Inserting Objects into Any Background Images via Implicit Parametric Representation

Qi Zhang, Guanyu Xing, Mengting Luo, Jianwei Zhang, and Yanli Liu

***Abstract*—Inserting an object into a background scene has wide applications in image editing and mixed reality. However, existing methods still struggle to seamlessly adapt the object to the background while maintaining its individual characteristics. In this article, we propose to fine-tune a pre-trained diffusion-based insertion model such that it learns to establish a unique correspondence between a few weights and the target object, given as input few-shot images of an object. A novel individualized feature extraction (IFE) module is designed to extract the individual detail features from few-shot object images. Then, the individual features of the target object, together with the semantic features of the target object and the background context features extracted by the pre-trained image encoders are injected into the cross-attention modules of the latent diffusion model, enabling it to learn the correlation information of the target object and the background scene through the attention mechanism. The weights obtained by fine-tuning implicitly serve as an alternative representation of the target object, with which the object can be easily inserted into any background images. Extensive comparative experiments validate the superiority of the proposed method to the state-of-the-art insertion methods in maintaining the individual details of the inserted object and adapting it to background scenes, including allowing the interaction between the inserted object and the background scene, correctly handling their occlusion relationship, maintaining the consistency of their viewpoints and poses.**

***Index Terms*—Object insertion, implicit parametric representation, latent diffusion model, insertion strategy.**

I. Introduction

THE task of inserting objects aims to seamlessly integrate target objects with background scenes, achieving realistic results comparable to real-world scenes. This technology holds significant application value in computer vision and computer graphics, particularly in mixed reality, content generation, and image editing.

Current studies on object insertion into scenes can be divided into two main categories: methods based on three-dimensional (3D) models [8], [9], [10], [11], [12] and methods based on two-dimensional images [1], [2], [13], [14], [15], [16], [17]. The methods in the first category require 3D modeling of target objects, which can be obtained by structure from motion (SFM) [18] and the neural radiance field (NeRF) [19]. Furthermore, lighting estimation and rendering [9], [10], [11], [12] require calibrating the camera and inferring information about complex lighting conditions, material properties, the 3D geometry of the scene, and strong assumptions about the light source, which complicate object insertion based on 3D models and are time-consuming, and cumbersome. In contrast, the methods in the second category regard target objects as two-dimensional images, copying objects and pasting them into suitable regions of background images. These methods refine and enhance the pasted images, aiming to achieve more realistic insertion results. For example, image harmonization [13], [14] and shadow generation [2], [15] methods focus on issues of inconsistent lighting, image blending methods [20] primarily address the image stitching gap, and foreground region localization [16] methods are used for object placement. However, these methods still have problems since two-dimensional images contain limited planar information and lack geometric and depth information. For instance, these methods cannot present geometric occlusion between objects and background scenes well. As shown in the first row of Fig. 1, the inserted object result of methods based on two-dimensional images (i.e., HDNet and SGRNet) shows the bunny hanging outside the red bucket, resulting in an unnatural visual phenomenon.

Recently, diffusion models have shown convincing performance in the field of artificial intelligence-generated content (AIGC) [3], [6], [7], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34]. Several methods based on diffusion models have been proposed to effectively insert objects into scenes, which typically utilize the object features as conditions and inject them into the latent diffusion model (LDM) model. For instance, Paint by Example (PbE) [3] is a framework that involves a single forward of the diffusion model without any iterative optimization. To prevent the insertion model from simply copying and pasting the object image, it drops the spatial tokens and only regards the global image embedding of the object as the condition. Since only the high-level semantics of the object image are extracted and injected into LDM, PbE suffers from a severe drawback of restricted object fidelity. Objects that have been inserted are prone to color distortion, shape distortion, loss of texture details, etc. As shown in the 4th column of Fig. 1, the appearance of CG bunny significantly deviates from the target one. Recently, ControlCom [4], Anydoor [5] and

Received 21 February 2024; revised 16 October 2024; accepted 10 November 2024. Date of publication 14 November 2024; date of current version 1 August 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62472297 and under Grant 62172290, and in part by the Sichuan Science and Technology Program under Grant 2023YFS0454. Recommended for acceptance by J. Liao. *(Corresponding author: Yanli Liu.)*

Qi Zhang, Mengting Luo, Jianwei Zhang, and Yanli Liu are with the College of Computer Science, Sichuan Universit, Chengdu 610065, China (e-mail: yanliliu@scu.edu.cn).

Guanyu Xing is with the School of Cyber Science and Engineering, Sichuan University, Chengdu 610065, China.

Our code is available at https://github.com/dorianzhang7/TVCG-insertion.

Digital Object Identifier 10.1109/TVCG.2024.3498065

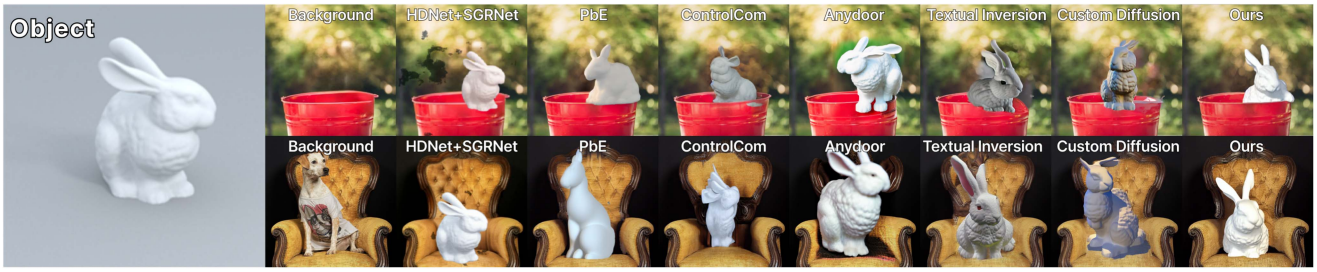

Fig. 1. The results of inserting a CG bunny into two background images. The third to ninth columns present respectively the results of HDNet [1], SGRNet [2], PbE [3], ControlCom [4], Anydoor [5], Textual Inversion [6], Custom Diffusion [7], and our method.

PhD [35] respectively improve the object fidelity by constructing a local enhancement module or injecting detailed features of the object with ControlNet [36] into the latent diffusion model. However, these methods typically show significant shortcomings in terms of posture adaptability and occlusion between objects and background scenes, as shown in the 5th and 6th columns of Fig. 1. To sum it up, the current insertion methods fail to preserve the identity and details of the inserted object and simultaneously adapt the inserted object to background scenes harmoniously. The main reason is that these feed-forward insertion methods only input an image and lack specific training mechanisms of the target object, leading to under-fits of the target object.

In the task of text-to-image, in order to achieve subject-driven generation, some fine-tuning methods have been proposed to train pre-trained LDMs with few-shot images of objects. For instance, Textual Inversion [6] learns to represent objects through new "words" in the embedding space of a frozen text-to-image model. Alternatively, Custom Diffusion [7] represents objects by optimizing some parameters in the text-to-image conditioning mechanism. Since these methods cannot be directly used for the object insertion task, we manage to combine them respectively with the blended diffusion [28], a general text-to-image insertion model, to perform object insertion. The inserted results of the two methods are shown in the 7th and 8th columns of Fig. 1, which demonstrate that they achieve remarkable performance in maintaining the shape, texture, and viewpoint of the bunny by fitting its few-shot object images. However, there are still obvious deficiencies in the color and lighting of the inserted bunny as well as the occlusion between the bunny and the red bucket.

In this paper, we propose to fine-tune the pre-trained PbE model [3] such that it learns to establish a unique correspondence between a few weights and the target object, given as input few-shot images of an object. To enable the LDM model [24] to learn the high-level semantics of the object, we adopt the structure of information bottleneck to discard the individualized detailed features of the object and only inject class features, i.e., the class tokens extracted by the CLIP-ViT [37] encoder with semantic features of the object, into the LDM. Then, in order to provide the network with sufficient detailed characteristics of objects, we replenish the individualized detailed features of objects by fine-tuning the network with information injection. Specifically, we design a novel individualized feature extraction (IFE) module to extract the individual detail features from few-shot object images in fine-tuning. Finally, the individualized features and the class features of the target object as well as the background context features are injected into the cross-attention modules of LDM, enabling the model to learn the correlation information from the target object and the background scene through the attention mechanism. The results of our model are shown in the 9th column of Fig. 1. Note that our inserted bunny matches the lighting and spatial conditions of the background scene while maintaining the identity and details of the bunny.

In contrast to the traditional methods that represent objects explicitly (e.g., 2D images or 3D models), the parameters obtained by fine-tuning implicitly serve as an alternative representation of the target object. Therefore, we refer to the parameters as implicit parameter representations for the insertion task, with which the object can be inserted into any background scenes at different positions. Furthermore, we construct a practical dataset, named InsertSet, which includes 101 image categories for training and 20 manual annotation categories for testing.

