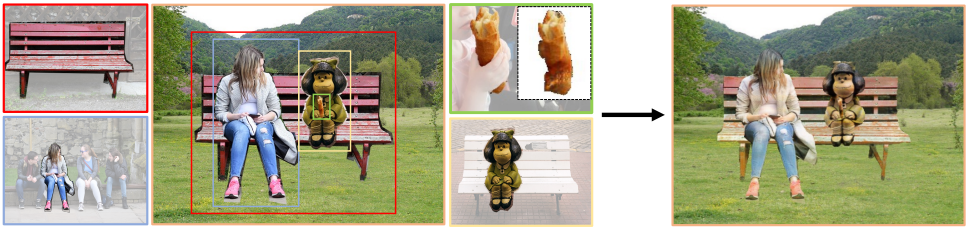
AMICO: Amodal Instance Composition

Peiye Zhuang[1]
peiye@illinois.edu
Jia-Bin Huang[23]
jbhuang@fb.com
Ayush Saraf[3]
ayush29feb@fb.com
Xuejian Rong[3]
xrong@fb.com
Changil Kim[3]
changil@fb.com
Denis Demandolx[3]
denisd@fb.com

[1] University of Illinois Urbana-Champaign
[2] University of Maryland, College Park
[3] Facebook



(a) Compositing components (b) Composed image

Figure 1: **Composition with imperfect instances.** Our method takes an ordered collection of imperfect instances (i.e., partially occluded and/or coarsely cropped) from multiple source images (instances identified by colored bounding boxes) and a background image as inputs *(a)*, and produces harmonized composition *(b)*. We achieve this via a unified framework that estimates the precise shape and content of each object and adjusts the object appearances to be mutually compatible.

Abstract

Image composition aims to blend multiple objects to form a harmonized image. Existing approaches often assume precisely segmented and intact objects. Such assumptions, however, are hard to satisfy in unconstrained scenarios. We present Amodal Instance Composition for compositing imperfect—potentially incomplete and/or coarsely segmented—objects onto a target image. We first develop object shape prediction and content completion modules to synthesize the amodal contents. We then propose a neural composition model to blend the objects seamlessly. Our primary technical novelty lies in using separate foreground/background representations and blending mask prediction to alleviate segmentation errors. Our results show state-of-the-art performance on public COCOA and KINS benchmarks and attain favorable visual results across diverse scenes. We demonstrate various image composition applications such as object insertion and de-occlusion.

1 Introduction

Image composition is a classic photo editing task that combines color-inconsistent objects from multiple source images into one composite image. Most existing approaches often assume intact (without occlusion) and precisely segmented instances [3, 8, 10, 26]. Such instances, however, may be challenging to obtain in real-world scenarios due to complex object shapes and occlusions in an image, e.g., bread in Figure 1(a).

We introduce the Amodal Instance Composition problem: compositing *imperfect* object instances (e.g., coarsely segmented or with incomplete shape/appearance due to occlusion) onto a target background image. The amodal instance composition poses several novel challenges for conventional image composition problems. First, given an object instance under occlusion, we need to perform amodal segmentation, estimating the object's full spatial extent beyond its visible regions, and then complete (hallucinate) the occluded regions of the selected object instance. Prior amodal instance completion methods [28, 57] directly applied classic image inpainting approach [30] for object completion. However, amodal object completion is different from image inpainting due to complex occlusion relationships, object shapes, and materials [2, 7, 44]. As a result, the prior methods [28, 57] often produce unrealistic results for object content completion.

Second, we need to adjust the appearance of the (completed) object instances to make them compatible with the background. When existing amodal instance completion methods [28, 57] demonstrate image manipulation tasks, e.g., to insert (completed) amodal instances to a new background, we observe that these methods take no consideration for color consistency. Unavoidably, they result in unrealistic composite images when the instances have distinct colors from the background. Moreover, the current composition methods [3, 8, 10, 26] often produce visible artifacts when imprecise masks are used as input.

In this paper, we present a fully automatic system to tackle the Amodal Instance Composition problem. Our method consists of three main modules tailored explicitly for addressing the above challenges. 1) *Object content completion*: Our object content completion module uses amodal and visible masks to synthesize the appearance of the missing regions. Our critical insight here is to use visible object regions only (instead of using the entire image as input). 2) *Image composition*: In contrast to prior harmonization methods that use a single image (with object instance copy-and-pasted onto the background) as input, our model takes a *separate* background image and object instance as inputs and produces RGBA (color and opacity) layers describing the appearance-adjusted object instance. 3) *Amodal mask prediction*: An amodal mask prediction module is trained offline and applied during inference to detect and recognize the occluded region of a given object. The estimated missing region will then be completed by our object content completion module. We validate that our proposed design leads to favorable performance on the publicly available COCOA and KINS datasets.

We summarize our main contributions as follows.

- We introduce the amodal instance composition task and present a learning-based system and the corresponding training strategies.
- We show favorable results against existing approaches on representative benchmarks and demonstrate various practical applications of amodal image composition.

2 Related work

Image composition seeks to compose and blend objects from multiple source images such that the new composite image appears photorealistic by harmonizing the colors of the fore-

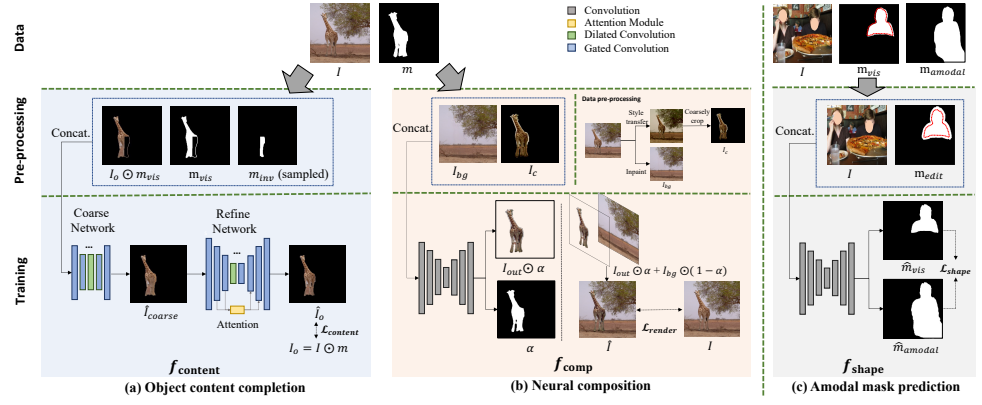


Figure 2: **Overall framework composed of three modules.** (a) An object content completion net f_{content} targets to generate missing regions for an object given triplet images as input, including a masked object $I_o \odot m_{vis}$ and two mutually exclusive binary masks m_{vis} and m_{inv} marking the *visible* and *invisible* region of the object ("$\odot$" denotes element-wise product). The generator in f_{content} has two stages and a discriminator (not shown due to limited space) is used for training the generator. (b) A neural composition net f_{comp} takes as input a background image I_{bg} and edited object I_c (refer to Sections 3-4 for details) and predicts RGBA layers, I_{out} and α, for the object. With the estimated alpha map α, we obtain the combined image, $\hat{I}$, via alpha-blending. (c) An amodal mask prediction net f_{shape} (trained offline) takes as input an image I and a mask m_{edit} that marks the visible region of an object in I.

