TABLE 3.2: **Quantitative evaluation on multi-modal image translation task**. LPIPS distance [222] measures the diversity of generated images by comparing the features of two images, while (D&C) [137] evaluates the diversity and fidelity by matching whole features in the generated and real datasets.

larger diversity (higher LPIPS score) on all tasks by adding noise to all decoders through the U_Net [160], but the tradeoffs are worse visual results (the highest FID score), due to the larger noise being directly added to the last generative layer. In contrast, we only added noise to the middle layers of generation, through AdaIN.

In Figure 3.8, we show qualitative comparisons of our method to MUNIT [84] on *winter* → *summer*, and *night* → *day* tasks. As can be seen, our model not only generated higher quality translated results, but also produced more diverse solutions for these multi-modal tasks. We believe this is because the formulation of our F/LSeSim will only maintain structural fidelity, and does not impose penalties on appropriate appearance modifications in the target domain.

Ablation Experiments To understand the influence of different components for the proposed spatially-correlative loss, we ran a number of ablations. The quantitative results are reported in Table 3.3 for both single- and multi-modal image translation. In this table, **row A** shows the performance of the baseline method [105] which utilizes self-similarity as content loss. However, it calculates the similarity

	Configuration	Horse → Zebra		Night → Day		
		FID ↓	Mem(GB) ↓	FID ↓	LPIPS ↑	D & C ↑
A	STROTSS [105] (random SeSim)	70.1	2.68	262.7 ± 3.6	0.162	0.289 / 0.554
B	Baseline (global SeSim on single layer)	53.7	2.97	**173.2 ± 2.2**	0.168	0.303 / **0.664**
C	(B): Global → Patch (32 × 32)	45.8	**2.61**	231.3 ± 2.5	0.181	0.317 / 0.634
D	(C): Single → Multi (`relu3_1`, `relu4_1`)	43.3	2.65	229.4 ± 2.1	0.177	0.311 / 0.646
E	(D): l1loss → 1 - *consine*	40.4	2.65	234.3 ± 2.8	0.203	0.332 / 0.638
F	Ours LSeSim	**38.0**	2.92	224.9 ± 2.0	**0.215**	**0.347** / 0.652

TABLE 3.3: **Ablation study on both single- and multi-modal image translation**. Refer to ablation experiments in main text for details.

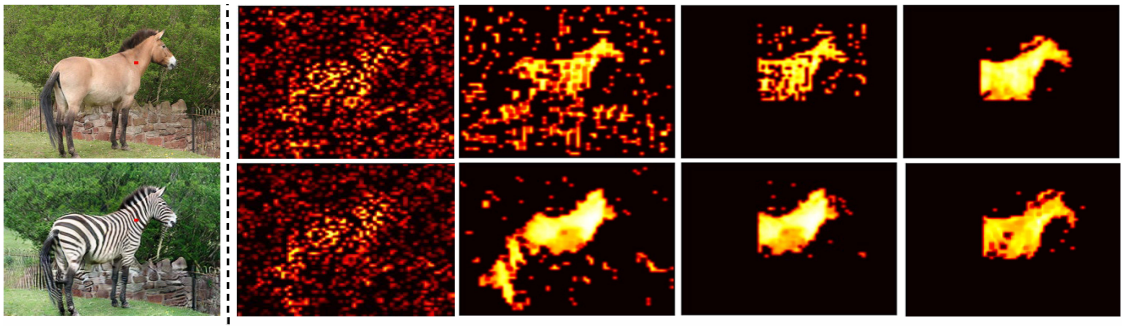


FIGURE 3.9: **Ablation study on self-similarity maps**. A, B, D and F correspond to the settings in Table 3.3, respectively.

using random sampled features and does not have an explicit connection to spatial structure. **Row B** is a global attention map. While this version performed well and ran faster by avoiding sampling, it has two main limitations. First, the original global attention module is memory intensive and *cannot* be applied to multiple scales nor to large feature spaces. Second, as evident from Figure 3.9, the spatially distant correlation is essentially noise (as is also the case for the Random baseline of [105]), which is detrimental to the results. Compared to the global version, **row C** largely improved the performance as clearer shapes are captured in the local patches. In **row D**, we applied local attention to multiple layers with a fixed path size. This results in the spatially-correlative maps having different receptive fields, which further improves the performance. **Row E** replaces the l_1 distance with cosine distance. While the improvement in image quality is not obvious, the diversity scores increased substantially. This is due to the cosine similarity supporting only the correlation between the two spatially-correlative maps without encouraging the maps to be fully same. **Row E** shows the performance of the full model (same as in Tables 3.1 and 3.2), where LSeSim of **row F** improved on many metrics, and had better visual results.

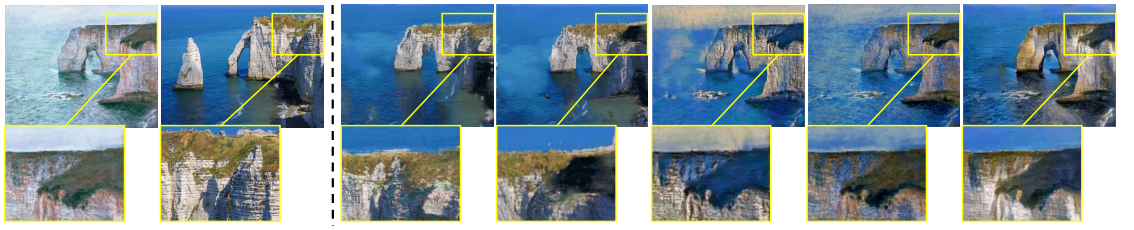


FIGURE 3.10: **High-resolution painting to photorealisitc image** on single-image translation.

3.4.3 *Single-Image* Unpaired Image Translation

To further test the generalization ability, we applied the FSeSim to a high-resolution single-image translation task. Here, only one source and one target image are provided for training, but they are unpaired. This task is conceptually similar to the style transfer [56, 93, 128], except that here we trained a SinGAN-like [168, 171] model that captures the distribution of a single image through the adversarial learning, rather than using a fixed style loss [93].

Implementation details The single-image translation setting is based on the CUT [143] method, except that the PatchNCE loss is replaced by our FSeSim loss. In detail, the StyleGAN2-based generator and discriminator [97] with the gradient penalty [96, 134] are used. To further increase simplicity, we removed the identity loss in CUT [143], and only used a GAN loss in conjunction with the proposed FSeSim loss to assess the appearance and structure separately. As these 64×64 cropped patches have to be taken from a high-resolution image for training here, it becomes less useful to further subsample "positive" and "negative" patches. Therefore, we only use our FSeSim to train the model, without using LSeSim with contrastive loss.

Results In Figure 3.10, we show qualitative results from the CUT [143] paper on the *painting→photo* task. As evident in the highlighted regions, our model generated not only higher quality results, but they were also closer to the target image style than existing methods, including classical style transfer models, such as WCT^2 [210] and STRORSS [105], as well as the latest single-image translation CUT [143] model.

3.5 Limitations and Discussion

In this chapter, we introduced F/LSeSim, a new structure consistency loss that focuses only on spatially-correlative relationships, without regard to visual appearances. The proposed F/LSeSim is naturally suitable for tasks that require structure consistency, and can be easily applied to existing architectures. We demonstrated its generality to various unpaired I2I translation tasks, where a simple replacement of the existing content losses with F/LSeSim led to solid performance improvements.

As demonstrated in experiments 3.4, the proposed spatially-correlative loss can easily be integrated into existing network architectures and thus allows wide applicability. However, the proposed structure loss only models the *content / structure* representation in this work, leaving the *style / appearance* to be judged by an auxiliary discriminator, which is not always stable in some situations. A future step is to better model the *style*, and to effectively incorporate *content* and *style* losses in translation.

So far, in the last two chapters, I have investigated the specific problem of unpaired I2I translation. In Chapter 2, a system for synthetic-to-realistic translation was proposed that solves for single image depth estimation. In this chapter, a general spatially-correlative loss was introduced for various I2I translation tasks, where the *structure* representation was explicitly modeled. These works mainly focus on modifying the appearance, which is a basic operation in visual synthesis. Next, in Part II, we go further to explore the content modification in visual synthesis, which requires a high-level semantic perception of a scene, instead of purely changing low-level appearance.

Part II

Generating Semantic Content: Image Completion

Chapter 4

Pluralistic Image Completion

This chapter covers a classic generative task: image inpainting / completion [13]. At the time of publication[1], the previous approaches produced only one result for a given masked image, although there may be many reasonable possibilities. In this chapter, a new perspective is presented, **pluralistic image completion** – the task of generating *multiple* and *diverse* plausible solutions. Although there had been some earlier works on multiple solutions in image generation and translation, it is significantly harder for image completion as all the multiple solutions need to seamlessly fit with the unmasked regions of the input image. A novel and probabilistically principled framework with two parallel paths is proposed. One is a reconstructive path that utilizes the only one ground truth to get a prior distribution of missing patches and rebuild the original image from this distribution. The other is a generative path for which the conditional prior is coupled to the distribution obtained in the reconstructive path. Experiments show that our method not only yields better results in various datasets than existing state-of-the-art methods, but also provides multiple and diverse outputs. This work was followed by many researchers [36, 147, 190, 224] to explore the multiple and diverse results for this highly subjective task.

The rest of this chapter is organized as follows: We introduce and discuss the motivation and previous works in Sections 4.1 and 4.2. Next, we describe the proposed probabilistically principled framework and the improved attention module in Section 4.3. Section 4.4 presents the interface for users to freely edit images. We then describe and discuss the experiments in Section 4.5, and conclude in Section 4.6.

[1]This work was published as *Pluralistic Image Completion* in CVPR, 2019 [227].

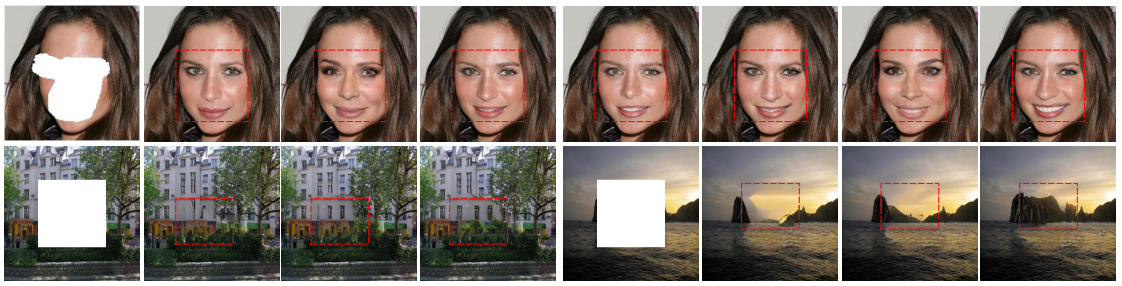


FIGURE 4.1: **Example completion results of our method** on images of a face, a building, and natural scenery with various masks (masks shown in white only for visual purpose). For each group, the masked input image is shown left, followed by sampled multiple results from our model without any post-processing.

4.1 Introduction

Image completion involves filling alternative content into the missing parts of images. It can be used for restoring damaged paintings, removing unwanted objects, and generating new content for incomplete scenes. Many approaches have been proposed for this non-trivial task, including diffusion-based methods [8, 13, 14, 113], patch-based methods [10, 32, 33, 89]) and learning-based methods [86, 123, 139, 146, 208, 214]. While these approaches rapidly improve the completion results, they produce only one "optimal" result for a given masked image and do *not* have the capacity to generate a variety of semantically meaningful results. It remains a challenging problem to provide *multiple* and *diverse* plausible results for this highly subjective problem.

Supposing you were shown the images with various missing regions in Figure 4.1, what would you *imagine* to be occupying these holes? Bertalmio *et al.* [13] related how expert conservators would restore damaged art by: 1) imagining the semantic content to be filled based on the overall scene; 2) ensuring structural continuity between the masked and unmasked regions; and 3) filling in visually realistic content for missing regions. Nonetheless, each expert will independently end up creating *substantially different details*, such as various shapes and colors of eyes, even if they may universally agree on high-level semantics, such as general placement of eyes and mouth on a damaged portrait.

Based on this observation, the main research goal in this chapter is thus to generate *multiple* and *diverse* plausible results when presented with a masked image. We refer to this task as **pluralistic image completion** (depicted in Figure 4.1). This is as opposed to existing works that attempt to generate only a single "guess" for this ill-posed problem.

To obtain a diverse set of results for a given input, some methods utilize conditional variational auto-encoders (CVAE) [9, 47, 176, 189], a conditional extension of variational auto-encoders (VAE) [103], which explicitly code for a distribution that can be sampled. However, specifically for an image completion scenario, the standard single-path formulation usually leads to grossly underestimating variances. This is because when *the condition label is itself a masked image*, the number of ground truth instances in the training data that match the label is *typically only one – the original complement of the masked image*. Hence, the estimated original conditional distributions tend to have very limited variation since they were trained to reconstruct the single original image.

An important insight we will use is that *partial images (patches)*, as a superset of full images, may also be considered as generated from *a latent space with smooth prior distributions* [168]. This provides a mechanism for alleviating the problem of having scarce samples per conditional masked image. To do so, we introduce a **P**luralistic **I**mage **C**ompletion **Net**work, called **PICNet**, with two parallel but linked training pipelines. The first pipeline is a VAE-based reconstructive path that not only utilizes the full instance ground truth, but also imposes smooth priors for the latent space of missing partial image. The second pipeline is a generative path that learns to predict the latent prior distribution for the missing regions only based on the visible pixels, from which can be sampled to generate diverse results. The training process for the latter path does *not* attempt to steer the output towards reconstructing the instance-specific results at all, instead allowing the reasonableness of results being driven by an auxiliary discriminator network [65]. This leads to substantially great variability in generation. To further utilize the information from the visible partial images as much as possible [10, 214], we also introduce an enhanced *short+long term patch attention* layer, a generic attention mechanism that allows information flowing from visible regions to missing holes.

We comprehensively evaluate and compare our approach with existing state-of-the-art methods on a large variety of scenes (Section 4.5.2), where various masks, including regular and free-form irregular masks, are used to erode the images. We additionally present many interesting applications of our model on free-form image editing (Section 4.5.3), *e.g.* object removal, face editing, and scene content-aware move. The extensive experimental results demonstrate that our proposed PICNet not only generates higher-quality completion results, but also produces multiple diverse solutions for this subjective processing task.

4.2 Background

Existing work on image completion either uses information from within the image [13, 14], or information from a large image dataset [70, 146]. Most approaches generate only one result per masked image, which is precisely the downside we want to address in this chapter.

4.2.1 Intra-Image Completion

Traditional intra-image completion works (also known as "inpainting" [13]) mainly propagate, copy and realign the background regions to missing regions, focusing only on the steps 2 and 3 mentioned above, by assuming that the holes should be filled with similar appearance to that of the visible regions. One category of intra-image completion methods are diffusion-based image synthesis [8, 13, 14, 113]. These methods fill the surrounded backgrounds to the missing regions by propagating the local colors. They only work well on the small and narrow holes. Another category of intra-image completion methods are patch-based approaches [10, 32, 33, 89]. They fill the holes by copying information from similar visible regions, which produce high-quality texture-consistent result. However, these intra-image methods cannot capture global semantics to hallucinate new content for large holes (as in step 1), which is significant for real image completion.

4.2.2 Inter-Image Completion

To hallucinate semantically new content, inter-image completion borrows information from a large dataset. Hays and Efros [70] first present an image completion method using millions of images. Recently, learning-based approaches are proposed. Initial works [104, 158] focus on small and thin holes. Then, Pathak *et al.* [146] proposed the Context Encoders (CE) to handle 64×64-sized holes. Iizuka *et al.* [86] built upon [146] by combining global and local discriminators (GL) as adversarial loss. Wang *et al.* [195] designed Multi-column CNNs and a cosine similarity based loss for high-quality image inpainting. More recent, Liu *et al.* [123] introduced "partial convolution" for free-form irregular mask image completion.

Some work has also explored additional information for semantically image completion. In [207], the "closest" features in the latent space for the masked image are searched to generate an image. Li *et al.* [115] introduced additional face parsing loss to ensure the semantic consistency of completed images. Song *et al.* [179]

proposed SPG-Net that simultaneously does semantic map and RGB appearance completion. Moreover, sketches and color are used in the latest Faceshape [150], DeepFillv2 [215], EdgeConnect [139] and SC-FEGAN [92]. A common drawback of these methods is that they utilize the visible information only through local convolutional operations, which creates distorted structures and blurry textures inconsistent with the visible regions, especially for large holes.

4.2.3 Combing Intra- and Inter-Image Completion

To mitigate the blurry problems, Yang *et al.* [204] proposed multi-scale neural patch synthesis, which generates high-frequency details by copying patches from mid-layer features. More recently, several works [178, 202, 208, 214] exploit spatial attention [88, 233] to get high-frequency details. Yu *et al.* [214] proposed a contextual attention layer to produce high-frequency details by copying similar features from visible regions to missing regions. Yan *et al.* [202] and Song *et al.* [178] proposed PatchMatch-like ideas on feature domain. Yi *et al.* [208] proposed contextual residual aggregation for very high resolution (8K) image inpainting. However, these methods identify similar features by comparing features of holes and visible regions, which is somewhat contradictory as feature transfer is unnecessary when two features are very similar, but when needed the features are too different to be matched easily. Furthermore, distant information is not used for new content that differs from visible regions. Our model solves it by extending self-attention to harness abundant context.

4.2.4 Image Generation

Image generation has progressed significantly using methods such as VAE [103] and GANs [65]. These have been applied to conditional image generation tasks, such as image translation [87, 234], synthetic to realistic [172, 226], future prediction [131], and 3D models [142]. Perhaps most relevant in spirit to us are conditional VAEs (CVAE) [176, 189] and CVAE-GAN [9], but these are not specially targeted for image completion. CVAE-based methods are most useful when the conditional labels are few and discrete, and there are sufficient training instances per label. Some recent work utilizing these in image translation can produce diverse output [111, 235], but in such situations the condition-to-sample mappings are more local (*e.g.* pixel-to-pixel), and only change the visual appearance without generating new content. This is untrue for image completion, where the conditional label

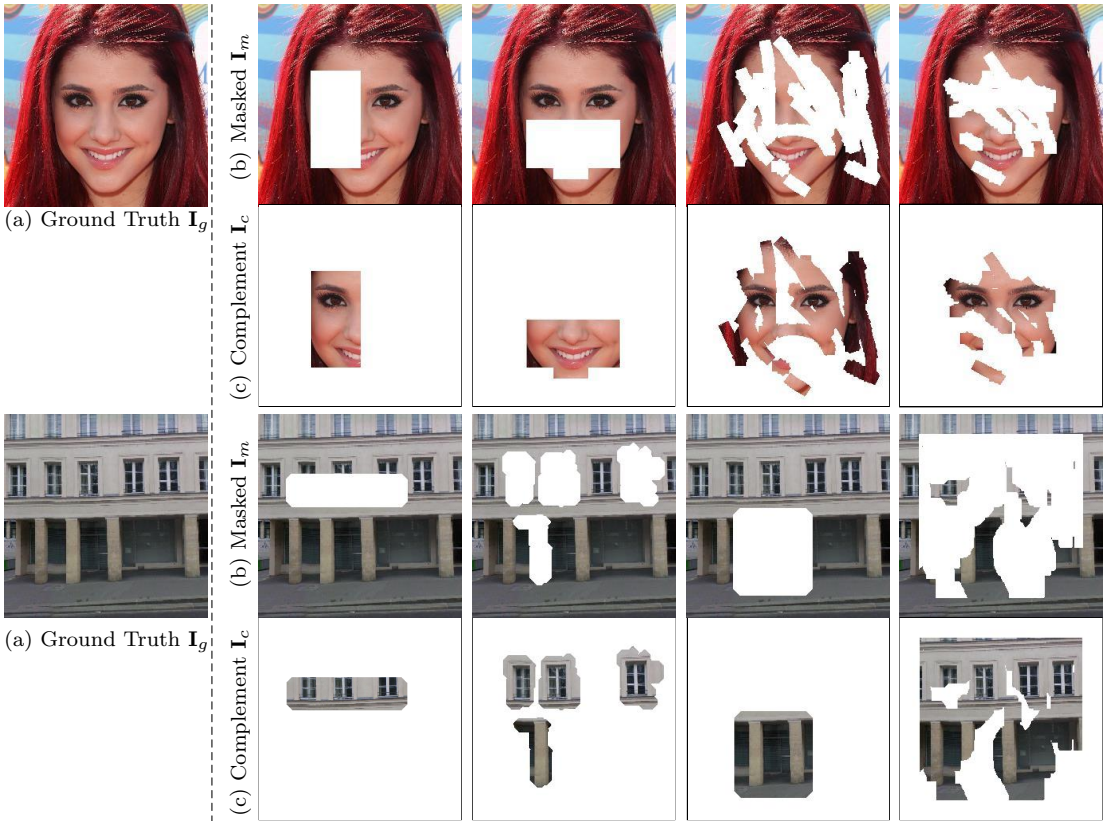


FIGURE 4.2: **Examples of different degraded images.** (a) Ground truth image $\mathbf{I}_g$. (b) Masked image $\mathbf{I}_m$. (c) The corresponding complement image $\mathbf{I}_c$ to each top masked image $\mathbf{I}_g$. It is often not reasonable to strongly enforce the completed masked regions to be identical to the ground truth, especially in cases when large variations in the completed content can still be perfectly consistent to the visible regions, *e.g.* when the entire mouth or both eyes are masked.

is the masked image itself, with only one training instance of the original holes. In [27], different outputs were obtained for face completion by specifying facial attributes (*e.g.* smile), but this method is very domain specific, requiring targeted attributes. In contrast, our proposed probabilistically principled framework produces multiple and diverse plausible in various datasets, which does not need any label information for training.

4.3 Approach

Suppose we have an image, originally ground truth $\mathbf{I}_g$ (Figure 4.2 (a)), but degraded by a number of missing pixels to become $\mathbf{I}_m$ (Figure 4.2 (b)), *masked partial image* comprising the visible pixels. We also define $\mathbf{I}_c$ (Figure 4.2 (c)) as its *complement partial image* comprising the missing pixels.

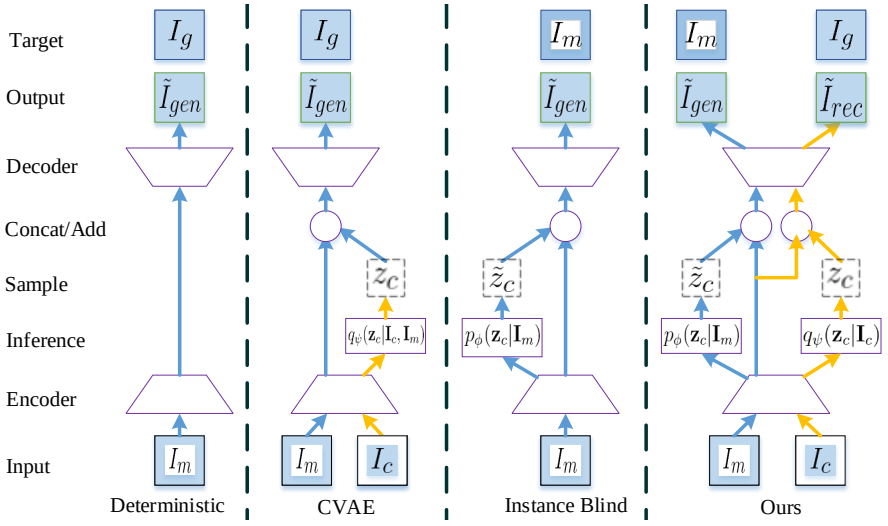


FIGURE 4.3: **Completion strategies given masked image.** (Deterministic) structure directly predicts the ground truth instance. (CVAE [189]) adds in random sampling to diversify the output, but is still trained on the single ground truth. (Instance Blind) only matches the masked instance, but training is unstable. (Ours) uses a generative path during testing, but is guided by a parallel reconstructive path during training. Note that, yellow path is only used for training.

Prior image completion methods [86, 139, 146, 214] attempt to reconstruct the original unmasked image $\mathbf{I}_g$ in a deterministic fashion from $\mathbf{I}_m$ (see Figure 4.3 "Deterministic"). However, this rigid approach has several limitations. First, while it is fine to rebuild the original image $\mathbf{I}_g$ when visible regions tightly constrain the completed content, *e.g.* when only the left half of a face is masked in Figure 4.2, it is unnecessarily limiting when visible regions allow a much greater range of perceptually consistent completion, *e.g.* with many different mouth expressions or building door appearances equally acceptable in Figure 4.2. Second, deterministic methods can only generate a single solution and are not able to recover a richer distribution of reasonable possibilities. Instead, our goal is to *sample* from $p(\mathbf{I}_c|\mathbf{I}_m)$ and we reconstruct the original image only when the corresponding complement partial images $\mathbf{I}_c$ are provided during the training.

4.3.1 Pluralistic Image Completion Network

4.3.1.1 Probabilistic Framework

In order to have a distribution to sample from, an approach is to employ the CVAE [176] which estimates a parametric distribution over a latent space, from which sampling is possible. This involves a variational lower bound of the conditional

log-likelihood:

$$\log p(\mathbf{I}_c|\mathbf{I}_m) \geq -\mathrm{KL}(q_\psi(\mathbf{z}_c|\mathbf{I}_c,\mathbf{I}_m)||p_\phi(\mathbf{z}_c|\mathbf{I}_m)) + \mathbb{E}_{q_\psi(\mathbf{z}_c|\mathbf{I}_c,\mathbf{I}_m)}[\log p_\theta(\mathbf{I}_c|\mathbf{z}_c,\mathbf{I}_m)] \quad (4.1)$$

where $\mathbf{z}_c$ is the latent vector of missing patches, $q_\psi(\mathbf{z}_c|\mathbf{I}_c,\mathbf{I}_m)$ is the recognition network, $p_\phi(\mathbf{z}_c|\mathbf{I}_m)$ is the conditional prior, and $p_\theta(\mathbf{I}_c|\mathbf{z}_c,\mathbf{I}_m)$ is the likelihood, with ψ, ϕ and θ being the deep network parameters of their corresponding functions. This lower bound is maximized *w.r.t.* all parameters. The detail proofs are provided in Appendix A.

For our purposes, the chief difficulty of using CVAE [176] directly is that the high DoF of recognition network $q_\psi(\mathbf{z}_c|\mathbf{I}_c,\mathbf{I}_m)$ and conditional prior network $p_\phi(\mathbf{z}_c|\mathbf{I}_m)$ are not easily separable in equation (4.1). Besides, since the conditional prior network $p_\phi(\mathbf{z}_c|\mathbf{I}_m)$ is sufficiently unconstrained in equation (4.1), it will lean a narrow delta-like prior distribution of $p_\phi(\mathbf{z}_c|\mathbf{I}_m) \to \delta(\mathbf{z}_c - \mathbf{z}_c^*)$, where $\mathbf{z}_c^*$ is the maximum latent likelihood point of $p_\theta(\mathbf{I}_c|\mathbf{z}_c,\mathbf{I}_m)$. In this way, the variance σ^2 of the learned latent distribution is easily driven towards zero. Then it is approximately equivalent to maximizing $\mathbb{E}_{p_\phi(\mathbf{z}_c|\mathbf{I}_m)}[\log p_\theta(\mathbf{I}_c|\mathbf{z}_c,\mathbf{I}_m)]$, the "GSNN" variant in [176], in which they directly set the recognition network the same as the prior network, *i.e.*, $q_\psi(\mathbf{z}_c|\mathbf{I}_c,\mathbf{I}_m) = p_\phi(\mathbf{z}_c|\mathbf{I}_m)$. While this low variance prior may be useful in estimating a single solution, sampling from it will lead to *negligible diversity* in image completion results. When the CVAE variant of [189], which assumes conditional prior $p_\phi(\mathbf{z}_c|\mathbf{I}_m) = \mathcal{N}(\mathbf{0},\mathbf{I})$, is used instead, the network learns to ignore the latent sampling and directly estimates $\mathbf{I}_c$ from $\mathbf{I}_m$ for a fixed ground truth, also resulting in similar solutions. A possible way to diversify the output is simply to not incentivize the output to reconstruct the instance-specific $\mathbf{I}_g$ during training, only needing it to fit in with the training set distribution as deemed by a learned adversarial discriminator (see Figure 4.3 "Instance Blind"). However, this approach is unstable, especially for large and complex scenes [178]. A detail analysis is presented in Section 4.3.1.3.

Latent Priors of Holes In our approach, we require that missing partial images (patches), as a superset of full images, *to also arise from a latent space distribution* [168], with a smooth prior of $p(\mathbf{z}_c)$. The variational lower bound is:

$$\log p(\mathbf{I}_c) \geq -\mathrm{KL}(q_\psi(\mathbf{z}_c|\mathbf{I}_c)||p(\mathbf{z}_c)) + \mathbb{E}_{q_\psi(\mathbf{z}_c|\mathbf{I}_c)}[\log p_\theta(\mathbf{I}_c|\mathbf{z}_c)] \quad (4.2)$$

where in [103] the prior is set as $p(\mathbf{z}_c) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. However, we can be more discerning when it comes to partial images (patches) since they have different numbers of pixels. In particular, *a complement image* $\mathbf{I}_c$ *with more pixels (large holes for the masked image* $\mathbf{I}_m$*, as shown in the last column in Figure 4.2) should have greater prior variance than a complement image* $\mathbf{I}_c$ *with fewer pixels (small holes)* and in fact a masked partial image $\mathbf{I}_m$ with no pixels missing should be completely deterministic! Hence we generalize the prior $p(\mathbf{z}_c) = \mathcal{N}_m(\mathbf{0}, \sigma^2(n)\mathbf{I})$ to adapt to the number of missing pixels n, where $\sigma^2(n) = \frac{n}{H \times W} \in (0,1]$.

Prior-Conditional Coupling Next, we combine the latent priors into the conditional lower bound of (4.1). Since $\mathbf{z}_c$ represents the distributions of target missing partial image $\mathbf{I}_c$, $\mathbf{z}_c$ can be naturally inferred using the target missing image $\mathbf{I}_c$, that $q_\psi(\mathbf{z}_c|\mathbf{I}_c, \mathbf{I}_m) \approx q_\psi(\mathbf{z}_c|\mathbf{I}_c)$ when $\mathbf{I}_c$ is available in the training. Updating (4.1):

$$\log p(\mathbf{I}_c|\mathbf{I}_m) \geq -\mathrm{KL}(q_\psi(\mathbf{z}_c|\mathbf{I}_c)||p_\phi(\mathbf{z}_c|\mathbf{I}_m)) + \mathbb{E}_{q_\psi(\mathbf{z}_c|\mathbf{I}_c)}[\log p_\theta(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m)]. \tag{4.3}$$

However, unlike in (4.1), notice that $q_\psi(\mathbf{z}_c|\mathbf{I}_c)$ *is no longer freely learned during training, yet is tied to its presence in (4.2).* Intuitively, the learning of $q_\psi(\mathbf{z}_c|\mathbf{I}_c)$ is regularized by the prior $p(\mathbf{z}_c)$ in (4.2), while the learning of the conditional prior $p_\phi(\mathbf{z}_c|\mathbf{I}_m)$ is in turn regularized by $q_\psi(\mathbf{z}_c|\mathbf{I}_c)$ in (4.3).

Reconstruction vs Creative Generation One issue with (4.3) is that the sampling is taken from $q_\psi(\mathbf{z}_c|\mathbf{I}_c)$ during training, but is not available during testing, whereupon sampling must come from $p_\phi(\mathbf{z}_c|\mathbf{I}_m)$ which may not be adequately learned for this role. In order to mitigate this problem, we modify (4.3) to have a blend of formulations *with and without importance sampling.*

As is typically the case for image completion, there is only one training instance of $\mathbf{I}_c$ for each unique $\mathbf{I}_m$. This means that for function $q_\psi(\mathbf{z}_c|\mathbf{I}_c, \mathbf{I}_m)$, $\mathbf{I}_c$ can be learned into the network as a hard-coded dependency of the input $\mathbf{I}_m$, so $q_\psi(\mathbf{z}_c|\mathbf{I}_c, \mathbf{I}_m) \cong \hat{q}_\psi(\mathbf{z}_c|\mathbf{I}_m)$. Assuming that the network for $p_\phi(\mathbf{z}_c|\mathbf{I}_m)$ has similar or higher modeling power and there are no other explicit constraints imposed on it, then in training $p_\phi(\mathbf{z}_c|\mathbf{I}_m) \to \hat{q}_\psi(\mathbf{z}_c|\mathbf{I}_m)$, and the KL divergence in (4.1) goes to zero. Then we get the following function:

$$\log p(\mathbf{I}_c|\mathbf{I}_m) \geq \mathbb{E}_{p_\phi(\mathbf{z}_c|\mathbf{I}_m)}[\log p_\theta(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m)] \tag{4.4}$$

the "GSNN" version in [176]. However, unlike [176], the variance σ^2 of the learned distribution $p_\phi(\mathbf{z}_c|\mathbf{I}_m)$ in our method will not be zero as mentioned above. This $\mathbf{z}_c$ for missing regions is sampling from the visible regions $\mathbf{I}_m$, we call this *without importance sampling*, contrary to the *importance sampling* $q_\psi(\mathbf{z}_c|\mathbf{I}_c)$. Finally, we combine (4.3) and (4.4) to obtain the reconstruction and creative generation function:

$$\log p(\mathbf{I}_c|\mathbf{I}_m) \geq \lambda \left\{ \mathbb{E}_{q_\psi}[\log p_\theta^r(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m)] - \mathrm{KL}(q_\psi||p_\phi) \right\} + (1-\lambda)\, \mathbb{E}_{p_\phi}[\log p_\theta^g(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m)] \tag{4.5}$$

where $\lambda \in [0,1]$ is implicitly set by training loss coefficients in Section 4.3.1.2 (see details in Appendix A). When sampling from the importance function $q_\psi(\mathbf{z}_c|\mathbf{I}_c)$, the missing instance information is available and we formulate the likelihood $p_\theta^r(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m)$ to be focused on *reconstructing* $\mathbf{I}_c$. Conversely, when sampling from the learned distribution $p_\phi(\mathbf{z}_c|\mathbf{I}_m)$ which does not contain $\mathbf{I}_c$, we will facilitate *creative generation* by having the likelihood model $p_\theta^g(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m) \cong \ell_\theta^g(\mathbf{z}_c, \mathbf{I}_m)$ be *independent of the original instance* of $\mathbf{I}_c$. Instead it only *encourages generated samples to fit in with the overall training distribution.*

Joint Unconditional and Conditional Variational Lower Bounds Our overall training objective may then be expressed as jointly maximizing the lower bounds in (4.2) and (4.5). This can be done by unifying the likelihood in (4.2) to that in (4.5) as $p_\theta(\mathbf{I}_c|\mathbf{z}_c) \cong p_\theta^r(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m)$, in which the $\mathbf{z}_c$ is sampling from the *important sampling* $q_\psi(\mathbf{z}_c|\mathbf{I}_c)$ that can be used for rebuild the original missing regions $\mathbf{I}_c$. We can then define a combine function as our maximization goal:

$$\begin{aligned} \mathcal{B} =& \beta\, \mathcal{B}_1 + \mathcal{B}_2 \\ =& -\left[\beta \mathrm{KL}(q_\psi||p_{z_c}) + \lambda \mathrm{KL}(q_\psi||p_\phi)\right] + (\beta+\lambda)\mathbb{E}_{q_\psi} \log p_\theta^r + (1-\lambda)\mathbb{E}_{p_\phi} \log p_\theta^g \end{aligned} \tag{4.6}$$

where $\mathcal{B}_1$ is the lower bound related to the unconditional log likelihood of missing partial image $\mathbf{I}_c$, and $\mathcal{B}_2$ relates to the log likelihood of missing regions $\mathbf{I}_c$ conditioned on $\mathbf{I}_m$. Note that this function holds a key different with hybrid objective function in [176] that *the conditional prior network* $p_\phi(\mathbf{z}_c|\mathbf{I}_m)$ *and the recognition network* $q_\psi(\mathbf{z}_c|\mathbf{I}_c)$ *are no longer freely learned, but are constrained by a mask related prior* $p(\mathbf{z}_c) = \mathcal{N}_m(\mathbf{0}, \sigma^2(n)\mathbf{I})$. Furthermore, our *without importance sampling*, also the testing sampling, does not learn to predict a fixed instance during the training, which encourages larger diversity.