The main innovative contributions of our method are listed as follows:

1) We propose a novel insertion model that initially extracts the high-level semantics of an object and then replenishes its individual details by fine-tuning PbE with an IFE module, given a few images of the object. By establishing a correspondence between a few weights and the target object, the proposed model inserts the object into any background scene seamlessly and harmoniously. The weights implicitly serve as an alternative representation of a target object in contrast to the traditional insertion methods that represent objects explicitly, e.g., 2D images or 3D models.
2) We design a new strategy of learning the correlation information between the target object and the background scene through the attention mechanism. With the strategy, the expanded individual features of the object, the semantics features of the target object, as well as the background context features, are injected into the cross-attention modules of LDM, which enables the proposed model to flexibly adapt the object to the background scenes with correct occlusion relationships and consistent viewpoints, while maintaining its individual characteristics.
3) We construct a practical dataset named InsertSet for the task of object insertion. It comprises 101 categories of objects for training and 20 manual annotation categories of objects for testing.

II. Related Work

A. Object Insertion

Object insertion refers to seamlessly integrating objects into background scenes. Image harmonization [13], [14], shadow generation [2], [15] and object placement methods [16] have been used to address the problems of lighting and placement. However, these methods have limitations in maintaining the object's geometric shape and handling occlusion between the foreground and background scenes. The structure from motion (SFM) [18] and neural radiance field (NeRF) [19] methods obtain 3D structures, cooperated with lighting estimation and rendering methods [9], [10], [11], [12] can achieve reasonable insertion results. Nevertheless, these methods generally require calibrating the camera and inferring complex environmental information. As a balance, the methods of 3D-aware image synthesis [38], [39] are also used for object insertion, but they require exploring 3D layout estimations for scenes and fail to insert objects in complex background scenes. [35], [40] utilize the generative models to achieve object insertion, but have shortcomings in terms of adaptability and authenticity. Object insertion based on diffusion models can be applied to the task of object insertion. For instance, Blended diffusion [28] leverages and combines a pre-trained language-image model (CLIP) to steer the edit toward a text prompt, generating natural-looking results. Paint by example (PbE) [3] and Objectstitch [21] treat the object image as references, and they are capable of inserting any object into scenes. However, these methods fail to comprehensively demonstrate the detail of objects and have an issue of identity shift. Recently, Anydoor [5] utilizes the encoder of ControlNet [36] to inject the high-frequency features of an object image into LDM [24]. ControlCom [4] based on PbE [3] introduces a local enhancement module to increase the object details. Imprint [41] introduces a novel context-agnostic ID-preserving training, demonstrating superior appearance preservation. While these methods are superior in preserving the identity of the inserted objects, they have limitations in posture adaptability and harmony between objects and background scenes.

B. Subject-Driven Text-to-Image Generation Based on Diffusion Models

Diffusion models [22], [23], [24], [25] define a Markov chain of diffusion steps to slowly add random noise to the data and then learn to reverse the diffusion process to construct desired data samples from the noise. Stable Diffusion [24] further applies the powerful pre-trained autoencoders to convert images into latent space and reaches a near-optimal point between complexity reduction and detail preservation for the first time. Currently, in the field of subject-driven generation, individualized fine-tuning diffusion models play an important role. For instance, the DreamBooth model [27] uses the few-shot images of objects to train the generated model to fit the data more fully but it has a high consumption of time and resources. DreamBooth-LoRA [42], Textual Inversion [6], Custom Diffusion [7] and SVDiff [29] are proposed as effective solutions for training specific parameters and the main difference between these methods lies in training distinct parameters. Moreover, [30] accelerates personalization from dozens of minutes to seconds, while preserving quality. [31] builds a well-defined celeb basis from the embedding space to showcase a better concept combination ability. In addition, feed-forward methods based on diffusion models also can achieve the task of subject-driven generation. For instance, Ip-adapter [33] designs a decoupled cross-attention mechanism that separates cross-attention layers for text features and image features, which is an effective and lightweight adapter to achieve specific image generation. Elite [34] consists of global and local mapping networks for fast and accurate customized text-to-image generation.

III. Method

A. Overview

Fig. 2 shows the pipeline of the proposed object insertion method, in which the diffusion mechanism is divided by a dashed line into the forward diffusion and the reverse diffusion.

For the forward diffusion, the background image x is converted into a high-dimensional latent variable z_0. Then, z_T is obtained from a forward Markov chain: $z_t = \sqrt{\alpha_t} z_0 + \sqrt{1-\alpha_t}\epsilon$, where α_t decreases with the timestep t, $t \in [1, T]$, and T is the total number of steps. $\epsilon \sim N(0, 1)$ is Gaussian distribution noise, which is added during the diffusion process.

For reverse diffusion, the proposed model is a probabilistic model designed to learn a data distribution of the target object by gradual denoising of a normally distributed variable, which corresponds to learning the reverse process of a fixed Markov chain of length T. A region of insertion (ROI) mask is used to constrain the placement position of the target object in the background image. Combining the ROI mask, the background image, and z_T to provide features of the background scene, we extract the features from the identity image and few-shot object images. The extracted features, used as object conditions, interact with scene features in the U-net of the proposed network. The result z_0 inferred by the proposed network is converted back into the image through the decoder, and $\hat{x}$ is the insertion result.

B. Network

In Fig. 3, the proposed network consists of three components: an Identity Support (IS) module, an Individualized Feature Extraction (IFE) module, and a U-Net. In the IS module, the model takes the identity image as input and applies the pre-trained CLIP-ViT [37] encoder to obtain the original feature. Same as the method in [3], the IS module with information bottleneck only selects class features, which are the class tokens extracted by a pre-trained CLIP image encoder with global semantic features. The IS module forces the network to understand the high-level semantics of objects and learn to maintain the semantic information. Correspondingly, the IFE module uses the CLIP-ViT encoder with shared parameters to extract the original feature from few-shot object images, which are captured under different poses, viewpoints, and lighting conditions of the target object. The feature encoder block (FEB), composed of convolutions makes full use of the original feature of the target object to obtain

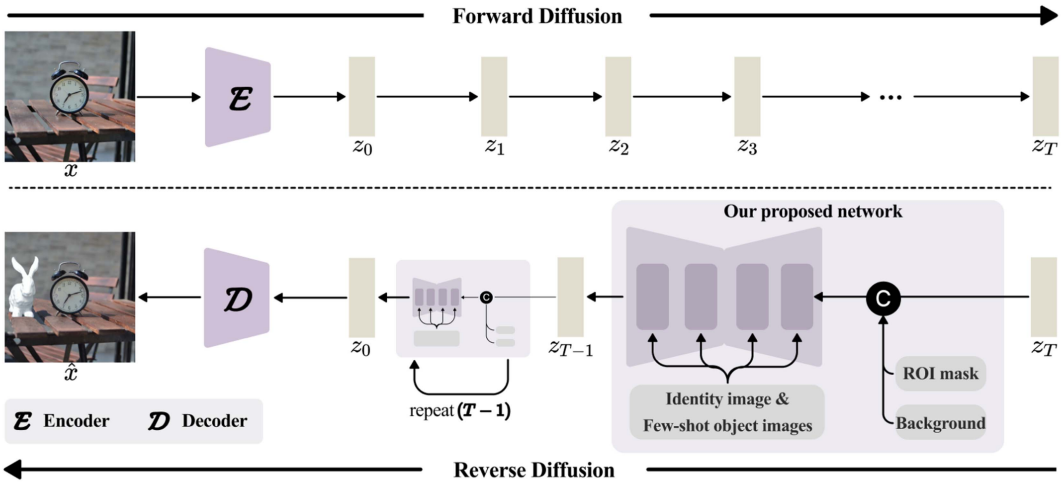

Fig. 2. The pipeline of object insertion in our proposed method. The background image x is converted into the low-dimensional latent feature z_T via forward diffusion, and the result inferred by our proposed model is converted back into the output image $\hat{x}$ via reverse diffusion.

