PAIR Diffusion: A Comprehensive Multimodal Object-Level Image Editor

Vidit Goel[1,2,*] Elia Peruzzo[3,*,†] Yifan Jiang [4] Dejia Xu[4] Xingqian Xu[2,1]
Nicu Sebe[3] Trevor Darrell[5] Zhangyang Wang[1,4] Humphrey Shi[1,2]
[1]Picsart AI Research (PAIR) [2]SHI Labs @ Georgia Tech & UIUC [3]University of Trento [4]UT Austin [5]UC Berkeley

https://github.com/Picsart-AI-Research/PAIR-Diffusion

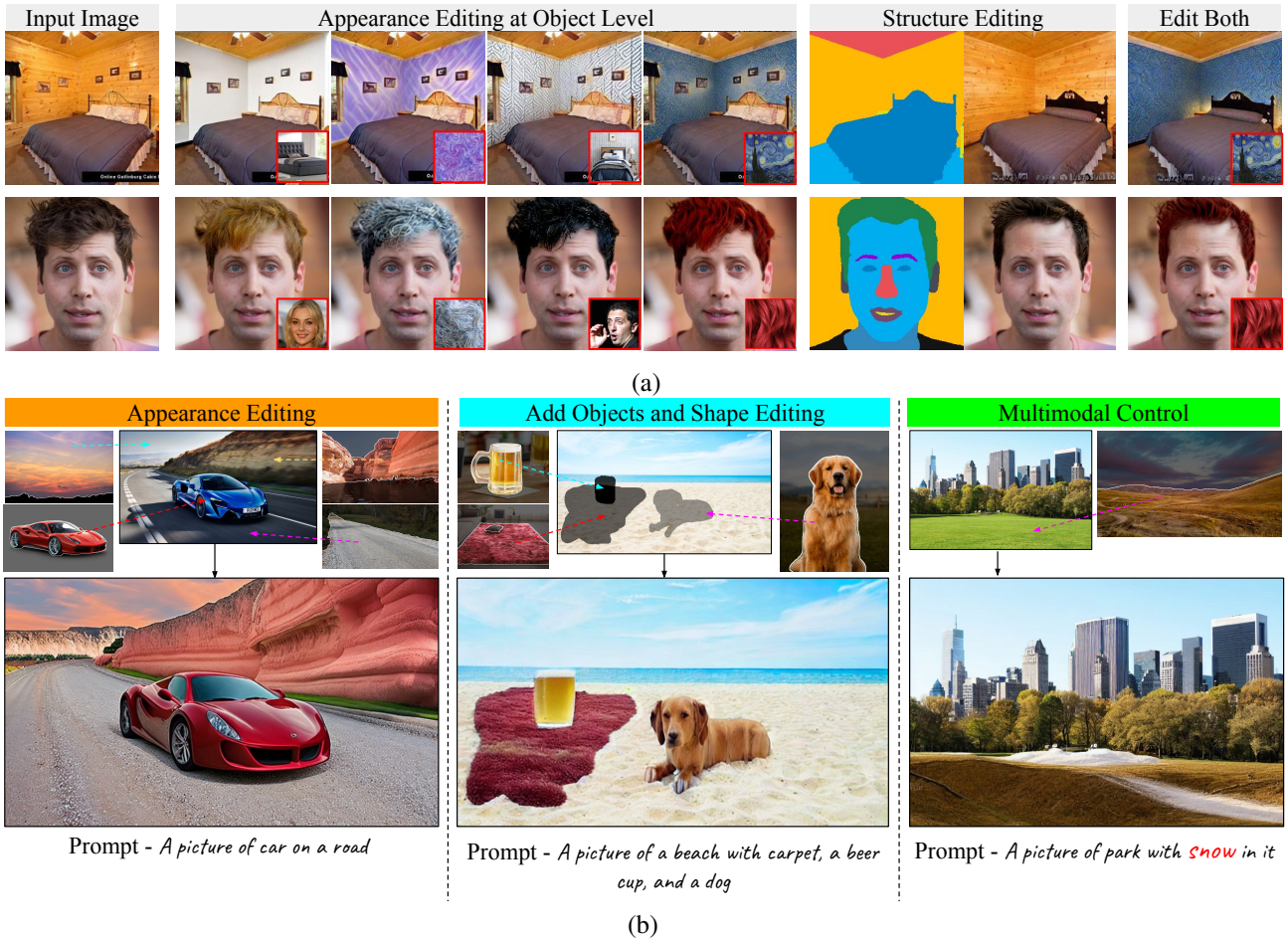

Figure 1. PAIR Diffusion framework allows the appearance and structure editing of an image at the object level. Our framework is general and can enable object-level editing capabilities in both (a) unconditional diffusion models and (b) foundational diffusion models. Using our framework with a foundational diffusion model allows for comprehensive in-the-wild object-level editing capabilities.