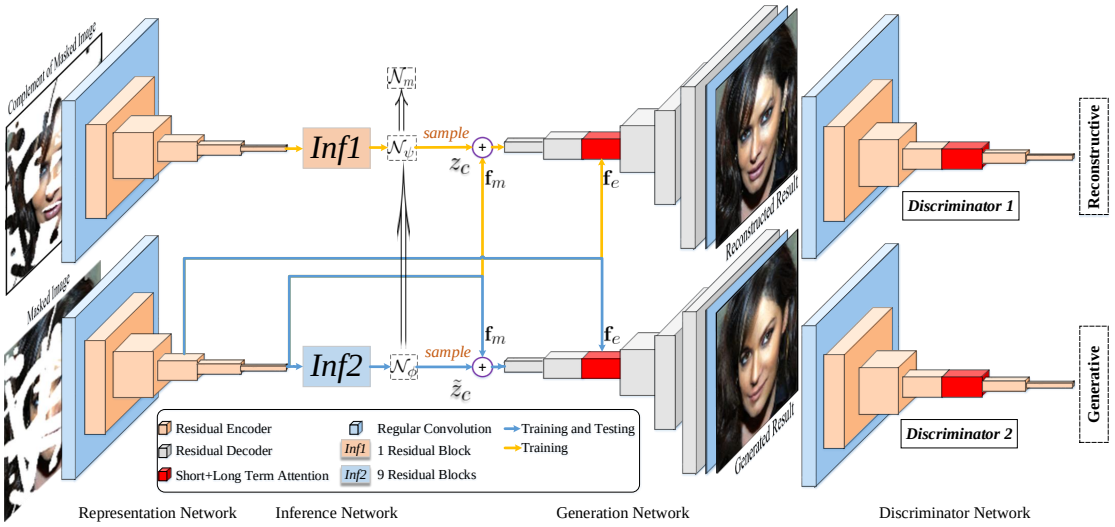


FIGURE 4.4: **Overview of our architecture with two parallel pipelines.** The top **reconstructive** pipeline (yellow line) combines information from $\mathbf{I}_m$ and $\mathbf{I}_c$, which is used only for training. The bottom **generative** pipeline (blue line) infers the conditional distribution of hidden regions, that can be sampled during testing. The two representation networks and generation networks in top and bottom share identical weights.

4.3.1.2 Network Structure and Training Loss

The formula in (4.6) is implemented as our dual pipeline, illustrated in Figure 4.4. This consists of representation, inference, generation, and auxiliary discriminator networks in two paths. The upper pipeline is the reconstruction path used in training that corresponds to the lower bound $\mathcal{B}_1$, in which $\mathbf{z}_c$ contains information of missing image $\mathbf{I}_c$. Hence when combined with the conditional feature $\mathbf{f}_m$, we can easily train this path to rebuild the original image $\mathbf{I}_g$. In contrast, the lower path, used in both training and testing, is responsible for the lower bound $\mathcal{B}_2$, where the missing information is inferred only from the masked image $\mathbf{I}_m$, resulting in a less restrictive prediction.

We transfer the lower bound terms in (4.6) as the corresponding loss function. During training, jointly maximizing the lower bounds is then minimizing a total loss $\mathcal{L}$, which consists of three groups of component losses:

$$\mathcal{L} = \alpha_{\text{KL}}(\mathcal{L}^r_{\text{KL}} + \mathcal{L}^g_{\text{KL}}) + \alpha_{\text{app}}(\mathcal{L}^r_{\text{app}} + \mathcal{L}^g_{\text{app}}) + \alpha_{\text{ad}}(\mathcal{L}^r_{\text{ad}} + \mathcal{L}^g_{\text{ad}}) \tag{4.7}$$

where the $\mathcal{L}_{\text{KL}}$ group regularizes consistency between pairs of distributions in terms of KL divergences, the $\mathcal{L}_{\text{app}}$ group encourages appearance matching fidelity, and the $\mathcal{L}_{\text{ad}}$ group forces sampled images to fit in with the training set distribution. Each of the groups has a separate term for the reconstructive and generative paths.

Distributive Regularization The typical interpretation of the KL divergence term in a VAE is that it regularizes the learned importance sampling function $q_\psi(\mathbf{z}_c|\mathbf{I}_c)$ to a latent prior $p(\mathbf{z}_c)$. Defining both as Gaussians, we get:

$$\mathcal{L}_{\text{KL}}^{r,(i)} = \text{KL}(q_\psi(\mathbf{z}|I_c^{(i)})||\mathcal{N}_m(\mathbf{0}, \sigma^{2,(i)}(n)\mathbf{I})). \tag{4.8}$$

For the generative path, the appropriate interpretation is *reversed*: the learned conditional prior $p_\phi(\mathbf{z}_c|\mathbf{I}_m)$, also a Gaussian, is regularized to $q_\psi(\mathbf{z}_c|\mathbf{I}_c)$.

$$\mathcal{L}_{\text{KL}}^{g,(i)} = \text{KL}(q_\psi(\mathbf{z}|I_c^{(i)}))||p_\phi(\mathbf{z}|I_m^{(i)}))). \tag{4.9}$$

Note that the conditional prior uses $\mathbf{I}_m$, while the importance function has access to the missing regions $\mathbf{I}_c$.

Appearance Matching Loss The likelihood term $p_\theta^r(\mathbf{I}_c|\mathbf{z}_c)$ is interpreted as probabilistically encouraging appearance matching to the missing regions $\mathbf{I}_c$. However, our framework also auto-encodes the masked image $\mathbf{I}_m$ (via $\mathbf{f}_m$) deterministically, and the loss function needs to cater for this reconstruction. As such, the per-instance loss here is:

$$\mathcal{L}_{\text{app}}^{r,(i)} = ||I_{\text{rec}}^{(i)} - I_g^{(i)}||_1 \tag{4.10}$$

where $I_{\text{rec}}^{(i)}$=$G(z_c, f_m)$ and $I_g^{(i)}$ are the reconstructed and original full images, respectively. The purpose of this loss is to bias the representation towards the actual visible information. In contrast, for the generative path, the latent distribution $\mathcal{N}_\phi$ of the missing regions $\mathbf{I}_c$ is inferred based only on the visible $\mathbf{I}_m$. This would be significantly less accurate than the inference in the upper path. Thus, we ignore instance-specific appearance matching for $\mathbf{I}_c$, and only focus on reconstructing $\mathbf{I}_m$:

$$\mathcal{L}_{\text{app}}^{g,(i)} = ||M * (I_{\text{gen}}^{(i)} - I_g^{(i)})||_1 \tag{4.11}$$

where $I_{\text{gen}}^{(i)}$=$G(\tilde{z}_c, f_m)$ is the generated image, and M is the binary mask selecting visible pixels.

Adversarial Loss The formulation of $p_\theta^r(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m)$ and the instance-blind $p_\theta^g(\mathbf{I}_c|\tilde{\mathbf{z}}_c, \mathbf{I}_m)$ also incorporates the use of adversarially learned discriminators D_1 and D_2 to judge whether the generated images fit into the training set distribution. Inspired by [9],

	Diversity (LPIPS)		Image Quality ($\mathbf{I}_{out}$)			
	$\mathbf{I}_{out}$ ↑	$\mathbf{I}_{out(m)}$ ↑	ℓ_1 loss ↓	SSIM ↑	PSNR ↑	FID ↓
CA [214]	-	-	0.031	0.820	23.57	9.53
EC [139]	-	-	0.030	0.819	23.47	8.01
MEDFE [81]	-	-	0.028	0.830	24.38	7.85
CVAE [176]	0.004	0.014	0.023	0.847	24.02	9.96
Instance Blind	0.015	0.049	0.025	0.852	23.77	9.48
BicycleGAN [235]	0.020	0.060	0.026	0.845	23.71	11.56
PICNet	**0.024**	**0.071**	**0.021**	**0.867**	**24.69**	**6.43**

TABLE 4.1: **Quantitative comparisons of different network structures** on CelebA-HQ testing set [95, 126] with center masks. ↓ = lower is better, ↑ = higher is better. $\mathbf{I}_{out}$ is the completed output image and $\mathbf{I}_{out(m)} = (1 - M) \times \mathbf{I}_{out}$ is extracted for the missing regions.

we use a mean feature match loss in the reconstructive path for the generator,

$$\mathcal{L}_{\text{ad}}^{r,(i)} = ||f_{D_1}(I_{\text{rec}}^{(i)}) - f_{D_1}(I_g^{(i)})||_2 \tag{4.12}$$

where $f_{D_1}(\cdot)$ is the feature output of the final layer of D_1. This encourages the original and reconstructed features in the discriminator to be close together. Conversely, the adversarial loss in the generative path for the generator is:

$$\mathcal{L}_{\text{ad}}^{g,(i)} = [D_2(I_{\text{gen}}^{(i)}) - 1]^2. \tag{4.13}$$

This is based on the generator loss in LSGAN [130], which performs better than the original GAN loss [65] in our scenario. The discriminator loss for both D_1 and D_2 is also based on LSGAN.

4.3.1.3 Analysis

Effect of Network Structure We first investigated the influence of using our two-path training structure in comparison to other variants such as the CVAE of [189] and the "Instance Blind" structures in Figure 4.3. We also trained the state-of-the-art multi-model BicycleGAN [235] on Celeba-HQ dataset [95, 126] by setting $\mathbf{A} = \mathbf{I}_m$, $\mathbf{B} = \mathbf{I}_c$ with center mask.

We first computed the diversity score using the Learned Perceptual Image Patch Similarity (LPIPS) metric reported in [235]. LPIPS metric [222] calculates the average distance of samples in a deep feature domain. For each random pairs, a pre-trained deep network (*e.g.* VGG [174]) is used to extract the features of

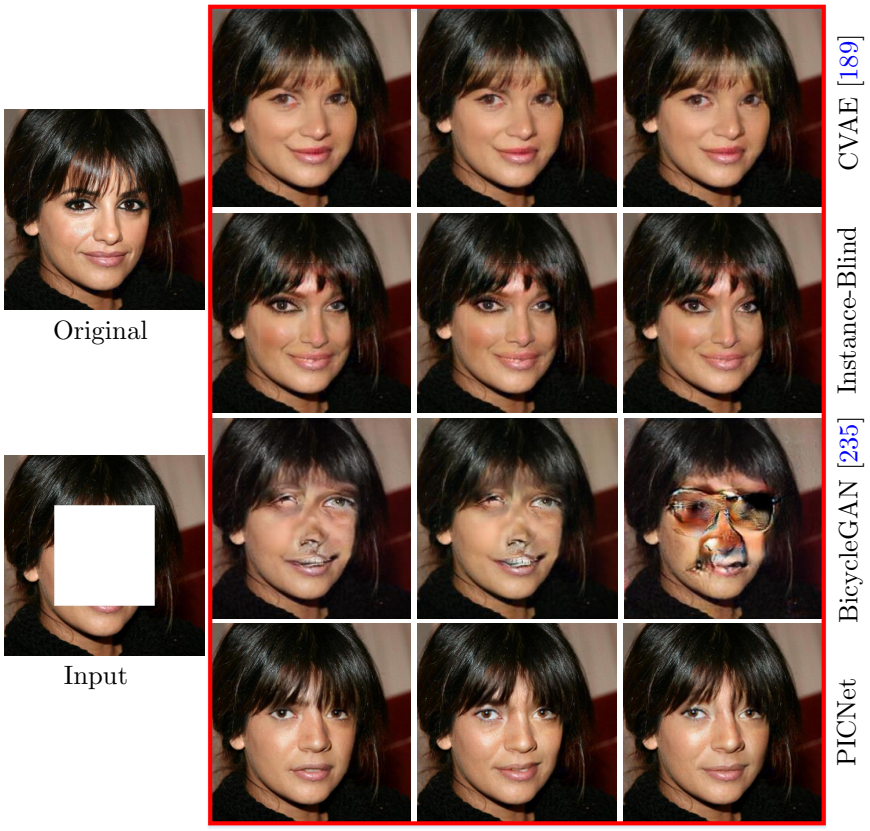


FIGURE 4.5: **Qualitative comparison results of different training strategies. First Column:** Original and Masked image. **Others:** the completed results of different methods. Our method provides diverse results, *i.e.* different hairstyles and mouth expressions, with realistic appearance.

images. Then, the distance of two vectors is calculated using ℓ_1 distance. The larger distance indicates the results are much more diverse, as the generated pairs far from each other. For each method, we sampled 50K pairs of randomly generated images from 1K center masked images. $\mathbf{I}_{out}$ and $\mathbf{I}_{out(m)}$ are the full output and the masked-regions' output, respectively. Furthermore, we used the popular Fréchet Inception Distance (FID) [77] to assess the visual quality of completed images by comparing the distance between distributions of completed and real images in a deep feature domain. As for the traditional pixel-level and patch-level image quality metrics, including the mean ℓ_1 loss, structural similarity (SSIM), and peak signal-to-noise ratio (PSNR), we select the closest generated image to the ground truth image for calculation, as these metrics are based on one-to-one pairing.

Table 4.1 shows diversity and image quality analysis for different network structures. We note that our method not only improved the image quality significantly (relative 18% improvement for FID), but also generated multiple and diverse completion results. Here, BicycleGAN obtained relatively higher diversity scores than

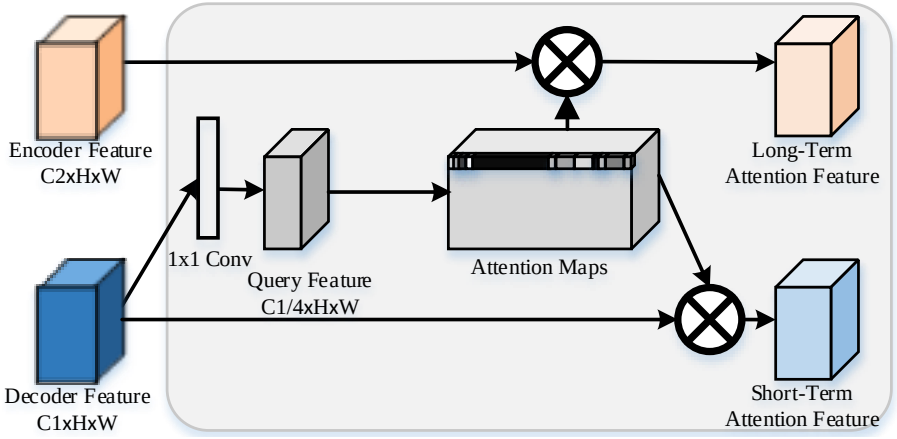


FIGURE 4.6: **Our short+long term patch attention layer**. The attention map is directly computed on the decoder features to estimate the content similarity in the same domain. After obtaining the self-attention scores, we use these to compute self-attention on decoder features, as well as contextual flow on encoder features.

our baseline framework by using cycle loss instead of reconstruction loss. However, the completed images are of low quality (as shown in Figure 4.5), which suggests that despite increased diversity, its network structure is not directly suitable for image completion.

Figure 4.5 shows some sampled examples of each structure. We observe that CVAE [189] obtains reasonable results, yet with little variation. The framework has likely learned to ignore the sampling and predicted a deterministic outcome as it always tries to rebuild the original ground truth during the training no matter what masks are used to degrade the input. As for "Instance Blind", If we enforced the generated image back to the original "ground truth" I_g, the experience will be similar to the CVAE [189]. The visual results of BicycleGAN are much worse than other methods. In their model, the latent code $\mathbf{z}$ to the encoder is replicated from $1 \times 1 \times Z$ to $H \times W \times Z$, where the different spatial position holds the same random value that does not represent any semantic meaning. On the contrary, our latent code $\mathbf{z}$ is inferred from the visible pixels during the testing, which includes the predicted semantic information from the visible pixels.

4.3.2 Short+Long Term Patch Attention

A weakness of purely convolutional operations is that they have limited spatial ranges, and cannot efficiently exploit distant correlation. Extending beyond the Self-Attention in SAGAN [220], we propose a novel short + long term patch attention layer that not only to use the self-attention within a decoder layer to harness

distant spatial context, but also to further capture *feature-feature context* between encoder and decoder layers. Our *key novel insight* is: doing so would allow the network a choice of attending to the finer-grained visible features in the encoder or the more semantically generative features in the decoder, depending on circumstances. Our proposed structure is shown in Figure 4.6.

4.3.2.1 Self-Patch-Attention Map

Feature attention has been widely used in image completion task [178, 202, 208, 214]. They calculate the attention map by comparing low-frequency decoder features of holes and high-frequency encoder feature of visible regions. Then, the high-frequency features are copied from visible regions to the missing holes based on the similarity score. However, this is a little contradictory as *feature transfer is unnecessary when two features are very similar, but when needed the features are too difficult to be matched easily.*

To address this, we calculate the content similarity in itself feature domain, the decoder feature. Our attention map calculates the response at a position in a sequence by paying attention to other position in the *same sequence*. Given the features $\mathbf{f}_d$ from the previous decoder layer, we first calculate the point attention score of:

$$\mathbf{A}_{j,i} = \frac{\exp(s_{i,j})}{\sum_{i=1}^{N} \exp(s_{i,j})}, \text{where } s_{i,j} = \theta(f_{di})^{\top}\theta(f_{dj}), \tag{4.14}$$

where $\mathbf{A}_{j,i}$ represents the similarity of i^{th} location to the j^{th} location. $N = H \times W$ is the number of pixels, while θ is a 1x1 convolution filter for refining the feature.

Inspired by PatchMatch [10], we further ensure the consistency of attention maps by fusing the similarity score in a square patch:

$$\hat{\mathbf{A}}_{j,i} = \sum_{j' \in U_j, i' \in U_i} \mathbf{A}_{j',i'} \tag{4.15}$$

where U_j and U_i are the neighborhood patch sets at j^{th} and i^{th} locations separately. We fixed the square size as 3×3 throughout this chapter.

4.3.2.2 Short-Term Attention from Decoder Full Regions

After we obtain the attention map, the non-local information is fused in the decoder features. This leads to the short-term intra-layer attention feature (**Short-Term**

(a) Input (b1) Output (c1) Attention Map (b2) Output (c2) Attention Map (b3) Output (c3) Attention Map

FIGURE 4.7: **Texture flow (white arrow) for diversely generated contents** with the same mask. (a) Masked input image. (b*) Multiple and diverse results as well as one query point (red dot). (c*) The corresponding attention maps (unsampled to original image size for visualization) for the query points in the output. The high-quality textures are copied from different visible regions (blue rectangles) to the generated regions (white rectangles), depending on what content has been generated.

Attention in Figure 4.6) and the output $\mathbf{y}_d$:

$$c_{dj} = \sum_{i=1}^{N} \hat{\mathbf{A}}_{j,i} f_{di} \ , \quad \mathbf{y}_d = \gamma_d \mathbf{c}_d + \mathbf{f}_d \tag{4.16}$$

where, we use a scale parameter γ_d to balance the weights between attention feature $\mathbf{c}_d$ and decoder feature $\mathbf{f}_d$. The initial value of γ_d is set to zero.

4.3.2.3 Long-Term Attention from Encoder Visible Regions

In addition, specifically for image completion task, we not only need the high-quality results for missing holes, but also need to ensure the appearance consistency of the generated patches of missing parts and the original patches of visible parts. Then, we introduce a long-term inter-layer attention feature (**Long-Term Attention** in Figure 4.6), in which the response attends to visible encoded features $\mathbf{f}_e$. Therefore, the output $\mathbf{y}_e$ is given by:

$$c_{ej} = \sum_{i=1}^{N} \hat{\mathbf{A}}_{j,i} f_{ei} \ , \quad \mathbf{y}_e = \gamma_e (1 - M)\mathbf{c}_e + M\mathbf{f}_e. \tag{4.17}$$

As before, a scale parameter γ_e is used to combine the encoder feature $\mathbf{f}_e$ and the attention feature $\mathbf{c}_e$. However, unlike the decoder feature $\mathbf{f}_d$ which has information for generating a full image, the encoder feature $\mathbf{f}_e$ only represents visible parts $\mathbf{I}_m$. Hence, a binary mask M (1 denotes visible regions, and 0 represents the holes) is used. In this way, the high-quality visible features are flowed to the holes based on the content similarity. Finally, both the short- and long-term attention features are aggregated and fed into further decoder layers.

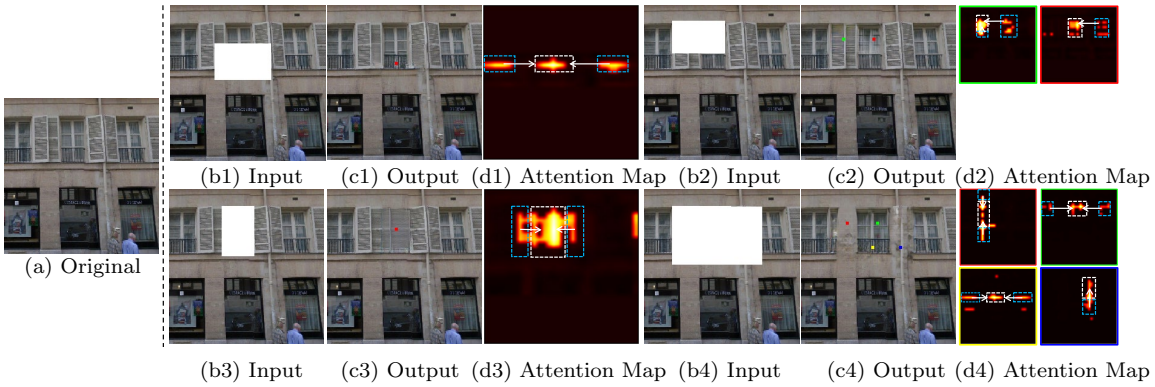


FIGURE 4.8: **Texture flow (white arrow) for different masked regions.** (a) Original image. (b*) Masked input images with different degraded regions. (c*) The completed results as well as query points (denoted by color dots). (d*) The corresponding attention maps for the query points in the output. The results attend to different visible regions (blue rectangles) based on the different visible content.

4.3.2.4 Analysis

Readers may wonder why the proposed short-long term attention layer would achieve better performance than existing contextual attention layers [208, 214]. Here, we show that the proposed module is able to exploit non-local information from *both* visible and generated regions for the holes, instead of purely copying high-frequency information from visible regions.

In Figures 4.7 and 4.8, completed results, along with corresponding attention maps for query points, are presented. Here, only points with the highest attention scores are highlighted. We use white arrows to explicitly show the texture flow, or how the attention layer copies information from high-quality visible features (blue rectangles) to the originally masked regions (white rectangles). In Figure 4.7, we find that the proposed attention layer attends to different visible regions for differently generated content, as sampled from our model. In this way, the model ensures appearance consistency between the diversely generated appearance and the visible pixels. Figure 4.8 shows other examples of texture flow from visible regions to masked regions. When we mask different regions of the window, the proposed attention layer learns to copy high-quality pixels from corresponding visible regions (blue rectangles) to the missing holes (white rectangles).

We also compare the proposed attention layer to previous methods, including contextual attention (**CA**) [214] and self-attention (**SA**) [220] for image completion. As shown in Figure 4.9, our proposed attention layer borrows features from different positions, rather than directly copying similar features from one visible position like

(a) Original (b) Input (c) CA (d) SA (e) Short-Long Term Attention (f) Attention Map

FIGURE 4.9: **Comparison of various attention modules.** (a) Original Image. (b) Masked input image. (c) Results of contextual attention [214]. (d) Results of self-attention [220]. (e) Multiple results of our method with short-long term patch attention. (f) The corresponding attention maps for the query points, *e.g.* hair (red), skin (green), eye (yellow) and teeth (blue) on the face.

CA. In the building scene, CA's result is of similar high quality to our method, due to the presence of repeated structures. However, in the case of faces, if the mask regions are large, both CA and SA are unable to generate high-quality results. It is worth mentioning that CA can copy high-quality pixels for skin (purple rectangle) from the visible skin, yet obtaining unrealistic eyes (blue rectangle). This is because when two eyes are masked, they cannot copy non-local similar patches from other visible parts. Conversely, SA only copies features in the decoder network, ignoring high-quality visible features. While it generates plausible appearances for skin and eyes, the generated skin is inconsistent to the visible skin. Our attention module is able to utilize both decoder features (which do not have masked parts) and encoder features appropriately. In completing the left eye, information is distantly shared from the decoded right eye. When it comes to completing a point in a masked hair region, it will focus on encoded features from visible hairs.

4.4 User Interface

We designed a real-time interactive system[2] that allows the user to easily explore and edit the image by creating free-form or regular masks.

As shown in Figures 4.10 and 4.11, the interface is composed of a button ("Random") to load in an input image, a button ("Mask Type") to select the mask type (*free-from* or *regular*), a button ("Fill") to fill into reasonable content as well as

[2]The local version is available on https://github.com/lyndonzheng/Pluralistic-Inpainting, while the online real-time system is available on http://www.chuanxiaz.com/project/pluralistic/.

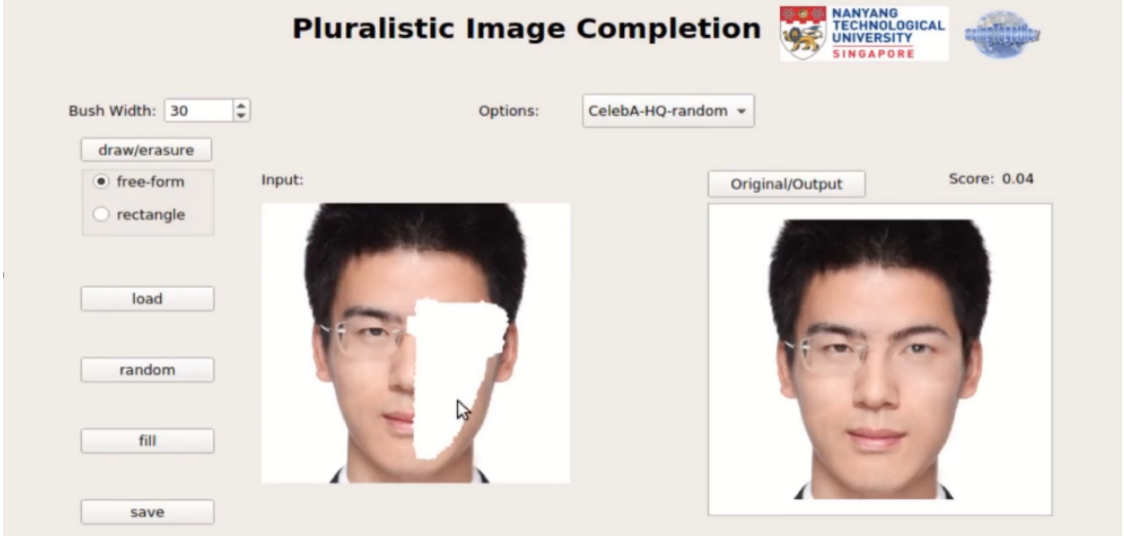


FIGURE 4.10: **Local interface for free-form image editing.** We produce a local interface on the GitHub.

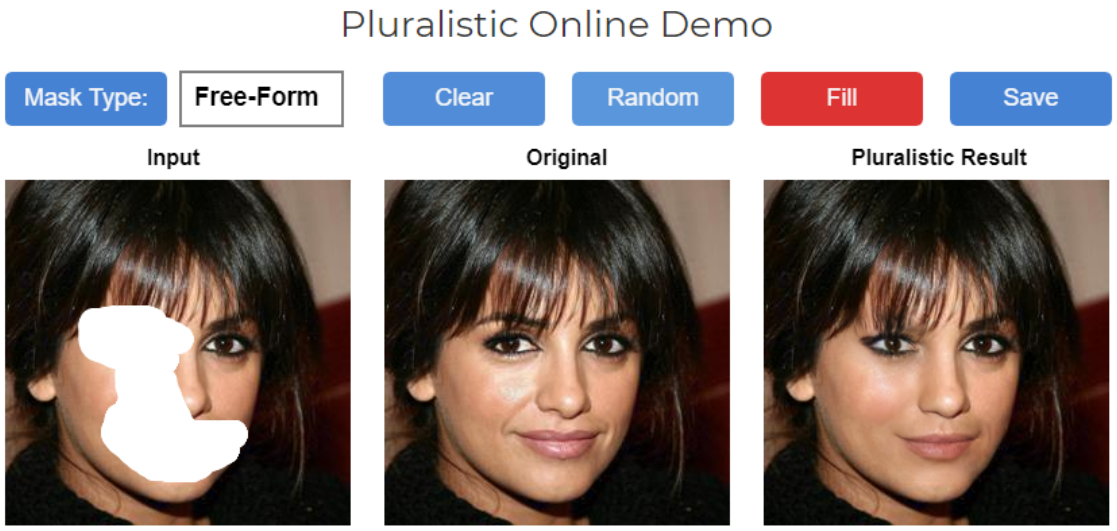


FIGURE 4.11: **Online interface for free-form image editing.** We produce an online interface on the Website that can be used to edit the image for diverse outputs.

visually realistic appearance. After the user makes an edit, the interface delivers the corresponding output. If the user hits the "Fill" button many times, it will randomly output a different result each time. Some of our results are presented in Section 4.5 by using this user interface.

4.5 Results and Applications

4.5.1 Experimental Details

Datasets We evaluated the proposed **PICNet** with arbitrary mask types on various datasets, including Paris [41], CelebA-HQ [95, 126], ImageNet [162] and Places2 [232]. Here, we only train one model to evaluate both the general free-form irregular masks and the center regular mask.

Metrics Quantitative evaluation is tricky for the pluralistic image completion task, as our goal is to get diverse but reasonable solutions for a given masked image. The original image is only one solution of many, and comparisons should not be made only based on this image. Therefore, we first used the Fréchet Inception Distance (FID) [77] and Inception Score (IS) [163] to assess the quality of the completed image, as they are measured on learned features over the whole test set. Following [123, 139], we then reported the traditional pixel- and patch-level image quality metrics, including ℓ_1 loss, structure similarity index (SSIM) and peak signal-to-noise ratio (PSNR). We additionally compared the visual realism of all results using human judgment, as previously proposed [221] and widely adopted for image generation [87, 139, 144, 234, 235].

Training PICNet is implemented in PyTorch v1.4. The missing regions take value 0 in the input. We highlight the missing regions as white in the figures only for visual purposes. Each mini-batch has 16 images per NVIDIA V100 GPU and each input has 1 reconstructive and 1 generative output. For the binary masks, we used randomly regular and irregular holes. However, allowing unrestricted mask sizes is more difficult than keeping to center masks as in our prior work [227]. In order to train the networks to convergence, two training steps were used: first, the completion network was trained using only the losses for the top reconstructive path, which has full information from both visible and missing regions. To do this, we estimated the missing regions' distributions that relate to different mask sizes. After we obtained the distribution of missing regions through the reconstructive path, the bottom generative path was trained to infer the distribution of missing holes based on the visible parts, from which we can generate multiple results.

Inference At test time, only the bottom generation path will be applied to generate *multiple* and *diverse* results based on the visible information. We sampled

50 images for each masked input image $\mathbf{I}_{\text{m}}$. Note that the distribution we sampled from is also learned from the visible regions, rather than a fixed distribution used in previous works [176, 189]. The visual results were automatically selected based on the higher discriminator scores.

4.5.2 Comparison with Existing Work

We mainly compare our method with 6 methods:

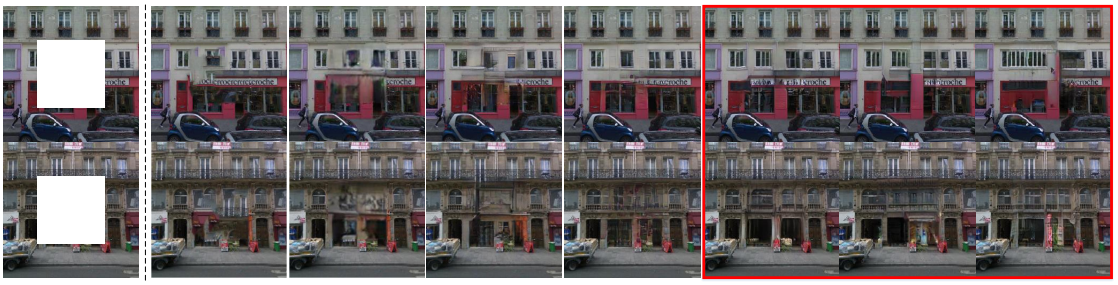
- **PM:** PatchMatch [10], the state-of-the-art non-learning based approach.
- **CE**[3]**:** Context Encoder [146], the first learning-based method for large holes.
- **GL**[4]**:** Globally and Locally [86], the first learning-based method for arbitrary regions.
- **CA**[5]**:** Contextual Attention [214], the first method combining learning- and patch-based methods.
- **PConv**[6]**:** Partial Convolution [123], the first learning-based method for free-form irregular holes.
- **EC**[7]**:** EdgeConnect [139], the latest works using auxiliary edge information.

Compared to these approaches, our **PICNet** is the first work considering multiple solutions on various datasets for this ill-posed problem. For fair comparison among learning-based methods, we mainly reported the results with *each model trained on the corresponding dataset*. We consider the released models on the respective authors' websites to be their best performing models.