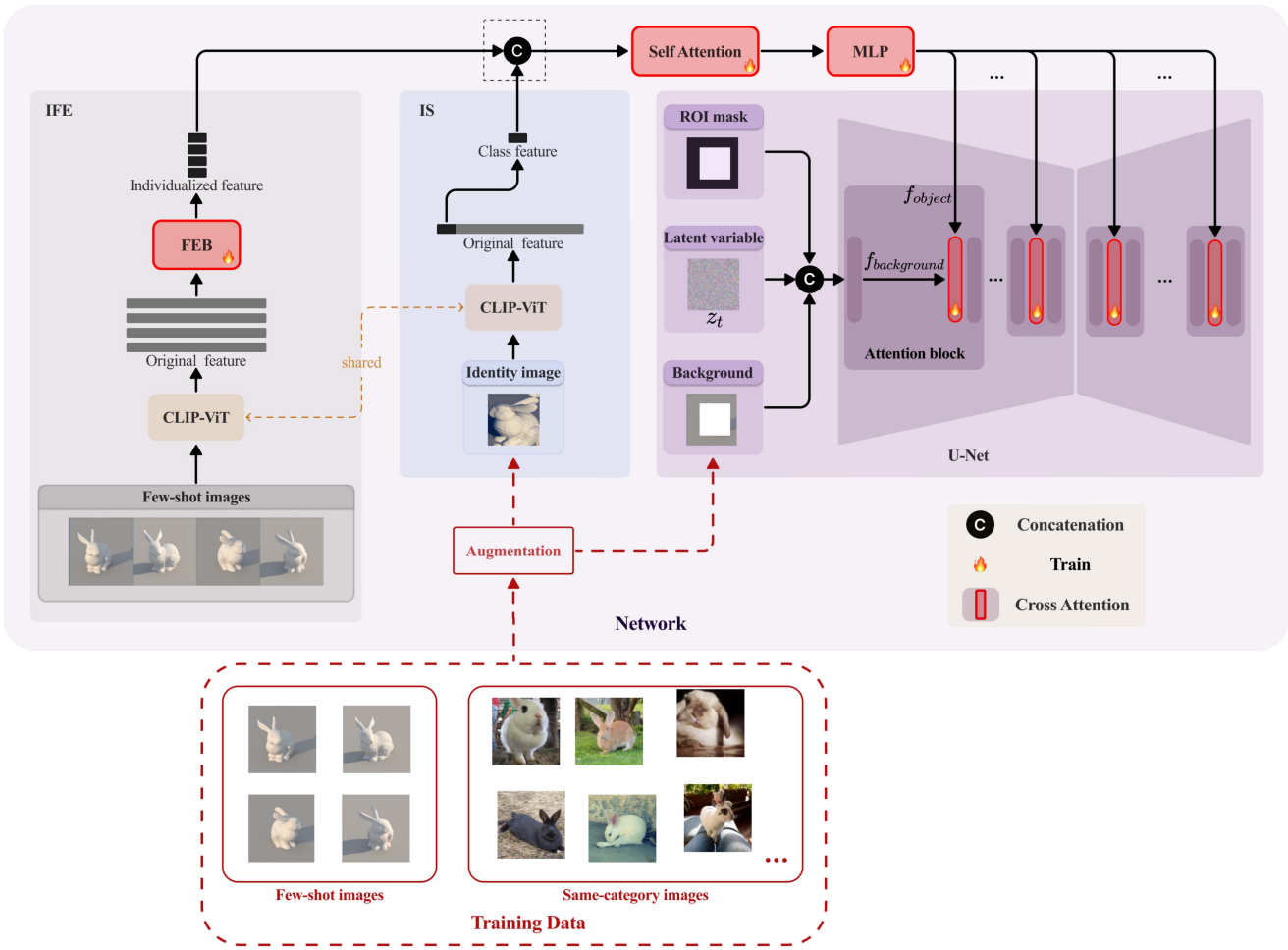

Fig. 3. The proposed network architecture and the pipeline of training. The detailed architecture of our network includes an IFE module, an IS module, and a U-Net. The parameters of the modules with red borders and flame symbols are trained, which are represented as implicit parametric representations of the target object.

the individualized feature, containing local detailed features of the target object comprehensively. The IFE module provides the model exhibiting sufficient detailed characteristics and demonstrating excellent authenticity. The U-Net, parameterized by θ, denoises the noisy latent representation by predicting the noise in the model. The U-Net contains several attention blocks and each block consists of a self-attention layer, a cross-attention layer, and a linear layer.

1) Object Information Injection: Existing methods [3], [5] cannot achieve an excellent performance of detail and harmony. The major reason is that the injection manners are limited and object information cannot be effectively embedded in the pre-trained generative model and interact with background scene information well. In our network, the IFE and IS modules extract the identity and individualized features of the object specifically. We concatenate the two features and feed them into a self-attention and multilayer perceptron (MLP). Then, we inject the features extracted by the MLP into attention blocks in U-Net. The injected features contain individual properties of the target object, which are represented as f_{object}. Correspondingly, the latent variable, the background, and the ROI mask are fed into the attention block to be denoted as $f_{background}$, which contains the information of the background scenes. The information of the target object as well as the background context information are injected into the cross-attention layer of U-Net, enabling the model to learn the correlation information from the target object and the background scene through the attention mechanism. The keys K and values V are derived from f_{object}, while the queries Q are obtained from $f_{background}$. The operation is formulated as follows:

$$Q = W_q f_{background},\ K = W_k \times f_{object},\ V = W_v \times f_{object}$$
$$F_{att} = \frac{QK^T}{\sqrt{d}}, \qquad F_o = softmax(F_{att})V, \tag{1}$$

where W_q, W_k, and W_v are the query, key, and value matrices, respectively, F_{att} is the attention feature, and F_o is the output feature.

It is worth noting the dimension of f_{object}. Suppose we only extract the class feature as object information without the IFE module. $f_{object} = [c_1 \cdots c_k]$, which is a $1 \times k$ vector. $f_{background}$ is an $m \times n$ vector, and QK^T is calculated as an $m \times 1$ vector, leading to all $softmax(F_{att})$ values being 1. We can conclude as follows:

$$F_o = softmax\left(F_{att}\right) V = V,\ where\ f_{object} = [c_1 \ldots c_k]\,. \tag{2}$$

For backpropagation, the network only optimizes W_v without W_q or W_k, and the injection of object information is seriously obstructed. Therefore, it is necessary to expand the dimension of f_{object} to $t \times k$, where $t > 1$. After introducing the IFE module, the individualized feature is extracted as a $num \times k$ vector, where num is the number of few-shot images. Then, the individualized feature is concatenated with the class feature, resulting in a $(num + 1) \times k$ vector. We confirm that $num \geq 1$ and $(num + 1) > 1$.

2) Implicit Parametric Representation: In our proposed model, we focus on optimizing the parameters of some important modules while keeping the remaining parameters of the model locked. As illustrated in Fig. 3, the modules to be trained are denoted by red boxes with flame symbols, including four parts: the FEB, the self-attention, the multilayer perceptron (MLP), and all cross-attention layers in U-Net.

Specifically, the FEB is a critical component of the IFE module that extracts and aggregates the original individualized features to obtain more comprehensive information. The self-attention and MLP extract high-dimensional features from all the features as object information. The cross-attention layers serve as interactive interfaces with object information in U-Net. We observe that the above four modules effectively control the extraction and injection of object information in our network, and they only occupy twenty percent of all the parameters. Therefore, we fine-tune the parameters of these modules and divide all the parameters into two parts: trained parameters and locked parameters. There is a one-to-one correspondence between trained parameters and target objects, and we refer to them as the implicit parameterized representations for objects. During the insertion process, we replace the implicit parameters of objects in the model to insert different objects into background scenes.

C. Training

1) InsertSet: We propose a practical dataset for training and testing, named InsertSet, and the training data consists of two parts: few-shot images and same-category images, which are shown in Fig. 3.

(a) Few-shot images. The few-shot images in the training data are the same as those used in the IFE module. For real objects, object images are obtained by multimedia devices such as cameras, and they are captured from multiple views to provide more comprehensive details of the target object. For virtual objects, object images are rendered by graphic design software such as 3ds Max. By randomly changing the lighting angle and intensity of the environment, varied illumination can be obtained from few-shot images. The object bounding boxes are detected by Faster R-CNN [43]. In this paper, we provide sample images and bounding boxes of real and virtual target objects, and additional data on objects can be created via customization.

(b) Same-category images. To provide richer global semantics and avoid overfitting in fine-tuning, we add same-category images of the target object to the training data. In our work, we build a set of same-category images that comprises 101 common categories consisting of a total of 20,882 data pairs (object images and bounding boxes). Each category contains approximately 200 to 400 data pairs. The object images are collected from OpenImage [44], COCO [45], and the Internet. We carefully select the desired images, especially those containing single, clear, moderately sized, unobstructed target objects. The bounding boxes of objects are detected by Faster R-CNN [43] or manually annotated.