ground instances. Earlier work uses transparency maps [35] or performs linear blending over multiple frequency bands [5, 6]. These methods, however, do not handle scenarios where the appearances of the object instances are not *compatible* with the background. To harmonize the composites, previous methods use color matching techniques such as applying color gradient-domain compositing [22, 34, 42] and statistical features [21, 37, 47]. Data-driven approaches (e.g., [19]) retrieve images with similar layouts from a large-scale database for compositing. Recent learning-based image composition methods demonstrate favorable performance [10, 39, 43, 54, 58]. Our work also focuses on learning-based color-consistent harmonization. Unlike existing work that uses a single composed image as input, we show that using *layered inputs* (e.g., separate foreground/background images) helps boost the harmonization quality for imperfect inputs.

The mask refinement step in our proposed composition module is also relevant to **image matting**. Concretely, the image matting task estimates an accurate alpha matte that separates the foreground instance from the background given manually created trimaps by users [1, 23, 24, 41, 46]. In contrast, our proposed approach takes as input imperfect instances and a background image (i.e., no trimaps) and produces RGBA layers with the aim of photorealistic composition.

Another line of research focuses on automatically placing an instance into a target background in a *geometrically consistent* manner by applying geometric transformations on the selected instance [26, 29, 56]. Our work focuses on color consistency instead.

Image completion focuses on synthesizing missing contents within one target image. Early methods search the good matching patches from valid image regions [4, 11, 12, 17] or large-scale datasets [16] to complete the holes. Recent deep learning-based approaches [18, 30, 32, 33, 48, 49, 53, 55, 59] build upon the Context Encoder approach [33], which extends early

Figure 3: **Samples of object completion for occluded object instances.** From left to right: original images from COCO test dataset with available visible and amodal masks annotated by [60] (*Column 1*); objects with occluded region in yellow and background region in white (*Column 2*); results from the baseline methods and ours (*Columns 3–7*) .

CNN-based inpainting to large masks using Generative Adversarial Networks [14]. Among which, partial convolution [30] and gated convolution [53] are proposed to address the issues of visual artifacts using vanilla convolutions. Further, auxiliary semantic information is leveraged to enhance inpainting performance such as edges [32], segmentation [25, 40] and foreground object contours [45]. In [28, 57], a partial convolution network [30] was employed to synthesize the *appearance* of the occluded content given predicted object amodal masks (the task also defined as **amodal instance completion**). Consequently, we compare our object content completion network with several representative image inpainting methods [30, 32, 49, 53] and amodal instance completion methods [28, 57] in Section 4.

3 Our Method

We address the problem of composing multiple input images with undesirable properties, such as, those that are incomplete, coarsely cropped, occluded, or having inconsistent colors. For this, we introduce an object content completion network f_{content} (Section 3.1), a neural compositing network f_{comp} (Section 3.2), and an amodal mask prediction network f_{shape} (Section 3.3). We show the overall framework, training strategies and inference procedures in Figure 2 and describe them accordingly.

To begin with, we denote an instance segmentation dataset as $\mathcal{D} = \{(I^{(i)}, m^{(i)}, m^{(i)}_{amodal})\}_{i=1}^{N}$, where $I^{(i)} \in \mathbb{R}^{H\times W\times 3}$ refers to an image, and $m^{(i)}$ and $m^{(i)}_{amodal} \in \{0,1\}^{H\times W\times 1}$ are two binary masks that mark the visible and the intact (i.e., both the visible and the invisible) region of an object in $I^{(i)}$, respectively. N is the size of the dataset. To avoid clutter, we use simplified notations, I, m, and m_{amodal} in the following sections.

3.1 Object content completion network

One critical step to compose occluded objects into a new background image is to complete the appearance of the invisible regions. Since there are no available paired images of the same object with and without occlusion, we manually construct our own paired data to train an object completion network, f_{content}, in a self-supervised manner.

Concretely, we mask out part of an object and train f_{content} to recover the missing content. As shown in Figure 2 *(a)*, we separate the object mask m into two mutually exclusive parts: 1) a visible mask m_{vis}, and 2) an invisible mask m_{inv}, where $m = m_{vis} \cup m_{inv}$. The input of f_{content} is a triplet consisting of a masked object image $I_o \odot m_{vis}$, and the two binary masks m_{vis} and m_{inv}, where I_o refers to the instance image, defined as $I_o = I \odot m$. To obtain m_{inv}, we

randomly sample a mask from dataset $\mathcal{D}$ that has overlapped with m. The overlapped region is captured by m_{inv} with the overlapping ratio $\frac{\|m_{inv}\|_1}{\|m\|_1} \in [0.1, 0.9]$.

Our object completion network f_{content} consists of a two-state generator G and a discriminator D, inspired by image inpainting techniques [49, 52, 53]. We use a discriminator D to distinguish the input image source as either real or synthetic. In the generator G, a coarse network produces a rough completion result, notated as $\hat{I}_{coarse}$, and a refine network generates a finer completed image, namely $\hat{I}_o$. More detailed model architecture is presented in the supplementary. Therefore, we train the generator G with the loss function $\mathcal{L}_{\text{content}}$:

$$\mathcal{L}_{\text{content}} = \lambda_1 \mathcal{L}_{\text{refine}} + \lambda_2 \mathcal{L}_{\text{coarse}} + \lambda_3 \mathcal{L}_{\text{adv}}, \tag{1}$$

where λ_1, λ_2, and λ_3 are coefficients of the loss terms.

Firstly, $\mathcal{L}_{\text{refine}}$ is the object reconstruction loss for $\hat{I}_o$:

$$\mathcal{L}_{\text{refine}} = \mathbb{E}_{(I_o,m)\sim\mathcal{D},\hat{I}_o\sim\mathcal{D}'|(I_o,m)}[\|\hat{I}_o - I_o\|_1 \odot m_{inv} + \beta\|\hat{I}_o - I_o\|_1 \odot m_{vis}]. \tag{2}$$

where, $\beta = 5$. We refer to the distribution of $\hat{I}_o$ via $\mathcal{D}'$ conditioned on (I,m), formally written as $\mathcal{D}'|(I,m)$. $\mathcal{L}_{\text{coarse}}$ is the same reconstruction loss except computing with $\hat{I}_{coarse}$.