4.5.2.1 Center Region Completion

Qualitative Results In Figure 4.12, we first show the visual results on the Paris dataset [41]. **PM** works by coping similar patches from visible regions and obtains good results on this dataset with repetitive structures. **CE** generates reasonable structures with blurry textures. **Shift-Net** produces better results by copying

[3]https://github.com/pathak22/context-encoder
[4]https://github.com/satoshiiizuka/siggraph2017_inpainting
[5]https://github.com/JiahuiYu/generative_inpainting
[6]https://github.com/NVIDIA/partialconv
[7]https://github.com/knazeri/edge-connect



(a) Input (b) PM (c) CE (d) Shift-Net (e) EC (f) PICNet

FIGURE 4.12: **Qualitative results on Paris val set** [41] for center region completion. Here, we compare with **PM** [10], **CE** [146], **Shift-Net** [202] and **EC** [139]. Note that, our **PICNet** generates different numbers of windows and varying door size.

(a) Input (b) PM (c) CE (d) GL (e) CA (f) PICNet

FIGURE 4.13: **Qualitative results and comparisons with the PM [10], CE [146], GL [86] and CA [214] on the ImageNet [162]**. Our **PICNet** tries to generate some semantic result for the animals, when the significant semantic information is missing.

feature from visible regions to holes, which is similar to **CA** (**CA** did not release model for Paris). **EC** provides a single reasonable solution. Compared to these, our **PICNet** model not only generates more natural images with high-quality, but also provides multiple results, *e.g.* different numbers of windows and varying door sizes.

Next, we report the performance on the more challenging ImageNet dataset [162]. For a fair comparison, we also used a subset of 100K training images of ImageNet to train our model as previous works [86]. Visual results on a variety of objects from the validation set are shown in Figure 4.13. These visual test images are those chosen in [86]. We note that, while learning-based methods **CE**, **GL** and **CA** provide correctly semantic results, our model is able to infer the content quite effectively. We observe that our model tries to generate full body for the first dog,

	Size	GL [86]	CA [214]	PConv [123]	EC [139]	PICNet
FID†	[0.01, 0.1]	10.40	12.63	11.59	**8.78**	9.33
	(0.1, 0.2]	26.42	24.63	26.46	16.75	**15.93**
	(0.2, 0.3]	50.37	39.87	47.32	28.37	**22.74**
	(0.3, 0.4]	79.01	57.44	77.16	43.74	**36.23**
	(0.4, 0.5]	108.37	76.10	91.29	63.15	**53.14**
	(0.5, 0.6]	125.41	93.55	113.62	93.43	**78.53**
IScore⋆	[0.01, 0.1]	34.66	37.33	**38.62**	38.57	38.18
	(0.1, 0.2]	31.94	34.95	31.97	**35.59**	35.36
	(0.2, 0.3]	24.26	28.79	25.53	31.06	**32.95**
	(0.3, 0.4]	17.00	22.52	18.43	26.27	**28.73**
	(0.4, 0.5]	12.13	18.35	12.43	18.94	**21.20**
	(0.5, 0.6]	8.12	13.37	10.2	12.84	**16.99**

TABLE 4.2: **Quantitative comparisons on ImageNet [162]** with free-form masks provided in [123]. † = lower is better. ⋆ = higher is better. Here, we used the top 10 samples (ranked by the discriminator score) in our models for the latest learning-based feature-level image quality evaluation.

and the mouth for the second dog. Meanwhile, our **PICNet** provides *multiple* and *diverse* results, from which we can choose different realistic results.

4.5.2.2 Free-form Region Completion

We further evaluate our model on various datasets with irregular holes as proposed by Liu *et al.* [123]. In this testing dataset, they generated 6 categories of free-form masks with different hole-to-image area ratios: [0.01, 0.1], (0.1, 0.2], (0.2, 0.3], (0.3, 0.4], (0.4, 0.5], (0.5, 0.6]. Each has 2,000 irregular masks. Results are compared against the current state-of-the-art approaches both qualitatively and quantitatively. Results of **GL** and **CA** were obtained from their released models, which were trained only on regular random masks. Results of **EC** were also generated from their released model, which was trained on the same images and masks as ours. As **PConv** only provided the partial convolutional operation, we reproduced the model with the same masks.

Quantitative Results In Table 4.2, we first report the FID and IS results on the ImageNet test set [162]. In this setting, we used our top 10 samples of the 50 generated images for the evaluation (automatically voted using the discriminator score). As can be seen, while our multiple results are slight worse than **EC** on small mask sizes, we improve FID and IS significantly on large mask ratios, *e.g.* "78.53" *vs* "93.43" (*16% relative improvement*) FID for mask ratio (0.5, 0.6]. This

	Size	GL [86]	CA [214]	PConv [123]	EC [139]	PICNet
ℓ_1(%)†	[0.01, 0.1]	0.023	0.024	0.021	0.020	**0.010**
	(0.1, 0.2]	0.035	0.034	0.030	0.025	**0.016**
	(0.2, 0.3]	0.050	0.047	0.042	0.033	**0.025**
	(0.3, 0.4]	0.066	0.061	0.057	0.042	**0.035**
	(0.4, 0.5]	0.081	0.075	0.073	0.051	**0.046**
	(0.5, 0.6]	0.095	0.093	0.099	0.068	**0.064**
SSIM*	[0.01, 0.1]	0.915	0.908	0.917	0.923	**0.963**
	(0.1, 0.2]	0.853	0.845	0.859	0.878	**0.914**
	(0.2, 0.3]	0.767	0.765	0.782	0.820	**0.852**
	(0.3, 0.4]	0.682	0.691	0.704	0.760	**0.785**
	(0.4, 0.5]	0.600	0.613	0.622	0.693	**0.712**
	(0.5, 0.6]	0.529	0.532	0.513	0.599	**0.618**
PSNR*	[0.01, 0.1]	28.42	26.85	28.79	29.47	**32.26**
	(0.1, 0.2]	24.41	23.18	24.67	26.25	**27.33**
	(0.2, 0.3]	21.33	20.44	21.63	23.82	**24.44**
	(0.3, 0.4]	19.11	18.63	19.39	21.95	**22.32**
	(0.4, 0.5]	17.56	17.30	17.75	20.44	**20.71**
	(0.5, 0.6]	16.48	16.08	15.68	18.53	**18.72**

TABLE 4.3: **Quantitative comparisons over Places2** [232] on free-form masks provided in [123]. † = lower is better. * = higher is better. Here, the closest to the original ground truth samples in our method are selected for the traditional pixel- and patch-level image quality evaluation.

suggests that when the mask ratios are small, it is sufficient to predict a *single* best result based on the neighboring visible pixels, yet it is not reasonable when the mask ratios are large. The latter requires our approach of generating multiple and diverse results that match the testing set distribution.

Traditional pixel- and patch-level comparison results are reported on the Places2 test set [232] in Table 4.3. As these metrics require one-to-one matched images for the evaluation, we selected one sample from our multiple results, with the best balance of quantitative measures for comparison. Without bells and whistles, all instantiations of our model outperform the existing state-of-the-art models, indicating that our random samples include the close example to the original image. While the prior works [86, 123, 139, 214] strongly enforce the generated images to be the same as the original images via a reconstruction loss, the testing images are not in the training set.

Qualitative Results. Qualitative comparison results are visualized in Figures 4.14, 4.15 and 4.16. Our PICNet is able to achieve good results for multiple solutions

(a) Original (b) Input (c) PM (d) PConv (e) EC (f) PICNet

FIGURE 4.14: **Comparison of qualitative results on Paris val set [41] with free-form masks from PConv [123].** (a) Original image. (b) Masked input. (c) Results of **PM** [10]. (d) Results of **PConv** [123]. (e) Results of **EC** [139]. (f). Our multiple and diverse results.

(a) Original (b) Input (c) PM (d) CA (e) EC (f) PICNet

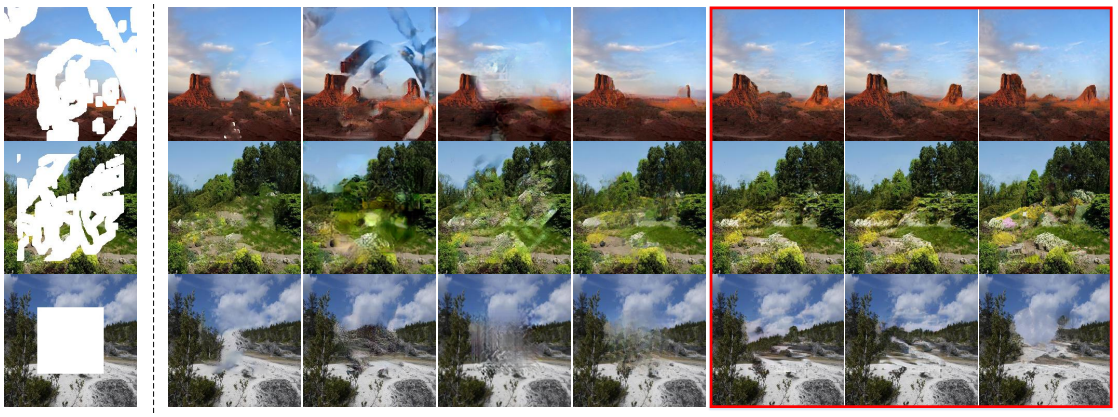
FIGURE 4.15: **Qualitative results on CelebA-HQ testing set [95, 126] with free-form masks from PConv [123].** (a) Original image. (b) Masked input. (c) Results of **PM** [10]. (d) Results of **CA** [214]. (e) Results of **EC** [139]. (f). Our multiple and diverse results.

even under challenging conditions.

In Figure 4.14, we show some results on Paris dataset. We can see that **PM** and **PConv** fail to synthesize semantic structure for large holes. The **EC** works well on the obvious structure by utilizing the auxiliary edge. Our method was explicitly trained to copy information from visible parts, leading to better visual results on repetitive structures, *e.g.* the window in the first row. Furthermore, our model provides multiple and diverse results for one given masked image. More results are available online[8].

Figure 4.15 shows some results on the Celeba-HQ dataset. We can see that the non-learning-based method **PM** is unable to generate reasonable semantic content in the images. While the **CA** is able to generate novel content on the face, it is not as suitable for large holes. **EC** results in reasonable semantic structure but blurry and inconsistent images. Our approach was explicitly trained for variable

[8]https://github.com/lyndonzheng/Pluralistic-Inpainting



(a) Input (b) PM (c) CA (d) PConv (e) EC (f) PICNet

FIGURE 4.16: **Qualitative results on Place2 testing set [232] with various masks.** (a) Masked input. (b) Results of **PM** [10]. (c) Results of **CA** [214]. (d) Results of **PConv** [123]. (e) Results of **EC** [139]. (f). Our multiple and diverse results.

results, rather than strongly enforcing the completed image to be close to the original. Hence, our **PICNet** can provide multiple plausible results with different expressions. The online demo is also provided on our project page[9].

In Figure 4.16, we further show results on the more challenging Places2 dataset. The non-learning-based **PM** fills in reasonable pixels for natural scenes by copying similar patches from visible parts to missing holes. The **CA** only works well on regular masks as their released model was only trained on random regular masks. **EC** results are not as realistic. Instead, we can select plausible images from PICNet's multiple sampled results. Furthermore, it is hard to identify the filled-in areas in our completed images, as our short-long term patch attention copies non-local information from visible regions based on correctly predicted content.

4.5.2.3 Visual Turing Tests

We additionally compared the perceived visual fidelity of our model against existing approaches using human perceptual metrics, as proposed in [221]. We conducted two types of user surveys: *2 alternative forced choice* (2AFCs) and *visual fidelity and perceived quality* (VFPQ). In particular, for 2AFCs, we randomly presented a generated image from an undisclosed method to the participants, and asked them to decide whether the presented image was real or fake. For quality control, we also inserted a number of real images to avoid negative testing. For VFPQ, we gave the participants a masked input and the corresponding results from all

[9]http://www.chuanxiaz.com/project/pluralistic/

	GL [86]	**CA** [214]	**EC** [139]	**PICNet**	**Real**
2AFC(%)	15.1±1.8	17.8±2.0	44.2±3.7	57.0±4.5	90.44±1.5

TABLE 4.4: **2-alternative-forced-choice (2AFCs) score on CelebA-HQ [95, 126] testing set.** All testing images were degraded by center masks. Here, the participants were required to judge whether a randomly displayed image was real *or* fake. The reported values are the percentages of images generated by each method that were judged "real".

	VFPQ(%)			
	[0.01, 0.1]	(0.1, 0.2]	(0.2, 0.3]	(0.3, 0.4]
GL [86]	23.3 ± 4.3	9.8 ± 1.6	6.4 ± 1.0	4.1 ± 0.4
CA [214]	11.4 ± 2.1	9.7 ± 1.3	7.6 ± 0.9	8.6 ± 0.9
PConv [123]	27.8 ± 4.0	13.5 ± 1.6	11.0 ± 1.4	5.3 ± 0.7
EC [139]	42.3 ± 6.0	38.8 ± 4.9	33.6 ± 3.2	26.7 ± 3.3
PICNet	57.5 ± 3.6	63.0 ± 4.0	69.9 ± 3.8	71.4 ± 3.4

TABLE 4.5: **Visual fidelity and perceived quality (VFPQ) score on Places2 [232] test set.** All testing images were degraded by free-form masks provided in PConv [123]. Participants selected the most realistic image from among blinded methods for the same masked input, with multiple selections allowed. Headers are ranges of mask sizes (as fraction of image). For each method, we report the percentage of trials for which it was selected, and the 95% margin of error.

methods (blinded), and asked the participants to choose the image that was the most visually realistic. The participants were allowed to vote for multiple images simultaneously, if they felt the images were equally realistic. For each participant, we randomly presented 100 questions, consisting of 60 2AFCs examples and 40 VFPQ questions. We collected 47 valid surveys with 4,700 answers.

We first show the 2FACs evaluation results in Table 4.4. Most participants correctly identified the real image during the evaluation, showing that they made conscientious discerning judgement. Our model achieved better realism scores than existing state-of-the-art methods. Table 4.5 shows the VFPQ evaluation results. We found that the participants strongly favored our completed results for all mask ratios, and especially so on the challenging large mask ratios. This suggests that once the visible regions do not impose strong constraints, our multiple and diverse results were naturally varied but mostly realistic and reasonable.

4.5.3 Additional Results

We show additional results of our proposed PICNet in Figures 4.17, 4.18 and 4.19. Our approach is suitable for a wide range of applications, *e.g.* face editing, scene

FIGURE 4.17: **Additional results of our PICNet on the CelebA-HQ test set [95, 126] for free-form image editing.** (a) Original image. (b) Masked input image. (c) Output of our **PICNet**. In the first two columns, we erased eyeglasses. Wrinkles and facial hair were removed in the next two columns. Finally, we freely changed mouth expressions. Note that due to the provision of multiple and diverse results, the users can easily select their favorite result. We refer readers to our online demo for testing.

recomposition, object removal and outpainting.

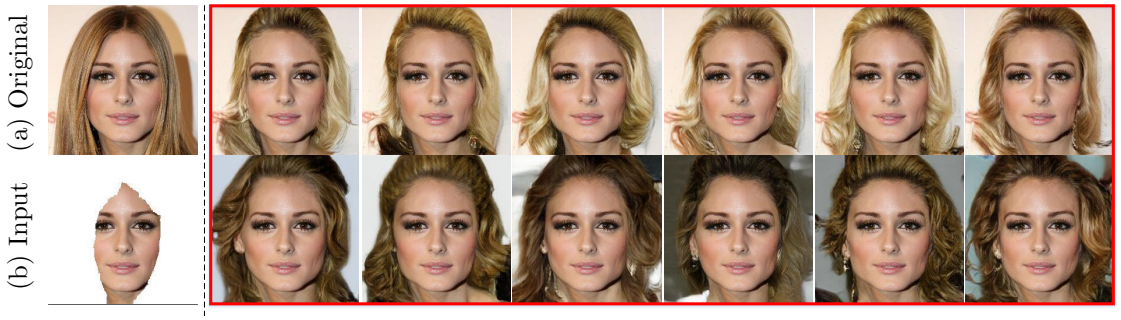
Face Editing We first show free-form image editing on face images in Figure 4.17. Our model works well for conventional object removal, *e.g.* removing eyeglasses in the first two columns. Next, we smoothed faces by removing wrinkles and facial hair. Finally, we changed mouth expressions by selecting an example among our multiple and diverse completed results.

High-Resolution Natural Image Editing The basic PICNet did not handle high resolution (HR) image completion, because the generation from a random vector $\mathbf{z}$ only works for a fixed feature size [97]. However, following the *two-stage* image completion approaches [139, 178, 204, 208, 214, 215, 218], we trained another encoder-decoder framework to refine the fixed resolution output of our PICNet. Since this work does *not* focus on HR images, we used a simple design for the refinement network by directly reapplying the PICNet framework in the second refinement stage, but without the sampling process. Note that the multiple and diverse solutions were seeded by the first content generation stage.

As can be seen in Figure 4.18, our approach produces diverse results as well as visually realistic appearance for HR natural image editing, *e.g.* reshaping the

(a) Original (b) Input (c) Outputs

FIGURE 4.18: **Additional results of PICNet on the Places2 test set [232] for free-form image editing.** (a) Original image. (b) Input masked image. (c) Multiple and diverse outputs of our **PICNet**. Here, we show examples of reshaping the mountain ridge and subject removal, but, unlike conventional inpainting, we can provide multiple and diverse choices and on high-resolution images.



(c) Outputs

FIGURE 4.19: **Outpainting examples of our models.** (a) Original image. (b) Masked input. (c) Multiple and diverse results of our **PICNet**. Note that, it provides different hairstyles for the users.

mountain ridge and generating various mountain streams. This demonstrates that our model works well for HR images.

Outpainting In our dual pipeline framework, the masked image $\mathbf{I}_m$ and its corresponding complement image $\mathbf{I}_c$ can be easily swapped. Therefore, we randomly reversed the input mask during training on Celeba-HQ. Figure 4.19 shows examples where information is missing from the image border regions. This "outpainting" is a challenging task as these regions have much larger uncertainty [86]. Note that

(a) Original (b) Input (c) Failure Outputs

FIGURE 4.20: **Failure cases of our PICNet.** (a) Original image. (b) Masked input. (c) Failure results of our **PICNet**, where the semantic information is heavily masked, *e.g.* only four legs are visible of the dogs.

the subject's hair can be significantly varied during completion, suggesting that our model is applicable to style editing. Our structure has been extended to other related tasks, such as spherical image generation [69].

4.6 Limitations and Discussion

In this chapter, a novel solution is presented for the image completion task. Unlike existing methods [86, 139, 146, 208, 214, 215], our probabilistically principled framework can generate multiple and diverse solutions with plausible content for a given masked image. The resulting **PICNet** shows that prior-conditional lower bound coupling is significant for conditional image generation, leading to a more reasonable two-branch training than the current deterministic structure. We also introduce an enhanced short+long term patch attention layer, which improves realism by automatically attending to both high quality visible features and semantically correct generated features.

Experiments on a variety of datasets demonstrated that the multiple solutions were diverse and of high quality. On the latest learning based feature-level metrics and traditional pixel- and patch-level metrics, we demonstrated that PICNet outperformed the single-solution approaches [86, 123, 139, 214], especially for large mask ratios with large uncertainty. We further showed in studies that users

strongly favored our completed results when compared to the results in existing approaches. We additionally demonstrated that our PICNet is suitable for many interesting free-form image editing, *e.g.* object removal, expression changing, and scene recomposition. These multiple and diverse results can also be easily extended to HR image editing.

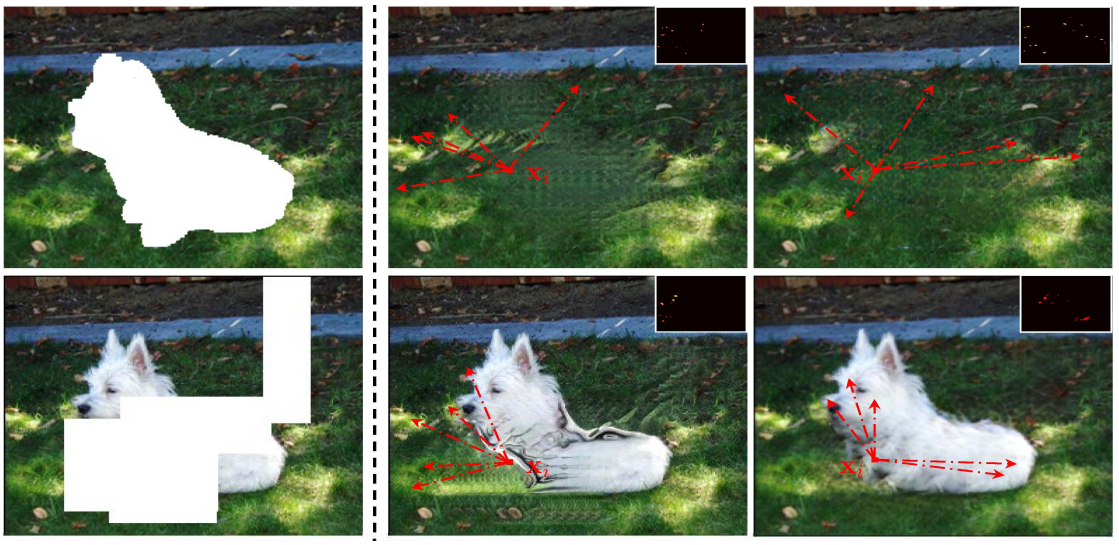
Although the proposed model achieved better results than prior methods on various datasets by selecting images from the number of diverse sampling results, the model does not cope well with heavily structured objects with important information missing, as shown in Figure 4.20. As semantic image completion is as yet an immature task that builds upon conventional image inpainting, a full understanding of semantic image content remains a challenge. In Figure 4.20(top), we can see that although the four legs of the dog are visible, the model cannot generate a complete dog even after multiple sampling. This is a significant issue in current image completion task. When the important semantic regions, such as head or body of human, are missing heavily, the current deep learning models cannot correctly imagine these missed contents. In the bottom image, if the content is not correctly generated, our attention model fails to provide high-quality visual results. In Chapter 5, I aim to address these issues.

Chapter 5

Image Completion via Transformer

The previous chapter introduces multiple and diverse results for the high subjective image completion task, but it has some failed cases, especially when the mask is very large. Bridging distant context interactions is important for high-quality image completion with large masks. However, previous methods that attempt this via deep or large receptive field (RF) convolutions cannot escape from the dominance of nearby interactions, which may tend to inferior results. In this chapter, I propose treating image completion as a directionless sequence-to-sequence prediction task, and deploy a transformer to directly capture long-range dependence in the encoder in a first phase. Crucially, a *restrictive CNN* with small and non-overlapping RF is employed for token representation, which allows the transformer to explicitly model the long-range context relations with equal importance in all layers, without implicitly confounding neighboring tokens when larger RFs are used. In a second phase, to improve appearance consistency between visible and generated regions, a novel attention-aware layer (AAL) is introduced to better exploit distantly related features and also avoid the insular effect of standard attention.

This chapter is organized as follows: I first introduce the motivation in Section 5.1 and then discuss the related works in Section 5.2. Next, I describe the proposed transformer-based completion framework in Section 5.3. Section 5.5 demonstrates how models with a transformer can be used to improve the completion performance. Finally, I will discuss the limitations and further directions in Section 5.6.



(a) Masked 1nput (b) HiFill [208] (c) Ours-TFill

FIGURE 5.1: **An example of information flow in image completion with free-form masks.** The position $\mathbf{x}_i$'s response (flow) is calculated by inferring the *Jacobian* matrix between it to all pixels in the given masked input. Here, only the highest flows are shown. Our TFill correctly captures long-range visible context flow, even with a large mask splitting two semantically important zones.

5.1 Motivation

In Section 4.1, we introduce how expert conservators would restore damaged art, where the first and most important step is to imagine the semantic content to be filled based on the overall visible scene. However, *bridging and exploiting visible information globally, after it had been degraded by arbitrary masks* remains a main challenge in this task. As depicted in Figure 4.1 top row, when the entire dog is masked, the natural expectation is to complete the masked area based on the visible background context. In contrast, in the bottom row, when the free-form regular mask covers the bulk of the dog but leaves the head and tail visible, it is necessary but highly challenging to globally capture *long-range* dependencies between the two separated foreground regions, so that the masked area can be completed in not just a photorealistic, but also semantically correct, manner.

To achieve this goal, many *two-stage* approaches [139, 208, 214, 217] have been proposed, consisting of a *content inference network* and an *appearance refinement network*. They typically infer a coarse image or edge/semantic map based on globally visible information in a first phase, and then fill in visually realistic appearance in a second phase. However, this global perception is achieved by repeated *local* convolutional operations, which have several limitations. First, due to translation

equivariance in convolutions, the information flow tends to be predominantly local, with global information only shared gradually through heat-like propagation across multiple layers. Second, during inference, the elements between adjacent layers are connected via learned but fixed weights, rather than input-dependent adaptive weightings. These issues mean long-distance messages are only delivered inefficiently in a very deep layer, resulting in a strong inclination for the network to fill holes based on nearby rather than distant visible pixels (Figure 5.1 (b)).

In this chapter, we propose an alternative perspective by treating image completion as a *directionless sequence-to-sequence* prediction task. In particular, instead of modeling the global context using deeply stacked convolutional layers, we design a new content inference model, called TFill, that uses a **T**ransformer-based architecture to **Fill** reasonable content into the missing holes. An important insight here is that a transformer directly exploits long-range dependencies at every encoder layer through the attention mechanism, which *creates an equal flowing opportunity for all visible pixels, regardless of their relative spatial positions*. This reduces the proximity-dominant influence that can lead to semantically incoherent results.

Our design is motivated by the transformer literature in natural language processing (NLP) [37, 154, 155, 188]. However, it remains a challenge to directly apply these transformer models to visual generation tasks. Particularly, unlike the NLP that naturally treats each word as a vector for token embedding, it is unclear what a good token representation should be for the visual task. If we use every pixel as a token, the memory cost will make this infeasible except for very small images [22]. To mitigate this issue, our model embeds the masked image into an intermediate latent space for token representation, an approach also broadly taken by recent vision transformer models [18, 48, 231, 236]. However, unlike these models that use traditional CNN-based encoders to embed the tokens, we propose a *restrictive CNN* for token representation, which has a profound influence on how the visible information is connected in the network. To do so, we ensure the individual tokens represent visible information independently, each with a *small* and *non-overlapping* receptive field (RF). This forces *the long-range context relationships between tokens to be explicitly and co-equally perceived in every transformer encoder layer*, without neighboring tokens being entangled by implicit correlation through overlapping RF. As a result, each token will *not* be gradually affected by neighboring regions, retaining equal probability of being captured in every layer.

While the proposed transformer-based architecture can achieve better results than state-of-the-art methods [48, 208, 214, 227], by itself it only works for a fixed sequence length because of the position embedding (Figure 5.2(a)). To allow our approach to flexibly scale to images of different sizes, a fully convolutional encoder-decoder network (Figure 5.2(b))) is subsequently applied to refine the visual appearance built upon the coarse content previously inferred. We also design a novel **A**ttention-**A**ware **L**ayer (AAL) between the encoder and decoder that adaptively balances the attention paid to visible and generated content, leading to semantically superior feature transfer.

5.2 Background

5.2.1 Image Completion

As this related research has been discussed above, we refer readers to Section 4.2.

5.2.2 The Transformer Family

The transformer architecture is first proposed by Vaswani *et al.* [188], and has later become the de facto standard backbone in NLP tasks [37, 154, 155]. It merges the information between the inputs solely through attention [7], which directly models the long-range dependence by calculating the similarity of two points, regardless of their spatially relative position.

The early application of attention in computer vision is only in one deep layer to learn the long-range dependence, such as the attention in Non-local Neural Networks [194] and self-attention GAN [220]. This is because the 2D image has a quadratic cost in the number of pixels than the 1D sequence sentence with words. It will require complex engineering and high GPU memory to implement the vision transformer efficiently on hardware accelerators.

To mitigate this issue, recent works have explored different visual token representation methods to directly apply a standard transformer for visual tasks, such as image classification [22, 42], object detection [18, 236], semantic segmentation [198, 231], image generation and translation [19, 48, 85, 91]. As illuminated in Section 5.3, compared to these general token representations, our *restrictive CNN* is particularly well suited due to its compact representation that limits implicit correlation.

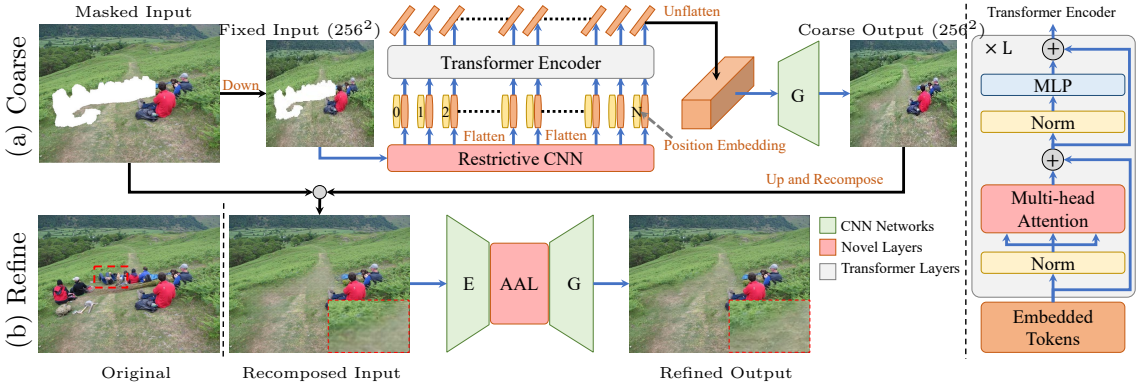


FIGURE 5.2: **The overall pipeline of the proposed method.** (a) Masked input is resized to a fixed low resolution (256^2) and it is then fed into the transformer to generate semantically correct content. (b) The inferred content is merged with the original high-resolution image and passed to a refinement network with an **A**ttention-**A**ware **L**ayer (**AAL**) to transfer high-quality information from both visible and masked regions. Note the recomposed input has repeating artifacts, which are resolved in our refined network. Zoom in to see the details.

5.3 Approach

Given a masked image $\mathbf{I}_m$, degraded from a real image $\mathbf{I}$ by a free-form mask, our goal is to learn a model Φ to infer the content for missing regions, as well as filling in with visually realistic appearance. To achieve this, our image completion framework, illustrated in Figure 5.2, consists of a content inference network and an appearance refinement network. The former is responsible for capturing the global context through a transformer encoder at a fixed scale. The embedded tokens have small receptive fields (RF) and limited capacity, preventing their states from being implicitly dominated by visible pixels nearby than far. While similar transformer-based architectures have recently been explored for visual tasks [18, 19, 22, 42, 48, 198, 231, 236], we believe our work is the first to explore this for free-form image completion, where we discover *how the token representation has a profound effect on the flow of visible information in the network, in spite of the supposedly global reach of transformers*. The latter network is designed to refine visual appearance by utilizing high-resolution visible features, and also frees the limitation to fixed image sizes.

5.3.1 Transformer-based Architecture

Background We begin by briefly reviewing the transformer [188]. As depicted on Figure 5.2 (right), a transformer encoder layer consists of multihead `self-attention`

(MSA) and `Multi layer Perception` (MLP) blocks (see Appendix B.2.1). The MSA is responsible for capturing long-range dependencies, while the MLP is applied to further transform merged features. The `LayerNorm` (LN) is used for non-linear projection. These are expressed by:

$$\mathbf{z}_0 = [\mathbf{x}^1; \mathbf{x}^2; \ldots; \mathbf{x}^N] + \mathbf{E}_{pos} \tag{5.1}$$

$$\mathbf{z}'_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1} \tag{5.2}$$

$$\mathbf{z}_\ell = \text{MLP}(\text{LN}(\mathbf{z}'_\ell)) + \mathbf{z}'_\ell \tag{5.3}$$

where $\mathbf{z} \in \mathbb{R}^{N \times C}$ is the 1D sequence of N tokens $\mathbf{x}$ with C channels, and $\mathbf{E}_{pos} \in \mathbb{R}^{N \times C}$ is the position embedding.

Transformer-Encoder In order to feed a 2D masked image $\mathbf{I}_m$ into the transformer, we first downsample the high-resolution image to a fixed size, *e.g.* 256^2. However, it is *not* feasible to run the transformer model if we directly *flatten* image pixels into a 1D sequence with 196,608 tokens. To achieve independent token representation and reduce its length, a projection is implemented using our proposed *restrictive CNN*, a decision we will analyze in Section 5.5.2. After that, we obtain a 2D feature map with size $\frac{256}{16} \times \frac{256}{16} \times C$, and then flatten it to a 1D sequence of $256 \times C$, where 256 is the sequence length and C=512 is the feature dimension. As shown in Figure 5.2 (a), once we embed the image to a 1D sequence, a transformer encoder distills long-range relationships between all tokens in every layer.

To encourage the model to *bias* to the important visible values, we replace the self-attention layer with the *masked* self-attention layer, in which a weight is applied to scale the attention scores. The initial weight $w_{key} \in (0.02, 1.0]$ is obtained by calculating the fraction of visible pixels in a small RF, *e.g.* $192/16^2$ means 3/4 of the region in the 16^2 RF contains visible pixels. It will then be gradually amplified by updating $w_{key} \leftarrow \sqrt{w_{key}}$ after every encoder layer, to *reflect* visible information flow. This initial ratio for each token is efficiently implemented in our restrictive CNN encoder using a modified partial convolution layer [123]. The implementation details can be found in Appendix B.2.

CNN-based Decoder While a one-layer non-linear projection may be used to directly map the output features back to a completed image, the visual appearance is slightly worse than using a stacked decoder. Therefore, following existing

(a) Ground Truth (b) Input (c) TFill-*Coarse* (d) TFill-*Refined*

FIGURE 5.3: **Coarse and Refined results**. (a) Ground truth. (b) Masked input degraded by free-form masks. (c) Coarse output. (d) Refined output. We can see that the refinement network not only increased image quality to a high resolution (256^2 *vs* 512^2), but also encourages the left eyeball to be consistent with the visible right eyeball using our attention-aware layer.

works [208, 214, 227], a gradual upsampling decoder is implemented to generate photorealistic images.

5.3.2 Attention-Aware Layer (AAL)

Although our TFill-*Coarse* model correctly infers reasonable content by equally utilizing the global visible information in every layer, two limitations remain. First, it is *not* suitable for high-resolution input due to the fixed length position embedding. One solution is to follow the directional sequence-to-sequence methods [22, 48] that only use the top-left context to predict the next token, in an auto-regressive manner. However, this will not adequately capture the global visible information needed for image completion. Second, the realistic completed results may not be fully consistent with the original visible appearances, *e.g.* the generated left eye having a different shape and color to the visible right eye in Figure 5.3 (c). This is because the embedded tokens are extracted from a 16^2 resolution feature map, where important high-frequency details may be lost.

To mitigate these issues, a CNN-based encoder-decoder refinement network, trained on high-resolution images, is proposed (Figure 5.2 (b)). In particular, to further utilize the visible high-frequency details, an **A**ttention-**A**ware **L**ayer (AAL) is designed to capture long-range dependencies.