For the testing dataset in InsertSet, we manually draw reasonable ROI masks for 1,000 background scenes to generate

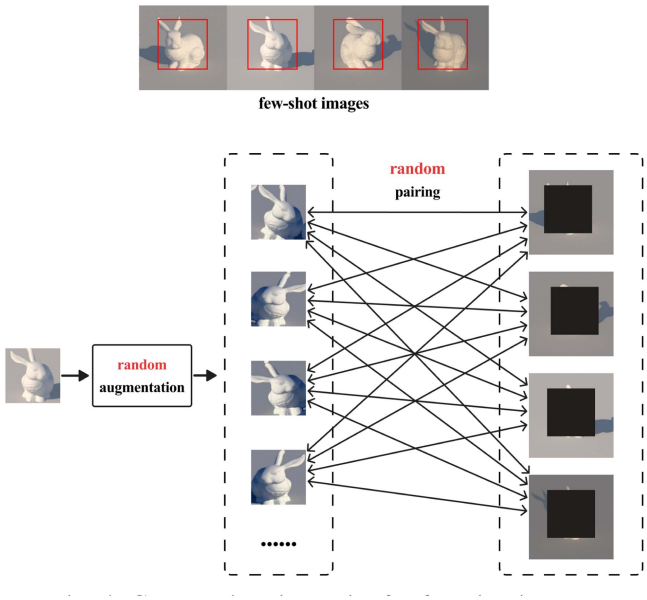

Fig. 4. Constructing data pairs for few-shot images.

semantically correct insertion images. These images encompass a wide range of background environments, such as indoor, outdoor, real, virtual, and cartoon images, showcasing remarkable diversity.

2) Data Augmentation: To enable the model to generate inserted results with posture adaptability and harmony, such as correct light and shadow, we develop an effective data augmentation method of few-shot images and same-category images to construct a broad array of object-scene combinations, which offers rich lighting and shadow information required for seamlessly inserting objects into backgrounds. As shown in Fig. 3, we randomly select an image I from the dataset and use a bounding box mask M to divide it into a foreground image I_f and a background image I_b. For foreground same-category images and few-shot images, we follow the PbE model [3] and apply diverse transformations with random probability to generate various foreground object images, called identity images I_{id}. The transformations used in our method are the same as those of PbE, including flipping, rotation, blurring, and elastic transforms. Finally, the background images and the identity images are combined to construct a series of data pairs $\{I_{id}, I_b\}$ for fine-tuning.

Specifically, we observe that the number of few-shot images is markedly lower than that of same-category images, which fails to provide sufficient information of the target object during fine-tuning. In order to increase the diversity of object-scene combinations and balance the number of few-shot images and same-category images, we take each image in background few-shot images and a randomly select augmented foreground few-shot image to construct a data pair, as shown in Fig. 4. With this method, 3-10 few-shot images of an object are augmented to 200 data pairs.

3) Self-Supervised Learning: It is difficult to obtain Ground-truths of insertion, and we treat the image I as the inserted result in self-supervised learning. The background image and the ROI mask are converted into low-dimensional information z_0 by the encoder, and we add t times ϵ to z_0 and obtain the result z_t, which can be seen as a mixed distribution of noise and the target object. The identity image I_{id}, few-shot object images I_{few}, and ROI mask M are treated as conditions $c(I_{id}, I_{few}, M)$. z_t is fed into the prediction model ϵ_θ to obtain the output $\epsilon_\theta(z_t, c(I_{id}, I_{few}, M), t)$. For each object, the fine-tuning loss L_{obj} of our diffusion model can be represented as

$$L_{obj} = \mathbb{E}_{\epsilon, z_0, c, t}\left[w_t \left\|\epsilon - \epsilon_\theta\left(z_t, c\left(I_{id}, I_{few}, M\right), t\right)\right\|_2^2\right], \quad (3)$$

where w_t is a time-dependent weight on the loss. Given a random number of t, the prediction model ϵ_θ can learn to establish a correspondence between any mixed distribution of the target object and the Gaussian distribution during fine-tuning.

D. Inserting Multi-Objects Into Large-Scale Scenes

Inserting objects into high-resolution backgrounds involves various challenges, including image distortion and time costs. In this paper, we extend our insertion method for high-resolution large-scale scene images such as 2k and 4k images based on local regions. Inserting objects into high-resolution image scenes consumes almost the same computational resources and time as accessing low-resolution images.

The method is illustrated in Fig. 5. The information of background scenes is crucial for insertion. We consider that the harmony of the inserted result is mainly reflected between the object and its surrounding background image. Therefore, we crop a 512x512 region surrounding the target object to replace the entire background scene. The local region and the target object are fed into our proposed model to generate the inserted result, and we replace the corresponding region in the original high-resolution image with the result. Since the pixel values of the content generated by the diffusion model exhibit differences from the original image, typically within the range of 0-1 grayscale values. The inserted result shows subtle differences at the edges of the ROI after cropping and replacing. Therefore, we use the morphological gradient to select the area and apply the Gaussian smoothing filter to eliminate it. These operations are code integration in our work.

Our insertion method based on local region achieves outstanding insertion results in high-resolution large-scale scene images, as shown in Fig. 6. In the 2K indoor living room scene, we insert multiple objects (including a teddy bear, a cat, and a wooden pot) into appropriate positions. The results demonstrate that the three objects are seamlessly inserted into the background scene and exhibit excellent performance in posture adaptability, harmony, and details.

IV. Experiments

A. Implementation Details

Our experimental setup includes a 13th Gen Intel(R) Core(TM) i5-13600KF CPU, 32 GB of RAM, and an NVIDIA GeForce RTX 4090 GPU. The environment comprises CUDA version 11.3, Python version 3.8, and PyTorch version 1.11.0.

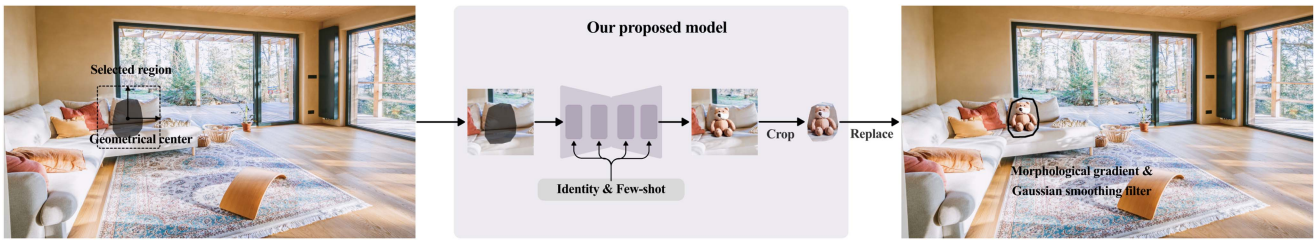

Fig. 5. The steps of inserting an object into a large-scale scene. First, we calculate the geometric center of the ROI in the large-scale image and center crop the image with 512×512 resolution. Then, we replace the corresponding region in the original large-scale image with the result generated by the proposed model. Finally, we use the morphological gradient to select the defective area and apply the Gaussian smoothing filter to eliminate it.

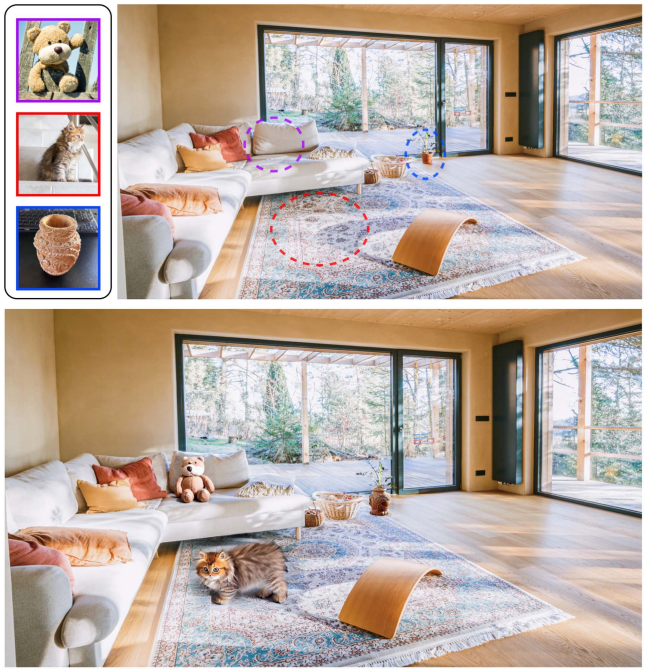

Fig. 6. The results of inserting multiple objects into a large-scale scene. We insert a teddy bear, a cat, and a wooden pot into appropriate positions in a high-resolution large-scale image scene.

We resize all images to the 512×512 resolution, and the batch size is 1. The IS module with information bottleneck in our model is the same as the PbE model [3], we use the PbE as the base model, which contains additional channels for ROI masks and background images. Then we construct the IFE module with information injection to replenish features and introduce the augmented few-shot images to fine-tune the partial weights of the model on InsertSet. For fine-tuning, we select 3-10 few-shot images and 200-400 same-category images for each target object. We train the model with 4,000 iterations for target objects with complex structures and textures (most real objects), which takes approximately 14 minutes, and we train the model with 2,000 iterations for target objects with simple structures and textures (most virtual objects), which takes approximately 7 minutes. We adopt a 50-step PLMS sampler during inference, which requires approximately 3-4 seconds.