We use the WGAN-GP [15] loss to update both G (via $\mathcal{L}_{\text{adv}}$) and D (via $\mathcal{L}_{\text{d}}$):

$$\mathcal{L}_{\text{adv}} = -\mathbb{E}_{\hat{I}_o\sim\mathcal{D}'|(I_o,m)}[D(\hat{I}_o)], \tag{3}$$

$$\mathcal{L}_{\text{d}} = \mathbb{E}_{\hat{I}_o\sim\mathcal{D}'|(I_o,m)}[D(\hat{I}_o)] - \mathbb{E}_{I_o\sim\mathcal{D}}[D(I_o)] + \sigma_1 \mathbb{E}_{\hat{I}_o\sim\mathcal{D}'|(I_o,m)}[\,\|\nabla_{\hat{I}_o} D(\hat{I}_o)\|_2 - 1\,]^2, \tag{4}$$

where $\sigma_1 = 10$ is a weight for the gradient penalty term. D and G are trained alternatively.

Figure 4: **Visual comparison of object composition on synthetic examples.** We first coarsely crop the object and apply color transfer to simulate different lighting conditions *(Column 2)*. Then we compose the objects onto the *same* inpainted background images shown at *Column 1* (thus the ground truth images are available for evaluation). Our method produces more plausible compositions than prior approaches.

3.2 Neural compositing network

We propose a neural compositing network, f_{comp}, shown in Figure 2 *(b)*, to blend multiple objects into a single coherent image. The composition network f_{comp} should be robust to objects that could be imperfectly cropped and have inconsistent appearances with the background image. For this, f_{comp} takes a background image I_{bg} and an edited object I_c as input and generates RGBA layers, including I_{out} and α, for the object. With the output layers, we obtain a reconstructed image $\hat{I}$ through standard alpha blending. In this case, I_{out} contains the color-transferred object with its appearance close to the background and an α map helps refine the object shape. We introduce the implementation details in following subsections.

Data preparation. We train f_{comp} in a self-supervised manner. For this, we employ two off-the-shelf modules: 1) an *image inpainting module* [49], denoted by Inpainting$(\cdot)$, for background completion; and 2) a *color transfer module* [50], denoted by ColorTransfer$(\cdot)$, for foreground color modification. Formally, we have

$$I_{bg} = \text{Inpaint}(I \odot (1-m), m), I_c = \text{ColorTransfer}(I, I_{ref}) \odot \text{Dilate}(m, iter). \tag{5}$$

Inpainting$(\cdot)$ completes the masked background region marked in m, and the ColorTransfer$(\cdot)$ transfers colors from a randomly sampled reference image $I_{ref} \in \mathcal{D}$ to the target I. Dilate$(\cdot)$ simulates the coarsely cropping step by randomly enlarging the cropped region for the object with $iter \sim \mathcal{U}\{0, \cdots, 25\}$ pixels.

Model optimization . As we obtain the compositing result $\hat{I}$ via alpha blending:

$$\hat{I} = I_{out} \odot \alpha + I_{bg} \odot (1-\alpha),$$

we use three loss terms to optimize f_{comp}: a reconstruction loss $\mathcal{L}_{\text{recon}}$, a mask loss $\mathcal{L}_{\text{mask}}$, and a regularization loss $\mathcal{L}_{\text{reg}}$ for α.

First, $\mathcal{L}_{\text{recon}}$ assesses how well the neural compositing model reconstructs I. We express it via ℓ_1 loss:

$$\mathcal{L}_{\text{recon}} = \mathbb{E}_{(I,m)\sim\mathcal{D}} \|\hat{I} - I\|_1. \tag{6}$$

Second, a mask loss $\mathcal{L}_{\text{mask}}$ on the α layer is used to encourage the learned α layer to match the exact object segment. Formally, we have

$$\mathcal{L}_{\text{mask}} = \mathbb{E}_{(I,m)\sim\mathcal{D}} \frac{\|m \odot (1-\alpha)\|_1}{2\|m\|_1} + \frac{\|(1-m) \odot \alpha\|_1}{2\|1-m\|_1}. \tag{7}$$

Last, inspired by [31], we apply a regularization loss, $\mathcal{L}_{\text{reg}}$, consisting of an L_1 norm and an approximation L_0 norm, to encourage α to be spatially sparse:

$$\mathcal{L}_{\text{reg}} = \mathbb{E}_{(I,m)\sim\mathcal{D}} \gamma \|\alpha\|_1 + 2 \cdot \text{Sigmoid}(5 \cdot \alpha) - 1, \tag{8}$$

where γ controls the relative weight ratio between the two terms.

Hence, the total loss of the neural compositing model is

$$\mathcal{L}_{\text{comp}} = \mathcal{L}_{\text{recon}} + \lambda_4 \mathcal{L}_{\text{mask}} + \lambda_5 \mathcal{L}_{\text{reg}}, \tag{9}$$

where λ_4 and λ_5 are loss weights. We approximate all aforementioned expectations by empirical sampling.

3.3 Amodal mask prediction network

We employ an amodal mask prediction network, defined as f_{shape}, that is trained offline and applied during inference to predict the visible and the intact regions of an object, defined as $\hat{m}_{vis}$ and $\hat{m}_{amodal}$, respectively. The estimated missing region is then completed by the content completion network f_{content}. As shown in Figure 2 *(c)*, the input of the amodal mask prediction network, f_{shape}, is an image I and a binary m_{edit} that roughly indicates the visible region. We obtain m_{edit} from m_{vis} via editing (e.g., dilation and erosion) to increase the robustness of f_{shape} for imperfect inference cases where the mask of the visible area may not be accurate. We formulate the loss function for optimizing f_{shape} as follows:

$$\mathcal{L}_{\text{shape}} = \mathbb{E}_{\mathcal{D}} \ \ell(\hat{m}_{vis}, m_{vis}) + \ell(\hat{m}_{amodal}, m_{amodal}), \tag{10}$$

where ℓ is a weighted binary cross-entropy (BCE) loss computed on both regions inside and outside of the object area:

$$\ell(m_1, m_2) = \omega \cdot \text{BCE}(m_1 \odot m_2, m_2) + \text{BCE}((1-m_1) \odot (1-m_2), 1-m_2), \tag{11}$$

where $\omega = 5$ and $\odot$ represents element-wise product.

Original De-occ. [57] Ours (w/o composition) Ours

Figure 5: **Visual comparison of amodal instance composition.** We show the scene manipulation results for an amodal object: the original object and its ground truth amodal mask *(Column 1)*, the amodal object completion and insertion by [57] (*Column 2*), ours without using the composition net f_{comp} (*Column 3*), and ours via the full method (*Column 4*).