As depicted in Figure 5.4, given a decoded feature $\mathbf{x}_d$, we first calculate the attention score of:

$$\mathbf{A} = \phi(\mathbf{x}_d)^\intercal \theta(\mathbf{x}_d) \tag{5.4}$$

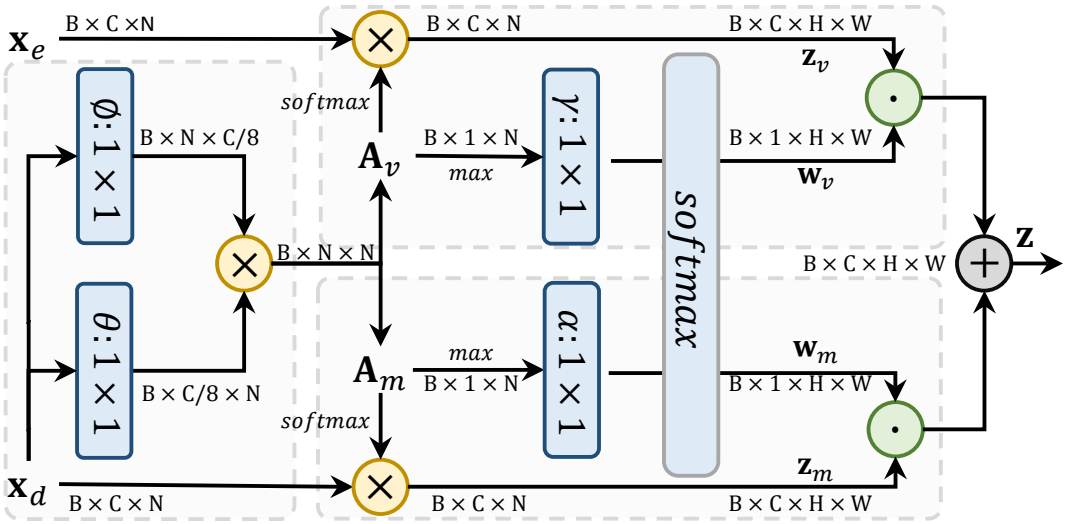


FIGURE 5.4: **Attention-aware layer.** The feature maps are shown as tensors. "$\otimes$" denotes matrix multiplication, "$\odot$" denotes element-wise multiplication and "$\oplus$" is element-wise sum. The blue boxes denote 1×1 convolution filters that are learned.

where $\mathbf{A}_{ij}$ represents the similarity of the i^{th} patch to the j^{th} patch, and ϕ, θ are 1×1 convolution filters.

Interestingly, we discover that using $\mathbf{A}$ directly in a standard self-attention layer is suboptimal, because the $\mathbf{x}_d$ features for visible regions are generally distinct from those generated for masked regions. Consequently, *the attention tends to be insular*, with masked regions preferentially attending to masked regions, and vice versa. To avoid this problem, we explicitly handled the attention to visible regions separately from masked regions. So before `softmax` normalization, $\mathbf{A}$ is split into two parts: $\mathbf{A}_v$ — similarity to visible regions, and $\mathbf{A}_m$ — similarity to generated masked regions. Next, we get long-range dependencies via:

$$\mathbf{z}_v = \texttt{softmax}(\mathbf{A}_v)\mathbf{x}_e \quad , \quad \mathbf{z}_m = \texttt{softmax}(\mathbf{A}_m)\mathbf{x}_d \tag{5.5}$$

where $\mathbf{z}_v$ contains features of contextual flow [214] for copying high-frequency details from the encoded high-resolution features $\mathbf{x}_e$ to masked regions, while $\mathbf{z}_m$ has features from the self-attention that is used in SAGAN [220] for high-quality image generation.

Instead of learning fixed weights [227] to combine $\mathbf{z}_v$ and $\mathbf{z}_m$, we learn the *weights mapping* based on the largest attention score in each position. Specifically, we first obtain the largest attention score of $\mathbf{A}_v$ and $\mathbf{A}_m$, respectively. Then, we use the 1×1 filter γ and α to *modulate* the ratio of the weights. `Softmax` normalization

is applied to ensure $\mathbf{w}_v$+$\mathbf{w}_m$=1 in every spatial position:

$$[\mathbf{w}_v, \mathbf{w}_w] = \texttt{softmax}([\gamma(\texttt{max}(\mathbf{A}_v)), \alpha(\texttt{max}(\mathbf{A}_m)])) \tag{5.6}$$

where `max` is executed on the attention score channel. Finally, an attention-balanced output $\mathbf{z}$ is obtained by:

$$\mathbf{z} = \mathbf{w}_v \cdot \mathbf{z}_v + \mathbf{w}_m \cdot \mathbf{z}_m \tag{5.7}$$

where $\mathbf{w}_v, \mathbf{w}_m \in \mathbb{R}^{B\times 1\times H\times W}$ hold different values for various positions, dependent on the largest attention scores in the visible and masked regions, respectively.

5.3.2.1 Discussion on prior art

While contextual attention [214] has recently been widely applied in image completion [178, 202, 208, 214], it is fundamentally different from the attention in our transformer-based architecture — the contextual attention is used to refine visual appearance by copying high-frequency information from visible regions to masked holes, rather than capturing and modeling long-range context for content inference. In addition, our AAL focuses on automatically selecting features from both visible and generated features, instead of copying only from visible regions [178, 202, 208, 214] or selecting through fixed weights [227].

5.4 User Interface

I designed a real-time interactive system that allows the user to easily explore and edit the high-resolution image by creating input masks. As shown in Figure 5.5, this user interface is built upon the interface in Section 4.4. Here, the resolution of the input image can be in multiples of $2^5 = 32$, *e.g.* 960×640, instead of the fixed resolution (256×256) in Chapter 4. In addition, two buttons, "load mask" and "random mask", are added to load the free-form masks provided by Liu *et al.* [123]. In this way, the user can directly load the free-form masks to assess the robustness of various methods.

5.5 Results and Applications

Datasets We evaluated our TFill with arbitrary mask types on various datasets, including CelebA-HQ [95, 126], FFHQ [96], Places2 [232], and ImageNet [162].

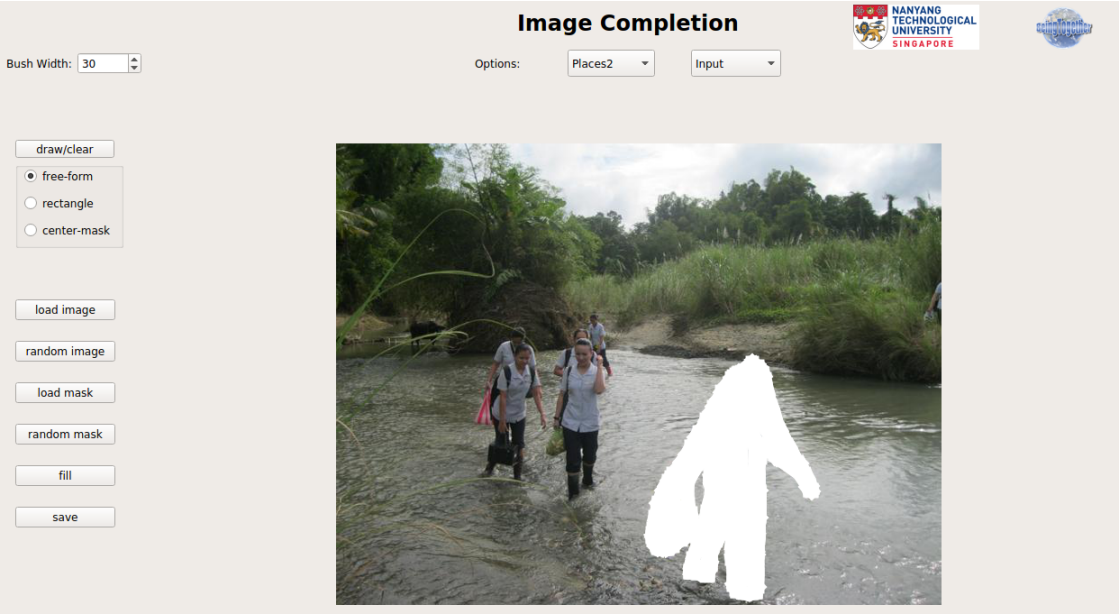


FIGURE 5.5: **Local interface for free-form high-resolution image editing.**

Metrics As proposed in previous works [214, 227], it is not reasonable to require the completed image to be exactly the same as the original image. Hence, we only report the LPIPS [222] and the FID [77] scores in the main text, leaving the traditional pixel- and patch-level evaluation results, *e.g.* the mean ℓ_1 loss, in Appendix B.1.

Implementation details Our model is trained in two stages: **1)** the content inference network is first trained for 256^2 resolution; and **2)** the visual appearance network is then trained for 512^2 resolution. Both networks are optimized using the loss $L = L_{pixel} + L_{per} + L_{GAN}$, where L_{pixel} is the ℓ_1 reconstruction loss, L_{per} is the perceptual loss [93], and L_{GAN} is the discriminator loss [65].

5.5.1 Comparison with Existing Work

Here we compared with these image completion methods: **PM** [10], a classical approach; **GL** [86], the first learning-based method for arbitrary regions; **CA** [214], the first method combining learning and patch-based methods; ours **PICNet** [227] in Chapter 4, the first work considering multiple solutions; **HiFill** [208], the latest very high-resolution (8K) method. Our TFill introduces a transformer-based architecture for this challenging image completion problem.

Table 5.1 shows quantitative evaluation results on Place2 [232], in which the images were degraded by free-form masks provided in the PConv [123] testing

	Size	GL [86]	CA [214]	PICNet [227]	HiFill [208]	TFill
LPIPS	[0.01, 0.1]	0.057	0.083	0.037	0.056	**0.027**
	(0.1, 0.2]	0.112	0.134	0.074	0.105	**0.055**
	(0.2, 0.3]	0.185	0.195	0.118	0.163	**0.092**
	(0.3, 0.4]	0.254	0.249	0.167	0.226	**0.133**
	(0.4, 0.5]	0.319	0.306	0.225	0.305	**0.180**
	(0.5, 0.6]	0.370	0.364	0.330	0.412	**0.259**
FID	[0.01, 0.1]	16.86	10.21	7.04	9.10	**5.22**
	(0.1, 0.2]	26.11	18.93	13.58	16.72	**9.67**
	(0.2, 0.3]	39.22	30.31	21.62	26.89	**15.28**
	(0.3, 0.4]	53.24	40.29	29.59	38.40	**19.99**
	(0.4, 0.5]	68.46	53.39	41.60	56.24	**25.88**
	(0.5, 0.6]	74.95	59.85	61.17	83.36	**34.58**

TABLE 5.1: Quantitative comparisons on Places2 [232] with free-form masks [123]. Without bells and whistles, TFill outperformed all traditional CNN-based models. The results are reported on 256^2 resolution, as earlier works were trained only on this scale.

(a) Original (b) Input (c) CA [214] (d) PICNet [227] (e) TFill-*Coarse* (f) TFill-*Refined*

FIGURE 5.6: **Completion results on CelebA-HQ [95, 126] testing set among CA [214], PICNet [227] and Ours.** Our results are reported for 512^2 resolution. While ours previous PICNet [227] works well for frontal facing faces, it may generate more uncanny faces with mismatched features at larger angles, *e.g.* the examples in third and last row.

set. The size column denotes the range of masking proportion applied to the images. We observe that our transformer-based model improved both LPIPS and FID quite significantly over the CNN-based state-of-the-art models in all mask

(a) Original (b) Input (c) GL [86] (d) CA [214] (e) PICNet [227] (f) HiFill [208] (g) TFill-*Refined*

FIGURE 5.7: **Completion results on ImageNet [162] testing set among GL [86], CA [214], PICNet [227], HiFill [208] and Ours.** Our TFill model generated better visual results even under very challenging situations, *e.g.* the heavily masked chicken in the second last row.

scales. Specifically, it achieves relative 27% and 21% improvement for LPIPS at scales of [0.01, 0.1] and (0.5, 0.6], respectively. Furthermore, our completed images form closer distributions to the real testing set, with FID scores averaging 32% relative improvement on all mask scales.

The qualitative comparisons are visualized in Figures 5.6 and 5.7. TFill achieved superior visual results even under challenging conditions. In Figure 5.6, we compare with CA and our previous PICNet trained on CelebA-HQ dataset. Our TFill generates photorealistic high-resolution (512^2) results, even when significant semantic information is missing due to large free-form masks. Figure 5.7 shows visual results on natural images that were degraded by random masks. GL and CA, while good at object removal, failed to infer shapes needed for object completion. The PICNet proposed in the last chapter produced multiple diverse results in which some shapes were correct but of limited quality. Our results are evaluated in higher resolution, with the short side at 512 pixels and the long side at multiples of 2^5, *e.g.* 640. Our TFill model generated better visual results even under very challenging situations, *e.g.* the heavily masked chicken in the second last row.

(a) Original (b) Input (c) Edited Output

FIGURE 5.8: **Free-form editing results on ImageNet [162].**

	Method	CelebA-HQ		FFHQ	
		LPIPS↓	FID↓	LPIPS↓	FID↓
	CA [214]	0.104	9.53	0.127	8.78
	PICNet [227]	0.061	6.43	0.068	4.61
	MEDFE [81]	0.067	7.01	-	-
A	Traditional *Conv*	0.060	6.29	0.066	4.12
B	+ Attention in G	0.059	6.34	0.064	4.01
C	+ Restrictive *Conv*	0.056	4.68	0.060	3.87
D	+ Transformer	0.051	4.02	0.057	3.66
E	+ Masked Attention	0.050	3.92	0.057	3.63
F	+ Refine Network	**0.048**	**3.86**	**0.053**	**3.50**

TABLE 5.2: Learned Perceptual Image Patch Similarity (LPIPS) and Fréchet Inception Distance (FID) for various completion networks on center masked images. Here, we calculate the LPIPS and FID using all images in the corresponding test sets.

5.5.2 Results and Analysis for Token Representation

Results We first demonstrate experimentally that the transformer-based model outperforms previous CNN-based models. Table 5.2 shows Learned Perceptual

(a) Original (b) Edited Output (a) Original (b) Edited Output

FIGURE 5.9: **Qualitative results on CelebA-HQ [95, 126] and FFHQ [96] testing set for free-form mask editing.** All results are reported at 512^2 resolution.

Image Patch Similarity (LPIPS)[1] [222] and Fréchet Inception Distance (FID) [77] for various image completion architectures on CelebA-HQ [95, 126] and FFHQ [96] datasets degraded by center masks. The traditional image quality results are given in Appendix B.1. Here, we compared with three CNN-based models, for which CA [214] and PICNet [227] had the appropriate pretrained models available, while the latest MEDFE [81] was reproduced using their publicly available code. All scores are reported for 256^2 resolution. Without bells and whistles, our TFill-*Coarse* with configuration (E) improved LPIPS (18% relative improvement) and FID (39% relative improvement) quite significantly on CelebA-HQ, despite only using the transformer-based content inference network, without our refinement network.

[1]While multi-modal generation tasks had previously been evaluated with LPIPS [84, 227, 235] in Chapters 3 and 4, it was used to measure diversity. Here, we apply it to measure the similarity between completed images and original ground-truth. A smaller value means the completed image is closer to the ground-truth image w.r.t. the learned perceptual similarity, rather than pixel-level reconstruction. We refer readers to [222] for details.

FIGURE 5.10: **Example completion results of our method (config E) on face datasets.** Here, a center mask was used for all input images. The corresponding quantitative results are reported in Tables 5.2 and 5.3. One center masked example input is shown top-left.

Figure 5.10 shows the visual results of our TFill on CelebA-HQ and FFHQ datasets. Here, all images are center masked in order to demonstrate its ability to go beyond object removal and to generate reasonable semantic content for large missing regions. As can be seen, the completed images are on average of high quality. Even for some challenging cases, such as when eyeglasses are center masked, our TFill can correctly repair the face *with* eyeglasses. Furthermore, it generally works well for varied skin tones, poses, expressions, ages, and illumination.

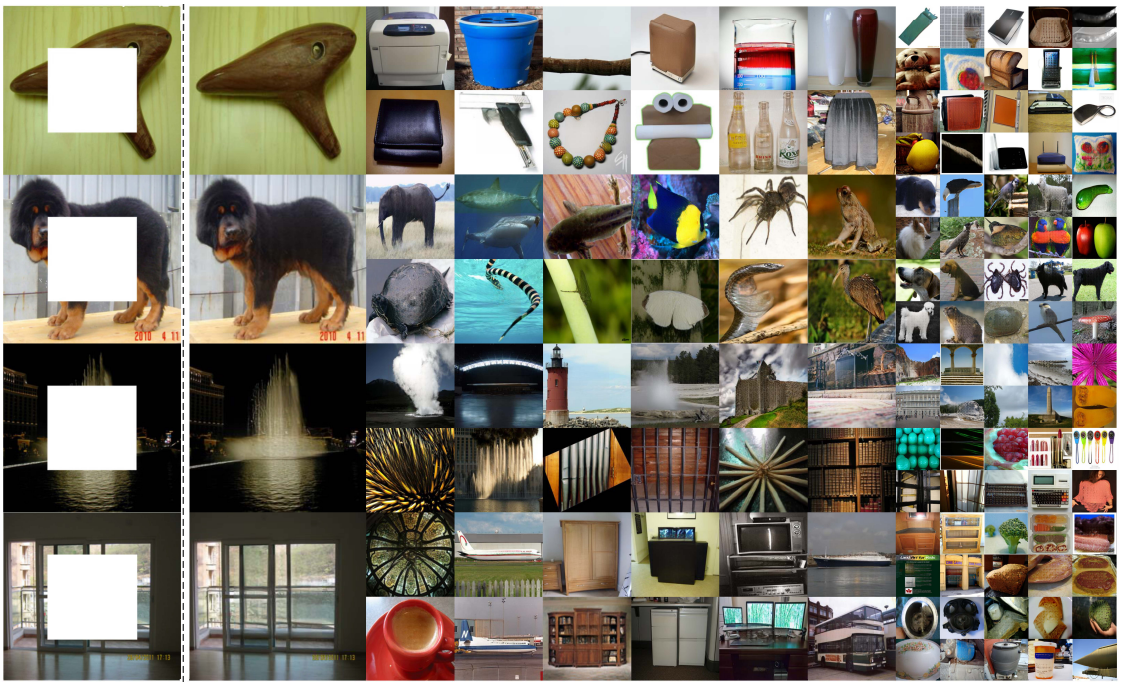


FIGURE 5.11: **Completion results of our method (config E) on ImageNet datasets [162].** All images come from the corresponding testing set that were **degraded by center masks**. Here, we show results for various categories, such as commodity, animal, plant, natural scene, building, food, furniture and so on.

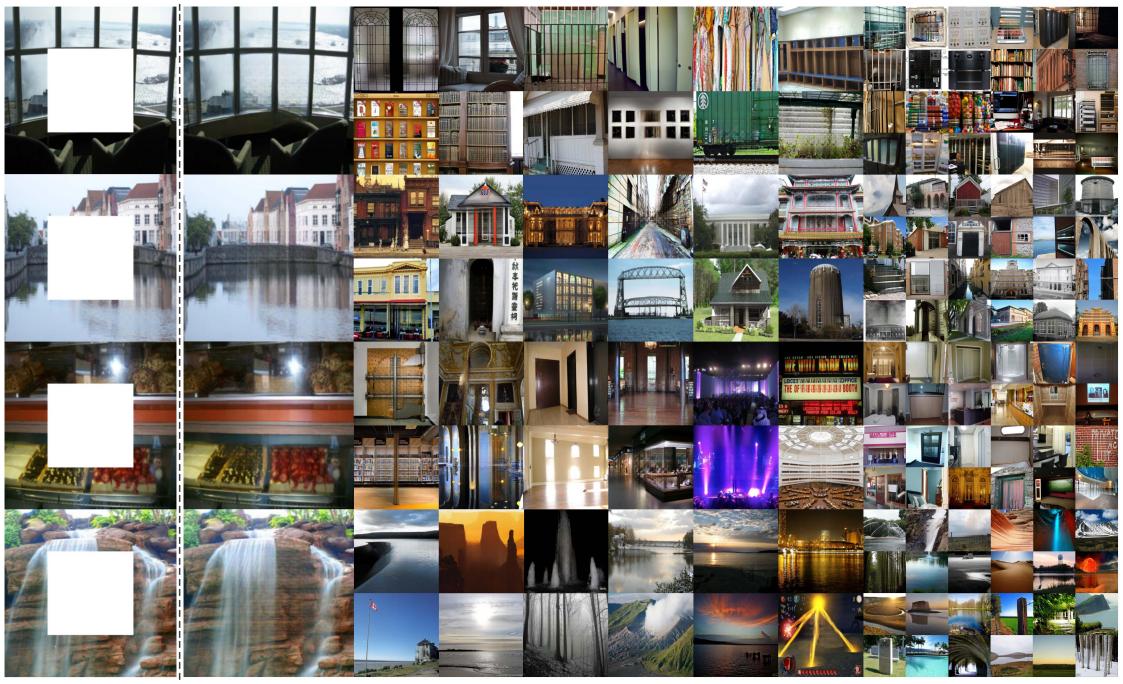


FIGURE 5.12: **Completion results of our method (config E) on Places2 datasets [222].** All images come from the corresponding testing set that were **degraded by center masks**.

	Method	LPIPS↓	FID↓	Mem↓	Time↓
	IGPT [22] (RF 1)	0.609	148.42	3.16	26.45
	VIT [42] (RF 16)	0.062	5.09	1.16	0.167
	VQGAN [48]	0.226	11.92	2.36	4.29
B	*Conv* (RF 229)	0.064	4.01	0.99	0.162
C	Ours *R-Conv* (RF 16)	0.060	3.87	**0.90**	**0.157**
	T-based (RF 229)	0.062	3.92	1.25	0.188
E	T-based (RF 16)	**0.057**	**3.63**	1.15	0.180

TABLE 5.3: **The effect of restrictive token embedding and transformer block** in our transformer-based completion network on FFHQ dataset. "RF" indicates the Receptive Field size. "Mem" denotes the memory (GB) cost during testing and "Time" is the testing time (s) for each center masked image.

In Figure 5.11, we show more examples for object completion, such as the various items and animals on the top half. In Figure 5.12, we display the completed images for various natural scenes. These examples are good evidence that our TFill model is suitable for both *foreground* object completion and *background* scene completion, where it can synthesize semantically consistent content with visually realistic appearance based on the presented visible pixels.

Analysis Our baseline configuration (A) used the same encoder-decoder structure as VQGAN [48], except here attention layers were removed for a pure CNN-based version. When combined with the powerful discriminator of StyleGANv2 [97], the performance was comparable to PICNet [227], in which the best results were selected from 50 diverse samples. We first added the attention layer to the decoder (Generator, G) in (B), but the performance remained similar to baseline (A). In contrast, when we use our proposed *restrictive CNN* in (C), the performance improved substantially, especially for FID. This suggests that the input feature representation is significant for the attention layer to equally deliver all messages, as explained later. We then improved this new baseline by adding the transformer encoder (D), which benefits from globally delivered messages at every layer. Finally, we introduced masked weights to each attention layer of the transformer (E), improving results further.

To study the influence of the token representation, we conducted two experiments that compared with recent visual transformer works [22, 42, 48] and provided an ablation study by controlling the RF in Table 5.3.

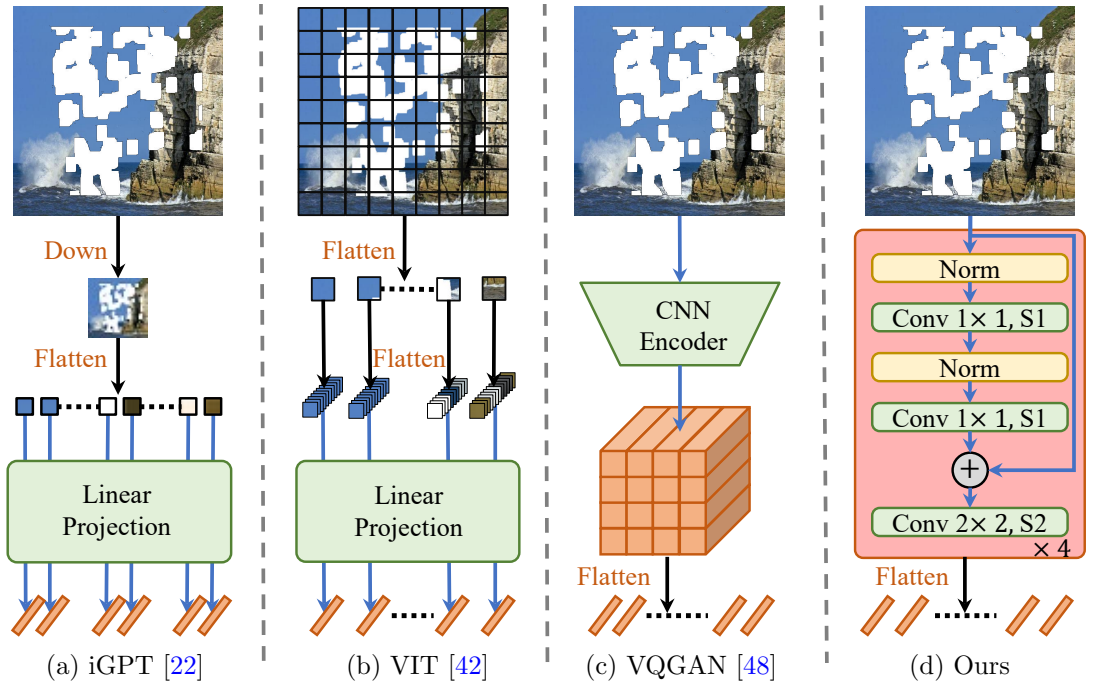


FIGURE 5.13: **Token representation.** (a) Pixel to token. (b) Patch to token. (c) Feature to token. (d) Restrictive **R**eceptive **F**ield (RF) feature to token. Note our token has a small and non-overlapping RF like VIT [42], but uses a complex CNN embedding. Each token represents locally isolated contexts, leaving the long-range relationship to be cleanly modeled in the transformer encoder.

As illustrated in Figure 5.13, iGPT [22] downsamples the image to a fixed scale, *e.g.* 32^2 resolution, and embeds *each pixel to a token*. While this may not impact the original classification task, which is robust to low resolutions [184], it has a large negative effect on generating high-quality images. Furthermore, the auto-regressive form resulted in the completed image being inconsistent with the bottom-right visible region (iGPT in Figure 5.14), and each image runs an average of 26.45s during the testing. This is because the conditional sequence generation can only utilize the top-left visible pixels, generating new pixels one-by-one. In contrast, VIT [42] divides an image to a set of fixed patches and embeds *each patch to a token*. As shown in Table 5.3 and Figure 5.14, it can achieve relatively good quantitative and qualitative results. However, some details are perceptually poor, *e.g.* the strange eyes in Figure 5.14, possibly due to the limited one-layer linear projection. Finally, VQGAN [48] employs a traditional CNN to encode an image to the feature domains and then *quantizes each feature as a token* through a learned codebook [157, 187]. Figure 5.14 shows the generated images using tokens embedded from ground truth (VQ Rec), and tokens extracted from the center

FIGURE 5.14: **Comparing results under different token representations.** All transformers are based on the same transformer backbone [188]. For VQGAN [48], we report reconstruction (Rec) image, completed (Comp) image and recomposed output image. TFill-*Coarse* is our model with configuration E in Tables 1 and 2, *i.e.* TFill without the refinement network. Please see main text for details.

masked image (VQ Comp). While it generates the content of missing regions sequentially conditioned only on top-left visible tokens, we found the completed pixels to be consistent with the bottom-right region, even though these tokens were *not* used to infer missing content in the transformer encoder. We believe this is due to the large RF in the CNN-based encoder causing each token to capture extended dependencies in a deep layer. However, this leads to two issues: **1)** even the original visible tokens are modified, resulting in different appearances for the visible regions *e.g.* see VQ Rec *vs* VQ Comp in Figure 5.14; **2)** inferred tokens are unduly influenced by implicit CNN-based correlation to nearby tokens, and cannot establish ties cleanly to important but distant tokens. Thus it generates a visually realistic completion, but when pasted to the original masked input (VQ Output in Figure 5.14), there is an obvious gap between generated and visible pixels.

In contrast to [22, 42, 48], our token representation is extracted using a *restrictive CNN* (Figure 5.13(d)). In particular, the 1×1 filter and `layernorm` is applied for non-linear projection, followed by a partial convolution layer [123] that uses a 2×2 filter with stride 2 to extract visible information and reduce feature resolution simultaneously. For instance, if half of the pixels in a window are masked, we only embed the other 50% comprising visible pixels as our token representation, and establish an initial weight of 0.5 for the *masked* self-attention layer. To do this, we ensure each token represents only the visible information in a small RF, *leaving the long-range dependencies to be explicitly modeled by the transformer encoder in every layer*, without cross-contamination from implicit correlation due to larger CNN RF. To demonstrate the impact of RF, a thorough ablation study result is reported in Table 5.3, in which we find the small RF CNN improves both LPIPS and FID significantly, with the added benefit of low memory cost. Furthermore, our model runs at 180ms per image on an Nvidia GTX 1080Ti (+21ms CPU time

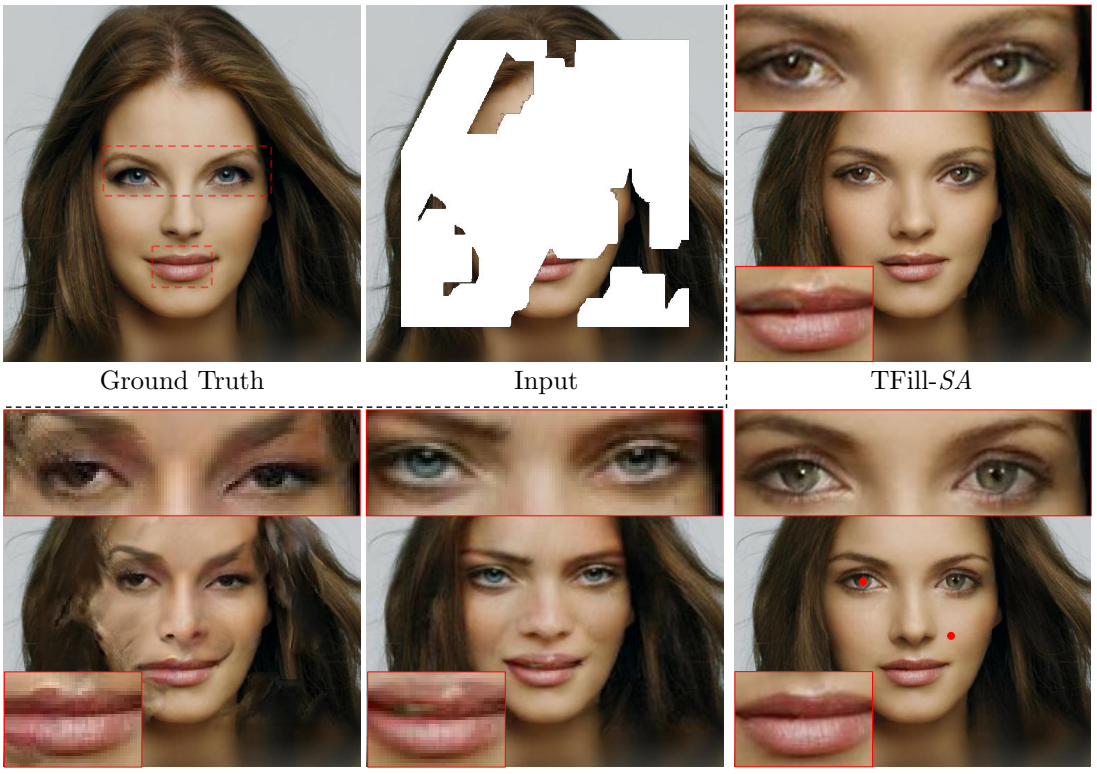


FIGURE 5.15: **Results with different attention modules** in various methods. Our attention-ware layer is able to adaptively select the features from both visible and generated content. In this example, the ratio for the two query points is $\mathbf{w}_v/\mathbf{w}_m = 0.77/0.23$ (skin) and $\mathbf{w}_v/\mathbf{w}_m = 0.08/0.92$ (eye), respectively.

for resizing input and storing output), due to predicting all output heads together, rather than auto-regressively as in existing work [22, 48].

5.5.3 Results and Analysis for AAL

Mask Type	Metric	SA [220]	CA [214]	SLTA [227]	Ours-AAL
center	LPIPS	0.058	0.061	0.056	**0.053**
	FID	3.62	3.86	3.61	**3.50**
random	LPIPS	0.047	0.044	0.045	**0.041**
	FID	2.69	2.66	2.64	**2.57**

TABLE 5.4: The effect of various attention layers on FFHQ dataset. "center" denotes the center mask, "random" denotes the random regular mask and "SA" is the basic self-attention layer. These attention layers were implemented within our TFill refinement framework.

We ran ablations to analyze our proposed AAL by replacing it with existing contextual attention models of SA [188, 220], CA [214] and SLTA [227]. As shown

in Table 5.4, SA showed similar performance to the coarse results in Table 5.2, due to the insular attention problem mentioned earlier. CA [214] performed worse on large center masks than random regular masks (even worse than the coarse results of (E) in Table 5.2), as it borrows context from visible regions only. When important context is not visible, *e.g.* when both eyes are missing in Figure 5.15, it is unable to find the right context to copy. While our previous PICNet [227] focuses on both visible and invisible regions, selection was done by *fixed* weights learned during training. This is also inferior, and in some cases we observed that it can have difficulty in selecting the best features for generation, especially on free-form masks. In contrast, our AAL selects features based on the largest attention scores, using weights *dynamically mapped* during inference. For instance, in Figure 5.3, only the left eye was masked, and it had a large attention score to the visible right eye, resulting in a ratio of $\mathbf{w}_v/\mathbf{w}_m = 0.91/0.09$. Conversely, when two eyes were masked in Figure 5.15, the attention score between the two eyes was still high, but the ratio was correctly flipped to $\mathbf{w}_v/\mathbf{w}_m = 0.08/0.92$ for the left eye.

5.5.4 Additional Results

Following Chapter 4, I also show interesting applications of the proposed TFill model for free-form image editing on various higher resolution datasets.

As shown in Figure 5.8, I edit the natural scene with object removal being the main task, as it is the main use case for image inpainting. Here, I enforce the input image size to be multiples of 32, *e.g.* 512×384 and provide the high-resolution results on the corresponding image size. As we can see, our TFill-*Refined* model is able to handle high-resolution images for object removal in traditional image inpainting task. More results can be found in our online project[2].

In Figure 5.9, more examples are shown for face editing at 512×512 resolution. For conventional object removal, *e.g.* watermark removal, the proposed TFill addresses them easily. Furthermore, the TFill can handle more extensive face editing, such as removing substantial facial hair and changing mouth expressions.