B. Evaluation Metrics

The task of object insertion aims to achieve authentic and seamlessly integrated results. Therefore, we follow the DreamBooth [27] model to calculate the CLIP score [37] and DINO score [46], as these metrics could reflect the consistency and similarity between the generated region and the target object. The FID score [47] evaluates the authenticity of the inserted results. In addition, around 20 anonymous voters are asked to rate the similarity, harmony, adaptability, and authenticity of the insertion results, and the average values of all voters called human vision (HV) score.

C. Comparative Experiments

We conduct comparative experiments, including two fine-tuning-based baselines (Textual Inversion (TI) [6], Custom Diffusion (CD) [7]) and three feed-forward models (Paint by Example (PbE) [3], ControlCom (CCom) [4], Anydoor [5]).

Textual Inversion [6] and Custom Diffusion [7] are proposed to perform the text-to-image task. For the task of object insertion, the two fine-tuning-based methods need to be combined with the blended diffusion [28], which is a general text-to-image insertion model. Specifically, for Textual Inversion [6], we employ the blended diffusion as the generation model and feed the learned embedding for inserted objects into the model. Since all the comparative data in our paper are 512-resolution, we apply the official pipeline provided by diffusers [48] to the Textual Inversion and use the parameters of Stable Diffusion v1.5 [24] for the blended diffusion. For Custom Diffusion [7], we also take the blended diffusion as the backbone model and use the official pipeline of Custom Diffusion provided by diffusers to train the specific parameters for the inserted objects. Since Paint by Example [3], ControlCom [4] and Anydoor [5] models are feed-forward methods for 512-resolution images, we use their open-source codes provided by authors for experimentation directly.

Specifically, we select 20 target objects including 10 real objects and 10 virtual objects, which cover a diverse range of fields, including living and nonliving objects, indoor and outdoor objects as well as cartoon objects. For each target object, we select 50 reasonable background scene images and corresponding ROIs for testing. Finally, we obtain 1,000 insertion results for each comparison model.

Fig. 7. Visual comparison of insertion results between the proposed model and 5 state-of-the-art insertion models. From left to right: target objects (including real objects and virtual objects), ROIs, the results of the Paint by Example (PbE) [3], Anydoor [5], ControlCom (CCom) [4], Textual Inversion (TI) [6], Custom Diffusion (CD) [7] and our proposed model.

1) Qualitative Analysis: Fig. 7 shows the comparison results of the 6 insertion models. PbE [3] model suffers from significant identity drift issues. For the objects with individualized characteristics, such as the wooden pot in the 1st row of the left column, and the foxman in the 4th row of the left column, the inserted results show a serious loss of details. Since Anydoor [5] injects the detailed information into the generated network by ControlNet [36] to preserve detailed features, it lacks high-level semantic interaction between the foreground object to be inserted and the background scene, resulting in the poor adaptability of the inserted object to the background. It can be seen that the guitar in the 7th row of the left column and the panda in the 10th row of the right column of Anydoor [5] fail to interact with the girl and the bear in the background scenes. Based on PbE [3], ControlCom [4] constructs an additional local enhancement module to preserve detailed features of the object. Due to the limitation of the local information injection method, the objects after insertion tend to have poor fidelity. As it can be seen, in the 9th row of the left column the inserted object is difficult to identify as a light tower. In the 8th row of the left column, the background area behind the penguin is modified by ControlCom [4]. As Textual Inversion [6] and Custom Diffusion [7] reasonably establish the connections between the objects and the modifier tokens that are new "words" in the embedding space of the blended diffusion model [28], they improve the adaptability of objects to the background images. However, the combination with the blended diffusion [28] makes their inserted objects suffer from the problem of identity drifts. For instance, the bird in the 6th row of the right column and the chair in the 3rd row of the right column lose their identity, and the dinosaur in the 7th row of the right column and the snake in the 9th row of the right column lack facial textures and colors. From the results, our model shows excellent harmony between objects and background scenes as well as the high fidelity of the inserted objects. Especially, for objects with distinctive individualized characteristics, such as the wooden pot in the 1st row of the left column, the cartoon foxman in the 4th row of the left column, and the monkey in the 7th row of the right column, the proposed model demonstrates significant advantages over other models.

2) Quantitative Analysis: We calculate the CLIP score and DINO score of the inserted results of the 6 models and present the scores in Table I, in which the bold numbers indicate the best evaluation values. Among all the comparison models, our model achieves the best CLIP score and DINO score. Especially, the DINO score of our model is 10.5% higher than that of the second-best model Anydoor [5]. For the FID scores, we only consider the inserted results of the 10 real objects and we evaluate a total of 500 inserted results for each comparison model. Since we have no Ground-truths of insertion, the FID score in Table I is calculated by comparing the distribution of

TABLE I
QUANTITATIVE ANALYSIS BETWEEN THE PROPOSED MODEL AND 5 STATE-OF-THE-ART INSERTION MODELS

Model	CLIP (↑)	DINO (↑)	FID (↓)	HV (↑)
Paint by example (PbE)	76.6619	53.6138	20.1576	2.7186
ControlCom (CCom)	62.5943	51.0840	**18.7870**	3.0104
Textual inversion (TI)	71.9841	51.2806	21.0122	3.1740
Custom diffusion (CD)	67.1280	59.0878	21.7092	3.5050
Anydoor	77.6401	64.5575	24.3018	3.7801
Ours	**79.1908**	**71.3763**	22.7116	**4.2906**

the inserted images with that of the original background images alternatively. Note that both Anydoor and our model achieve high CLIP and DINO scores, but not low FID scores. In contrast, ControlCom [4] has the lowest CLIP and DINO scores, but the best FID score. Combined with the visual comparison in Fig. 7, it can be seen that the FID scores of the models with poor fidelity of the inserted object are low, while the FID scores of the models with high fidelity are high. This is probably because we do not have Ground-truths when calculating FID, and the models with high fidelity produce insertion results quite different from the content of the original background image, subsequently, their pixel distributions are different, which causes the FID score to be high. Moreover, around 20 anonymous voters are asked to rate the similarity, harmony, adaptability, and fidelity of the insertion results on a scale of 1 to 5, where a higher value indicates a better result. The average values of all voters called human vision (HV) for the 6 models, are shown in Table I. It can be observed that our proposed model yields the highest value of human vision and achieves the most reasonable insertion results.

D. Ablation Experiments

1) The Object Information Injection Manner: To validate the effectiveness of the object information injection manner of our model, we design 5 different models based on the backbone (PbE [3]) for ablation experiments, as shown in Fig. 8.

Model A: We combine the backbone and the fine-tuning method from Custom Diffusion [7] directly, which only has the IS module in the experiment.

Model B: Since the input of the IS module and the background image together form a data pair for self-supervised learning in our method, the IS module cannot be removed from our model during training. Consequently, we keep the IS module and the IFE module during training, and remove the IS module and only keep the IFE module during inference.

Model C: To avoid obstructing the injection of object information, we apply a naive method and add a shared feature to expand the dimension based on Model A.

Model D: We attempt to combine the features extracted by the IS module with the detailed features from the IFE module using the weighted addition method while expanding dimensions.

Model E: Inspired by the IP-Adapter model [33], we use a decoupled cross-attention structure, where the class features extracted by the IS module and the detailed features extracted by the IFE module are input into different attention modules. Then the weighted outputs of the attention modules are added as the condition. Only the newly added modules for detailed features (in red color) are trained while the pre-trained model for the IS branch is frozen.

Our model: We use the concatenation method to combine the IFE and the IS module in our proposed network, which achieves dimensional expansion and injects features of objects comprehensively.

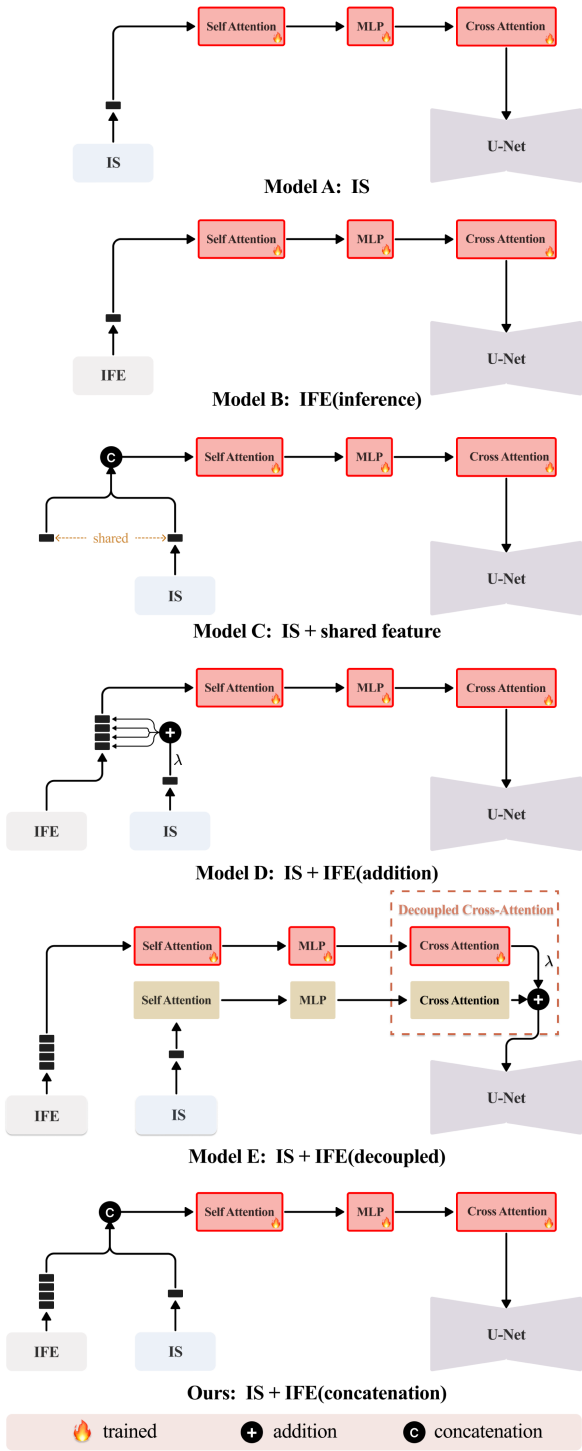

Fig. 8. The structures of 6 models in the ablation experiment titled the object information injection manner.