3.4 Model inference

Given a background image I_{bg}, and object images $X = \{x^{(1)}, x^{(2)}, ..., x^{(M)}\}$, where $x^{(j)} = (I^{(j)}, m^{(j)}_{vis})$, $j \in \{1, 2, ..., M\}$ and $M = |X|$. Note that $m^{(j)}_{vis}$ is not required to be precise during inference. For occluded objects (if any), f_{shape} and f_{content} are applied in sequence to predict and synthesize the invisible regions. Afterwards, the objects are composed with the background image iteratively by f_{comp}, and the background image is progressively updated with each object composition. More detailed algorithms are presented in the supplementary.

4 Experimental results

Datasets. We use the COCOA dataset [60] and KINS [36] dataset for our evaluation. The COCOA dataset [60] contains amodal segmentation annotations for 5,000 images from MS COCO 2014 dataset [27]. We follow the official data split: 22,163 instances from 2,500 images for training, and 12,753 instances from 1,323 images for validation. The KINS dataset [36] was derived from KITTI [13], and contains 95,311 instances from 7,474 images for training, and 92,492 instances from 7,517 images for testing.

Implementation details. We note that our three modules could fit with a wide range of backbone architectures. In practice, we adopt U-Net [33] as the backbone of f_{shape}, f_{comp}, and the coarse net of f_{content}. Gated convolution layers [53] and dilated convolution layers [51] are applied in f_{content}. We also employ the contextual attention module from [52] in the refine-net of f_{content} learning to focus on object areas. Architecture details are presented in the supplementary. We choose $\lambda_1 = \lambda_2 = 1.2$, and $\lambda_3 = 10^{-3}$ to optimize f_{content}. For f_{comp}, we use $\lambda_4 = 50$ for the first 150K iterations and decrease it to 5 subsequently. Since instances in COCOA dataset [60] is annotated in polygons, i.e., the annotations only approximate and may not present the exact object shapes, we use slightly eroded masks m (with 3 pixels) after 100K iteration training to compute the first term of $\mathcal{L}_{\text{mask}}$ (m used in the first 100K iterations). We also empirically choose the hyper-parameters $\lambda_5 = 0.005$ and $\gamma = 2$. All models are trained using the two datasets at the 256×256 resolution.

4.1 Results of amodal instance completion

Occluded object completion. The object completion net, f_{content}, targets to hallucinate missing content of objects. There is no good numerical metric to evaluate content completion for the occluded object; thus, prior work [28, 57] primarily focuses on qualitative evaluations. In an attempt to address this concern, we automate quantitative evaluations based on a property that an amodal instance completion method should faithfully reconstruct an object

Table 1: **Quantitative evaluation for amodal instance completion**. Three groups of baselines are compared:1) classic inpainting approaches [49, 53], 2) an inpainting method with auxiliary foreground edges [32], and 3) an amodal instance completion method [57].

Method	COCOA [60]			KINS [36]		
	$\ell_1\downarrow$	PSNR↑	SSIM↑	$\ell_1\downarrow$	PSNR↑	SSIM↑
GConv [53]	61.45	29.40	0.981	47.06	25.52	0.935
CRA [49]	53.92	30.96	0.983	43.82	25.55	0.940
EdgeCon. [32]	41.02	31.54	0.983	37.38	26.64	0.939
De-occ. [57]	49.58	23.49	0.876	40.72	26.19	0.927
Ours	**37.11**	**31.91**	**0.985**	**34.99**	**26.93**	**0.965**

Table 2: **Quantitative comparison for image compositing** with varying object-to-image area ratios (ranges indicated in the *second row*). Here we compare our neural compositing model against Poisson Blending (PB) [34], Deep Image Blending (DIB) [58], and DoveNet [10]. Note that we dilate the original object masks by 5–10 pixels and crop the objects using its dilated mask to simulate "coarse cropping" during evaluation.

Method	COCOA [60]									KINS [36]								
	(0.05, 0.2]			(0.2, 0.4]			(0.4, 0.5]			(0.05, 0.2]			(0.2, 0.4]			(0.4, 0.5]		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
PB [34]	29.20	0.982	0.021	24.87	0.960	0.045	23.21	0.950	0.058	33.07	0.991	0.012	28.77	0.978	0.029	26.20	0.967	0.038
DIB [58]	26.80	0.958	0.049	23.94	0.920	0.102	21.08	0.881	0.151	29.91	0.978	0.027	26.26	0.957	0.055	23.76	0.941	0.069
DoveNet [10]	35.28	0.994	0.011	31.62	0.987	0.022	30.24	0.983	0.028	37.47	0.995	0.007	36.17	0.994	0.009	35.52	0.993	0.010
Ours	**37.06**	**0.996**	**0.008**	**32.87**	**0.991**	**0.018**	**31.56**	**0.986**	**0.022**	**38.60**	**0.996**	**0.004**	**37.44**	**0.996**	**0.007**	**37.28**	**0.996**	**0.008**

with part of which manually occluded. We note that the proposed evaluation strategy is commonly used in image inpainting [32, 49, 49, 52, 53]. Concretely, we employ three groups of representative baselines: 1) classic inpainting approaches [49, 53]; 2) inpainting methods with auxiliary information (e.g., instance contours) [32]; and 3) amodal instance completion methods [57]. We report our evaluation in terms of the L_1 loss, PSNR, and SSIM. The quantitative evaluation results with 500 test images are presented in Table 1. The numerical results indicate superior performance of our object completion module f_{content} than the alternative [32, 49, 53, 57] in this case. We show additional results and applications in the supplementary.

4.2 Results of amodal instance composition

Prior amodal completion work [28, 57] show applications such as inserting an amodal object into a new background. However, they do *not* consider color-inconsistency issues between the foreground object and the new background image. Moreover, such issues could be common as lights and shadows in the wild are exceptionally changeable. We proposed our composition model f_{comp} for appearance adjustment to address this concern. We show a qualitative evaluation in Figure 5, where the occluded white car (*Column 1*) was placed in the shadow regions of the new background image. We observe that De-occ. [57] produced unrealistic compositing results (*Column 2*) due to color inconsistency. In contrast, our entire method provided remarkable photo-realistic results (*Column 4*).

As precise segmentations for the intact and the visible regions of an object may be unavailable, we expect our composition net f_{comp} to be robust to the imperfect instances. For this, we verify the effectiveness of our composition module f_{comp} on coarsely cropped COCOA validation set [60] to simulate defective amodal instances. In this case, Poisson blending (PB) [34], and DIB [58] are employed as baseline algorithms in that they can blend

Table 3: **Quantitative evaluation for object amodal mask prediction using mIOU.** *Raw*: no amodal mask prediction; *Convex*: computing convex hull of the visible object region.