5.6 Limitations and Discussion

Through the detailed analyses and experiments, I demonstrate that the transformer based architecture has exciting potential for image completion, due to its

[2]More results are available on http://www.chuanxiaz.com/publication/tfill/

capacity for effectively modeling connections between distant image content. Unlike recent vision transformer models that either use shallow projections or large receptive fields for token representation, our *restrictive CNN projection* provides the necessary separation between explicit attention modeling and implicit RF correlation that leads to substantial improvement in results. I also introduced a novel attention-aware layer that adaptively balances the attention for visible and masked regions, further improving the completed image quality.

While this *TFill* model generates reasonable content as well as realistic appearance, it provides only one "optimal" result for this highly subjective task. It will be much more interesting if we can explore the transformer-based architecture for multiple and diverse results. The more recent work [190] has made an initial step towards this goal. We would like to explore more in future work.

Part III

Modeling Shape and Appearance: Completed Scene Decomposition

Chapter 6

Visiting the Invisible

The methods in Part II can produce plausible results given a masked image by filling into reasonable content as well as visually realistic appearance. However, these systems depend on manual masks as input, rather than automatically understanding the full scene. In this chapter, we present a higher-level structural scene decomposition and completion system, which has the ability to *decompose* a scene into individual objects, *infer* their underlying occlusion relationships and moreover *imagine* what occluded objects may look like, while *using only an image as input*. In order to disentangle the occluded relationships of all objects in a complex scene, we use the fact that the front object, being free from occlusion, is easy to be identified, detected, and segmented. Our system interleaves the two tasks of instance segmentation and scene completion through multiple iterations, solving for objects layer-by-layer. We first provide a thorough experiment using a new realistically rendered dataset, where ground-truth is available for all invisible regions. To bridge the domain gap to real imagery, where ground-truth is not available, we then train another model with pseudo-ground-truths generated from our previously trained synthesis model. We demonstrate results on a wide variety of datasets and show significant improvement over the state-of-the-art.

The rest of this chapter is organized as follows. We first describe our motivation in Section 6.1. Then, we discuss the related work in Section 6.2, and describe our rendered dataset in Section 6.3. In Section 6.4 we present our layer-by-layer CSDNet method. We then show the experiment results on this synthetic dataset as well as the results on real-world images in Section 6.5, followed by a conclusion in Section 6.6.

6.1 Motivation

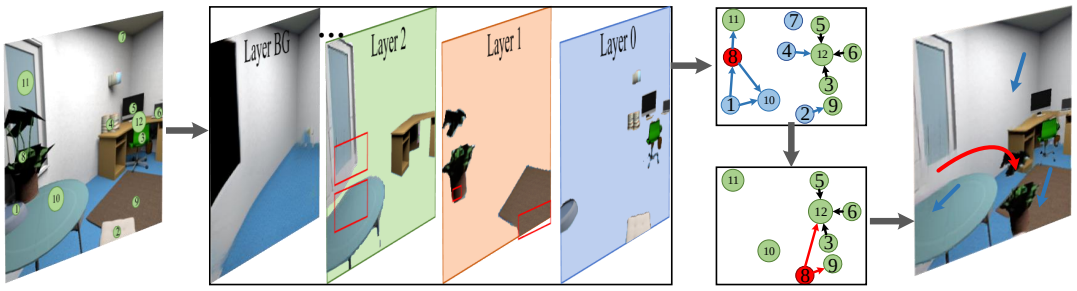
The vision community has made rapid advances in scene understanding tasks, such as object classification and localization [61, 73, 159], scene parsing [5, 21, 127], instance segmentation [20, 72, 148], and layered scene decomposition [66, 205, 223]. Despite their impressive performance, these systems deal only with *visible* parts of scenes without trying to exploit *invisible* regions, which results in an uncompleted representation of real objects.

In parallel, significantly progress for the generation task has been made with the emergence of deep generative networks, such as GAN-based models [65, 67, 96], VAE-based models [103, 186, 187], and flow-based models [39, 40, 102]. Empowered by these techniques, image completion [86, 214, 227] and object completion [44, 120, 219] have made it possible to create the plausible appearances for occluded objects and backgrounds, as shown in above chapters. However, these systems depend on manual masks or visible ground-truth masks as input, rather than automatically understand the full scene.

In this chapter, we will present a system that has the ability to *decompose* a scene into individual objects, *infer* their underlying occlusion relationships, and moreover *imagine* what occluded objects may look like, *while using only an image as input*. This novel task involves the classical recognition task of instance segmentation to predict the geometry and category of all objects in a scene, and the generation task of image completion to reconstruct invisible parts of objects and backgrounds. After a full decomposition of the given scene, users can freely edit the instances in the original 2D image, such as deleting object, moving object's positions, and further changing their occlusion relationships 6.12.

To decompose a scene into instances with completed appearances in one pass is extremely challenging. This is because realistic natural scenes often consist of a vast collection of physical objects, with complex scene structure and occlusion relationships, especially when one object is occluded by multiple objects, or when instances have deep hierarchical occlusion relationships.

Our core idea is from the observation that *it is much easier to identify, detect and segment foreground objects than occluded objects*. Motivated by this, we propose a **C**ompleted **S**cene **D**ecomposition **Net**work (**CSDNet**) that learns to segment and complete each object in a scene layer-by-layer consecutively. As shown in Figure 6.1, our layered scene decomposition network only segments the fully visible objects out in each layer (Figure 6.1(b)). If the system is able to properly



(a) Input (b) Layered Completed Scene Decomposition (c) Pairwise Order (d) Recomposed Image

FIGURE 6.1: **Example results of scene decomposition and recomposition.** (a) Input. (b) Our model structurally decomposes a scene into individual completed objects. Red rectangles highlight the original *invisible* parts. (c) The inferred pairwise order (top graph) and edited order (bottom graph) of the instances. Blue nodes indicate the deleted objects while the red node is the moved object. (d) The new recomposed scene.

segment the foreground objects, it will automatically learn which parts of occluded objects are actually invisible that need to be filled in. The completed image is then passed back to the layered scene decomposition network, which can again focus purely on detecting and segmenting visible objects. As the interleaving proceeds, a structured instance depth order (Figure 6.1(c)) is progressively derived by using the inferred absolute layer order. The thorough decomposition of a scene along with spatial relationships allows the system to freely recompose a new scene (Figure 6.1(d)).

Another challenge in this novel task is the lack of data: there is no complex, realistic dataset that provides intact ground-truth appearance for originally occluded objects and backgrounds in a scene. While latest works [114, 219] introduced a self-supervised way to tackle the amodal completion using only visible annotations, they can not do a fair quantitative comparison as no real ground-truths are available. To mitigate this issue, we constructed a high-quality rendered dataset, named **C**ompleted **S**cene **D**ecomposition (**CSD**), based on more than 2k indoor rooms. Unlike the datasets in [38, 44], our dataset is designed to have more typical camera viewpoints, with near-realistic appearance.

As elaborated in Section 6.5.2, the proposed system performs well on this rendered dataset, both qualitatively and quantitatively outperforming existing methods in completed scene decomposition, in terms of instance segmentation, depth ordering, and amodal mask and content completion. To further demonstrate the generalization of our system, we extend it to real datasets. As there are no ground truth annotations and appearance available for training, we created pseudo-ground-truths for real images using our model that is purely trained on **CSD**, and then

fine-tuned this model accordingly. This model outperforms state-of-the-art methods [151, 219, 237] on amodal instance segmentation and depth ordering tasks, despite these methods being specialized to their respective tasks rather than our holistic completed scene decomposition task. While we are unable to quantitatively evaluate real-image scene completion without ground truth appearance for occluded objects, our method is able to create visually reasonable layer-by-layer decomposition results, and we further demonstrate its effectiveness in real scene recomposition.

In summary, we propose a layer-by-layer scene decomposition network that jointly learns structural scene decomposition and completion, rather than treating them separately as the existing works [38, 44, 219]. To our knowledge, it is the first work that proposes to complete objects based on the global context, instead of tackling each object independently. To address this novel task, we render a high-quality rendered dataset with ground-truth for all instances. We then provide a thorough ablation study using this rendered dataset, in which we demonstrate that the method substantially outperforms existing methods that address the task in isolation. On real images, we improve the performance to the recent state-of-the-art methods by using pseudo-ground-truth as weakly-supervised labels. The experimental results show that our **CSDNet** is able to acquire a full decomposition of a scene, *with only an image as input*, which conduces to a lot of applications, *e.g.* object-level image editing.

6.2 Background

A variety of scene understanding tasks have previously been proposed, including layered scene decomposition [206], instance segmentation [72], amodal segmentation [114], and scene parsing [21]. In order to clarify the relationships of our work to the relevant literature, Table 6.1 gives a comparison based on three aspects: what the goals are, which information is used, and on which dataset is evaluated.

6.2.1 Inmodal Perception

The layered scene decomposition for visible regions has been extensively studied in the literature. Shade *et al.* [167] first proposed a representation called a layered depth image (LDI), which contains multiple layers for a complex scene. Based on this image representation that requires occlusion reasoning, the early works

Paper	Outputs	Inputs	Data
	SP, O	I	LabelMe, PASVOC, others
Yang *et al.* [206]	In, O	I	PASVOC
Tighe *et al.* [182]	SP, O	I	LabelMe, SUN
Zhang *et al.* [223]	In, O	I	KITTI
Guo *et al.* [68]	AS	I	StreetScenes, SUN, others
Kar *et al.* [94]	AB	I	PASVOC, PAS3D
Liu *et al.* [121]	AS, O	I, D	NTUv2-D
Li *et al.* [114]	A	I, In	PASVOC
Zhu *et al.* [237]	A, O	I	COCOA (from COCO)
Follmann *et al.* [50]	A	I	COCOA, COCOA-cls, D2S
Qi *et al.* [151]	A	I	KINS (from KITTI)
Hu *et al.* [82]	A	I	Synthesis video
Ehsani *et al.* [44]	A, O, IRGB	I, In	DYCE, PAS3D
Zhan *et al.* [219]	A, O, IRGB	I, In	KINS, COCOA
Ling *et al.* [120]	A, IRGB	I, In	KINS
Yan *et al.* [201]	A, IRGB	I	Vehicle
Burgess *et al.* [16]	In, IRGB	I	Toy
Dhamo *et al.* [38]	A, D, IRGB	I	SUNCG, Stanford 2D-3D
Ours	A, O, IRGB	I	KINS, COCOA, SUNCG

TABLE 6.1: **Comparison with related work based on three aspects: outputs, inputs and data**. I: image, In: inmodal segmentation, O: occlusion order, SP: scene parsing, AB: amodal bounding box, AS: amodal surface, A: amodal segmentation, D: depth, IRGB: intact RGB object.

focused on ordering the semantic map as occluded and visible regions. Winn and Shotton [197] proposed a LayoutCRF to model several occlusions for segmenting partially occluded objects. Gould *et al.* [66] decomposed a scene into semantic regions together with their spatial relationships. Sun *et al.* [181] utilized an MRF to model the layered image motion with occlusion ordering. Yang *et al.* [205, 206] formulated a layered object detection and segmentation model, in which occlusion ordering for all detected objects was derived. This inferred order for all objects has been used to improve scene parsing [182] through a CRF. Zhang *et al.* [223] combined CNN and MRF to predict instance segmentation with depth ordering. While these methods evaluate occlusion ordering, their main goal is to improve the inmodal perception accuracy for object detection, image parsing, or instance segmentation using the spatial occlusion information. In contrast to these methods, our method *not* only focuses on visible regions with structural inmodal perception, but also tries to solve for amodal perception. *i.e.* to learn *what is behind the occlusion.*

6.2.2 Amodal Image/Instance Perception

Some initial steps have been taken toward amodal perception, exploring the invisible regions. Guo and Hoiem [68] investigated background segmentation map completion by learning relationships between occluders and background. Subsequently, [121] introduced the Occlusion-CRF to handle occlusions and complete occluded surfaces. Kar *et al.* [94] focused on amodal bounding box completion, where the goal is to predict the intact extent of the bounding box. The common attribute in these earlier amodal perception works is using piecemeal representations of a scene, rather than a full decomposition that infers the amodal shapes for all objects.

The success of advanced deep networks trained on large-scale annotated datasets has recently led to the ability to get more comprehensive representations of a scene. Instance segmentation [34, 116, 148, 149] deal with detecting, localizing and segmenting all objects of a scene into individual instances. This task combines the classical object detection [60, 61, 73, 159] and semantic segmentation [5, 21, 127]. However, these notable methods typically segment the scene only into visible regions, and do *not* have an explicit structural representation of a scene. We believe a key reason is the lack of large-scale datasets with corresponding annotations for amodal perception and occlusion ordering. The widely used datasets, such as Pascal VOC 2012 [49], NYU Depth v2 [173], COCO [119], KITTI [57], and CityScapes [31], contain only annotations for the visible instances, purely aiming for 2D inmodal perception.

To mitigate the lack of annotated datasets, Li *et al.* [114] presented a self-supervised approach by pasting occluders into an image. Although reasonable amodal segmentation results are shown, a quantitative comparison is unavailable due to the lack of ground-truth annotations for invisible parts. In more recent works, the completed masks for occluded parts are provided in COCOA [237] and KINS [151], which are respectively a subset of COCO [119] and KITTI [57]. However, their annotations for invisible parts are manually labeled, which is highly subjective [44, 219]. Furthermore, these datasets are mainly used for the task of inferring amodal semantic maps and are not suitable for the task of RGB appearance completion, since the ground truth RGB appearance for occluded parts are not available. In contrast, we jointly address these two amodal perception tasks using our constructed CSD dataset.

6.2.3 Amodal Perception for both Mask and Appearance

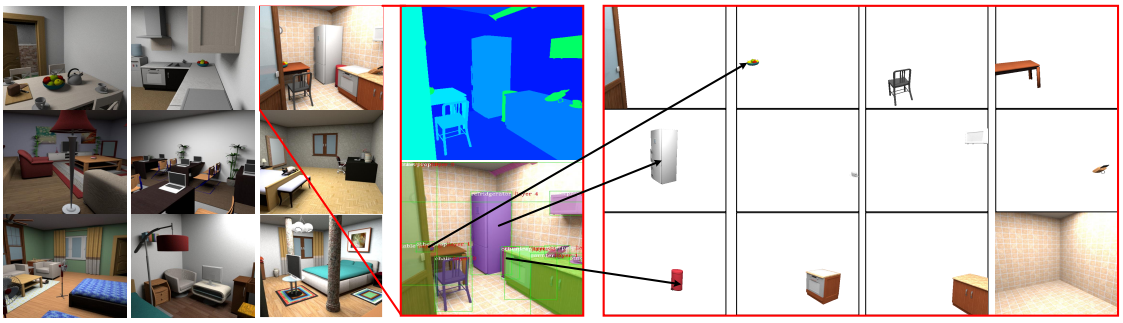
The problem of generating the amodal RGB appearance for the occluded parts is highly related to semantic image completion. The latest methods [86, 139, 146, 204, 214, 227] extend GANs [65] and CGANs [135] to address this task, generating new imagery to fill in partially erased image regions. However, they mainly focus on object removal, needing users to interactively annotate the objects to be removed.

SeGAN [44] involved an end-to-end network that sequentially infers amodal masks and generates complete RGB appearances for instances. The instance depth order is estimated by comparing the areas of the full and visible masks. PCNet [219] used a self-supervised learning approach to recover masks and content using only visible annotations. However, these works mainly present results in which the ground truth visible mask is used as input, and are sensitive to errors in this visible mask. As stated in [219], their focus is on amodal completion, rather than a scene understanding for amodal perception. While the recent work of Yan *et al.* [201] tried to visualize the invisible from a single input image, it only tackles the occluded "vehicle" category, for which there is much less variation in amodal shape and RGB appearance, and thus easier to model.

There are two recent works that attempt to learn structural scene decomposition with amodal perception. MONet [16] combined an attention network and a CVAE [103] for jointly modeling objects in a scene. While it is nominally able to do object appearance completion, this unsupervised method has only been shown to work on simple toy examples with minimal occlusions. Dhamo *et al.* [38] utilized Mask-RCNN [72] to obtain visible masks, and conducted RGBA-D completion for each object. However, depth values are hard to accurately estimate from a single image, especially in real images without paired depth ground-truths. Besides, they still considered the decomposition and completion separately. In practice, even if we use domain transfer learning for depth estimation, the pixel-level depth value for all objects are unlikely to be consistent in a real scene. Therefore, our method uses an instance-level occlusion order, called the "2.1D" model [206], to represent the structural information of a scene, which is easier to be inferred and manipulated.

6.3 Data Collection

Large datasets with complete ground-truth appearances for all objects are very limited. Burgess *et al.* [16] created the *Objects Room dataset*, but only with toy



(a) Rendered Images (b) Annotations (c) Full RGBA Ground-Truth for Instance and Background

FIGURE 6.2: **Our rendered dataset.** (a) High quality rendered RGB images. (b) Semantic map and instance annotations with bbox, category, ordering and segmentation map. (c) Intact RGBA ground-truth for instances and background.

objects. Ehsani *et al.* [44] and Dhamo *et al.* [38] rendered synthetic datasets. However, the former only includes 11 rooms, with most viewpoints being atypical of real indoor images. The latter's OpenGL-rendered dataset appears to have more typical viewpoints with rich annotations, but the OpenGL-rendered images have low realism. Recently, Zhan *et al.* [219] explored the *amodal completion* task through self-supervised learning without the need of amodal ground-truth. However, a fair quantitative comparison is not possible as no appearance ground-truth is available for invisible parts.

To mitigate this issue, we rendered a realistic dataset with Maya [3], instead of the OpenGL-rendering used in Chapter 2. We can train the supervised model and test the unsupervised model on this synthetic data with masks and RGB appearance ground-truths for all occluded parts.

Data Rendering Our rendered data is based on a total of 10.2k views inside over 2k rooms (CAD models from SUNCG [177]) with various room types and lighting environments (see Figure 6.2(a)). To select the typical viewpoints, we first sampled many random positions, orientations and heights for the camera. Only when a view contains at least 5 objects will we render the image and the corresponding ground-truth of each instance. To avoid an excessive rendering workload, we separately rendered each isolated object, as well as the empty room, as shown in Figure 6.2(c). This allows us to then freely create the ground-truths of each layer by compositing these individual objects and background using the instance occlusion order. The rendering details and statistics of the dataset can be found in the Appendix C.2.

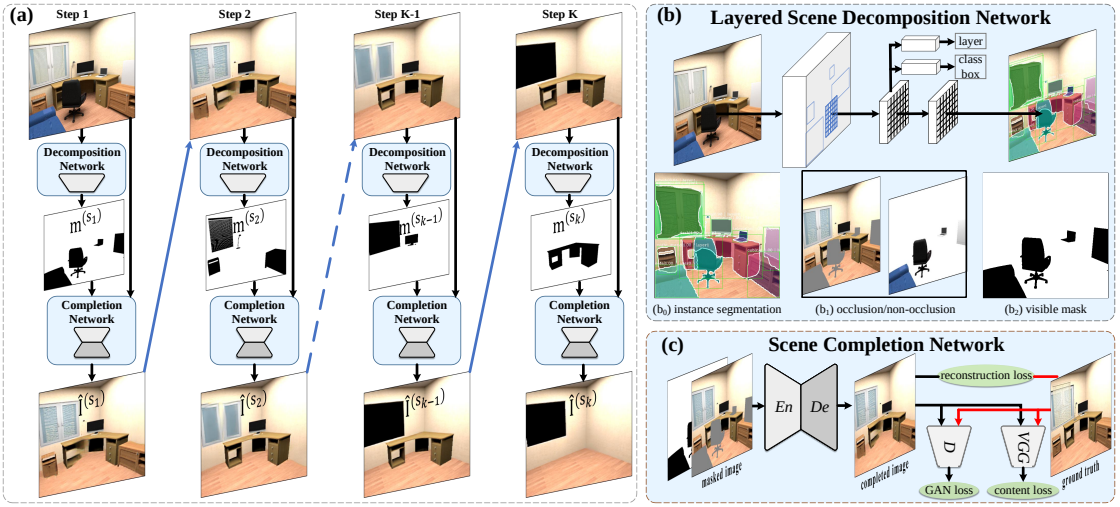


FIGURE 6.3: **An illustration of the CSDNet framework.** (a) Overall layer-by-layer completed scene decomposition pipeline. In each step, the layered decomposition network selects out the fully visible objects. The completion network will complete the resultant holes with appropriate imagery. The next step starts again with the completed image. (b) The layered scene decomposition network estimates instance masks and binary occlusion relationships. (c) The completion network generates realistic content for the original *invisible* regions.

Data Annotation Each rendered scene is accompanied by a global semantic map and dense annotations for all objects. As shown in Figure 6.2(b) and Figure 6.2(c), the intact RGB appearances are given, as well as categories (the 40 classes in NYUDv2 [138]), bounding boxes and masks for complete objects, as well as for only the visible regions. Furthermore, the absolute layer order and pairwise occlusion order shown in Figure 6.4 are also defined in our rendered dataset. The detail examples are presented in Appendix C.2.2.

6.4 Approach

In this work, we aim to derive a higher-level structural decomposition of a scene. When given a single RGB image **I**, our goal is to decompose all objects in it and infer their fully completed RGB appearances, together with their underlying occlusion relationships (As depicted in Figure 6.1). Our system is designed to carry out inmodal perception for *visible* structured instance segmentation, and also solve the amodal perception task of completing shapes and appearances for originally *invisible* parts.

Instead of directly predicting the invisible content and decoupling the occlusion relationships of all objects at one pass, we use the fact that foreground objects are

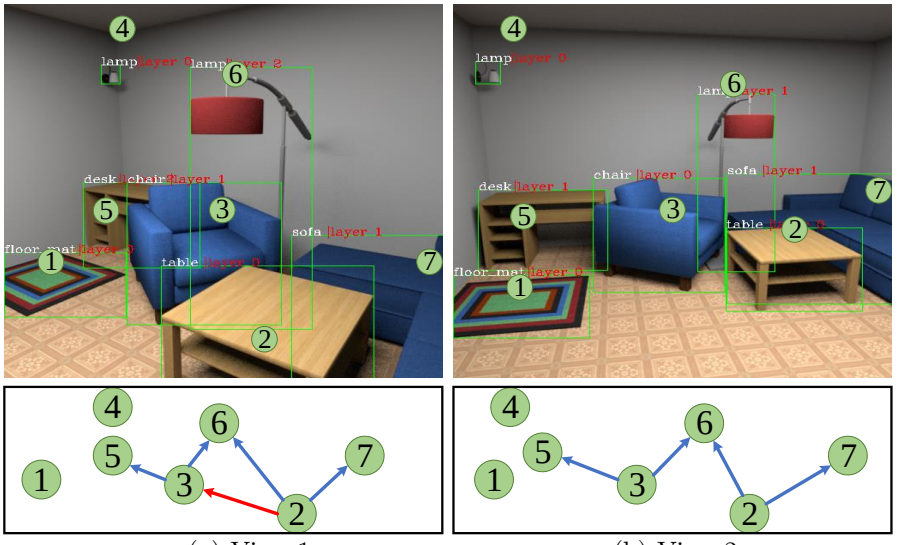
more easily identified, detected and segmented without occlusion. Our CSDNet decomposes the scene layer-by-layer. As shown in Figure 6.3(a), in each step s_{k-1}, given an image $\mathbf{I}^{(s_{k-1})}$, the layered segmentation network creates masks as well as occlusion labels for all detected objects. Those instances classified as fully visible are extracted out, and the scene completion network generates appropriate appearances for the invisible regions. The completed image $\hat{\mathbf{I}}^{(s_{k-1})}$ will then be resubmitted for layered instance segmentation in the next step s_k. This differs significantly from previous works [16, 38, 44, 120, 219], which do not adapt the segmentation process based on completion results.

Our *key novel insight* is that scene completion generates completed shapes for originally occluded objects by leveraging the global scene context, so that they are subsequently easier to be detected and segmented without occlusion. Conversely, better segmented masks are the cornerstones to complete individual objects by precisely predicting which regions are occluded. Furthermore, this interleaved process enables extensive *information sharing between these two networks*, to holistically solve for multiple objects, and produces a structured representation for a scene. This contrasts with existing one-pass methods [16, 38, 44, 120, 219], where the segmentation and completion are processed separately and instances are handled independently. Together with the benefit of occlusion reasoning, our system is able to explicitly learn *which parts of the objects and background are occluded that need to be completed*, instead of freely extending to arbitrary shapes.

6.4.1 Layered Scene Decomposition

As shown in Figure 6.3, our layered scene decomposition network comprehensively detect objects in a scene. For each candidate instance, it outputs a class label, a bounding-box offset and an instance mask. The system is an extension of Mask-RCNN [72], which consists of two main stages. In the first stage, the image is passed to a *backbone network* (*e.g.* ResNet-50-FPN [118]) and next to a *region proposal network* (RPN [159]) to get object proposals. In the second stage, the network extracts features using *RoIAlign* from each candidate box, for passing to object classification and mask prediction. We refer readers to [20, 72] for details.

To determine if an object is fully visible *or* partially occluded, a new parallel branch for this binary occlusion classification is added, as shown in Figure 6.3(b). This decomposition is done consecutively layer-by-layer, where at each step it is applied to a single RGB derived from the counterpart scene completion step. While



(a) View 1 (b) View 2

FIGURE 6.4: **Instance depth order representation.** Top images show absolute layer order [151] in different views. Bottom directed graphs give the pairwise order between objects.

only binary decisions are made in each step, after a full set of iterations, a comprehensive layered occlusion ordering is obtained. The following parts describe how this is done, looking first at the instance depth order representation, followed by how the occlusion head is designed.

Instance depth order representation Absolute layer order and pairwise occlusion order are two standard representations for occlusion reasoning in a scene [181, 182]. As shown in Figure 6.4, the definition for our *absolute layer order* $\mathcal{L}$ follows [151], where fully visible objects are labeled as 0, while other occluded objects have 1 order higher than the highest-order instance occluding them (see top images in Figure 6.4). We interpret the *pairwise occlusion order matrix* as a directed graph $G = (\Omega, W)$ (see bottom graphs in Figure 6.4), where Ω is a discrete set of all instances with number N, and W is a $N \times N$ matrix. $W_{i,j}$ is the occlusion relationship of instance i to instance j. We use three numbers to encode the order — $\{-1$: occluded, 0 : no relationship, 1: front$\}$. For example, the chair (instance #3) is occluded by the table (instance #2), so the pairwise order for the chair is $W_{3,2} = -1$, while the pairwise order for the table is inversely labeled as $W_{2,3} = 1$.

Occlusion head In practice, we find it hard to directly predict these instance depth orders. The absolute layer order index $l \in \mathcal{L}$ cannot be predicted purely from local features in a bounding box, since it depends on the global layout of all

objects in a scene. Furthermore, this index is very sensitive to viewpoints, *e.g.* in Figure 6.4, the desk (instance #5) is occluded by only one object (instance #3) in both views, but the absolute layer order indices of the desk are different: "2" *vs* "1". In contrast, pairwise ordering G captures the occlusion relationships between pairs of objects, but all pairs have to be analyzed, leading to scalability issues in the current instance segmentation network. As R-CNN-based system creates 2,000 candidate objects, this pairwise analysis requires building an unwieldy 2k×2k features. Even if we were to restrict these to the 100 highest scoring detection boxes, it will still be very memory intensive.

We circumvent these problems as our occlusion classifier only predicts a *binary occlusion label*: $\{0, 1\}$ in each step, where 0 is fully visible, and 1 is occluded, following the setting of absolute layer order. During training, each ground-truth binary occlusion label is determined from the pairwise order of the actual objects present in the scene (see details in the Appendix C.1). The occlusion head in our layered scene decomposition network is a *fc* layer, which receives aligned features from each RoI and predicts the binary occlusion label.

Decomposition Loss The multi-task loss function for layered scene decomposition is defined as follows:

$$L_{\text{decomp}} = \sum_{t=1}^{T} \alpha_t (L^t_{\text{cls}} + L^t_{\text{bbox}} + L^t_{\text{mask}} + L^t_{\text{occ}}) + \beta L_{\text{seg}} \tag{6.1}$$

where classification loss L^t_{cls}, bounding-box loss L^t_{bbox}, mask loss L^t_{mask} and semantic segmentation loss L_{seg} are identical to those defined in HTC [20], and L^t_{occ} is the occlusion loss at the cascade refined stage t (three cascade refined blocks in HTC [20]), using binary cross-entropy loss [127] for each RoI.

6.4.2 Visiting the Invisible by Exploring Global Context

In our solution, we treat visiting the invisible as a *semantic image completion* [146] problem. As illustrated in Figure 6.3, in step s_{k-1}, after removing the front visible instances, the given image $\mathbf{I}^{(s_{k-1})}$ is degraded to become $\mathbf{I}_m^{(s_{k-1})}$. Our goal is to generate appropriate content to complete these previously *invisible* regions (being occluded) for the next layer $\mathbf{I}^{(s_k)}$. Unlike existing methods that complete each object independently [16, 38, 44, 120], our model completes multiple objects in

each step layer-by-layer, such that the information from earlier scene completions propagate to later ones. The global scene context is utilized in each step.

To visit the invisible, it is critical to know which parts are invisible that need to be completed. The general image completion methods use manually interactive masks as input, which differs from our goal. Recent related works [44, 120, 219] depend on the ground-truth visible masks as input to indicate which parts are occluded. In contrast, our system selects out fully visible objects and automatically learns which parts are occluded in each step. The holes left behind explicitly define the occluded regions for remaining objects, and thus the completed shapes for remaining objects must be deliberately *restricted to these regions*, instead of being allowed to grow freely using only the predicted visible masks.

We use our previous PICNet [227] framework to train our completion network. While our original PICNet was designed for diversity, here we only want to obtain the best result closest to the ground-truth. Therefore, we only use the encoder-decoder structure, and eschew the random sampling aspect.

Completion Loss The overall scene completion loss function is given by

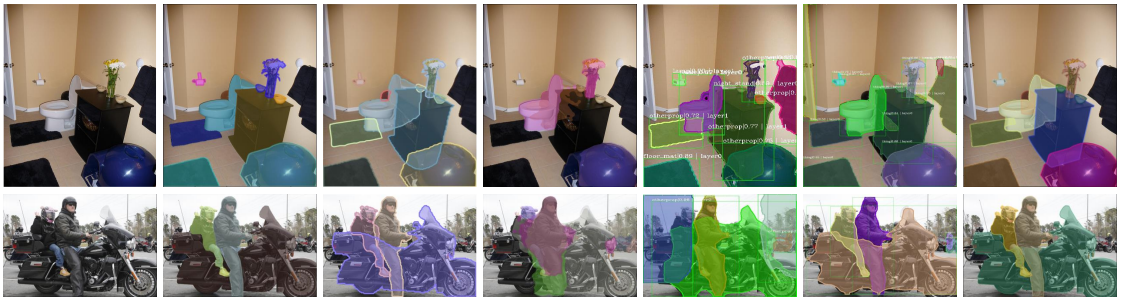
$$L_{\text{comp}} = \alpha_{\text{rec}} L_{\text{rec}} + \alpha_{\text{ad}} L_{\text{ad}} + \alpha_{\text{per}} L_{\text{per}} \tag{6.2}$$

where reconstruction loss L_{rec} and adversarial loss L_{ad} are identical to those in PICNet [227] proposed in Chapter 4. The perceptual loss $L_{\text{per}} = |\mathbf{F}^{(l)}(\hat{\mathbf{I}}^{(s_k)}) - \mathbf{F}^{(l)}(\mathbf{I}^{(s_k)})|$ [93], based on a pretrained VGG-19 [174], is the l_1 distance of features $\mathbf{F}$ in l-th layer between the generated image $\hat{\mathbf{I}}^{(s_k)}$ and ground-truth $\mathbf{I}^{(s_k)}$.

6.4.3 Inferring Instance Pairwise Occlusion Order

As discussed in Section 6.4.1, absolute layer order $\mathcal{L}$ is sensitive to errors. If one object is incorrectly selected as a front object in an earlier step, objects behind it will have their absolute layer order incorrectly shifted. Hence in keeping with prior works [44, 219], we use the pairwise occlusion order $G = (\Omega, W)$ to represent our final instance occlusion relationships for evaluation.

Given a single image $\mathbf{I}$, our model decomposes it into instances with completed RGB appearances $A_{\Omega}^{S_K}$. Here, A denotes the amodal perception instance (inclusive of both mask and appearance), Ω specifies instances in the scene, and S_K indicates which layers are the instances in (selected out in step s_k). When two segmented



(a) Input (b) Inmodal (c) PCNet (d) AmodalMask (e) Ours-Syn (f) Ours-Real (g) GT

FIGURE 6.5: **Amodal instance segmentation results on the COCOA validation set.** Our model trained on synthetic dataset (Ours-syn) achieves visually reasonable results in the similar real indoor scenes (example in top row), but it fails in some dissimilar real scenes (example in bottom row). After training on the real data with "pseudo ground-truths", the model (Ours-Real) performs much better. Note that, unlike the PCNet [219] that need visible inmodal ground-truth masks as input, our system decomposes a scene using only an RGB image.

amodal masks $A^{s_i}_{\omega_i}$ and $A^{s_j}_{\omega_j}$ overlap, we infer their occlusion relationship based on the order of the object-removing process, formally:

$$W_{\omega_i,\omega_j} = \begin{cases} 0 & \text{if } O(A^{s_i}_{\omega_i}, A^{s_j}_{\omega_j}) = 0 \\ 1 & \text{if } O(A^{s_i}_{\omega_i}, A^{s_j}_{\omega_j}) > 0 \text{ and } s_i < s_j \\ -1 & \text{if } O(A^{s_i}_{\omega_i}, A^{s_j}_{\omega_j}) > 0 \text{ and } s_i \geq s_j \end{cases} \tag{6.3}$$

where $O(A^{s_i}_{\omega_i}, A^{s_j}_{\omega_j})$ is the area of overlap between instances ω_i and ω_j. If they do not overlap, they share no pairwise depth-order relationship in a scene. If there is an overlap and the instance ω_i is first selected out with a smaller layer order, the inferred pairwise order is $W_{\omega_i,\omega_j} = 1$; otherwise it is labeled as $W_{\omega_i,\omega_j} = -1$. Hence the instance occlusion order only depends on the order (selected out step) of removal between the two instances, and do not suffer from shift errors.

6.4.4 Training on Real Data with Pseudo Ground-truth

Real-world data appropriate for a completed scene decomposition task is difficult to acquire, because ground truth shapes and RGB appearance for occluded parts are hard to collect without very extensive manual interventions, *e.g.* deliberate physical placement and removal of objects. Although our proposed model trained on the high-quality rendered data achieves visually reasonable results in some real scenes that share similarities to the rendered dataset (*e.g.* indoor scene in top row of Figure 6.5), it does not generalize well to dissimilar real scenes (*e.g.* outdoor

(a) Input (b) Masked Image (c) Completed Image

FIGURE 6.6: **Pseudo RGB ground-truth.** (a) Input. (b) Masked image by selecting out the fully visible objects. (c) Pseudo ground-truth generated from our model trained on synthetic data.

scene in bottom row of Figure 6.5). These are caused by: 1) differences in labeled categories between synthetic and real datasets, especially between indoor and outdoor scenes; and 2) inconsistencies between synthetically trained image completion of masked regions and fully visible real pixels.