Fig. 9. Visual comparison of insertion results in the ablation experiment titled the object information injection manner. From left to right: target objects (including real objects and virtual objects), ROIs, the results of, Model A, Model B, Model C, Model D, Model E, and our model.

Fig. 9 shows comparison results for the 5 ablation models and our model. The results of Model A show minimal differences compared to the backbone [3], with only minor changes in facial areas, such as the horse in the 4th row of the right column and the monkey in the 5th row of the right column. This is because directly fine-tuning the backbone with the IS module seriously obstructs the injection of object information. From the results of Model B, we believe that the features extracted by the IFE module contain sufficient information for the target object, making it possible for the model to achieve decent insertion results in inference, but the model lacks the class features provided by the IS module, leading to some results with identity deviation, such as the teddybear in the 2nd row of the left column, and the horse in the 4th row of the right column. The results of Model C demonstrate that concatenating the shared feature to expand dimensions effectively injects information into the model. However, for the objects with distinctive characteristics, the results lose details and exhibit disharmonies, such as the foxman in the 4th row of the left column, and the bird in the 6th row of the right column. Model D, Model E, and our model incorporate the same IFE module by different combination methods. Model E retains the original parameters of the base model and fine-tunes the parameters of the newly added IFE module branch, performing better in posture adaptability but worse in similarity. For instance, the penguin in the 8th row of the left column interacts with the background scene well and has differences in appearance compared to the target object. Model D obtains excellent inserted results but still has problems in localized areas of results, such as the tail of the dog in the 2nd row of the right column, and the face of the panda in the 10th row of the right column. From the results, we can observe that our proposed model performs optimal performance in posture adaptability and harmony of results while maintaining the details of objects.

TABLE II
QUANTITATIVE ANALYSIS OF 6 MODELS IN THE ABLATION EXPERIMENT TITLED THE OBJECT INFORMATION INJECTION MANNER

Model	Structure	CLIP (↑)	DINO (↑)	HV (↑)
PbE	Backbone	76.6619	53.6138	2.7186
A	IS	76.7811	53.4576	2.8205
B	IFE (Inference)	74.3931	54.1168	3.3242
C	IS + Shared feature	78.0723	66.6321	3.9805
D	IS + IFE (Addition)	78.2991	66.9590	4.1365
E	IS + IFE (Decoupled)	77.6099	61.5815	3.8702
Ours	IS + IFE (Concatenation)	**79.1908**	**71.3763**	**4.2906**

Furthermore, we calculate the CLIP, DINO, and HV scores of the results of the 6 models. The scores are presented in Table II, in which the bold numbers indicate the best values. From the results of the backbone and Model A, we consider that the simple combination of the PbE model and cross-attention fine-tuning from Custom Diffusion cannot effectively inject the object information into the generated model. From the results

Fig. 10. Diverse insertion results with different random seeds of the initial Gaussian noise in inference. From left to right: backgrounds and 5 insertion results with 5 random seeds.

of Model A and Model C, it can be inferred that dimension expansion is important for information injection in the model. The results of Model B and our model indicate that the IS module effectively avoids identity deviation in inference and improves the quality of results. From the results of Model C and our model, it is evident that the IFE module is beneficial for fully extracting the information from the few-shot images and it further improves performance in details and harmony. In particular, by comparing Model D, Model E, and our model, we found that the concatenation for the IFE module is a simple and suitable method for integrating detailed features of objects.

2) Diversity of Model Generation Results: Our method could produce multiple insertion results with the target object in different poses by adjusting the random seeds of the initial Gaussian noise of diffusion inference. Fig. 10 shows the results of inserting a cartoon monkey into 2 background images by respectively using 5 different seeds. As can be seen, 5 poses of the monkey in each row are produced, demonstrating the rich diversity of the insertion.

Additionally, in our method, both the target object features and the background context features are injected into the cross-attention modules of LDM [24], enabling the model to learn the correlation information from the target object and the background scene through the attention mechanism. Therefore, our method excellently adapts the target objects with various poses, views, and lighting to different background scenes. Fig. 11 shows all few-shot images of a cat, some same-category images of the cat, and the results generated by our model. It can be seen that our method flexibly adapts the cat to different background scenes while maintaining the individual characteristics of the cat. Moreover, we observe that the inserted results are different from few-shot images and same-category images in terms of the cat's poses and lighting. Specifically, we carefully compare 200 same-category images of the target object with our results, of which 10 are listed in Fig. 11. It indicates that our method effectively injects the texture, posture, lighting, and expression from the training data into the base model [3], and our inserted results are not limited by the few-shot images and same-category images.

3) Consistency of Lighting and Shadows: We conduct an ablation experiment to validate the consistency of lighting and shadows. In the experiment, we compare the adaptability of the backbone, the model fine-tuned with few-shot images (+F), and

Fig. 11. All few-shot images of a cat, some same-category images of the cat, and our insertion results.

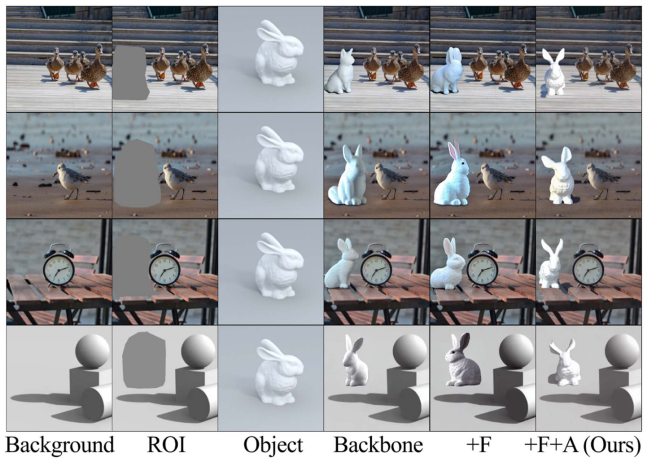

Fig. 12. Comparison results of inserting a CG bunny in 4 scenes with different methods. From left to right: background images, ROIs, object, the results of backbone, the results of the model fine-tuned with few-shot images (+F), and the results of the model fine-tuned with augmented few-shot images (ours).

the model fine-tuned with augmented few-shot images (ours). To observe the lighting conditions of objects intuitively, we choose a CG bunny with a glossy surface as the target object and selected background scenes, including real scenes and virtual scenes. The experimental results are shown in Fig. 12.

The results of the backbone not only lack texture details but also lose cast shadows of the inserted object. Compared with the backbone, the texture and lighting of the CG bunny generated by fine-tuning only with few-shot images (+F) are more realistic, but the shadows cast by the bunny are still lacking. Our model of fine-tuning with few-shot images and data augmentation

Fig. 13. The insertion result of our method for different lighting conditions in background scenes. We select a penguin doll as the target object and select background scenes of light sources with different directions and intensities.

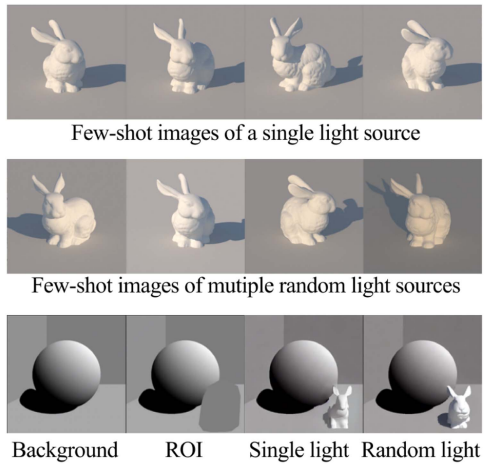

Fig. 14. The inserted results with the few-shot images in two different lighting conditions. The 1st row images are rendered by a single lighting source and the 2nd row images are rendered by random lighting sources. The insertion results are shown in the 3rd row.