Dataset	Raw [57]	Convex [57]	De-occ. [57]	Ours
COCOA [60]	0.655	0.744	0.814	**0.820**

inaccurately cropped objects. We also compare to a learning-based approach, DoveNet [10]. One distinct difference of DoveNet [10] compared to our composition module f_{comp} lies in the format of input and output, where DoveNet [10] takes as input the *composite* of a background and an object image and produces a single harmonized output. Since DoveNet [10] does not consider imprecise inputs, we fine-tuned it using the same training dataset [60]. Figure 4 presents the compositing results with the ground truth. We observe that the blending algorithms [34, 58] have limited performance when the compositing components have intense color contrast. DoveNet [10] can adjust the object colors to some extent; however, it still has difficulties in dealing with coarse object boundaries. Our f_{comp} performs stably well in terms of color and content consistency.

We quantitatively evaluate f_{comp} on $2,500$ pairs of coarsely segmented and color-transferred instances with their inpainted background. Since instance area ratios may affect the performance, we conducted the experiments on 3 disjoint ranges according to the ratio of an instance to the image, i.e., $(0.05, 0.2]$, $(0.2, 0.4]$, and $(0.4, 0.5]$. We report the results in Table 2, with variances presented in the supplementary. The statistics in Table 2 show that our model outperforms the baselines [10, 41, 58] on image reconstruction and photorealism. We also observe that the area ratio of an object is an influencing factor of the compositing performance, i.e., the metric scores decrease as the ratio increases. We show more qualitative results and applications in the supplementary.

4.3 Results of amodal mask prediction

We evaluate our amodal mask prediction net, f_{shape} using the mean Intersection over Union (mIOU) metric. We show the quantitative results in Table 3. Compared to the baselines [57], our amodal mask prediction net f_{shape} performs slightly better.

4.4 Ablation study

We visualize the composition result after each of our processing step in Figure 8. Concretely, we aim to insert the occluded person (the red box) to a new background. First, straightforward object insertion *(a)* results in clearly visible artifacts. Second, we complete the occluded regions of the person using our two modules, f_{shape} and f_{content}, and insert the completed person to the background image *(b)*. However, the completed object contain invalid pixels due to the imperfection of amodal mask prediction. Third, our composition net f_{comp} removes and harmonizes noise pixels in *(b)*, achieving more plausible composition *(c)*. Note that our composition net f_{comp} can compose the person and the bench iteratively in a proper occlusion order. Figure 8 suggests the necessity of our proposed pipeline.

We further employ an ablation study on f_{comp} to analyze how the input and output format affect compositing performance. In this case, we either compose the background I_{bg} and the object image I_c with respect to the mask or concatenate them as input, and produces either RGB or RGBA layers with the remaining architecture fixed. The results are shown in Table 4. The inferior results in the first two rows validate the necessity of the design of f_{comp}.

(a) Insertion (cut-n-paste) (b) Completion (w/o comp.) (c) Ours

Figure 6: **Ablation study.** We insert the coarsely cropped occluded person (marked in the red box) to a new background via *(a)* naive cut-and-paste, *(b)* amodal mask prediction and object completion, and *(c)* our method.

(a) Object GT Ours *(b)* Object Occ. region Ours *(c)* Comp. result

Figure 7: **Failure cases** of our amodal mask prediction net f_{shape} *(a)*, the object completion net f_{content} *(b)*, and the neural composition net f_{comp} *(c)*.

4.5 Failure cases and discussions

While achieving favorable results than the baseline approaches, our method has several limitations. Specifically, the amodal instance completion task remains challenging partially due to the limited availability of training data, and complex shapes and colors of various instances. As shown in Figure 6, our amodal mask prediction net f_{shape} failed to recognized the occluded body region under the table *(a)*, and the object completion net f_{content} generated unrealistic occluded content *(b)* for the motorcycle. Second, our approach does not explicitly model environmental lighting in the wild (e.g., in *(c)*, invalid lighting directions on hair after composition), and thus we leave it for future work.

Table 4: **Ablation study** in the image composition task using 500 KINS test images [36]. The input is either the *composition* of the background image and the object image with respect to the mask, or the *concatenation* of the two.

Input	Output	PSNR ↑	SSIM ↑	LPIPS ↓
Composition	RGB	34.56	0.992	0.012
Composition	RGBA	36.23	0.993	0.011
Concatenation	RGBA	**37.36**	**0.996**	**0.007**

5 Conclusions

We proposed a fully automatic system for image composition that is capable of handling imperfect and heterogeneous amodal inputs. Experimental results demonstrate that our approach outperforms various baselines for dealing with imperfect and heterogeneous amodal instances. From an application perspective, our method has a broad impact on amodal instance manipulation and style harmonization. Beyond that, we leave automatic adjustments for spatial transformations and better handling of lighting for future work.

References

[1] Yagiz Aksoy, Tunc Ozan Aydin, and Marc Pollefeys. Designing effective inter-pixel information flow for natural image matting. In *CVPRn*, 2017.

[2] Pablo Arbeláez, Bharath Hariharan, Chunhui Gu, Saurabh Gupta, Lubomir Bourdev, and Jitendra Malik. Semantic segmentation using regions and parts. In *CVPR*, 2012.

[3] Samaneh Azadi, Deepak Pathak, Sayna Ebrahimi, et al. Compositional gan: Learning conditional image composition. *IJCV*, 2020.

[4] Connelly Barnes, Eli Shechtman, Adam Finkelstein, et al. Patchmatch: A randomized correspondence algorithm for structural image editing. *TOG*, 2009.

[5] Matthew Brown, David G Lowe, et al. Recognising panoramas. In *Int. Conf. Comput. Vis.*, 2003.

[6] Peter J Burt and Edward H Adelson. The laplacian pyramid as a compact image code. In *Readings in computer vision*, pages 671–679. Elsevier, 1987.

[7] Joao Carreira and Cristian Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *TPAMI*, 2011.

[8] Bor-Chun Chen and Andrew Kae. Toward realistic image compositing with adversarial learning. In *CVPR*, 2019.

[9] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint*, 2015.

[10] Wenyan Cong, Jianfu Zhang, Li Niu, et al. Dovenet: Deep image harmonization via domain verification. In *CVPR*, 2020.

[11] A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 2004.

[12] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001.

[13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, et al. Generative adversarial nets. In *NeurIPS*, 2014.

[15] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, et al. Improved training of wasserstein gans. In *NeurIPS*, 2017.

[16] James Hays and Alexei A Efros. Scene completion using millions of photographs. *TOG*, 2007.

[17] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. Image completion using planar structure guidance. *ACM Transactions on graphics (TOG)*, 33(4): 1–10, 2014.

[18] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ToG*, 2017.

[19] Micah K Johnson, Kevin Dale, Shai Avidan, et al. Cg2real: Improving the realism of computer generated images using a large collection of photographs. *IEEE Transactions on Visualization and Computer Graphics*, 2010.

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2014.

[21] J. Lalonde and A. A. Efros. Using color compatibility for assessing image realism. In *ICCV*, 2007.

[22] Anat Levin, Assaf Zomet, Shmuel Peleg, and Yair Weiss. Seamless image stitching in the gradient domain. In *European Conference on Computer Vision*, pages 377–389, 2004.