One alternative is to simply use an image completion network trained only on real images. From our experience, this performs poorly in a scene decomposition task. The reason is that while real-trained image completion methods are able to create perceptually-pleasing regions and textures for a single completion, they do not appear to have the ability to adhere to consistent object geometry and boundaries when de-occluding, which is crucial for object shape completion. As a result, errors accumulate even more dramatically as the decomposition progresses.

Our *key motivating insight* is this: instead of training the model entirely without ground-truth in the completion task, we train it in a semi-supervised learning manner, exploiting the scene structure and object shape knowledge that has been gained in our synthetically-trained CSDNet. As shown in Figure 6.6, this synthetic completion model is able to generate visually adequate appearance, but more importantly it is better able to retain appropriate geometry and shapes. We can use this to guide the main image completion process in real-word data, while allowing a GAN-based loss to increase the realism of the output.

Specifically, for a real image $\mathbf{I}$, we first train the layered decomposition network using the manual annotated amodal labels. In a step, after segmenting and selecting

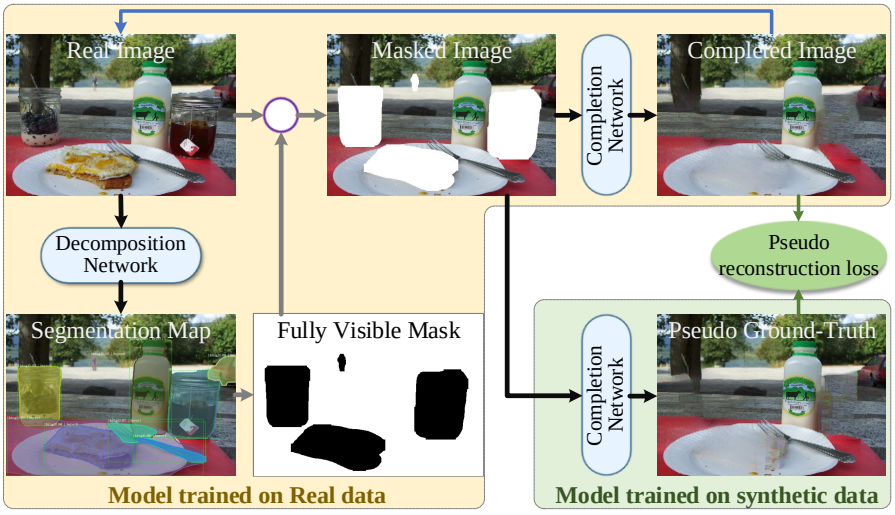


FIGURE 6.7: **Training pipeline for real images.** We introduce a semi-supervised learning method for real data by providing *pseudo* RGB ground-truth for originally invisible regions.

out the foreground objects, we obtain $\hat{\mathbf{I}}_{syn}^{(s_k)} = G(\mathbf{I}_m^{(s_k)}; \theta_{\text{syn}})$ to serve as "pseudo ground-truth" (green box in Figure 6.7) through the completion model trained on synthetic data. We then train the completion network $G(\mathbf{I}_m^{(s_k)}; \theta_{\text{real}})$ using the loss function of equation (6.2) by comparing the output $\hat{\mathbf{I}}_{real}^{(s_k)}$ to "pseudo ground-truth" $\hat{\mathbf{I}}_{syn}^{(s_k)}$. Like [166], we also reduce the weights of reconstruction loss L_{rec} and perceptual loss L_{per} to encourage the output to be biased towards the real image distribution via the discriminator loss L_{ad}. It is worth noticing that the completed image is *passed back* to the layered decomposition network in the next layer, where the decomposition loss L_{decomp} in equation (6.1) will be backpropagated to the completion network. This connection allows the completion network to *learn to complete real-world objects that might not be learned through the synthetic data.*

6.5 Results and Applications

6.5.1 Setup

Datasets We evaluated our system on three datasets: **COCOA** [237], **KINS** [151] and the rendered **CSD**. **COCOA** is annotated from COCO2014 [119], a large scale natural image datasets, in which 5,000 images are selected to manually label with pairwise occlusion orders and amodal masks. **KINS** is derived from the outdoor traffic dataset KITTI [57], in which 14,991 images were labeled with absolute layer orders and amodal masks. **CSD** is our rendered synthetic dataset, which contains

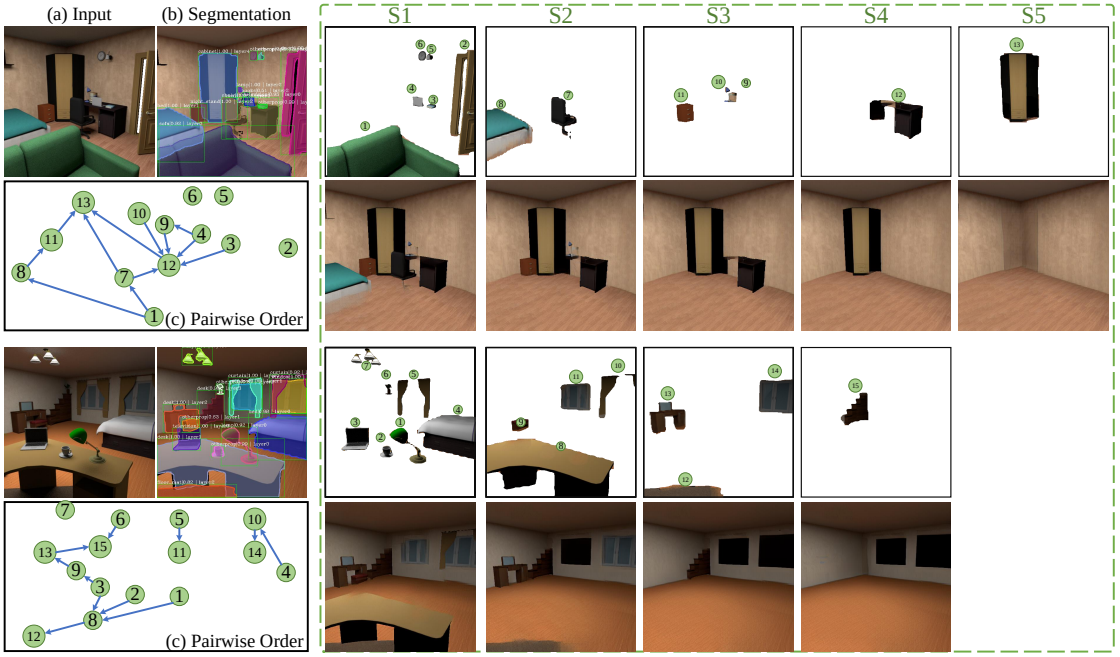


FIGURE 6.8: **Layer-by-Layer Completed Scene Decomposition** results on rendered CSD testing set. (a) Input RGB images. (b) Final amodal instance segmentation. (c) Inferred directed graph for pairwise order. (d) Columns labeled S1-5 show the decomposed instances (top) and completed scene (bottom) based on the predicted non-occluded masks. Note that the originally invisible parts are filled in with realistic appearance.

8,298 images, 95,030 instances for training and 1,012 images, 11,648 instances for testing. We conducted thorough experiments and ablation studies to assess the quality of the completed results for invisible appearance estimation (since the in-the-wild datasets lack ground-truth for the occluded parts).

Metrics For amodal instance segmentation, we report the standard COCO metrics [119], including AP (average over IoU thresholds), AP_{50}, AP_{75}, and AP_S, AP_M and AP_L (AP at different scales). Unless otherwise stated, the AP is for mask IoU. For appearance completion, we used RMSE, SSIM and PSNR to evaluate the quality of generated images. All images were normalized to the range $[0, 1]$.

Since the occlusion order is related to the quality of instance segmentation, we defined a novel metric for evaluating the occlusion order that uses the previous benchmark criterion for instance segmentation. Specifically, given a pairwise occlusion order $G = (\Omega, W)$ predicted by the model, we only evaluate the order for these valid instances that have IoU with ground-truth masks over a given threshold. For instance, if we set the threshold as 0.5, the predicted instance ω will be evaluated when we can identify a matched ground-truth mask with IoU ≥ 0.5. Hence we

	SegNet	box AP	mask AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask-RCNN [72]	Mask-RCNN	51.3	46.8	67.2	50.6	14.5	43.0	49.9
MLC [151]	Mask-RCNN	52.3	47.2	67.5	50.9	14.7	43.8	50.2
PCNet [219]	Mask-RCNN	-	43.6	59.1	43.4	11.5	40.4	46.0
HTC [20]	HTC	52.9	47.3	65.9	51.6	12.2	41.3	51.0
MLC [151]	HTC	53.6	47.9	66.1	52.3	13.1	41.9	51.7
PCNet [219]	HTC	-	45.7	60.6	49.2	10.2	39.3	48.4
CSDNet	Mask-RCNN	52.6	48.7	66.2	53.1	15.7	42.8	52.2
CSDNet	HTC	**56.3**	**50.3**	**67.7**	**53.4**	**17.4**	**44.2**	**53.1**
CSDNet-gt	Mask-RCNN	54.9	53.1	66.5	56.9	21.4*	49.9	57.0
CSDNet-gt	HTC	60.3*	56.0*	67.9*	59.3*	19.6	53.4*	59.5*

TABLE 6.2: **Amodal Instance Segmentation on CSD testing sets**. Mask-RCNN [72] and HTC [20] are the state-of-the-arts of the COCO segmentation challenges. MLC [151] is the latest amodal instance segmentation work for outdoor scene. PCNet [219] is the self-supervised amodal completion work. The **CSDNet**-gt holds same training environment as **CSDNet**, but is tested with completed ground-truths images $\mathbf{I}^{s*}$ in each step. Best results used ground-truth annotations are marked with *, while best results only used RGB images are in bold.

can measure the **occlusion average precision** (OAP) as assessed with different thresholds.

6.5.2 Results on Synthetic CSD Dataset

We first present results that we obtained from our framework when experimenting on our synthetic CSD dataset.

6.5.2.1 Main Results

Completed scene decomposition We show the qualitative results of CSDNet in Figure 6.8. Given a single RGB image, the system has learned to decompose it into semantically complete instances (*e.g.* counter, table, window) and the background (wall, floor and ceiling), while completing RGB appearance for *invisible* regions. Columns labeled S1-5 show the completed results layer-by-layer. In each layer, fully visible instances are segmented out, and after scene completion some previously occluded regions become fully visible in the next layer, *e.g.* the table in the second example. The final amodal instance segmentation results shown in Figure 6.8(b) consist of the fully visible amodal masks in each layer. Note that unlike MONet [16], our model does not need predefined slots. The process will stop when it is unable to detect any more objects.

Amodal instance segmentation We compare CSDNet to the state-of-the-art methods in amodal instance segmentation in Table 6.2. As the existing works

	Inputs		Ordering Algorithm	OAP	OAP_{50}	OAP_{75}	OAP_{85}	OAP_S	OAP_M	OAP_L
	Amodal	Ordering								
SeGAN [44]	$I + V_{gt}$	$V_{gt} + \hat{F}_{pre}$	IoU Area	68.4	-	-	-	-	-	-
SeGAN [44]	$I + \hat{V}_{pred}$	$V_{gt} + \hat{F}_{pre}$	IoU Area	66.1	50.2	65.6	70.4	10.6	65.0	63.8
HTC + MLC [151]	I	$V_{gt} + \hat{F}_{pre}$	IoU Area	76.5	70.3	77.1	79.8	11.6	69.8	78.2
HTC + MLC [151]	I	$\hat{F}_{pre}$ + layer	layer order[1]	51.9	44.3	50.8	54.6	11.7	60.8	46.2
HTC + PCNet [219]	$I + \hat{V}_{pred}$	$V_{gt} + \hat{F}_{pre}$	IoU Area	70.8	56.9	71.3	76.0	11.3	67.1	68.6
CSDNet	I	$\hat{F}_{pre}$	Area	44.7	45.3	45.7	45.1	17.4	34.5	41.5
CSDNet	I	$\hat{F}_{pre}$	Y-axis	62.0	60.1	61.2	62.7	**63.4**	58.6	66.1
CSDNet	I	$V_{gt} + \hat{F}_{pre}$	IoU Area	80.7	**77.2**	**81.0**	82.9	61.1	73.7	80.5
CSDNet	I	$\hat{F}_{pre}$ + layer	layer order	**81.7**	76.6	80.9	**84.6**	15.7	**75.9**	**82.6**
CSDNet-gt	I^{s*}	$\hat{F}_{pre}$ + layer	layer order	88.9*	85.2*	88.5*	90.1*	49.6	84.3*	89.9*

TABLE 6.3: **Instance depth ordering on CSD testing sets**. We report the pairwise depth ordering on occluded instance pairs OAP_{occ}. I^{s*} = ground-truth completed image in each step $s*$, V_{gt} = visible ground-truth mask, $\hat{V}_{pred}$ = visible predicated mask, and $\hat{F}_{pre}$ = full (amodal) predicated mask. layer order[1] only predicts the occlusion / non-occlusion labels in the original image (the first step in our model).

Mask-RCNN [72] and HTC [20] are aimed at inmodal perception for visible parts, we retrained their models for amodal perception task, by providing amodal ground-truths. We also retrained MLC [151] on our rendered dataset, which are the latest work for amodal perception. For PCNet [219], we used the predicted visible mask as input, rather than the original visible ground-truth annotations. While the HTC [20] improves on Mask-RCNN's [72] bounding box AP by about 1.6 points by refining the bounding box offsets in three cascade steps, the improvement for amodal mask segmentation is quite minor at 0.5 points. We believe this is an inherent limitation of methods that attempt amodal segmentation of occluded objects directly without first reasoning about occluding objects and masking their image features, as such the front objects' features will distract the network. In contrast, our CSDNet is able to improve the amodal mask segmentation accuracy by a relative 6.3% with the same *backbone segmentation network* (HTC), by jointing segmentation and completion with layer-by-layer decomposition.

To further demonstrate that better completed images improve amodal segmentation, we consider a scenario with a completion oracle, by using ground-truth appearances to repair the occluded parts in each step. This is denoted as the **CSDNet**-gt, for which amodal instance segmentation accuracy increases from 47.3% to 56.0% (relative 18.4% improvement). We also note that, while the **CSDNet**-gt using Mask-RCNN achieves lower bounding box score than our HTC **CSDNet** ("54.9" *vs* "56.3"), the mask accuracy is much higher ("53.1" *vs* "50.3"). This suggests that amodal segmentation benefits from better completed images.

	C1-F_{gt}				C2		
	RMSE	SSIM	PSNR		RMSE	SSIM	PSNR
SeGAN [44]	0.1246	0.8140	21.42	C2a-V_{gt}	0.2390	0.6045	16.03
PCNet [219]	0.1129	0.8267	23.16		0.2483	0.5931	15.54
DNT [38]	0.1548	0.7642	20.32	C2b	0.2519	0.5721	15.10
PICNet [227]	0.0927	0.8355	28.81		0.1401	0.7730	24.71
CSDNet	**0.0614**	**0.9179**	**35.24**		**0.0914**	**0.8768**	**30.45**

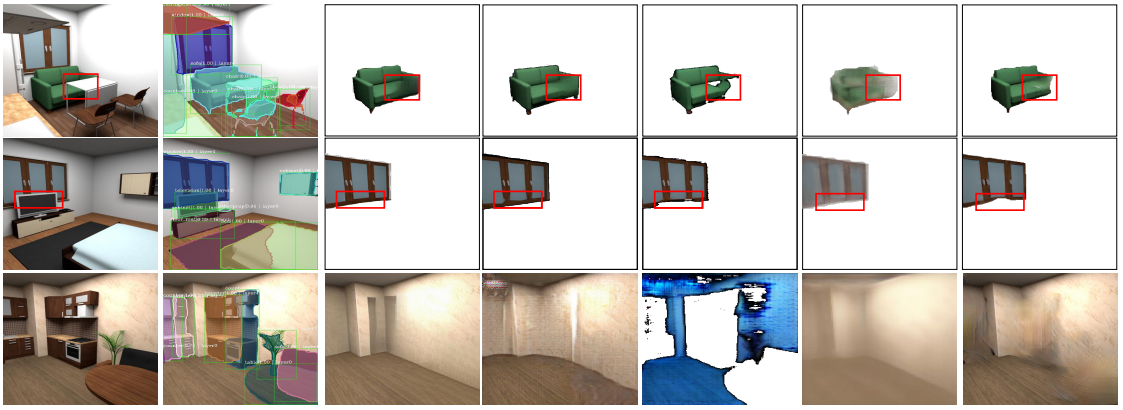
TABLE 6.4: **Object Completion**. F_{gt} = full ground-truth mask, V_{gt} = visible ground-truth mask. For methods provided with F_{gt}, we only evaluate the completion networks.

Instance depth ordering Following [237], we report the pairwise instance depth ordering for correctly detected instances in Table 6.3. The original SeGAN and PCNet used ground-truth visible masks V_{gt} as input. For a fair comparison, we first retrained them on our synthetic data using the same segmentation network (HTC [20]) for all models. After predicting amodal masks, we assessed various instance depth ordering algorithms: two baselines proposed in AmodalMask [237] of ordering by *area*[1] and by *y-axis* (amodal masks closest to image bottom in front), ordering by *incremental area* defined as the *IoU area* between visible and amodal masks[2], and our ordering by absolute *layer order* (Section 6.4.3).

As can be seen in Table 6.3, all instantiations of our model outperformed baselines as well as previous models. Unlike SeGAN [44] and PCNet [219], our final model explicitly predicts the occlusion labels of instances, which improved the OAP substantially. While MLC [151] predicts the instance occlusion order in a network, it only contains one layer for binary occlusion / non-occlusion labeling. In contrast, our method provides a fully structural decomposition of a scene in multiple steps. Additionally, we observed that our model achieves better performance with a higher IoU threshold for selecting the segmented mask (closer match to the ground-truth masks). We further observed that the occlusion relationships of small objects are difficult to infer in our method. However, the *Y-axis* ordering method had similar performance under various metrics as it only depends on the locations of objects. Note that our depth ordering does *not* rely on any ground-truth that is used in [44, 219].

[1]We used the heuristic in PCNet [219] — larger masks are ordered in front for KINS, and behind for COCOA and CSD.

[2]See details in [219], where the visible ground-truth masks V_{gt} are used for ordering.



(a) Input (b) Segmentation (c) Ours (d) SeGAN-F_{gt} (e) SeGAN-V_{gt} (f) DNT (g) PCNet-V_{gt}

FIGURE 6.9: Results for **Visiting the Invisible**. We show the input image, our amodal instance segmentation results, and the objects and background we try to visit. The red rectangles highlight the previously *invisible* regions of occluded objects.

Object completion. We finally evaluated the quality of generated appearances. We compared our results to those from SeGAN [44], Dhamo *et al.* [38] (abbrev. as DNT), PCNet [219] and our previous PICNet [227] (original point-attention) in Table 6.4. We evaluated different conditions of: C1) when the ground-truth full mask F_{gt} is provided to all methods, C2a) when the ground-truth visible mask V_{gt} is the input to SeGAN and PCNet, and C2b) when an RGB image is the only input to other methods. C2a-V_{gt} is considered because SeGAN and PCNet assumes that a predefined mask is provided as input.

In C1-F_{gt}, CSDNet substantially outperformed the other methods. In C2, even when given only RGB images *without* ground-truth masks, our method worked better than SeGAN and PCNet with V_{gt}. One important reason for the strong performance of CSDNet is the *occlusion reasoning* component, which constraints the completed shapes of partly occluded objects based on the global scene context and other objects *during testing*.

Qualitative results are visualized in Figure 6.9. We noted that SeGAN worked well only when ground-truth amodal masks F_{gt} were available to accurately label which parts were *invisible* that needed filling in, while DNT generated blurry results from simultaneously predicting RGB appearance and depth maps in one network, which is not an ideal approach [216]. The PCNet [219] can not correctly repair the object shape as it trained without ground-truth object shape and appearance. Our CSDNet performed much better on background completion, as it only masked fully visible objects in each step instead of all objects at a go, so that *earlier completed information propagates to later steps*.

	train	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
gt	-	56.0*	67.9*	59.3*	19.6*	53.4*	59.5*
w/o	-	36.8	52.6	38.3	10.8	31.6	38.2
PICNet-point	sep	40.8	63.0	43.5	12.6	37.2	43.7
PICNet-patch	sep	43.8	60.7	46.6	11.0	36.3	45.5
PICNet-point	end	47.7	63.2	50.6	14.9	41.7	51.3
PICNet-patch	end	**50.3**	**67.7**	**53.4**	**17.4**	**44.2**	**53.1**

(a) **Effect of different completion methods on instance segmentation (HTC-based decomposition)**. "sep" = separate training of the 2 networks, "w/o" = without any completion, and "end" = joint training.

	train	RMSE	SSIM	PSNR
gt	-	0.0614*	0.9179*	35.24*
M-RCNN[72]	sep	0.1520	0.7781	22.34
HTC [20]	sep	0.1496	0.7637	26.75
M-RCNN[72]	end	0.1345	0.7824	27.31
HTC [20]	end	**0.0914**	**0.8768**	**30.45**

(b) **Effect of different decomposition methods on scene completion (Patch-Attention PICNet)**. Better scene decomposition improved scene completion.

TABLE 6.5: **Ablations** for joint optimization. In each table, we fixed one model for one subtask and trained different models for the other subtask. Better performance in one task can improve the performance in the other, which demonstrates the joint training of two tasks with layer-by-layer decomposition contributes to each other.

6.5.2.2 Ablation Studies

To demonstrate the two tasks can contribute to a better scene understanding system, instead of solving them isolated, we ran a number of ablations.

Does better *completion* help decomposition? We show quantitative results for a fixed decomposition network (layered HTC [20] with two completion methods in Table 6.5(a). Without any completion ("w/o"), segmented results were naturally bad ("36.8" *vs* "50.3") as it had to handle empty regions. More interestingly, even if advanced methods were used to generate visual completion, the isolated training of the decomposition and completion networks led to degraded performance. This suggests that even when generated imagery looks good visually, there is still a domain or semantic gap to the original visible pixels, and thus flaws and artifacts will affect the next segmentation step. Our original PICNet with patch attention provides better completed results than the original point attention PICNet [227], resulting in a large improvement ("50.3" *vs* "47.7") of amodal instance segmentation.

Does better *decomposition* help completion? To answer this, we report the results of using different scene segmentation networks with a same completion network (Patch-attention PICNet [227]) in Table 6.5(b). We also first considered the ideal situation that ground-truth segmentation masks were provided in each decomposition step. As shown in Table 6.5(b), the completion quality significantly improved (RMSE: "0.0614", SSIM: "0.9179" and PSNR: "35.24") as occluded parts were correctly pointed out and the completion network precisely knows which parts need to be completed. HTC [20] provided better instance masks than

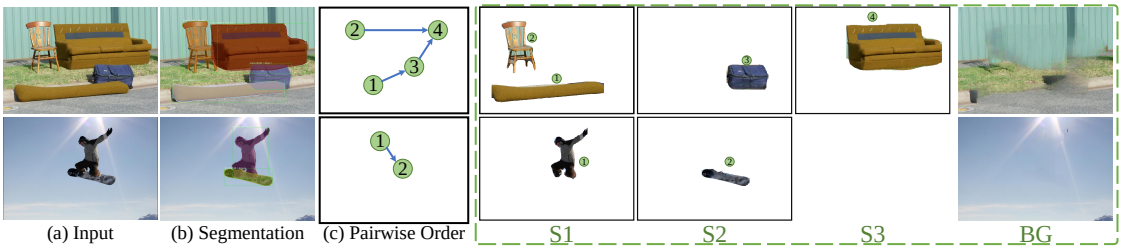


FIGURE 6.10: **Layer-by-layer completed scene decomposition on natural images**. (a) Inputs. (b) Final amodal instance segmentation. (c) Inferred directed graph for pairwise occlusion order. (d) Columns labeled S1-3 show the decomposed instances with completed appearance in each step.

	Inputs	SegNet	COCOA (%mAP)	KINS (%mAP)
Amodel [237]	I	Sharp [149]	7.7	-
Mask-RCNN [72]	I	Mask-RCNN [72]	31.8	29.3
ORCNN [50]	I	Mask-RCNN [72]	33.2	29.0
MLC [151]	I	Mask-RCNN [72]	34.0	31.1
MLC [151]	I	HTC [20]	34.4	31.6
PCNet [219]	I+$\hat{V}_{pred}$	Mask-RCNN [72]	30.3	28.6
PCNet [219]	I+$\hat{V}_{pred}$	HTC [20]	32.6	30.1
CSDNet	I	Mask-RCNN [72]	34.1	31.5
CSDNet	I	HTC [20]	**34.8**	**32.2**

TABLE 6.6: **Amodal Instance Segmentation on COCOA and KINS sets**. The gray color shows results reported in existing works and the others are our reported results by using the released codes and our CSDNet.

Mask-RCNN [72], which resulted in more accurately completed scene imagery. The best results were with end-to-end jointly training.

6.5.3 Results on Real Datasets

We now assess our model on real images. Since the ground-truth appearances are unavailable, we only provide the visual *scene manipulation* results in Section 6.5.4, instead of quantitative results for *invisible completion*.

Completed scene decomposition In Figure 6.10, we visualize the layer-by-layer completed scene decomposition results on real images. Our CSDNet is able to decompose a scene into completed instances with correct ordering. The originally occluded invisible parts of "suitcase", for instance, is completed with full shape and realistic appearance. Note that, our system is a fully scene understanding method that only takes an image as input, without requiring the other manual annotations as [44, 219].

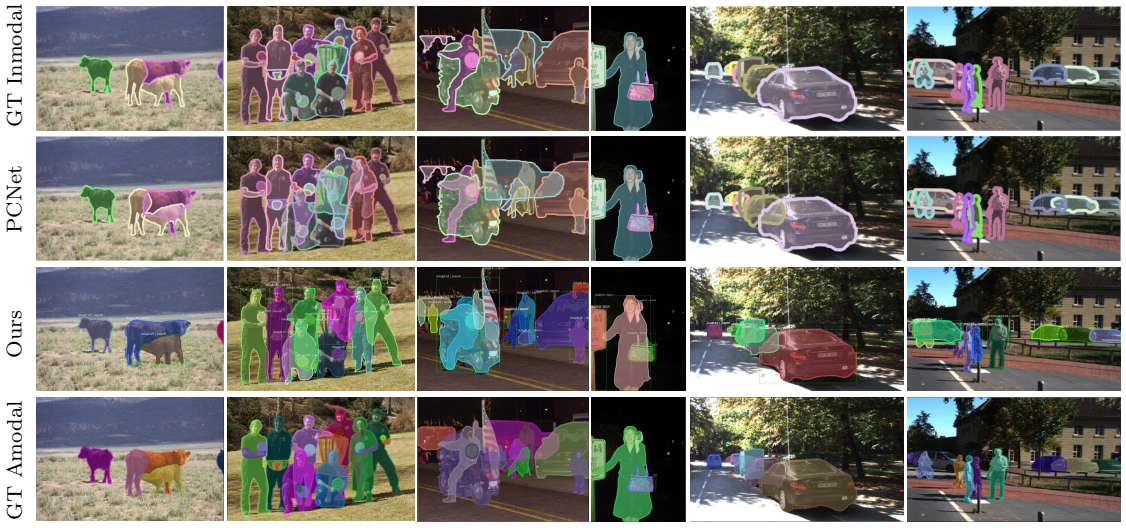


FIGURE 6.11: **Amodal instance segmentation results on natural images**. Our CSDNet learns to predict the intact mask for the occluded objects (*e.g.* animals and human). Note that, unlike PCNet [219], our model does *not* depend on the visible mask (first row) as input. Hence it can handle some objects without ground-truth annotation, such as two 'humans' in the third column and the 'smartphone' in the fourth column.

Amodal instance segmentation Next, we compare with state-of-the-art methods on amodal instance segmentation. Among these, AmodalMask [237] and ORCNN [50] were trained for the COCOA dataset, MLC [151] works for the KINS dataset, and PCNet [219] is focused on amodal completion (mask completion) rather than amodal instance segmentation (requiring precise visible masks). For a fair comparison, when these methods do not provide results on a given dataset, we trained their models using publicly released code. For COCOA, we only report the results for "thing" category (*e.g.* car, person, chair), because the "stuff" category (*e.g.* glass, cloud, water) does not have specific shapes.

Table 6.6 shows that our results (34.8 mAP and 32.2 mAP) are 0.4 points and 0.6 points higher than the recent MLC using the same segmentation structure (HTC) in COCOA and KINS, respectively. PCNet [219] considers amodal perception in two steps and assumes that visible masks are available. We note that their mAP scores were very high when the visible ground-truth masks were provided. This is because all initial masks were matched to the annotations (without detection and segmentation errors for instances, as shown in Figure 6.11). However, when we used a segmentation network to obtain visible masks $\hat{\mathrm{V}}_{pred}$ for PCNet, the amodal instance segmentation results became lower than other methods, suggesting that it is much harder to segment a visible mask and then complete it.

	Ordering Inputs	Ordering Algorithm	COCOA (OAP)	KINS (OAP)
OrderNet [237]	I+F_{gt}	Network	88.3	94.1
PCNet [219]	V_{gt} +$\hat{F}_{pre}$	IoU Area	84.6	86.0
MLC [151]	V_{gt} + $\hat{F}_{pre}$	IoU Area	80.3	82.3
CSDNet	V_{gt} + $\hat{F}_{pre}$	IoU Area	84.7	86.4
MLC [151]	$\hat{V}_{pred}$ + $\hat{F}_{pre}$	IoU Area	74.2	80.2
MLC [151]	$\hat{F}_{pre}$ + layer	layer order[1]	66.5	71.8
PCNet [219]	$\hat{V}_{pred}$ + $\hat{F}_{pre}$	IoU Area	72.4	79.8
CSDNet	$\hat{V}_{pred}$ + $\hat{F}_{pre}$	IoU Area	75.4	81.6
CSDNet	$\hat{F}_{pre}$ + layer	layer order	80.9	82.2

TABLE 6.7: **Instance depth ordering on COCOA and KINS sets**. The blue rows show the results that uses ground-truth annotations as inputs.

In Figure 6.11, we compare our CSDNet and PCNet [219]. PCNet only completes the given visible annotated objects which had visible masks. In contrast, our CSDNet produces more contiguous amodal instance segmentation maps even for some unlabeled objects, for instance, the two "humans" in the third column. Furthermore, our model can directly create a deep hierarchical representation of a scene, producing a layer order for each instance.

Instance depth ordering. Finally, we report the instance depth ordering results in Table 6.7. In order to compare with existing work, we considered two settings: ground-truths provided (blue rows in Table 6.7), and only RGB images given. The OrderNet obtained the best results as the ground-truth full masks F_{gt} were given. We note that PCNet and our model achieved comparable performance when the visible ground-truth masks were used. Note that, we only used V_{gt} for depth ordering, while PCNet utilized the visible mask as input for both mask prediction and depth ordering. Furthermore, when no ground-truth annotation was provided as input, our model performed better than MLCand PCNet.

6.5.4 Applications

We illustrate some image editing / re-composition applications of this novel task, after the system has learned to decompose a scene into isolated completed objects together with their spatial occlusion relationships. In Figure 6.12, we visualize some recomposed scenes on various datasets, including our CSD, real COCOA [237], KITTI [57] and NYU-v2 [138].

In these cases, we directly modified the positions and occlusion ordering of individual objects. For instance, in the first bedroom example, we *deleted* the

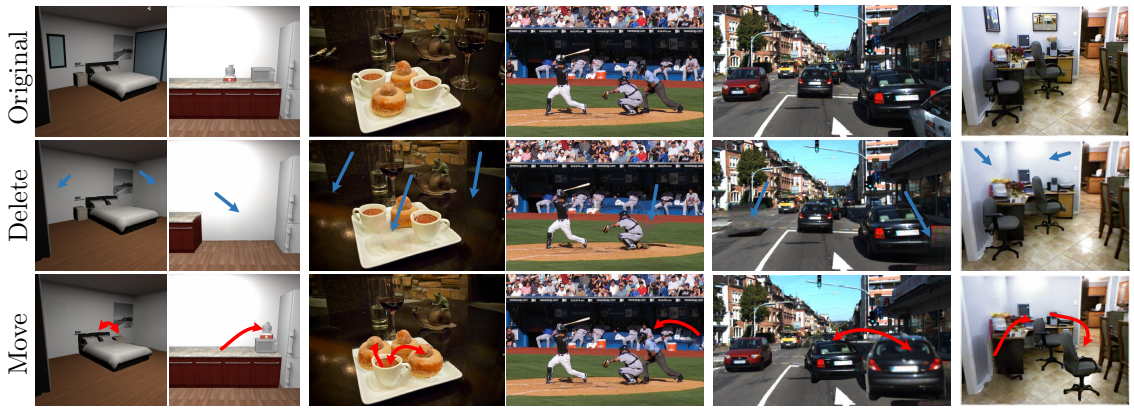


FIGURE 6.12: **Free editing based on the results of our system** on images from various datasets. Note that our method is able to automatically detect, segment and complete the objects in the scene, *without the need for manual interactive masking*, with interactive operations limited to only "delete" and "drag-and-drop". The blue arrows show object removal, while red arrows show object moving operations. We can observe that the originally *invisible* regions are fully visible after editing.

"window", and *moved* the "bed" and the "counter", which resulted in also *modifying* their occlusion order. Note that all original *invisible* regions were filled in with reasonable appearance. We also tested our model on real NYU-v2 [138] images which do *not* belong to any of the training sets used. As shown in the last column of Figure 6.12, our model was able to detect and segment the object and complete the scene. The "picture", for instance, is deleted and filled in with background appearance.

6.6 Limirations and Discussion

Building on previous inmodal and amodal instance perception work, we explored a higher-level structure scene understanding task that aims to decompose a scene into semantic instances, with completed RGB appearance and spatial occlusion relationships. We presented a layer-by-layer CSDNet, an iterative method to address this novel task. The main motivation behind our method is that fully visible objects, at each step, can be relatively easily detected and selected out without concern about occlusion. To do this, we simplified this complex task to two subtasks: instance segmentation and scene completion. We analyzed CSDNet and compared it with recent works on various datasets. Experimental results show that our model can handle an arbitrary number of objects and is able to generate

the appearance of occluded parts. Our model outperformed current state-of-the-art methods that address this problem in one pass. The thorough ablation studies on synthetic data demonstrate that the two subtasks can contribute to each other through the layer-by-layer processing.

Although we have achieved good results for visiting the invisible, there are some limitations to the proposed method. First, if there are too many objects in a complex scene, the progressively introduced artifacts in image completion will have an increasing impact on subsequent steps. Second, limited by GPU memory, the completion network currently operates at a lower resolution than the scene decomposition network. Besides, we freely remove and move objects in a natural scene, but it is still an operation in a 2D image. It will be much more interesting to do the free editing in 3D space, just like the real-world object interaction [203].