(+F+A) successfully generates realistic lighting and shadows of the inserted bunny and simultaneously maintains its color and textures. It is contributed that our data augmentation based on few-shot images constructs a broad array of object-scene combinations and offers rich lighting and shadow information required for seamlessly inserting objects into backgrounds.

4) Illumination of Background Images: Furthermore, we conduct an experiment to analyze the robustness of background lighting on the inserted results, and the inserted results are shown in Fig. 13. From the experimental results, it is evident that our method can generate accurate object lighting and shadows under various light source backgrounds, showing outstanding performance and robustness in adapting to background scenes.

5) Illumination of Few-Shot Images of Objects: We experiment to investigate the impact of lighting conditions of few-shot images on our insertion results. Fig. 14 shows the comparison results with few-shot images of two different lighting conditions. The few-shot images in the 1st row are illuminated by a single left-side lighting source and the ones in the 2nd row are illuminated by random lighting sources. We take a background image with the right-side lighting direction for testing, in which the lighting direction is opposite to the left-side lighting direction of few-shot images in the first row. The insertion results are shown in the 3rd row. It can be seen from the result generated with few-shot images in random lighting sources has better performance on the consistency of lighting and shadows than that with the few-shot images in a single light source. The results indicate that the insertion results of our method are affected by the lighting information of the few-shot images of the object.

6) Number of Same-Category Images: The use of same-category images (SCIs) makes our model focus on the global features of same-category objects, enabling it to capture enough semantics of objects with the same category as the inserted object. Moreover, we introduce the same-category images of the object as the training data to prevent overfitting while maintaining the semantic information. As shown in Fig. 15, the

Fig. 15. Insertion results of our model trained without SCIs compared to our model trained with SCIs. From left to right: the background scene, the ROI, the target object, the result of the backbone, the result of our model trained without SCIs, and the result of our method with SCIs.

TABLE III
QUANTITATIVE ANALYSIS OF THE IMPACT OF HYPERPARAMETERS ON INSERTED RESULTS

Influence factor	Number	CLIP (↑)	DINO (↑)
Same-category Image	w/o	81.4806	72.1841
	100	82.6619	75.1325
	200	82.1031	71.7950
	300	83.0494	75.8380
	400	**84.6550**	**76.2287**
Few-shot image	1	77.6363	42.4893
	4	**80.3431**	**64.4774**
	7	80.0300	60.2201
	10	79.8531	61.5671
Train Iteration	1000	76.9897	77.5590
	2000	75.7063	86.2776
	3000	78.9819	89.8284
	4000	**79.0575**	**90.9688**
	5000	75.1866	83.8998

insertion result suffers from unreasonable occlusion between the bunny and the red bucket and the inserted bunny takes on a pose identical to one of the few-shot images. On the contrary, our method effectively mitigates these problems and inserts the bunny into the red bucket harmoniously.

Furthermore, we conduct comparative experiments using 0, 100, 200, 300, and 400 SCIs and apply the five trained models to insert the same object into 100 different background images from InsertSet. The CLIP and DINO scores for the inserted results are shown in Table III, in which the bold numbers indicate the best evaluation values. It is demonstrated that the training dataset

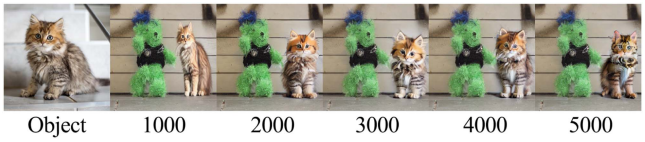

Fig. 16. Comparison results of object insertion with different training iterations, from left to right: background scenes, ROIs, target object, and the results of models trained for 1000, 2000, 3000, 4000, and 5000 iterations.

with SCIs enhances the performance of the inserted results. Considering the cost of image collection, we use 200-400 SCIs as the training data in our work.

7) Number of Few-Shot Images: The experiment is conducted to analyze the impact of the number of few-shot images on the inserted results. We conduct four comparative experiments, respectively inputting 1, 4, 7, and 10 few-shot images for training, and select 100 background images from InsertSet for testing. The CLIP and DINO scores are presented in Table III, in which the bold numbers indicate the best evaluation values. it can be seen that with the growing number of few-shot images, the insertion results fluctuate. Generally, training the proposed model with 3 to 10 few-shot images is suitable.

8) Number of Training Iterations: To verify the reasonability of training iterations in our method, we train the models with iterations of 1,000, 2,000, 3,000, 4,000, and 5,000. The results are presented in Fig. 16. As the number of training iterations increases, the inserted results gradually adapt to the background scenes and resemble the target object in shape. The model trained for 4,000 iterations generates the optimal inserted result, but the result exhibits an unnatural appearance such as the face and body of the cat when the number reaches 5,000. We consider that excessive training leads to model collapse, causing the model to focus on the features of the target object and lose its fundamental generative capability. We select 100 background images from InsertSet for testing and the CLIP and DINO scores of the results are shown in Table III, in which the bold numbers indicate the best values. The model trained for 4,000 iterations exhibits the optimal performance, aligning with the results illustrated in Fig. 16.

9) The Ways of Acquiring Same-Category Images: We experiment to investigate the impact of different ways of acquiring same-category images (SCIs) on insertion results. We train the models with real SCIs collected from InsertSet and generated SCIs by the pre-trained diffusion model (SD-v1.4 [24]) respectively. Fig. 17 shows some SCIs that are collected from InsertSet and generated by the pre-trained diffusion model. The distribution of generated images and the distribution of a set of real images are different, and the Fréchet inception distance (FID) of them is calculated as 22.3524. Then, we use the two trained models to insert a cat into 50 different background scenes.

The comparison results of insertion with different acquired ways of SCIs are demonstrated in Fig. 18. It can be seen that the inserted results of the model trained with generated SCIs maintain the individual details of the cat well and achieve excellent performance in posture adaptability and harmony between the cat and background scenes. However, there is a small difference

Fig. 17. Some same-category images of a cat acquired in different ways. The images on the left are generated by the pre-trained diffusion model [24], and the images on the right are collected from InsertSet.

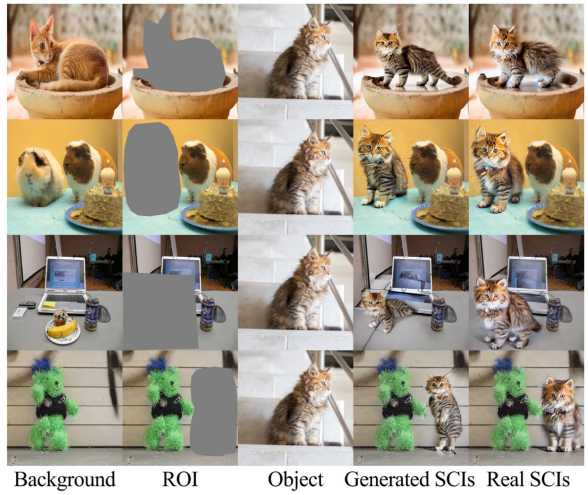

Fig. 18. Comparison results of insertion with different acquired ways for SCIs. From left to right: backgrounds, ROIs, target objects, the inserted results of the model trained with generated SCIs, and the inserted results of the model trained with real SCIs (the proposed method).

between the insertion results of the two models such as the limbs of the cat in the third and fourth rows of Fig. 18. The main reason is that the defects of the generated SCIs (some generated images have blurred and distorted limbs) make the model learn incorrect features. Additionally, the difference in the distribution of the generated image and the real image causes the model trained with generated SCIs to fail to fit real background images. Furthermore, we calculate the CLIP and DINO scores of all insertion results. The CLIP and DINO scores of the model trained with generated SCIs are 78.6613 and 86.6327, and the CLIP and DINO scores of the model trained with real SCIs are 79.6763 and 92.0616. The scores indicate that the model trained with generated SCIs achieves slightly lower CLIP and DINO scores than the model trained with real SCIs. With the quantitative and qualitative comparison results, we believe that the pre-trained diffusion model can serve as a good backup way to acquire same-category images, especially for objects that are not present in general datasets.

E. Limitations

One limitation of our method is that the insertion is performed based on two-dimensional images and lacks explicit 3D perception of object images and background images. As shown in

Fig. 19. A failure case of the proposed method.

Fig. 19, since the insertion model cannot perceive the depth of the background scenes, when we place a bunny in different locations in a scene, our method does not generate bunnies looking small in the distance and big on the contrary. Currently, we obtain the desired sizes of inserted objects by constraining the ROI masks.

In addition, for the insertion of objects with higher resolution such as 2 k/4k, our method is based on a fine-tuning method, and the resolution of the generated result is limited by the base generative model [24]. Therefore, our method is only applicable to high-resolution large-scale scene images where the inserted object has a small proportion in the background scenes.