[23] Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. *TPAMI*, 2007.

[24] Anat Levin, Alex Rav-Acha, and Dani Lischinski. Spectral matting. *TPAMI*, 2008.

[25] Liang Liao, Jing Xiao, Zheng Wang, Chia-wen Lin, and Shin'ichi Satoh. Guidance and evaluation: Semantic-aware image inpainting for mixed scenes. *ECCV*, 2020.

[26] Chen-Hsuan Lin, Ersin Yumer, Oliver Wang, Eli Shechtman, and Simon Lucey. St-gan: Spatial transformer generative adversarial networks for image compositing. In *CVPR*, 2018.

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, et al. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[28] Huan Ling, David Acuna, Karsten Kreis, Seung Wook Kim, and Sanja Fidler. Variational amodal object completion. *NeurIPS*, 2020.

[29] Zhang Lingzhi, Wen Tarmily, Min Jie, et al. Learning diverse object placement by inpainting for compositional data augmentation. *ECCV*, 2020.

[30] Guilin Liu, Fitsum A Reda, Kevin J Shih, et al. Image inpainting for irregular holes using partial convolutions. In *ECCV*, 2018.

[31] Erika Lu, Forrester Cole, Tali Dekel, Weidi Xie, et al. Cg2real: Improving the realism of computer generated images using a large collection of photographs. *SIGGRAPH*, 2010.

[32] Kamyar Nazeri, Eric Ng, Tony Joseph, et al. Edgeconnect: Generative image inpainting with adversarial edge learning. *ICCV*, 2019.

[33] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, et al. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

[34] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *SIGGRAPH*, 2003.

[35] Thomas Porter and Tom Duff. Compositing digital images. In *SIGGRAPH*, 1984.

[36] Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Amodal instance segmentation with kins dataset. In *CVPR*, 2019.

[37] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 2001.

[38] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[39] Konstantin Sofiiuk, Polina Popenova, and Anton Konushin. Foreground-aware semantic representations for image harmonization. *arXiv*, 2020.

[40] Yuhang Song, Chao Yang, Yeji Shen, et al. Spg-net: Segmentation prediction and guidance network for image inpainting. *BMVC*, 2018.

[41] Jian Sun, Jiaya Jia, Chi-Keung Tang, and Heung-Yeung Shum. Poisson matting. In *SIGGRAPH*. 2004.

[42] Michael W Tao, Micah K Johnson, and Sylvain Paris. Error-tolerant image compositing. *IJCV*, 2013.

[43] Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Xin Lu, and Ming-Hsuan Yang. Deep image harmonization. In *CVPR*, 2017.

[44] Enze Xie, Wenjia Wang, Wenhai Wang, Mingyu Ding, Chunhua Shen, and Ping Luo. Segmenting transparent objects in the wild. *ECCV*, 2020.

[45] Wei Xiong, Jiahui Yu, Zhe Lin, et al. Foreground-aware image inpainting. In *CVPR*, 2019.

[46] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. In *CVPR*, 2017.

[47] Su Xue, Aseem Agarwala, Julie Dorsey, et al. Understanding and improving the realism of image composites. *TOG*, 2012.

[48] Chao Yang, Xin Lu, Zhe Lin, et al. High-resolution image inpainting using multi-scale neural patch synthesis. In *CVPR*, 2017.

[49] Zili Yi, Qiang Tang, Shekoofeh Azizi, et al. Contextual residual aggregation for ultra high-resolution image inpainting. In *CVPR*, 2020.

[50] Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, et al. Photorealistic style transfer via wavelet transforms. In *ICCV*, 2019.

[51] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *ICLR*, 2016.

[52] Jiahui Yu, Zhe Lin, Jimei Yang, et al. Generative image inpainting with contextual attention. In *CVPR*, 2018.

[53] Jiahui Yu, Zhe Lin, Jimei Yang, et al. Free-form image inpainting with gated convolution. In *CVPR*, 2019.

[54] Mihai Zanfir, Elisabeta Oneata, Alin-Ionut Popa, et al. Human synthesis and scene compositing. In *AAAI*, 2020.

[55] Yu Zeng, Zhe Lin, Jimei Yang, et al. High-resolution image inpainting with iterative confidence feedback and guided upsampling. In *ICCV*, 2020.

[56] Fangneng Zhan, Jiaxing Huang, and Shijian Lu. Adaptive composition gan towards realistic image synthesis. *arXiv*, 2019.

[57] Xiaohang Zhan, Xingang Pan, Bo Dai, et al. Self-supervised scene de-occlusion. In *CVPR*, 2020.

[58] Lingzhi Zhang, Tarmily Wen, and Jianbo Shi. Deep image blending. In *WACV*, 2020.

[59] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic image completion. In *CVPR*, 2019.

[60] Yan Zhu, Yuandong Tian, Dimitris Metaxas, and Piotr Dollár. Semantic amodal segmentation. In *CVPR*, 2017.

Supplementary

In this supplementary document, we provide the network architectures and additional implementation details to complement the main paper. We also present additional quantitative and qualitative comparisons. We will make the source code and pretrained models publicly available to foster future research.

A.Implementation details

A.1 Network architecture

We adopt U-Net [38] as the backbone of the amodal mask prediction net f_{shape} and the composition net f_{comp}. We show the full backbone in Table 5. We use "zero-padding", batch normalization ("bn"), convolutional transpose ("convt"), and skip connection ("skipk" refers to a skip connection with layer k) in the neural compositing network. The full definition of the neural compositing network is defined as follows.

We modify the CRA model [49] as the backbone of our object content completion net f_{content}. We note that CRA's architecture is a common paradigm in image inpainting tasks, as similar structures were widely used in [45, 53, 55]. There are, however, several important differences between our object content completion net f_{content} and the CRA model [49]. First, the goal of the two methods is different: our object content completion net f_{content} aims to hallucinate the *invisible regions* of an object from the visible regions On the other hand, the CRA method [49] targets to recover the entire image (i.e., filling in the missing regions using all the remaining known pixels as contexts). Second, the inputs of the two methods are different. Specifically, the input of our object content completion module, f_{content}, is a triplet consisting of a masked object image ($I_o \odot m_{vis}$ in the main manuscript), and the two binary masks (m_{vis} and m_{inv} in the main manuscript). The CRA model [49] takes as input a masked image as well as the corresponding mask. Consequently, training data pre-processing for the object completion net f_{content} in our method is different from [49]. In our work, we randomly hide part of the target object to simulate amodal instance completion. In particular, we randomly sample an object masks m_{inv} from the dataset and use them to occlude part of the target object I_o. We apply basic transformations, e.g., scaling and translation, to the sampled object mask m_{inv} such that it has overlap with the target object I_o. Table 2 and Figure 3 in the main manuscript show effectiveness of the modifications on the object completion model f_{content} compared to the original CRA [49]. We use "same" padding and the Exponential Linear Unit (ELU) activation function [9] for all convolution layers. We show the full definition of the object content completion net f_{content} in Table 6-7.