Abstract

Generative image editing has recently witnessed extremely fast-paced growth. Some works use high-level conditioning such as text, while others use low-level conditioning. Nevertheless, most of them lack fine-grained control over the properties of the different objects present in the image, i.e. *object-level image editing. In this work, we tackle the task by perceiving the images as an amalgamation of various objects and aim to control the properties of each object in a fine-grained manner. Out of these properties, we identify structure and appearance as the most intuitive to understand and useful for editing purposes. We propose* ***PAIR Diffusion****, a generic framework that enables a diffusion model to control the structure and appearance properties of each object in the image. We show that having control over the properties*

[*]Denotes equal contribution.
†Work performed while interning at Picsart AI Research.

2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 979-8-3503-5300-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/CVPR52733.2024.00822

2575-7075/24/$31.00 ©2024 IEEE
DOI 10.1109/CVPR52733.2024.00822

of each object in an image leads to comprehensive editing capabilities. Our framework allows for various object-level editing operations on real images such as reference image-based appearance editing, free-form shape editing, adding objects, and variations. Thanks to our design, we do not require any inversion step. Additionally, we propose multi-modal classifier-free guidance which enables editing images using both reference images and text when using our approach with foundational diffusion models. We validate the above claims by extensively evaluating our framework on both unconditional and foundational diffusion models.

1. Introduction

Diffusion-based generative models have shown promising results in synthesizing and manipulating images with great fidelity, among which text-to-image models and their follow-up works have great influence in both academia and industry. When editing a real image a user generally desires to have intuitive and precise control over different elements (*i.e*. the objects) composing the image, and to manipulate them independently. We can categorize existing image editing methods based on the level of control they have over individual objects in an image. One line of work involves the use of text prompts to manipulate images [2, 15, 24, 27]. These methods have limited capability for fine-grained control at the object level, owing to the difficulty of describing the shape and appearance of multiple objects simultaneously with text. In the meantime, prompt engineering makes the manipulation task tedious and time-consuming. Another line of work uses low-level conditioning signals such as masks Hu et al. [18], Patashnik et al. [34], Zeng et al. [58], sketches [50], images [5, 47, 54] to edit the images. However, most of these works either fall into the prompt engineering pitfall or fail to independently manipulate multiple objects. Different from previous works, we aim to independently control the properties of multiple objects composing an image *i.e.* object-level editing. We show that we can formulate various image editing tasks under the object-level editing framework leading to comprehensive editing capabilities.

To tackle the aforementioned task, we propose a novel framework, dubbed Structure-and-Appearance **Pair**ed **Diffusion** Models (**PAIR Diffusion**). Specifically, we perceive an image as an amalgamation of diverse objects, each described by various factors such as shape, category, texture, illumination, and depth. Then we further identified two crucial macro properties of an object: structure and appearance. Structure oversees an object's shape and category, while appearance contains details like texture, color, and illumination. To accomplish this goal, PAIR Diffusion adopts an off-the-shelf network to estimate panoptic segmentation maps as the structure, and then extract appearance representation using pre-trained image encoders. We use the extracted per-object appearance and structure information to condition a diffusion model and train it to generate images. In contrast to previous text-guided image editing works [1, 2, 8, 39], we consider an additional reference image to control the appearance. Compared to text prompts that, although conveniently, can only vaguely describe the appearance, images can precisely define the expected texture and make fine-grained image editing easier. Having the ability to control the structure and appearance of an image at an object level gives us comprehensive editing capabilities. Using our framework we can achieve, localized free-form shape editing, appearance editing, editing shape and appearance simultaneously, adding objects in a controlled manner, and object-level image variation (Fig. 1). Moreover, thanks to our design we do not require any inversion step for editing real images.