V. Conclusion

In this paper, we introduce a new insertion method that first captures the semantics of the object to be inserted, followed by fine-tuning to replenish its specific details. Specifically, we construct a novel IFE module that extracts individual features from few-shot images of the object, ensuring the model has comprehensive information to generate high-quality inserted results. In addition, we develop an effective data augmentation to enable the model to learn to fit objects with more lighting conditions in the background scenes. These parameters obtained by fine-tuning serve as implicit parameter representations for objects. Our method is capable of inserting objects into any background scene and maintains excellent adaptability and harmony between the inserted objects and background scenes. It automates the traditional image editing workflow where artists perform tedious transformations upon image assets for coherent image blending.

In future work, we aim to address the shortcomings in perceiving the depth of the background scene. In addition, we plan to expand the application of our method to object insertion in videos and cartoon animation production, breaking through new and interesting challenges in real-time performance.

References

[1] H. Chen et al., "Hierarchical dynamic image harmonization," in *Proc. 31st ACM Int. Conf. Multimedia*, 2023, pp. 1422–1430.
[2] Y. Hong, L. Niu, and J. Zhang, "Shadow generation for composite image in real-world scenes," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 914–922.
[3] B. Yang et al., "Paint by example: Exemplar-based image editing with diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 18381–18391.
[4] B. Zhang et al., "ControlCom: Controllable image composition using diffusion model," 2023, *arXiv:2308.10040*.
[5] X. Chen, L. Huang, Y. Liu, Y. Shen, D. Zhao, and H. Zhao, "AnyDoor: Zero-shot object-level image customization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 6593–6602.
[6] R. Gal et al., "An image is worth one word: Personalizing text-to-image generation using textual inversion," 2022, *arXiv:2208.01618*.
[7] N. Kumari, B. Zhang, R. Zhang, E. Shechtman, and J.-Y. Zhu, "Multi-concept customization of text-to-image diffusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 1931–1941.
[8] P. Yu et al., "ShadowMover: Automatically projecting real shadows onto virtual object," *IEEE Trans. Vis. Comput. Graphics*, vol. 29, no. 5, pp. 2379–2389, May 2023.
[9] H. Cheng, C. Xu, J. Wang, Z. Chen, and L. Zhao, "Fast and accurate illumination estimation using LDR panoramic images for realistic rendering," *IEEE Trans. Vis. Comput. Graph.*, vol. 29, no. 12, pp. 5235–5249, Dec. 2023.
[10] J. Zhao, A. Chalmers, and T. Rhee, "Adaptive light estimation using dynamic filtering for diverse lighting conditions," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 11, pp. 4097–4106, Nov. 2021.
[11] F. Zhan et al., "EMLight: Lighting estimation via spherical distribution approximation," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 3287–3295.
[12] H. Kato, Y. Ushiku, and T. Harada, "Neural 3D mesh renderer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3907–3916.
[13] W. Cong et al., "DoveNet: Deep image harmonization via domain verification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8394–8403.
[14] K. Sofiiuk, P. Popenova, and A. Konushin, "Foreground-aware semantic representations for image harmonization," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2021, pp. 1620–1629.
[15] D. Liu, C. Long, H. Zhang, H. Yu, X. Dong, and C. Xiao, "ARShadowGAN: Shadow generative adversarial network for augmented reality in single light scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8139–8148.
[16] L. Zhang, T. Wen, J. Min, J. Wang, D. Han, and J. Shi, "Learning object placement by inpainting for compositional data augmentation," in *Proc. 16th Eur. Conf. Comput. Vis.*, Glasgow, U.K., Springer, 2020, pp. 566–581.
[17] J. Liang, L. Niu, and L. Zhang, "Inharmonious region localization," in *Proc. 2021 IEEE Int. Conf. Multimedia Expo*, 2021, pp. 1–6.
[18] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4104–4113.
[19] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," *Commun. ACM*, vol. 65, no. 1, pp. 99–106, 2021.
[20] L. Zhang, T. Wen, and J. Shi, "Deep image blending," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2020, pp. 231–240.
[21] Y. Song et al., "ObjectStitch: Generative object compositing," 2022, *arXiv:2212.00932*.
[22] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," 2020, *arXiv: 2010.02502*.
[23] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 6840–6851.
[24] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10684–10695.
[25] A. Nichol et al., "GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models," 2021, *arXiv:2112.10741*.
[26] Y. Cao, X. Meng, P. Mok, X. Liu, T.-Y. Lee, and P. Li, "AnimeDiffusion: Anime face line drawing colorization via diffusion models," 2023, *arXiv:2303.11137*.
[27] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 22500–22510.

[28] O. Avrahami, D. Lischinski, and O. Fried, "Blended diffusion for text-driven editing of natural images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18208–18218.
[29] L. Han, Y. Li, H. Zhang, P. Milanfar, D. Metaxas, and F. Yang, "SVDiff: Compact parameter space for diffusion fine-tuning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 7323–7334.
[30] R. Gal, M. Arar, Y. Atzmon, A. H. Bermano, G. Chechik, and D. Cohen-Or, "Encoder-based domain tuning for fast personalization of text-to-image models," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 1–13, 2023.
[31] G. Yuan et al., "Inserting anybody in diffusion models via celeb basis," 2023, *arXiv:2306.00926*.
[32] Q. Wang, X. Bai, H. Wang, Z. Qin, and A. Chen, "InstantID: Zero-shot identity-preserving generation in seconds," 2024, *arXiv:2401.07519*.
[33] H. Ye, J. Zhang, S. Liu, X. Han, and W. Yang, "IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models," 2023, *arXiv:2308.06721*.
[34] Y. Wei, Y. Zhang, Z. Ji, J. Bai, L. Zhang, and W. Zuo, "ELITE: Encoding visual concepts into textual embeddings for customized text-to-image generation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 15943–15953.
[35] X. Zhang, J. Guo, P. Yoo, Y. Matsuo, and Y. Iwasawa, "Paste, inpaint and harmonize via denoising: Subject-driven image editing with pre-trained diffusion model," 2023, *arXiv:2306.07596*.
[36] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 3836–3847.
[37] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2021, pp. 8748–8763.
[38] M. Niemeyer and A. Geiger, "GIRAFFE: Representing scenes as compositional generative neural feature fields," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11453–11464.
[39] Y. Xu et al., "DisCoScene: Spatially disentangled generative radiance fields for controllable 3D-aware scene synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 4402–4412.
[40] R. Hachnochi et al., "Cross-domain compositing with pretrained diffusion models," 2023, *arXiv:2302.10167*.
[41] Y. Song et al., "IMPRINT: Generative object compositing by learning identity-preserving representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 8048–8058.
[42] S. Ryu, "Low-rank adaptation for fast text-to-image diffusion fine-tuning," 2022. [Online]. Available: https://github.com/cloneofsimo/lora
[43] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[44] A. Kuznetsova et al., "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *Int. J. Comput. Vis.*, vol. 128, no. 7, pp. 1956–1981, 2020.
[45] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. 13th Eur. Conf. Comput. Vis.*, Zurich, Switzerland, Springer, 2014, pp. 740–755.
[46] M. Caron et al., "Emerging properties in self-supervised vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9650–9660.
[47] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6629–6640.
[48] P. von Platen et al., "Diffusers: State-of-the-art diffusion models," 2022. [Online]. Available: https://github.com/huggingface/diffusers

Qi Zhang received the MS degree from Sichuan University, Chengdu, China, in 2021. He is currently working toward the PhD degree with Sichuan University, Chengdu, China. His research interests include image processing, augmented reality, and computer vision.

Guanyu Xing received the PhD degree from Zhejiang University, Hangzhou, China, in 2013, he was a visiting scholar of Temple University, Philadelphia, Pennsylvania, in 2016. He is currently an associate professor with the School of Cyber Science and Engineering, Sichuan University, Chengdu, China. His research interests include augmented reality, computer vision, and multimedia.

Mengting Luo received the BS degree from Sichuan Agricultural University, Yaan, China, in 2020. She is currently working toward the PhD degree with Sichuan University, Chengdu, China. Her research interests include image processing, artificial intelligence, and computer vision.

Jianwei Zhang received the PhD degree from Sichuan University, Chengdu, China, in 2008. He has taught and conducted research with Sichuan University since 1993. He has published more than 30 articles. His research interests include air traffic management, and intelligent image analysis and processing. He received the National Science and Technology Progress Award in China.

Yanli Liu received the PhD degree from Zhejiang University, Hangzhou, China, in 2010. Thereafter, she was a postdoctoral researcher with the French National Institute for Research in Computer Science and Control (INRIA). She is currently a professor with the College of Computer Science, Sichuan University, Chengdu, China. Her research interests include augmented reality, image/video processing, and computer vision.