A.2 Training details

We use the Adam optimizer [20] to train the three networks. The initial learning rate is $1e$-3 for the amodal prediction net f_{shape}, $1e$-4 for the object content completion model f_{content}, and $2e$-4 for the neural composition net f_{comp}. We select beta1 $= 0.5$ and beta2 $= 0.9$ for all Adam optimizer [20].

A.3 Inference algorithm

We present the inference procedure in Algorithm 1.

Table 5: The backbone used in the amodal shape prediction net f_{shape} and the neural composition net f_{comp}. *bn*: batch normalization, *convt*: convolutional transpose, *skipk*: a skip connection with layer k.

layers	out channels	stride	activation
4×4 conv	64	2	leaky
4×4 conv, bn	128	2	leaky
4×4 conv, bn	256	2	leaky
4×4 conv, bn	256	2	leaky
4×4 conv, bn	256	2	leaky
4×4 conv, bn	256	1	leaky
4×4 conv, bn	256	1	leaky
skip5, 4×4 convt, bn	256	2	relu
skip4, 4×4 convt, bn	256	2	relu
skip3, 4×4 convt, bn	128	2	relu
skip2, 4×4 convt, bn	64	2	relu
skip1, 4×4 convt, bn	64	2	relu
4×4 conv	4	1	tanh

Table 6: The *coarse network* of the object content completion module f_{content}. *num* refers to the number of layers. *out channels* refers to the number of output channels after the layer. *stride* and *dilation* are the parameters of the convolution operation.

layers	num	out channels	stride	dilation	out shape
5×5 gconv	1	32	2	1	128×128
3×3 gconv	1	64	1	1	128×128
3×3 gconv	1	64	2	1	64×64
3×3 gconv	6	64	1	1	64×64
3×3 gconv	5	64	1	2	64×64
3×3 gconv	4	64	1	4	64×64
3×3 gconv	2	64	1	8	64×64
3×3 gconv	3	64	1	1	64×64
3×3 deconv	1	32	1	1	128×128
3×3 deconv	1	3	1	1	256×256

Table 7: The *refine network* of the object content completion module f_{content}. The notations in the first row are identical to Table 6. *attn* refers to Contextual Attention [53].

layers	num	out channels	stride	dilation	out shape
5×5 gconv	1	32	2	1	128×128
3×3 gconv	1	32	1	1	128×128
3×3 gconv	1	64	2	1	64×64
3×3 gconv	1	128	2	1	32×32
3×3 gconv	2	128	1	1	32×32
3×3 gconv	1	128	1	2	32×32
3×3 gconv	1	128	1	4	32×32
3×3 gconv	1	128	1	8	32×32
3×3 gconv	1	128	1	16	32×32
3×3 gconv + attn	1	128	1	1	32×32
3×3 deconv	1	64	1	1	64×64
3×3 gconv + attn	1	64	1	1	64×64
3×3 deconv	1	32	1	1	128×128
3×3 gconv + attn	1	32	1	1	128×128
3×3 deconv	1	3	1	1	256×256

Algorithm 1 Inference Procedure

Input: an amodal mask prediction model f_{shape}; an object completion model, denoted as f_{content}; a neural compositing model, denoted as f_{comp}; input background I_{bg} and an ordered collection of object images $X = \{x^{(1)}, x^{(2)}, ..., x^{(M)}\}$, where $x^{(j)} = (I^{(j)}, m_{vis}^{(j)})$, $j \in \{1, 2, ..., M\}$ and $M = |X|$ (note that $m_{vis}^{(j)}$ is not required to be precise during inference).

for $j = 1, \ldots, M$ **do**
  if $x^{(j)}$ is occluded **then**
    $\hat{m}_{vis}, \hat{m}_{amodal} = f_{\text{shape}}(x_j, m_{vis})$
    $\hat{I}_o = f_{\text{content}}(I, \hat{m}_{vis}, \hat{m}_{amodal})$
  else
    $\hat{I}_o = I \odot m_{vis}$
  end if
  $I_{out}, \alpha = f_{\text{comp}}(I_{bg}, \hat{I}_o)$
  $\hat{I} = \alpha \odot I_{out} + (1 - \alpha) \odot I_{bg}$
end for

Return: $\hat{I}$

B. Ablation study

We conduct an ablation study for the content completion model f_{content} where the new inputs are triplet images consisting of a masked image with the background region preserved ($I \odot m_{inv}$), a visible mask (m_{vis}), and an invisible mask (m_{inv}). In other words, we do *not* hide the background region during training. We train the new content completion model with an identical number of iterations as the prior model. We show the comparison results in Figure 8 which indicates that **removing background regions benefits object reconstruction**.

C. Additional results

C.1 Variance of the comparison for image composition.

We computed Table 3 in the main manuscript by repeating the experiments three times with the object styles transferred towards a randomly selected reference image. Here, we show the variances of the statistical results in Table 8.

Table 8: **Variances of the quantitative comparison for image compositing on the COCOA validation dataset [27].** Three baselines are compared: PB [34], DIB [58], and DovNet [10].

	(0.05, 0.2]			(0.2, 0.4]			(0.4, 0.5]		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
[34]	1e-2	2e-7	2e-7	5e-2	2e-7	9e-7	1e-1	4e-6	3e-6
[58]	6e-2	3e-6	3e-4	2e-1	3e-5	3e-4	3e-1	2e-4	2e-5
[10]	3e-3	3e-7	2e-8	2e-2	4e-8.	5e-7	2e-2	1e-7	2e-6
Ours	7e-2	2e-7	4e-7	1e-2	2e-8	8e-8	8e-2	1e-6	2e-6

Original Occ. object **Ours w/o bg** Ours w/ bg

Figure 8: **Ablation study of the object completion model f_{content}.** We predict occluded regions of the objects. *Column 1*: Original images from COCOA validation dataset [60] with visible and amodal masks annotated by [60]; *Column 2*: objects with occluded region marked in yellow and background region in white; *Column 3*: the object completion model with $I \odot m_{vis}$ as input (i.e., background pixels are empty); *Column 4*: the object completion model with $I \odot (1 - m_{inv})$ as input (i.e., background pixels are valid).