Chapter 7

Conclusion and Future Directions

In previous chapters, a few novel learning-based methods have been presented for visual synthesis and generation, including changing visual appearance for I2I translation (Chapters 2 and 3), generating semantic content for image completion (Chapters 4 and 5), and simultaneously modeling shapes and appearance for scene decomposition and completion (Chapter 6). In each chapter, a new model is introduced to advance the state-of-the-art in the corresponding task and I hope to bring some new perspectives for each task. The extensive experiments have demonstrated that the proposed approaches can generate reasonable content as well as visually realistic appearance results compared to previous methods.

For changing visual appearance in **Part I**, we found that the learned model mainly focused on modifying local patch textures, regardless of the global semantic information. In particular, when we aimed to generate multiple and diverse results in an I2I translation task, repeated noises tend to be added at different image or feature positions, such as in BicycleGAN [235] and MUNIT [84]. This resulted in unwanted texture modification, *e.g.* the zebra-stripe in background in *horse→zebra*. To generate semantic content in **Part II**, we needed to correctly model long-range dependencies, instead of focusing on local texture information. Therefore, the general network architecture involved downsampling the image to lower resolutions to extract global information, *e.g.* 5 times downsampling (Chapter 4) compared to 2 times in I2I translation (Chapters 2 and 3). The patch discriminator [237] is also replaced by the global discriminator (Chapters 4 and 5) that can model long-range relationships. Furthermore, transformer-based architectures (Chapter 5, [190]) have rapidly improved the image completion results for both single and multiple solutions, which further demonstrated that directly

modeling the long-range visible information is quite important for semantic content generation.

As for simultaneously modeling shapes and appearance in **Part III**, it remains quite a challenging problem, which requires global perception of a scene to be able to decompose all instances as well as infer their underlying occlusion relationships, with local texture modeling needed to generate visually realistic appearance for occluded regions. Furthermore, although we rendered a high-quality synthetic dataset with RGB ground truth for all instances and background in this thesis, the dataset still has a gap to real images. Building a publicly available dataset for this higher-level scene understanding task is still some distance away.

Next, I discuss several possible future research directions, building on our current visual synthesis and generation algorithms.

Visual Word Representation in Generator As mentioned above, while visually plausible results have been achieved in image translation and completion, there are a number of failure cases in all methods. A possible factor is the distribution of the training features and the testing features being different. Recently, vector quantization (VQ) has been re-used in the computer vision community and contributed to excellent performance in image generation [141, 147, 157]. Due to the quantization and online learned dictionary, training features and testing features will belong more closely to the same domain, which is naturally suitable for image generation, resulting in lower risk of mode "collapse". Furthermore, the quantized visual words can be processed with frameworks used in NLP, in which the transformer [48] has shown excellent performance.

Image Editing with Interactive Inputs Existing learning-based image editing approaches have achieved rapid improvement over a short period of time, but most of the results are not manual editable. While the latest EdgeConnect approach [139] provides edge input during the completion, it is difficult to train networks to recognize arbitrary edges, due to the gap between manual input edges and ground truth edges. However, is it enough to only provide edges? On the other hand, some recent works [145, 153] have succeeded in text-guided image generation and manipulation, where the content and style in the generative image is controllable using language. These breakthroughs may enable high-level controllable image editing applications. In particular, it may be possible to learn the joint distribution of visual words and language words in the future.

3D View Synthesis While working on 2D scene synthesis and generation, I realized that in addition to content generation in a 2D plane, the GAN-based method is a potentially powerful method to create different views from a single image or limited numbers of images. In particular, we can rebuild the 3D shapes and try to hallucinate unobserved parts based on prior knowledge and limited visible information, similar to the 2D image completion in **Part II**. However, the existing methods [30, 59, 62, 191] that learns to rebuild the 3D scene in 3D format, such as point cloud [117], voxel [59] and mesh [191], we would like to represent the 3D structure as a 3D feature in latent space, where a corresponding generator can be applied as a render simulator to generate visually realistic images from arbitrary views.

Appendix A

Proofs for Chapter 4

A.1 Mathematical Derivation and Analysis

A.1.1 Difficulties with Using the Classical CVAE for Image Completion

Here we elaborate on the difficulties encountered when using the classical CVAE formulation for pluralistic image completion, expanding on the shorter description in Section 4.3.1.1.

A.1.1.1 Background: Derivation of the Conditional Variational Auto-Encoder (CVAE)

The broad CVAE framework of Sohn *et al.* [176] is a straightforward conditioning of the classical VAE. Using the notation in Chapter 4, a latent variable $\mathbf{z}_c$ is assumed to stochastically generate the hidden partial image $\mathbf{I}_c$. When conditioned on the visible partial image $\mathbf{I}_m$, we get the conditional probability:

$$p(\mathbf{I}_c|\mathbf{I}_m) = \int p_\phi(\mathbf{z}_c|\mathbf{I}_m)p_\theta(\mathbf{I}_c|\mathbf{z}_c,\mathbf{I}_m)d\mathbf{z}_c \tag{A.1}$$

The variance of the Monte Carlo estimate can be reduced by importance sampling:

$$\begin{aligned} p(\mathbf{I}_c|\mathbf{I}_m) &= \int q_\psi(\mathbf{z}_c|\mathbf{I}_c,\mathbf{I}_m)\frac{p_\phi(\mathbf{z}_c|\mathbf{I}_m)}{q_\psi(\mathbf{z}_c|\mathbf{I}_c,\mathbf{I}_m)}p_\theta(\mathbf{I}_c|\mathbf{z}_c,\mathbf{I}_m)d\mathbf{z}_c \\ &= \mathbb{E}_{\mathbf{z}_c\sim q_\psi(\mathbf{z}_c|\mathbf{I}_c,\mathbf{I}_m)}\left[\frac{p_\phi(\mathbf{z}_c|\mathbf{I}_m)}{q_\psi(\mathbf{z}_c|\mathbf{I}_c,\mathbf{I}_m)}p_\theta(\mathbf{I}_c|\mathbf{z}_c,\mathbf{I}_m)\right] \end{aligned} \tag{A.2}$$

Taking logs and apply Jensen's inequality leads to

$$\log p(\mathbf{I}_c|\mathbf{I}_m) \geq \mathbb{E}_{\mathbf{z}_c\sim q_\psi(\mathbf{z}_c|\mathbf{I}_c,\mathbf{I}_m)}\left[\log p_\theta(\mathbf{I}_c|\mathbf{z}_c,\mathbf{I}_m) - \log\frac{q_\psi(\mathbf{z}_c|\mathbf{I}_c,\mathbf{I}_m)}{p_\phi(\mathbf{z}_c|\mathbf{I}_m)}\right]$$
$$\mathcal{V} = \mathbb{E}_{\mathbf{z}_c\sim q_\psi(\mathbf{z}_c|\mathbf{I}_c,\mathbf{I}_m)}\left[\log p_\theta(\mathbf{I}_c|\mathbf{z}_c,\mathbf{I}_m)\right] - \mathrm{KL}\left(q_\psi(\mathbf{z}_c|\mathbf{I}_c,\mathbf{I}_m)||p_\phi(\mathbf{z}_c|\mathbf{I}_m)\right) \tag{A.3}$$

The variational lower bound $\mathcal{V}$ totaled over all training data is jointly maximized *w.r.t.* the network parameters θ, ϕ and ψ in attempting to maximize the total log likelihood of the observed training instances.

A.1.1.2 Single Instance Per Conditioning Label

As is typically the case for image completion, there is only one training instance of $\mathbf{I}_c$ for each unique $\mathbf{I}_m$. This means that for the function $q_\psi(\mathbf{z}_c|\mathbf{I}_c,\mathbf{I}_m)$, $\mathbf{I}_c$ can be learned into the network as a hard-coded dependency of the input $\mathbf{I}_m$, so $q_\psi(\mathbf{z}_c|\mathbf{I}_c,\mathbf{I}_m) \cong \hat{q}_\psi(\mathbf{z}_c|\mathbf{I}_m)$. Assuming that the network for $p_\phi(\mathbf{z}_c|\mathbf{I}_m)$ has similar or higher modeling power and there are no other explicit constraints imposed on it, then in training $p_\phi(\mathbf{z}_c|\mathbf{I}_m) \to \hat{q}_\psi(\mathbf{z}_c|\mathbf{I}_m)$, and the KL divergence in (A.3) goes to zero.

In this situation of zero KL divergence, we can rewrite the variational lower bound and replace $\hat{q}_\psi(\mathbf{z}_c|\mathbf{I}_m)$ with $p_\phi(\mathbf{z}_c|\mathbf{I}_m)$ without loss of generality, as

$$\mathcal{V} = \mathbb{E}_{\mathbf{z}_c\sim p_\phi(\mathbf{z}_c|\mathbf{I}_m)}\left[\log p_\theta(\mathbf{I}_c|\mathbf{z}_c,\mathbf{I}_m)\right] \tag{A.4}$$

A.1.1.3 Unconstrained Learning of the Conditional Prior

We can analyze how $\mathcal{V}$ can be maximized, by using Jensen's inequality again (reversing earlier use)

$$\mathcal{V} \leq \log \mathbb{E}_{\mathbf{z}_c\sim p_\phi(\mathbf{z}_c|\mathbf{I}_m)}\left[p_\theta(\mathbf{I}_c|\mathbf{z}_c,\mathbf{I}_m)\right]$$
$$= \log\int p_\phi(\mathbf{z}_c|\mathbf{I}_m)p_\theta(\mathbf{I}_c|\mathbf{z}_c,\mathbf{I}_m)d\mathbf{z}_c \tag{A.5}$$

By further applying Hölder's inequality (*i.e.* $\|fg\|_1 \leq \|f\|_p \|g\|_q$ for $\frac{1}{p}+\frac{1}{q}=1$), we get

$$\mathcal{V} \leq \log\left[\left|\int |p_\phi(\mathbf{z}_c|\mathbf{I}_m)|\, d\mathbf{z}_c\right| \left|\int |p_\theta(\mathbf{I}_c|\mathbf{z}_c,\mathbf{I}_m)|^\infty\, d\mathbf{z}_c\right|^{\frac{1}{\infty}}\right] \quad \text{(by setting } p=1, q=\infty\text{)}$$
$$= \log\left[1\cdot\max_{\mathbf{z}_c} p_\theta(\mathbf{I}_c|\mathbf{z}_c,\mathbf{I}_m)\right] = \max_{\mathbf{z}_c}\log p_\theta(\mathbf{I}_c|\mathbf{z}_c,\mathbf{I}_m) \tag{A.6}$$

Assuming that there is a unique global maximum for $\log p_\phi(\mathbf{z}_c|\mathbf{I}_m)$, the bound achieves equality when the conditional prior becomes a Dirac delta function centered at the maximum latent likelihood point

$$p_\phi(\mathbf{z}_c|\mathbf{I}_m) \to \delta(\mathbf{z}_c - \mathbf{z}_c^*) \quad \text{where } \mathbf{z}_c^* = \arg\max_{\mathbf{z}_c} \log p_\theta(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m) \tag{A.7}$$

Intuitively, subject to the vagaries of stochastic gradient descent, the network for $p_\phi(\mathbf{z}_c|\mathbf{I}_m)$ without further constraints will learn a narrow delta-like function that sifts out maximum latent likelihood value of $\log p_\theta(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m)$.

As mentioned in Section 4.3.1.1, although this narrow conditional prior may be helpful in estimating a single solution for $\mathbf{I}_c$ given $\mathbf{I}_m$ during testing, this is poor for sampling a diversity of solutions. In our framework, the (unconditional) latent priors are imposed for the partial images themselves, which prevent this delta function degeneracy.

A.1.1.4 CVAE with Fixed Prior

An alternative CVAE variant [189] assumes that conditional prior is independent of the $\mathbf{I}_m$ and fixed, so $p(\mathbf{z}_c|\mathbf{I}_m) \cong p(\mathbf{z}_c)$, where $p(\mathbf{z}_c)$ is a fixed distribution (*e.g.* standard normal). This means

$$p(\mathbf{I}_c|\mathbf{I}_m) = \int p(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m)p(\mathbf{z}_c)d\mathbf{z}_c \tag{A.8}$$

Now we can consider the case for a fixed $\mathbf{I}_m = \mathbf{I}_m^*$, and rewrite (A.8) as

$$p_{\mathbf{I}_m^*}(\mathbf{I}_c) = \int p_{\mathbf{I}_m^*}(\mathbf{I}_c|\mathbf{z}_c)p(\mathbf{z}_c)d\mathbf{z}_c \tag{A.9}$$

Doing so makes it obvious we can then derive the standard (unconditional) VAE formulation from here. Thus, an appropriate interpretation of this CVAE variant is that it uses $\mathbf{I}_m$ as a "switch" parameter to choose between different VAE models that are trained for the specific conditions.

Once again, this is fine if there are multiple training instances per conditional label. However, in the image completion problem, there is only one $\mathbf{I}_c$ per unique $\mathbf{I}_m$, so the condition-specific VAE model will simply ignore the sampling "noise" and learn to predict the single instance of $\mathbf{I}_c$ from $\mathbf{I}_m$ directly, *i.e.* $p(\mathbf{I}_c|\mathbf{z}_c, \mathbf{I}_m) \approx p(\mathbf{I}_c|\mathbf{I}_m)$, which incidentally achieves equality for the variational lower bound. This results in negligible variation of output despite now sampling from $p(\mathbf{z}_c) = \mathcal{N}(0, 1)$.

Our framework resolves this in part by defining all (unconditional) partial images of $\mathbf{I}_c$ as sharing a common latent space with adaptive priors, with the likelihood parameters learned as an unconditional VAE, and further coupling on the conditional portion (*i.e.* the generative path) to get a more distinct but regularized estimate for $p(\mathbf{z}_c|\mathbf{I}_m)$.

A.1.2 Joint Maximization of Unconditional and Conditional Variational Lower Bounds

The overall training loss function (4.7) used in our framework has a direct link to jointly maximizing the unconditional and unconditional variational lower bounds, respectively expressed by (4.2) and (4.5). Using simplified notation, we rewrite these bounds respectively as:

$$\begin{aligned}\mathcal{B}_1 &= \mathbb{E}_{q_\psi} \log p_\theta^r - \mathrm{KL}(q_\psi||p_{z_c}) \\ \mathcal{B}_2 &= \lambda \left(\mathbb{E}_{q_\psi} \log p_\theta^r - \mathrm{KL}(q_\psi||p_{z_c})\right) + \mathbb{E}_{p_\phi} \log p_\theta^g \end{aligned} \tag{A.10}$$

To clarify, $\mathcal{B}_1$ is the lower bound related to the unconditional log likelihood of observing $\mathbf{I}_c$, while $\mathcal{B}_2$ relates to the log likelihood of observing $\mathbf{I}_c$ conditioned on $\mathbf{I}_m$. The expression of $\mathcal{B}_2$ reflects a blend of conditional likelihood formulations with and without the use of importance sampling, which are matched to different likelihood models, as explained in Section 4.3.1.1. Note that the $(1-\lambda)$ coefficient from (4.5) is left out here for simplicity, but there is no loss of generality since we can ignore a constant factor of the true lower bound if we are simply maximizing it. We can then define a combined objective function as our maximization goal

$$\begin{aligned}\mathcal{B} &= \beta\, \mathcal{B}_1 + \mathcal{B}_2 \\ &= (\beta+\lambda)\mathbb{E}_{q_\psi} \log p_\theta^r + \mathbb{E}_{p_\phi} \log p_\theta^g - [\beta \mathrm{KL}(q_\psi||p_{z_c}) + \lambda \mathrm{KL}(q_\psi||p_\phi)] \end{aligned} \tag{A.11}$$

with $\beta \geq 0$.

To understand the relation between $\mathcal{B}$ in (A.11) and $\mathcal{L}$ in (4.7), we consider the equivalence of:

$$-\mathcal{B} \cong \mathcal{L} = \alpha_{\mathrm{KL}}(\mathcal{L}_{\mathrm{KL}}^r + \mathcal{L}_{\mathrm{KL}}^g) + \alpha_{\mathrm{app}}(\mathcal{L}_{\mathrm{app}}^r + \mathcal{L}_{\mathrm{app}}^g) + \alpha_{\mathrm{ad}}(\mathcal{L}_{\mathrm{ad}}^r + \mathcal{L}_{\mathrm{ad}}^g) \tag{A.12}$$

Comparing terms

$$\mathcal{L}^r_{\mathrm{KL}} \cong \mathrm{KL}(q_\psi || p_{z_c}), \quad \mathcal{L}^g_{\mathrm{KL}} \cong \mathrm{KL}(q_\psi || p_\phi) \quad \Rightarrow \beta = \lambda = \alpha_{\mathrm{KL}} \tag{A.13}$$

For the reconstructive path that involves sampling from the (posterior) importance function $q_\psi(\mathbf{z}_c|\mathbf{I}_c)$ of (4.3), we can substitute $(\beta + \lambda) = 2\alpha_{\mathrm{KL}}$ and get the reconstructive log likelihood formulation as

$$-\mathbb{E}_{q_\psi} \log p^r_\theta \cong \frac{\alpha_{\mathrm{app}}}{2\alpha_{\mathrm{KL}}} \mathcal{L}^r_{\mathrm{app}} + \frac{\alpha_{\mathrm{ad}}}{2\alpha_{\mathrm{KL}}} \mathcal{L}^r_{\mathrm{ad}} \tag{A.14}$$

Here, $\mathbf{I}_c$ is available, with $\mathcal{L}^r_{\mathrm{app}}$ reconstructing both $\mathbf{I}_c$ and $\mathbf{I}_m$ as in (4.10), while $\mathcal{L}^r_{\mathrm{ad}}$ involves GAN-based pairwise feature matching (4.12).

For the generative path that involves sampling from the conditional prior $p_\phi(\mathbf{z}_c|\mathbf{I}_m)$, we have the generative log likelihood formulation as

$$-\mathbb{E}_{p_\phi} \log p^g_\theta \cong \alpha_{\mathrm{app}} \mathcal{L}^g_{\mathrm{app}} + \alpha_{\mathrm{ad}} \mathcal{L}^g_{\mathrm{ad}} \tag{A.15}$$

As explained in Sections 4.3.1.1 and 4.3.1.2, the generative path does not have direct access to $\mathbf{I}_c$, and this is reflected in the likelihood p^g_θ in which the instances of $\mathbf{I}_c$ are ignored. Thus $\mathcal{L}^g_{\mathrm{app}}$ is only for reconstructing $\mathbf{I}_m$ in a deterministic auto-encoder fashion as per (4.11), while $\mathcal{L}^g_{\mathrm{ad}}$ in (4.13) only tries to enforce that the generated distribution be consistent with the training set distribution (hence without per-instance knowledge), as implemented in the form of a GAN.

Appendix B

Supplementary Material for Chapter 5

B.1 Additional Quantitative Results

We further report quantitative results using traditional pixel-level and patch-level image quality evaluation metrics.

	Method	CelebA-HQ			FFHQ		
		$\ell_1$loss ↓	SSIM↑	PSNR↑	$\ell_1$loss ↓	SSIM↑	PSNR↑
	CA [214]	0.0310	0.8201	23.5667	0.0337	0.8099	22.7745
	PICNet [227]	0.0209	0.8668	24.6860	0.0241	0.8547	24.3430
	MEDFE [81]	0.0208	0.8691	24.4733	-	-	-
A	Traditional *Conv*	0.0199	0.8693	24.5800	0.0241	0.8559	24.2271
B	+ Attention in G	0.0196	0.8717	24.6512	0.0236	0.8607	24.4384
C	+ Restrictive *Conv*	0.0191	0.8738	24.8067	0.0220	0.8681	24.9280
D	+ Transformer	0.0189	0.8749	24.9467	0.0197	0.8751	25.1002
E	+ Masked Attention	0.0183	0.8802	25.2510	0.0188	0.8765	25.1204
F	+ Refine Network	**0.0180**	**0.8821**	**25.4220**	**0.0184**	**0.8778**	**25.2061**

TABLE B.1: Quantitative results for traditional pixel-level and patch-level metrics on center masked images.

Table B.1 provides a comparison of our results to state-of-the-art CNN-based models, as well as various alternative configurations for our design, on the center masked face testing set. This is an extension of Table 5.2 in the main text. All images were normalized to the range [0,1] for quantitative evaluation. While there is no necessity to strongly encourage the completed images to be the same as the original ground-truth images, our TFill model nonetheless achieved better performance on these metrics too, including ℓ_1 loss, structure similarity index (SSIM) and peak signal-to-noise ratio (PSNR), suggesting that our TFill model is more capable of generating closer content to the original unmasked images.

	Size	GL [86]	CA [214]	PICNet [227]	HiFill [208]	TFill
$\ell_1$loss†	[0.01, 0.1]	0.0233	0.0241	0.0097	0.0195	**0.0093**
	(0.1, 0.2]	0.0346	0.0338	0.0164	0.0282	**0.0153**
	(0.2, 0.3]	0.0500	0.0471	0.0249	0.0390	**0.0231**
	(0.3, 0.4]	0.0659	0.0612	0.0348	0.0513	**0.0322**
	(0.4, 0.5]	0.0808	0.0753	0.0456	0.0657	**0.0422**
	(0.5, 0.6]	0.0945	0.0925	0.0641	0.0885	**0.0591**
SSIM⋆	[0.01, 0.1]	0.9150	0.9079	0.9634	0.9245	**0.9695**
	(0.1, 0.2]	0.8526	0.8447	0.9137	0.8603	**0.9253**
	(0.2, 0.3]	0.7672	0.7652	0.8520	0.7838	**0.8686**
	(0.3, 0.4]	0.6823	0.6906	0.7850	0.7057	**0.8063**
	(0.4, 0.5]	0.5987	0.6133	0.7119	0.6193	**0.7391**
	(0.5, 0.6]	0.5185	0.5322	0.6077	0.5137	**0.6428**
PSNR⋆	[0.01, 0.1]	28.4151	26.8452	32.2579	28.3955	**33.0585**
	(0.1, 0.2]	24.4074	23.1766	27.3320	24.5495	**28.0670**
	(0.2, 0.3]	21.3296	20.4427	24.4423	22.0604	**25.0951**
	(0.3, 0.4]	19.1118	18.6337	22.3238	20.1451	**22.8942**
	(0.4, 0.5]	17.5594	17.2978	20.7146	18.4715	**21.2200**
	(0.5, 0.6]	16.4831	16.0824	18.7234	16.4998	**19.1040**

TABLE B.2: Quantitative comparisons on Places2 [232] with free-form masks [123]. †Lower is better. ⋆Higher is better. Without bells and whistles, TFill outperformed all traditional CNN-based models.

Table B.2 provides a comparison of our results to state-of-the-art methods on the Places2 [232] testing set with free-form masks [123]. This is an extension of Table 5.1 in the main text. As we can see in Figure 5.12, while our TFill model does *not* generate the same content as the original unmasked images, it filled the masked holes with semantically appropriate content of consistent realistic appearance. There were no obvious artifacts when the completed pixels were recomposed with the original visible pixels, resulting in quite a significant improvement in image quality.

B.2 Experiment Details

Here we first present the novel layers and loss functions used to train our model, followed by the training details.

B.2.1 Multihead *Masked* Self-Attention

Our transformer encoder is built on the standard **qkv** `self-attention` (SA) [188] with a learned position embedding in each layer. Given an input sequence $\mathbf{z} \in \mathbb{R}^{N \times C}$, we first calculate the pairwise similarity $\mathbf{A}$ between each two elements as

follows:

$$[\mathbf{q}, \mathbf{k}, \mathbf{v}] = \mathbf{W}_{qkv}\mathbf{z} \tag{B.1}$$

$$\mathbf{A} = \texttt{softmax}(\mathbf{q}\mathbf{k}^\top / \sqrt{C_h}) \tag{B.2}$$

where $\mathbf{W}_{qkv} \in \mathbb{R}^{C \times 3C_h}$ is the learned parameter to refine the features $\mathbf{z}$ for the query $\mathbf{q}$, the key $\mathbf{k}$ and the value $\mathbf{v}$. $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the dot similarity of N tokens, which is scaled by the square root of feature dimension C_h. Then, we compute a weighted sum over all values $\mathbf{v}$ via:

$$\mathrm{SA}(\mathbf{z}) = \mathbf{A}\mathbf{v} \tag{B.3}$$

where the value z in the sequence is connected through their learned similarity A, rather than purely depending on a fixed learned weight w.

The multihead `self-attention` (MSA) is an extension of SA, in which H heads are run in parallel to get multiple attention scores and the corresponding projected results. Then we get the following function:

$$\mathrm{MSA}(\mathbf{z}) = [\mathrm{SA}_1(\mathbf{z}); \mathrm{SA}_2(\mathbf{z}); \ldots; \mathrm{SA}_h(\mathbf{z})] \tag{B.4}$$

To encourage the model to *bias* to the important visible values, we further modify the MSA with a *masked* self-attention layer, in which a masked weight is applied to scale the attention score $\mathbf{A}$. Given a feature $\mathbf{x}$ and the corresponding mask $\mathbf{m}$ (1 denotes visible pixel and 0 is masked pixel). The original partial convolution operation is operated as:

$$x' = \begin{cases} \mathbf{W}_p(\mathbf{x}_p \bigodot \mathbf{m}_p)\frac{1}{\sum(\mathbf{m}_p)} + b, & \text{if } \sum(\mathbf{m}_p) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{B.5}$$

$$m' = \begin{cases} 1, & \text{if } \sum(\mathbf{m}_p) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{B.6}$$

where $\mathbf{W}_p$ contain the convolution filter weights, b is the corresponding bias, while $\mathbf{x}_p$ and $\mathbf{m}_p$ are the feature values and mask values in the current convolution window (*e.g.* 2×2 in our *restrictive CNN*), respectively. Here, we replace the m' as a float value:

$$m' = \frac{\sum(\mathbf{m}_p)}{S} \tag{B.7}$$

where S is the size of each convolution filter, 2×2 used in our *restrictive CNN*. To do this, each token only extracts the visible information. What's more, the final m for each token denotes the percentage of valid values in each token under a small RF. Then, for each sequence $\mathbf{z} \in \mathbb{R}^{N \times C}$, we obtain a corresponding masked weight $\mathbf{m} \in \mathbb{R}^{N \times 1}$ by flattening the updated mask. Finally, we update the original attention score by multiplying with the repeated masked weight $\mathbf{m} \in \mathbb{R}^{N \times 1}$:

$$\mathbf{A}_m = \mathbf{A} \bigodot \mathbf{m}_r \tag{B.8}$$

where $\mathbf{m}_r \in \mathbb{R}^{N \times N}$ is the extension of masked weight $\mathbf{m} \in \mathbb{R}^{N \times 1}$ in the final dimension.

Appendix C

Supplementary Material for Chapter 6

C.1 Experimental Details

Training We trained our model on the synthetic data into three phases: 1) the layered scene decomposition network (Figure 6.3(b)) is trained with loss L_{decomp} for 24 epochs, where at each layer, re-composited layered ground-truths are used as input. 2) Separately, the completion network (Figure 6.3(c)) is trained with loss L_{comp} for 50 epochs, wherein the ground-truth layer orders and segmented masks are used to designate the *invisible* regions for completion. 3) Both decomposition and completion networks were trained jointly for 12 epochs, *without relying on ground-truths as input* at any layer (Figure 6.3(a)). Doing so allows the scene decomposition network to *learn to cope with flaws* (*e.g.* texture artifacts) in the scene completion network, and vice versa. For each scene, the iteration ends when no more objects are detected, or a maximum 10 iterations is reached.

The training on real data only involved phases 1 and 3, as no ground-truth appearances are available for the invisible parts. The layered decomposition network is trained only for one layer (original image) in phase 1 due to *no* re-composed ground-truth images. Since phase 3 does not rely on ground-truths as input, we trained it layer-by-layer on real images by providing the "pseudo ground truth" appearances to calculate the reconstruction loss. To reduce the effect of progressively introduced artifacts in image completion, we used bounding boxes detected in the first layer as proposals for remaining decomposition steps.

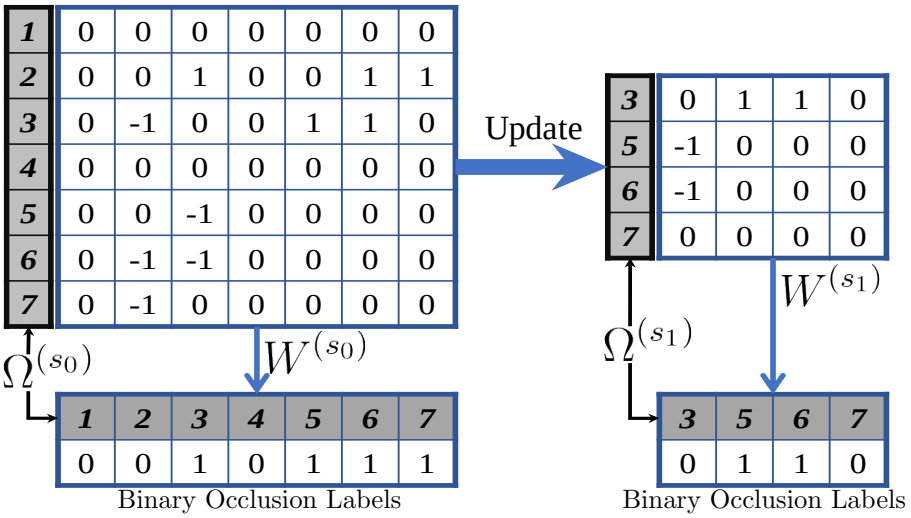


FIGURE C.1: **An illustration of obtaining the ground-truth binary occlusion labels from pairwise order graph $G = (\Omega, W)$ in each step s_k.** If the indegree of a vertex is 0, it will be labeled as 0, a fully visible instance. Otherwise, the instance will be labeled as 1, being occluded. When some objects are detected and selected out in the previous step, the object indexes and the corresponding occlusions will be eliminated.

FIGURE C.2: **Realistic rendered images in the CSD dataset** with various environment and lighting.

Inference During testing, fully visible instances were selected out and assigned an absolute layer order corresponding to the step index s_k. In each layer, the decomposition network selects the highest scoring 100 detected boxes for mask segmentation and non-occlusion predication. As we observed that higher object classification scores provided more accurately segmented boundaries, we only selected non-occluded objects with high object classification scores and non-occlusion scores (thresholds of 0.5 for synthetic images and 0.3 for real images) among these 100 candidates. We further observed that in some cases, we detected multiple objects with high object classification confidences, yet none were classified as fully visible due to low non-occlusion scores, especially in complex scenes with steps larger than 5. We will then choose the instance with the highest non-occlusion score so that *at least one object is selected at each layer*. When no objects are detected, the iteration stops.

Instance depth ordering update As illustrated in Figure C.1, we calculate the indegree $deg^-(\omega)$ (counts of -1) of each instance in the matrix. If $deg^-(\omega) = 0$, meaning no objects are in front of it, its binary occlusion label will be 0. Otherwise, the object is occluded, labeled as 1. At each step, the fully visible objects will be eliminated from the directed graph G, and the ground-truth binary occlusion labels will be updated in each step. So if the table (instance #2) was selected in the previous step, the vertex index Ω will be updated after the corresponding object ω_2 is deleted from the occlusion matrix.

C.2 Rendering Dataset

C.2.1 Data Rendering

Our **completed scene decomposition (CSD) dataset** was created using Maya [3], based on the SUNCG CAD models [177]. The original SUNCG dataset contains 45,622 different houses with realistically modeled rooms. As realistically rendering needs a lot of time (average 1 hour for each house), we only selected 2,456 houses in current work. The set of camera views was based on the original OpenGL-rendering method in SUNCG, but further filtered such that a camera view was only be picked when at least 5 objects appeared in that view. We then realistically rendered RGB images for the selected views. Eight examples are shown in Figure C.2 for various room types and lighting environments. Notice that our rendered images are much more realistic than the OpenGL rendered versions from the original SUNCG and likewise in [38].

To *visit the invisible*, the supervised method needs ground truth for the original occluded regions of each object. One possible way is to remove the fully visible objects in one layer and re-render the updated scene for the next layer, repeating this for all layers. However, during the training, the fully visible objects are not always correctly detected by the models. Thus, for more robust learning, we need to consider all different combinations of objects and the background. Given N objects, we would need to render 2^N images for each view. As we can see from the data statistics presented in Figure C.4, an average of 11 objects are visible in each view. Due to slow rendering, we do not have the capacity to render all such scenes (average $2^{11} = 2048$ images per view). Instead, we separately rendered each isolated object with the full RGB appearance, as well as the empty room.