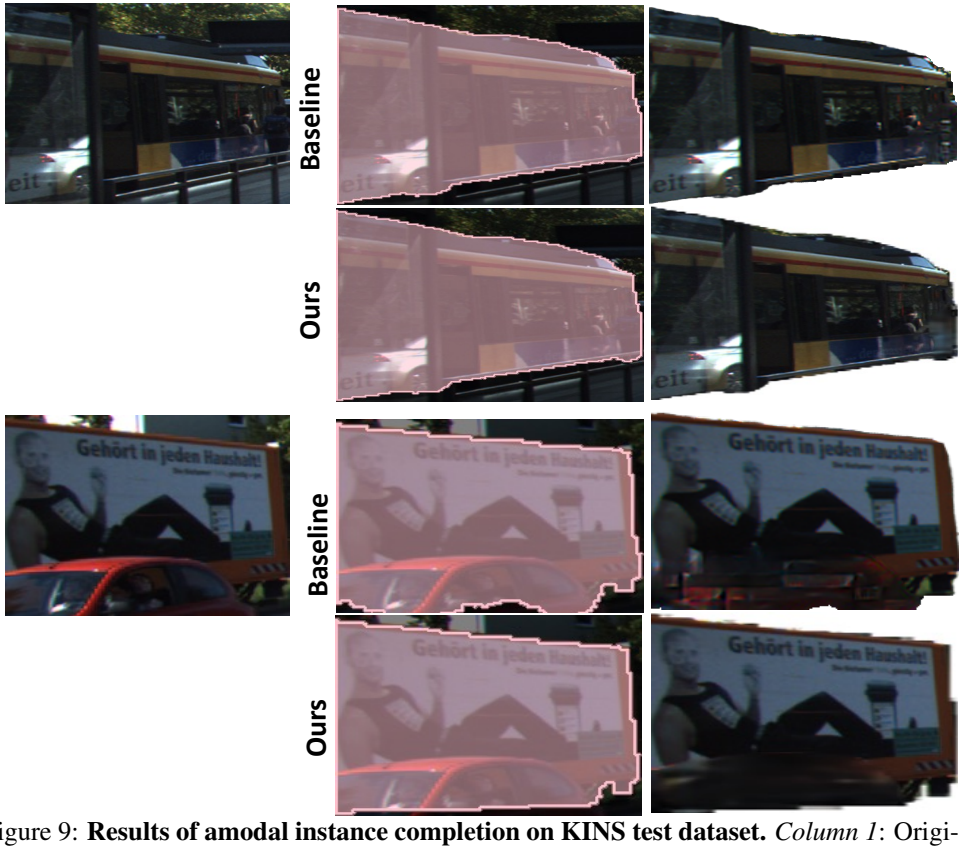


Figure 9: **Results of amodal instance completion on KINS test dataset.** *Column 1*: Original objects, *Column 2*: predicted amodal masks, *Column 3*: synthetic amodal objects.

De-occ. [57] PB [34] DoveNet [10] Ours

Figure 10: **Results of amodal instance composition on KINS test dataset.** We inserted the instances into the new background image (marked by red arrow). *De-occ.* [57] generated the intact object and then directly pasted the amodal instance into the new background. *PB* and *DoveNet* [10, 34] were used to harmonize into the new background the amodal instance by our amodal mask prediction net f_{shape} and object content completion net f_{content}. *Ours* were achieved by the proposed three modules.

C.2 Amodal instance composition on KINS dataset

We show additional amodal mask prediction and object content completion comparisons in Figure 9 and composition results of the identical instances in Figure 10. Specifically, in Figure 9, we present the predicted amodal masks and the hallucinated results by the baseline method [57] and our approach in column 2-3. Our amodal mask prediction net f_{shape} has a comparable performance with [57], and our content completion net f_{content} can hallucinate the occluded regions of the objects with fewer artifacts (see the occluded corners and the bottom of the vehicles). In Figure 10, we insert the completed objects into a new background image (marked by the red arrow). De-occ. [57] does not harmonize the amodal instances with the new background in its applications, leading to unrealistic compositing results. We also notice that PB [34] performs unsatisfied when the objects have obvious color contrast with the background. The same issues of PB [34] are discussed in prior work [42, 58]. In contrast, our composition net f_{comp} works stably well with fewer artifacts.

C.3 Additional applications

Here we demonstrate several additional applications using our method.

Object re-shuffling. Our object completion model enables re-shuffling objects. Figure 11 shows examples of re-shuffling objects to new locations in the images. A limitation of our model is that the occluded region is smooth, e.g., the blue luggage in Figure 11. We leave this for future improvement.

Object insertion with imperfect inputs. In Figure 12, we show the results of placing two dishes onto an indoor scene. Our method automatically adjusts the foreground instances' colors towards the background images with redundant pixels around objects removed in these cases.

Figure 11: **Object re-shuffling results**. *left*: original images. The arrows indicate object moving directions; *right*: results with objects re-shuffled.

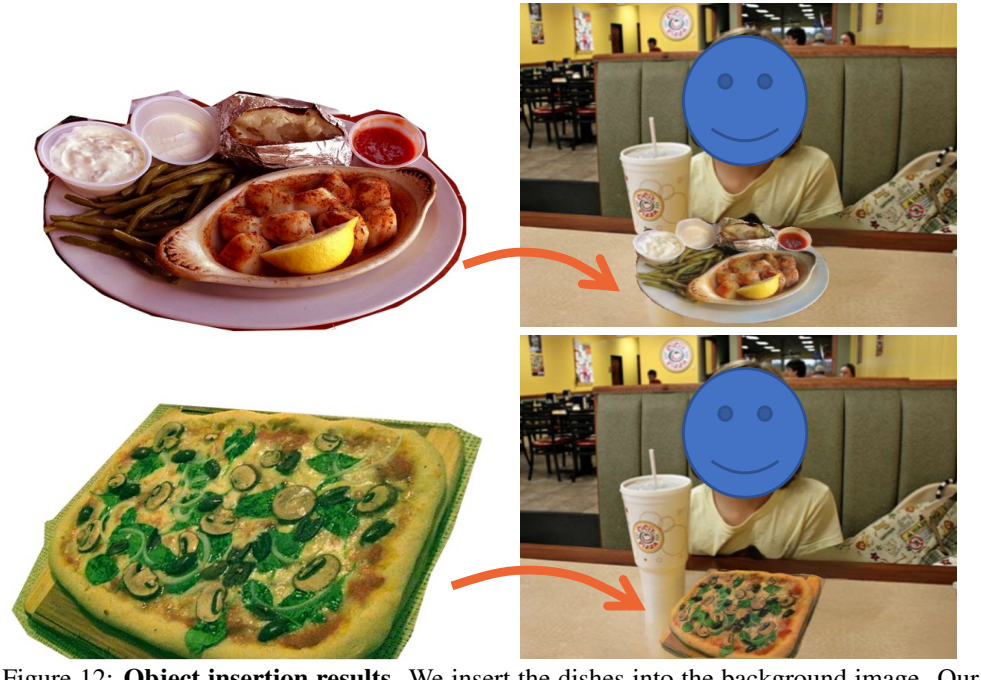


Figure 12: **Object insertion results**. We insert the dishes into the background image. Our method harmonizes the content and refines the imperfect masks.