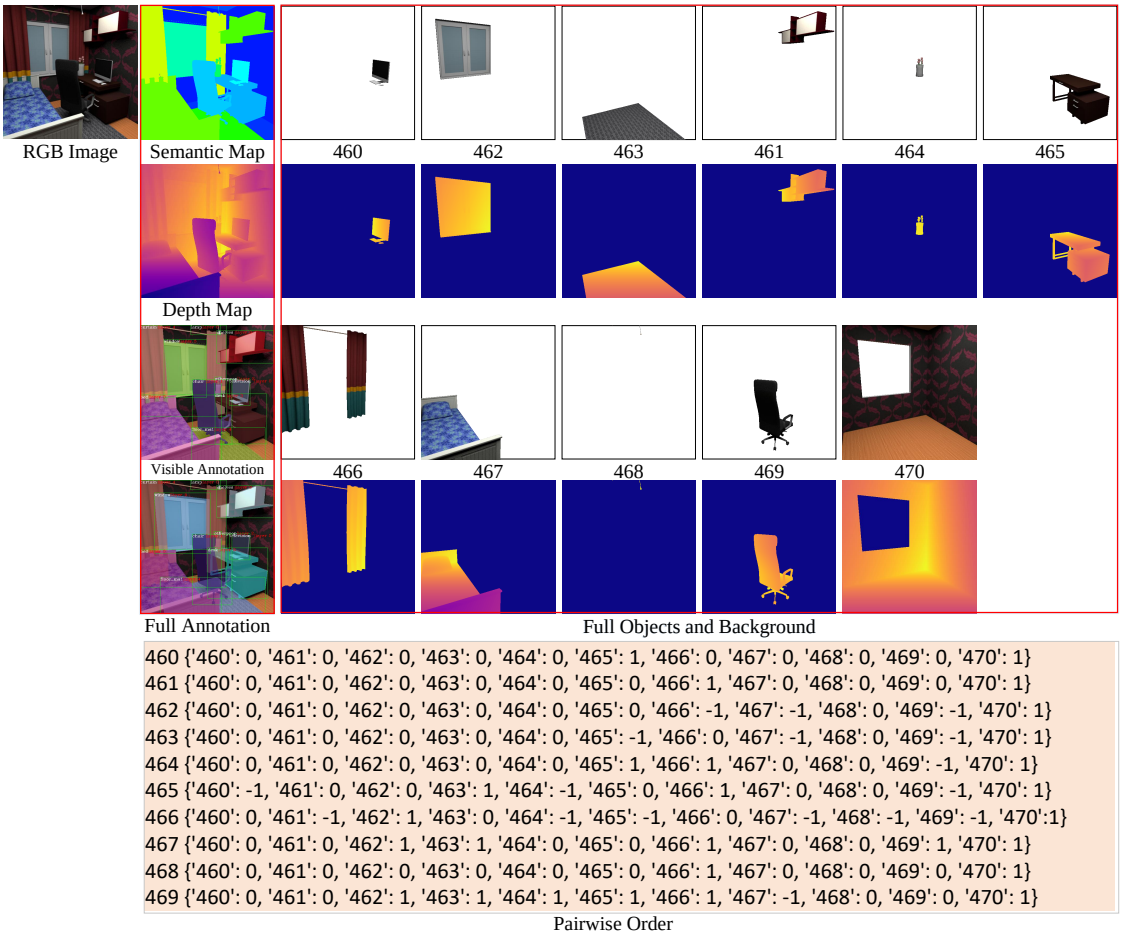
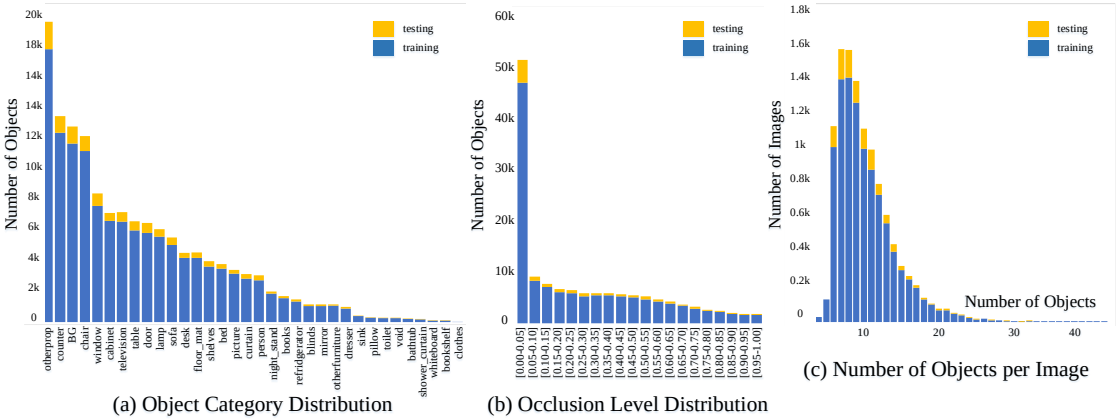


FIGURE C.3: **Illustration of Data Annotation.** For each rendered image, we have a corresponding semantic map, a depth map, and dense annotation, including class category, bounding box, instance mask, absolute layer order and pairwise order. In addition, for each object, we have a full RGBA image and depth map.

During training, the image of a scene is created by using a combination of the rendered images of these individual objects and the background to create a composed image, based on the remaining objects left after applying the scene decomposition network at each step. Since the room environment is empty for each individual objects during the rendering, the re-composited scenes have lower realism than the original scenes, due to missing shadows and lack of indirect illumination from other objects. In this project, we do not consider the challenges of working with shadows and indirect illumination, leaving those for future research.

C.2.2 Data Annotation

In Figure C.3, we show one example of a rendered image with rich annotations, consisting of a semantic map, a depth map, visible annotations and full (amodal)



(a) Object Category Distribution (b) Occlusion Level Distribution (c) Number of Objects per Image

FIGURE C.4: **Data Statistics.** Left: the object category distribution. Middle: the occlusion level distribution. Right: distribution of number of objects per image. On average there are 11 objects in each room.

annotations. For the semantic maps, we transferred the SUNCG class categories to NYUD-V2 40 categories so that this rendered dataset can be tested on real-world images. The depth map is stored in 16-bit format, with the largest indoor depth value at 20m. The class category and layer order (both absolute layer order and pairwise occlusion order) are included for visible annotations and full annotations. The visible annotations also contain the visible bounding-box offset and visible binary mask for each instance. Additionally, we also have the full (amodal) bounding-box offset and completed mask for each individual object.

Pairwise Occlusion Order The pairwise order for each object is a vector storing the occlusion relationship between itself and all other objects. We use three numbers $\{-1, 0, 1\}$ to encode the occlusion relationship between two objects — -1: occluded, 0 : no relationship, 1: front (*i.e.* occluding). As can be seen in Figure C.3, the computer (object number: #460) does not overlap the shelves (object number: #461), so the pairwise order is "0", indicating these two objects have no occlusion relationship. The computer is however on top of the desk (object number: #465), hence the pairwise order for $W_{460,465}$ is "1", and conversely the pairwise order for $W_{465,460}$ is "-1", representing that the desk is occluded by the computer.

C.2.3 Data Statistics

In total, there are 11,434 views encompassing 129,336 labeled object instances in our rendered dataset. On average, there are 11 individual objects per view. Among

these, 63.58% objects are partially occluded by other objects and the average occlusion ratio (average IoU between two objects) is 26.27%.

Object Category Statistics Figure C.4(a) shows the overall object category distribution in our CSD dataset. Overall, the distribution is highly similar to the object distribution of NYUD-V2 dataset [138], containing a diverse set of common furniture and objects in indoor rooms. "Other props" and "Other furniture" are atypical objects that do not belong in a common category. In particular, "Other props" are small objects that can be easily removed, while "Other furniture" are large objects with more permanent locations. Additionally, we merge floors, ceilings, and walls as "BG" in this work. If the user wants to obtain the separated semantic maps for these structures, these are also available.

Occlusion Statistics The occlusion level is defined as the fraction of overlapping regions between two objects (Intersection over Union, or IOU). We divide the occlusion into 20 levels from highly visible (denoted as [0.00-0.05] fraction of occlusion) to highly invisible (denoted as (0.95-1.00] fraction of occlusion), with 0.05 increment in the fraction of occlusion for each level. Figure C.4(b) shows the occlusion level in our dataset. In general, the distribution of occlusion levels is similar to the distribution in [237], where a vast number of the instances are slightly occluded, while only a small number of instances are heavily occluded.

Object Count Distribution Figure C.4(c) shows the distribution of the number of objects present per view. On average, there are more than 11 objects in each view. This supports the learning of rich scene contextual information for a completed scene decomposition task, instead of processing each object in isolation.

C.2.4 Data Encoding

After we get the views and corresponding dense annotations, we encode the data annotation to COCO format[1]. The annotations are stored using JSON, and the CSD API will be made available for visualizing and utilizing the rendered dataset. The JSON file contains a series of fields, including "categories", "images" and "annotations".

[1]http://cocodataset.org

List of Author's Publications

Conference Proceedings

- **Chuanxia Zheng**, Tat-Jen Cham, Jianfei Cai, "T^{2}net: Synthetic-to-realistic translation for solving single-image depth estimation tasks," in *Proceedings of the European Conference on Computer Vision (**ECCV**), 2018.*

- **Chuanxia Zheng**, Tat-Jen Cham, Jianfei Cai, "Pluralistic image completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (**CVPR**), 2019.*

- **Chuanxia Zheng**, Tat-Jen Cham, Jianfei Cai, "The Spatially-Correlative Loss for Various Image Translation Tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (**CVPR**), 2021.*

- Guoxian Song, Linjie Luo, Jing Liu, Chunpong Lai, **Chuanxia Zheng**, and Tat-Jen Cham, "Agilegan: Stylizing portraits by inversion-consistent transfer learning," in *ACM Transactions on Graphics (Proceedings of ACM **SIGGRAPH**), 2021.*

- Tianyi Zhang, Jingyi Yang, **Chuanxia Zheng**, Guosheng Lin, Jianfei Cai, Alex C Kot, "Task-in-all domain adaptation for semantic segmentation," in *IEEE Visual Communications and Image Processing (**VCIP**), 2019.*

Journals

- **Chuanxia Zheng**, Tat-Jen Cham, Jianfei Cai, "Pluralistic free-form image completion," *International Journal of Computer Vision (**IJCV**), 2021.*

- **Chuanxia Zheng**, Duy-Son Dao, Guoxian Song, Tat-Jen Cham, Jianfei Cai, "Visiting the invisible: layer-by-layer completed scene decomposition," *International Journal of Computer Vision (**IJCV**), 2021.*

Preprints

- **Chuanxia Zheng**, Tat-Jen Cham, Jianfei Cai, "Tfill: Image completion via a transformer-based architecture," in *(arXiv), 2021.*

Bibliography

[1] Youssef Alami Mejjati, Christian Richardt, James Tompkin, Darren Cosker, and Kwang In Kim. Unsupervised attention-guided image-to-image translation. In *Proceedings of the International Conference on Neural Information Processing Systems*, volume 31, pages 3693–3703, 2018. 38

[2] Matthew Amodio and Smita Krishnaswamy. Travelgan: Image-to-image translation by transformation vector learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8983–8992, 2019. 31, 32

[3] Autodesk Maya, 2019. `https://www.autodesk.com/products/maya/overview`. 20, 112, 149

[4] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 15535–15545, 2019. 32

[5] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017. 106, 110

[6] Kyungjune Baek, Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Hyunjung Shim. Rethinking the truly unsupervised image-to-image translation. *arXiv preprint arXiv:2006.06500*, 2020. 39

[7] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Processing of the 3rd International Conference on Learning Representations, ICLR 2015*, 2015. 84

[8] Coloma Ballester, Marcelo Bertalmio, Vicent Caselles, Guillermo Sapiro, and Joan Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE Transactions on Image Processing*, 10(8):1200–1211, 2001. 50, 52

[9] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Cvae-gan: Fine-grained image generation through asymmetric training. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2764–2773. IEEE, 2017. 51, 53, 60

[10] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (ToG)*, 28:24, 2009. 50, 51, 52, 64, 70, 71, 74, 75, 90

[11] Sagie Benaim and Lior Wolf. One-sided unsupervised domain mapping. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 752–762, 2017. 31, 32, 40

[12] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013. 2

[13] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pages 417–424. ACM Press/Addison-Wesley Publishing Co., 2000. 4, 49, 50, 52

[14] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, and Stanley Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*, 12(8):882–889, 2003. 50, 52

[15] Hervé Bourlard and Yves Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59(4):291–294, 1988. 2, 16

[16] Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. MONet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019. 109, 111, 114, 116, 122

[17] Yuanzhouhan Cao, Zifeng Wu, and Chunhua Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017. 14

[18] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 213–229. Springer, 2020. 83, 84, 85

[19] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. *arXiv preprint arXiv:2012.00364*, 2020. 84, 85

[20] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4974–4983, 2019. 106, 114, 116, 122, 123, 124, 126, 127

[21] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017. 106, 108, 110

[22] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *Proceedings of the International Conference on Machine Learning*, pages 1691–1703. PMLR, 2020. 83, 84, 85, 87, 97, 98, 99, 100

[23] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1511–1520, 2017. 30, 32, 39

[24] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 32, 35

[25] Xiaotian Chen, Yuwang Wang, Xuejin Chen, and Wenjun Zeng. S2r-depthnet: Learning a generalizable depth-specific structural representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3034–3043, 2021. 28

[26] Xinyuan Chen, Chang Xu, Xiaokang Yang, and Dacheng Tao. Attention-gan for object transfiguration in wild images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 164–180, 2018. 38

[27] Zeyuan Chen, Shaoliang Nie, Tianfu Wu, and Christopher G Healey. High resolution face completion with multiple controllable attributes via fully end-to-end progressive generative adversarial networks. *arXiv preprint arXiv:1801.07632*, 2018. 54

[28] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and

Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018. 32, 39

[29] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8188–8197, 2020. 39

[30] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of the European conference on computer vision (ECCV)*, pages 628–644. Springer, 2016. 135

[31] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Processing of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 24, 110

[32] Antonio Criminisi, Patrick Perez, and Kentaro Toyama. Object removal by exemplar-based inpainting. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–II. IEEE, 2003. 50, 52

[33] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13(9):1200–1212, 2004. 50, 52

[34] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3150–3158, 2016. 110

[35] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009. 30

[36] Ye Deng and Jinjun Wang. Image inpainting using parallel network. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 1088–1092. IEEE, 2020. 49

[37] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding.

arXiv preprint arXiv:1810.04805, 2018. 83, 84

[38] Helisa Dhamo, Nassir Navab, and Federico Tombari. Object-driven multi-layer scene decomposition from a single image. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 107, 108, 109, 111, 112, 114, 116, 124, 125, 149

[39] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014. 106

[40] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *Proceedings of the International Conference on Learning Representations*, 2017. 106

[41] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei Efros. What makes paris look like paris? *ACM Transactions on Graphics*, 31(4), 2012. 69, 70, 71, 74

[42] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Uszkoreit Jakob, and Houlsby Neil. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations*, 2020. 84, 85, 97, 98, 99

[43] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 658–666, 2016. 30, 32

[44] Kiana Ehsani, Roozbeh Mottaghi, and Ali Farhadi. SeGAN: Segmenting and generating the invisible. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6144–6153, 2018. 106, 107, 108, 109, 110, 111, 112, 114, 116, 117, 123, 124, 125, 127

[45] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2650–2658, 2015. 12, 22

[46] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 2366–2374, 2014. 12, 14, 21, 22, 23, 24, 25, 26

[47] S. M. Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S.

Morcos, Marta Garnelo, Avraham Ruderman, Andrei A. Rusu, Ivo Danihelka, Karol Gregor, David P. Reichert, Lars Buesing, Theophane Weber, Oriol Vinyals, Dan Rosenbaum, Neil Rabinowitz, Helen King, Chloe Hillier, Matt Botvinick, Daan Wierstra, Koray Kavukcuoglu, and Demis Hassabis. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018. 51

[48] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 83, 84, 85, 87, 97, 98, 99, 100, 134

[49] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 110

[50] Patrick Follmann, Rebecca Kö Nig, Philipp Hä Rtinger, Michael Klostermann, and Tobias Bö Ttger. Learning to see the invisible: End-to-end trainable amodal instance segmentation. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1328–1336. IEEE, 2019. 109, 127, 128

[51] Huan Fu, Bowen Cai, Lin Gao, Lingxiao Zhang, Cao Li, Zengqi Xun, Chengyue Sun, Yiyun Fei, Yu Zheng, Ying Li, et al. 3d-front: 3d furnished rooms with layouts and semantics. *arXiv preprint arXiv:2011.09127*, 2020. 2, 11

[52] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, Kun Zhang, and Dacheng Tao. Geometry-consistent generative adversarial networks for one-sided unsupervised domain mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2427–2436, 2019. 31, 32, 40

[53] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtualworlds as proxy for multi-object tracking analysis. In *Processing of the IEEE Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 4340–4349. IEEE, 2016. 20, 23, 24

[54] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Processing of the International Conference on Machine Learning (ICML)*, pages 1180–1189, 2015. 18

[55] Ravi Garg, Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *Proceedings*

of the European Conference on Computer Vision (ECCV), pages 740–756. Springer, 2016. 12, 14, 18, 24, 25

[56] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423. IEEE, 2016. 14, 32, 44

[57] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Processing of the IEEE Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012. 12, 20, 110, 120, 129

[58] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018. 6

[59] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *Proceedings of the European Conference on Computer Vision*, pages 484–499. Springer, 2016. 135

[60] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015. 110

[61] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014. 1, 106, 110

[62] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9785–9795, 2019. 135

[63] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recongition (CVPR)*, 2017. 12, 14, 18, 19, 24, 25

[64] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016. 1, 2

[65] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the International Conference on Neural Information Processing Systems*, volume 27, 2014. 3, 13, 17, 18, 30, 51, 53, 61,

90, 106, 111

[66] Stephen Gould, Richard Fulton, and Daphne Koller. Decomposing a scene into geometric and semantically consistent regions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–8, 2009. 106, 109

[67] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 5767–5777, 2017. 106

[68] Ruiqi Guo and Derek Hoiem. Beyond the line of sight: labeling the underlying surfaces. In *Proceedings of the European Conference on Computer Vision*, pages 761–774. Springer, 2012. 109, 110

[69] Takayuki Hara and Tatsuya Harada. Spherical image generation from a single normal field of view image by considering scene symmetry. *arXiv preprint arXiv:2001.02993*, 2020. 79

[70] James Hays and Alexei A Efros. Scene completion using millions of photographs. In *ACM Transactions on Graphics (TOG)*, volume 26, page 4. ACM, 2007. 52

[71] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 32, 35

[72] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017. 106, 108, 111, 114, 122, 123, 126, 127

[73] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, 2015. 106, 110

[74] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 19

[75] Philipp Heise, Sebastian Klose, Brian Jensen, and Alois Knoll. Pm-huber: Patchmatch with huber regularization for stereo matching. In *Processing of the IEEE International Conference on Computer Vision*, pages 2360–2367. IEEE, 2013. 18

[76] Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 32

[77] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 6626–6637, 2017. 6, 39, 40, 62, 69, 90, 94

[78] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *Proceedings of the International Conference on Learning Representations*, 2018. 32, 35

[79] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *Proceedings of the International Conference on Machine Learning*, pages 1989–1998, 2018. 12, 15, 32

[80] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Automatic photo pop-up. 24(3):577–584, 2005. 14

[81] Yibing Song Wei Huang Hongyu Liu, Bin Jiang and Chao Yang. Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In *Proceedings of the European Conference on Computer Vision*, 2020. 61, 93, 94, 143

[82] Yuan-Ting Hu, Hong-Shuo Chen, Kexin Hui, Jia-Bin Huang, and Alexander G Schwing. Sail-vos: Semantic amodal instance level video object segmentation-a synthetic dataset and baselines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3105–3115, 2019. 109

[83] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017. 41

[84] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 172–189, 2018. 31, 32, 40, 41, 42, 94, 133

[85] Drew A Hudson and C. Lawrence Zitnick. Generative adversarial transformers. *arXiv preprint*, 2021. 84

[86] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36(4):107, 2017. 50, 52, 55, 70, 71, 72, 73, 76, 78, 79, 90, 91, 92, 106, 111, 144

[87] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1125–1134, 2017. 4, 7, 17, 30, 39, 53, 69

[88] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 2017–2025, 2015. 53

[89] Jiaya Jia and Chi-Keung Tang. Inference of segmented color and texture description by tensor voting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):771–786, 2004. 50, 52

[90] Liming Jiang, Changxu Zhang, Mingyang Huang, Chunxiao Liu, Jianping Shi, and Chen Change Loy. Tsit: A simple and versatile framework for image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 32, 39

[91] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two transformers can make one strong gan. *arXiv preprint arXiv:2102.07074*, 2021. 84

[92] Youngjoo Jo and Jongyoul Park. Sc-fegan: Face editing generative adversarial network with user's sketch and color. *arXiv preprint arXiv:1902.06838*, 2019. 53

[93] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 694–711. Springer, 2016. 14, 30, 31, 32, 35, 36, 37, 38, 44, 90, 117

[94] Abhishek Kar, Shubham Tulsiani, Joao Carreira, and Jitendra Malik. Amodal completion and size constancy in natural scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 127–135, 2015. 109, 110

[95] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint*

arXiv:1710.10196, 2017. 61, 69, 74, 76, 77, 89, 91, 94

[96] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 31, 41, 44, 89, 94, 106

[97] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 31, 39, 44, 77, 97

[98] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth extraction from video using non-parametric sampling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 775–788. Springer, 2012. 14

[99] Dahun Kim, Donghyeon Cho, Donggeun Yoo, and In So Kweon. Learning image representations by completing damaged jigsaw puzzles. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 793–802. IEEE, 2018. 2

[100] Sunnie S. Y. Kim, Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich. Deformable style transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 39

[101] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1857–1865, 2017. 15, 30, 32

[102] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 10215–10224, 2018. 106

[103] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3, 51, 53, 57, 106, 111

[104] Rolf Köhler, Christian Schuler, Bernhard Schölkopf, and Stefan Harmeling. Mask-specific inpainting with deep neural networks. In *Proceedings of the German Conference on Pattern Recognition*, pages 523–534. Springer, 2014. 52

[105] Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich. Style transfer by relaxed optimal transport and self-similarity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10051–10060, 2019. 38, 42, 43, 44

[106] Yevhen Kuznietsov, Jörg Stückler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Processing of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6647–6655, 2017. 12, 14, 18, 19, 24

[107] L'ubor Ladický, Jianbo Shi, and Marc Pollefeys. Pulling things out of perspective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 89–96, 2014. 14, 22, 23

[108] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *Proceedings of the Fourth International Conference on 3D Vision (3DV)*, pages 239–248. IEEE, 2016. 12

[109] Barbara Landau, Linda B Smith, and Susan S Jones. The importance of shape in early lexical learning. *Cognitive Development*, 3(3):299–321, 1988. 6

[110] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. 1

[111] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 35–51, 2018. 32, 53

[112] Hsin-Ying Lee, Hung-Yu Tseng, Qi Mao, Jia-Bin Huang, Yu-Ding Lu, Maneesh Singh, and Ming-Hsuan Yang. Drit++: Diverse image-to-image translation via disentangled representations. *International Journal of Computer Vision*, pages 1–16, 2020. 32, 40, 41, 42

[113] Anat Levin, Assaf Zomet, and Yair Weiss. Learning how to inpaint from global image statistics. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, page 305. IEEE, 2003. 50, 52

[114] Ke Li and Jitendra Malik. Amodal instance segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 677–693. Springer, 2016. 107, 108, 109, 110

[115] Yijun Li, Sifei Liu, Jimei Yang, and Ming-Hsuan Yang. Generative face completion. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 5892–5900. IEEE, 2017. 52

[116] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2359–2367,

2017. 110

[117] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 135

[118] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017. 114

[119] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, pages 740–755. Springer, 2014. 110, 120, 121

[120] Huan Ling, David Acuna, Karsten Kreis, Seung Wook Kim, and Sanja Fidler. Variational amodal object completion. *Proceedings of the International Conference on Neural Information Processing Systems*, 33, 2020. 106, 109, 114, 116, 117

[121] Chen Liu, Pushmeet Kohli, and Yasutaka Furukawa. Layered scene decomposition via the Occlusion-CRF. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–173, 2016. 109, 110

[122] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(10):2024–2039, 2016. 12, 14, 22, 23, 25

[123] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 50, 52, 69, 70, 72, 73, 74, 75, 76, 79, 86, 89, 90, 91, 99, 144

[124] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Proceedings of the International Conference on Neural Information Processing Systems (NIPS)*, pages 700–708, 2017. 15

[125] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 469–477, 2016. 14

[126] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference*

on Computer Vision, pages 3730–3738, 2015. 61, 69, 74, 76, 77, 89, 91, 94

[127] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 106, 110, 116

[128] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4990–4998, 2017. 44

[129] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, and Zhen Wang. Multi-class generative adversarial networks with the l2 loss function. *CoRR, abs/1611.04076*, 2, 2016. 20

[130] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2813–2821. IEEE, 2017. 61

[131] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015. 53

[132] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. The contextual loss for image transformation with non-aligned data. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 768–783, 2018. 31, 32, 35, 36, 38

[133] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Processing of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 20, 24

[134] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *Proceedings of the International Conference on Machine learning (ICML)*, pages 3481–3490. PMLR, 2018. 44

[135] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 111

[136] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020. 2

[137] Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *Proceedings of the International Conference on Machine Learning*, 2020. 41,

42

[138] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *Proceedings of the European Conference on Computer Vision*, 2012. 113, 129, 130, 152

[139] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Structure guided image inpainting using edge prediction. In *Processing of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019. 7, 50, 53, 55, 61, 69, 70, 71, 72, 73, 74, 75, 76, 77, 79, 82, 111, 134

[140] Ori Nizan and Ayellet Tal. Breaking the cycle-colleagues are all you need. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7860–7869, 2020. 39

[141] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 32, 33, 134

[142] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *Processing of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 702–711. IEEE, 2017. 53

[143] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *Proceedings of the European Conference on Computer Vision*, 2020. 30, 31, 32, 33, 35, 36, 37, 38, 39, 40, 41, 44

[144] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019. 7, 32, 39, 69

[145] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. *arXiv preprint arXiv:2103.17249*, 2021. 134

[146] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016. 50, 52, 55, 70, 71, 79, 111, 116

[147] Jialun Peng, Dong Liu, Songcen Xu, and Houqiang Li. Generating diverse structure for image inpainting with hierarchical vq-vae. *arXiv preprint*

arXiv:2103.10022, 2021. 49, 134

[148] Pedro O Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 1990–1998, 2015. 1, 106, 110

[149] Pedro O Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. Learning to refine object segments. In *Proceedings of the European Conference on Computer Vision*, pages 75–91. Springer, 2016. 110, 127

[150] Tiziano Portenier, Qiyang Hu, Attila Szabo, Siavash Arjomand Bigdeli, Paolo Favaro, and Matthias Zwicker. Faceshop: Deep sketch-based face image editing. *ACM Transactions on Graphics (TOG)*, 37(4):99, 2018. 53

[151] Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Amodal instance segmentation with kins dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3014–3023, 2019. 108, 109, 110, 115, 120, 122, 123, 124, 127, 128, 129

[152] Weichao Qiu and Alan Yuille. Unrealcv: Connecting computer vision to unreal engine. In *Processing of the European Conference on Computer Vision*, pages 909–916. Springer, 2016. 12

[153] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 134

[154] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. Technical report, OpenAI, 2018. 83, 84

[155] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 83, 84

[156] René Ranftl, Katrin Lasinger, D Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 28

[157] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *arXiv preprint arXiv:1906.00446*, 2019. 98, 134

[158] Jimmy SJ Ren, Li Xu, Qiong Yan, and Wenxiu Sun. Shepard convolutional neural networks. In *Proceedings of the International Conference on Neural*

Information Processing Systems, pages 901–909, 2015. 52

[159] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 91–99, 2015. 106, 110, 114

[160] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Processing of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015. 19, 42

[161] Mihaela Rosca, Balaji Lakshminarayanan, David Warde-Farley, and Shakir Mohamed. Variational approaches for auto-encoding generative adversarial networks. *arXiv preprint arXiv:1706.04987*, 2017. 17

[162] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 69, 71, 72, 89, 92, 93, 96

[163] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 2234–2242, 2016. 6, 69

[164] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng. 3-d depth reconstruction from a single still image. *International Journal of Computer Vision*, 76(1):53–69, 2008. 14

[165] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–840, 2009. 12, 14, 25, 26

[166] Soumyadip Sengupta, Vivek Jayaram, Brian Curless, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Background matting: The world is your green screen. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2291–2300, 2020. 120

[167] Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. Layered depth images. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, pages 231–242, 1998. 108

[168] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4570–4580, 2019. 44,

51, 56

[169] Eli Shechtman and Michal Irani. Matching local self-similarities across images and videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 30

[170] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 30

[171] Assaf Shocher, Shai Bagon, Phillip Isola, and Michal Irani. Ingan: Capturing and retargeting the "dna" of a natural image. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 44

[172] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russ Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 12, 13, 15, 19, 26, 30, 32, 37, 53

[173] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Processing of the European Conference on Computer Vision (ECCV)*, pages 746–760. Springer, 2012. 12, 20, 22, 27, 110

[174] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations*, May 2015. 1, 30, 32, 33, 39, 61, 117

[175] Paul Smolensky. Information processing in dynamical systems: Foundations of harmony theory. Technical report, Colorado Univ at Boulder Dept of Computer Science, 1986. 2

[176] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 3483–3491, 2015. 51, 53, 55, 56, 58, 61, 70, 137

[177] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1746–1754, 2017. 2, 11, 20, 112, 149

[178] Yuhang Song, Chao Yang, Zhe Lin, Xiaofeng Liu, Qin Huang, Hao Li, and CC Jay. Contextual-based image inpainting: Infer, match, and translate. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages

3–19, 2018. 53, 56, 64, 77, 89

[179] Yuhang Song, Chao Yang, Yeji Shen, Peng Wang, Qin Huang, and C-C Jay Kuo. Spg-net: Segmentation prediction and guidance network for image inpainting. *arXiv preprint arXiv:1805.03356*, 2018. 52

[180] Hao Su, Charles R Qi, Yangyan Li, and Leonidas J Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2686–2694, 2015. 2

[181] Deqing Sun, Erik B Sudderth, and Michael J Black. Layered image motion with explicit occlusions, temporal consistency, and depth ordering. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 2226–2234, 2010. 109, 115

[182] Joseph Tighe, Marc Niethammer, and Svetlana Lazebnik. Scene parsing with object instances and occlusion ordering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3748–3755, 2014. 109, 115

[183] Alessio Tonioni, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano. Unsupervised adaptation for deep stereo. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1605–1613, 2017. 14

[184] Antonio Torralba, Rob Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008. 98

[185] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6924–6932, 2017. 32, 41

[186] Arash Vahdat and Jan Kautz. NVAE: A deep hierarchical variational autoencoder. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2020. 106

[187] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6309–6318, 2017. 98, 106

[188] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you

need. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 5998–6008, 2017. 83, 84, 85, 99, 100, 144

[189] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 51, 53, 55, 56, 61, 62, 63, 70, 139

[190] Ziyu Wan, Jingbo Zhang, Dongdong Chen, and Jing Liao. High-fidelity pluralistic image completion with transformers. *arXiv preprint arXiv:2103.14031*, 2021. 49, 102, 133

[191] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018. 135

[192] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille. Towards unified depth and semantic prediction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2800–2809, 2015. 14

[193] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018. 32, 39, 41

[194] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern recognition*, pages 7794–7803, 2018. 84

[195] Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Image inpainting via generative multi-column convolutional neural networks. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 331–340, 2018. 52

[196] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 6

[197] John Winn and Jamie Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 37–44. IEEE, 2006. 109

[198] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Masayoshi Tomizuka, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision. *arXiv preprint arXiv:2006.03677*, 2020. 84, 85

[199] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 32

[200] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International Conference on Machine Learning*, pages 2048–2057, 2015. 30

[201] Xiaosheng Yan, Feigege Wang, Wenxi Liu, Yuanlong Yu, Shengfeng He, and Jia Pan. Visualizing the invisible: Occluded vehicle segmentation and recovery. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7618–7627, 2019. 109, 111

[202] Zhaoyi Yan, Xiaoming Li, Mu Li, Wangmeng Zuo, and Shiguang Shan. Shift-net: Image inpainting via deep feature rearrangement. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 53, 64, 71, 89

[203] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *International Conference on Computer Vision (ICCV)*, October 2021. 131

[204] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 3, 2017. 53, 77, 111

[205] Yi Yang, Sam Hallman, Deva Ramanan, and Charless Fowlkes. Layered object detection for multi-class segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3113–3120, 2010. 106, 109

[206] Yi Yang, Sam Hallman, Deva Ramanan, and Charless C Fowlkes. Layered object models for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1731–1743, 2011. 108, 109, 111

[207] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark

Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *Proceedings of the Computer Vision and Pattern Recognition (CVPR)*, pages 6882–6890. IEEE, 2017. 52

[208] Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. Contextual residual aggregation for ultra high-resolution image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7508–7517, 2020. 50, 53, 64, 66, 77, 79, 82, 84, 87, 89, 90, 91, 92, 144

[209] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2849–2857, 2017. 15, 30, 31, 32

[210] Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha. Photorealistic style transfer via wavelet transforms. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9036–9045, 2019. 44

[211] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4651–4659, 2016. 1

[212] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *Processing of the International Conference on Learning Representations(ICLR)*, 2016. 19

[213] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 472–480, 2017. 39

[214] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5505–5514, 2018. 50, 51, 53, 55, 61, 64, 66, 67, 70, 71, 72, 73, 74, 75, 76, 77, 79, 82, 84, 87, 88, 89, 90, 91, 92, 93, 94, 100, 101, 106, 111, 143, 144

[215] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4471–4480, 2019. 53, 77, 79

[216] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern*

Recognition, pages 3712–3722, 2018. 125

[217] Yu Zeng, Zhe Lin, Huchuan Lu, and Vishal M Patel. Image inpainting with contextual reconstruction loss. *arXiv preprint arXiv:2011.12836*, 2020. 82

[218] Yu Zeng, Zhe Lin, Jimei Yang, Jianming Zhang, Eli Shechtman, and Huchuan Lu. High-resolution image inpainting with iterative confidence feedback and guided upsampling. In *Proceedings of the European Conference on Computer Vision*, pages 1–17. Springer, 2020. 77

[219] Xiaohang Zhan, Xingang Pan, Bo Dai, Ziwei Liu, Dahua Lin, and Chen Change Loy. Self-supervised scene de-occlusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3784–3792, 2020. 106, 107, 108, 109, 110, 111, 112, 114, 117, 118, 122, 123, 124, 125, 127, 128, 129

[220] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *Proceedings of the International Conference on Machine Learning*, pages 7354–7363. PMLR, 2019. 63, 66, 67, 84, 88, 100

[221] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Proceedings of the European Conference on Computer Vision*, pages 649–666. Springer, 2016. 7, 69, 75

[222] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 6, 32, 41, 42, 61, 90, 94, 96

[223] Ziyu Zhang, Alexander G Schwing, Sanja Fidler, and Raquel Urtasun. Monocular object instance segmentation and depth ordering with cnns. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2614–2622, 2015. 106, 109

[224] Lei Zhao, Qihang Mo, Sihuan Lin, Zhizhong Wang, Zhiwen Zuo, Haibo Chen, Wei Xing, and Dongming Lu. Uctgan: Diverse image inpainting based on unsupervised cross-space translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5741–5750, 2020. 49

[225] Shanshan Zhao, Huan Fu, Mingming Gong, and Dacheng Tao. Geometry-aware symmetric domain adaptation for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9788–9798, 2019. 28

[226] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018. 2, 7, 11, 53

[227] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic image completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1438–1447, 2019. 7, 49, 69, 84, 87, 88, 89, 90, 91, 92, 93, 94, 97, 100, 101, 106, 111, 117, 124, 125, 126, 143, 144

[228] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. The spatially-correlative loss for various image translation tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 7

[229] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Tfill: Image completion via a transformer-based architecture, 2021. 6, 7

[230] Chuanxia Zheng, Duy-Son Dao, Guoxian Song, Tat-Jen Cham, and Jianfei Cai. Visiting the invisible: Layer-by-layer completed scene decomposition, 2021. 7

[231] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H.S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers, 2020. 83, 84, 85

[232] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018. 69, 73, 75, 76, 78, 89, 90, 91, 144

[233] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *Proceedings of the European Conference on Computer Vision*, pages 286–301. Springer, 2016. 53

[234] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2223–2232, 2017. 4, 7, 15, 17, 19, 21, 27, 30, 31, 32, 38, 39, 40, 53, 69

[235] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Proceedings of the International Conference on Neural Information*

Processing Systems, pages 465–476, 2017. 7, 26, 32, 38, 39, 40, 41, 42, 53, 61, 62, 69, 94, 133

[236] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 83, 84, 85

[237] Yan Zhu, Yuandong Tian, Dimitris Metaxas, and Piotr Dollár. Semantic amodal segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1464–1472, 2017. 108, 109, 110, 120, 124, 127, 128, 129, 133, 152