The novelty of our work lies in the way we formulate the image editing tasks that lead to a general approach to enable comprehensive editing capabilities in various models. We show the efficacy of our framework on unconditional diffusion models and foundational text-to-image diffusion models. Lastly, we propose multimodal classifier-free guidance to reap the full benefits of the text-to-image diffusion models. It enables PAIR Diffusion to control the final output using both reference images and text in a controlled manner hence getting the best of both worlds. Thanks to our easy-to-extract representations we do not require specialized datasets for training and we show results on LSUN and Celeb-HQ datasets for unconditional models, and use the COCO dataset for foundational diffusion models. To summarize our contributions are as follows:

- We propose PAIR Diffusion, a general framework to enable object-level editing in diffusion models. It allows editing the structure and appearance of each object in the image independently.
- The proposed design inherently supports various editing tasks using a single model: localized free-form shape editing, appearance editing, editing shape and appearance simultaneously, adding objects in a controlled manner, and object-level image variation.
- Additionally, we propose a multimodal classifier-free guidance, enabling PAIR Diffusion to edit images using both reference images and text in a controlled manner when using the approach with foundational diffusion models.

2. Related Works

Diffusion Models. Diffusion probabilistic models [44] are a class of deep generative models that synthesize data through an iterative denoising process. Diffusion models utilize a forward process that applies noise into data distribution and then reverses the forward process to reconstruct the data itself. Recently, they have gained popularity for the task of image generation [17, 45]. Dhariwal *et al.* [9] introduced various techniques such as architectural improvements and classifier

guidance, that helped diffusion models beat GANs in image generation tasks for the first time. Followed by this, many works started working on scaling the models [31, 37, 38, 40] to billions of parameters, improving the inference speed [41] and memory cost [38, 49]. LDM [38] is one the most popular models which reduced the compute cost by applying the diffusion process to the low-resolution latent space and scaled their model successfully for text-to-image generation trained on webscale data. Other than image generation, they have been applied to various fields such as multi-modal generation [52], text-to-3D [35, 43], language generation [23], 3D reconstruction [14], novel-view synthesis [51], music generation [28], object detection [6], etc.

Generative Image Editing. Image generation models have been widely used in image editing tasks since the inception of GANs [10, 13, 20, 22, 26], however, they were limited to edit a restricted set of images. Recent developments in the diffusion model has enabled image editing in the wild. Earlier works [31, 37, 38] started using text prompts to control the generated image. This led to various text-based image editing works such as [12, 27, 29]. To make localized edits works such as [15, 33, 48] use cross-attention feature maps between text and image. InstructPix2Pix [2] further enabled instruction-based image editing. However, using only text can only provide coarse edits. Works such as [1, 58] explored explicit spatial conditioning to control the structure of generated images and used text to define the appearance of local regions. Works such as [8, 24] rely on input images and text descriptions to get the region of interest for editing. However, most of the mentioned works lack object-level editing capabilities and some still rely only on text for describing the appearance. Recent works such as [11, 30] have object-level editing capabilities, however, they are based on the classifier guidance technique at inference time which leads to limited precision. Further, they show results only on stable diffusion and require inversion to edit real images. Our framework is general and can be applied to any diffusion model. We also enable multimodal control of the appearances of objects in the image when using our framework with stable diffusion.

3. PAIR Diffusion

In this work, we aim to develop an image-editing framework that allows the editing of the properties of individual objects in the image. We perceive an image $x \in R^{3\times H\times W}$ as composition of objects $O = \{o_1, o_2, \ldots, o_n\}$ where o_i represents the properties of i^{th} object in the image. As discussed in Sec. 1, we focus on enabling control over the structure and the appearance of each object. Thus, let $o_i = (s_i, f_i)$ where s_i represents the structure, f_i represents the appearance. The distribution that we aim to model can be written as:

$$p(x|O, y) = p(x|\{(s_1\ f_1), \ldots, (s_n,\ f_n)\}, y) \quad (1)$$

We use y to represent any form of conditioning signal already present in the generative model, e.g. text, and develop our framework to enable new object-level editing capabilities while preserving the original conditioning. The rest of the method section is organized as follows. In Sec. 3.1, we describe the method to obtain s_i and f_i for every object in a given image. Next, in Sec. 3.2, we show that various image editing tasks can be defined in the scope of the proposed object-level formulation of images. Finally, in Sec. 3.3, we describe the usage of the representations to augment the generative models and inference techniques to achieve object-level editing in practice.

3.1. Structure and Appearance Representation

Given an image $x \in R^{3\times H\times W}$ we want to extract the structure and appearance of each object present in the image.

Structure. The structure oversees the object's shape and category and is represented as $s_i = (c_i, m_i)$ where c_i represents the category and $m_i \in \{0, 1\}^{H\times W}$ represents the shape. We extract the structure information using a panoptic segmentation map, as it readily provides each object's category and shape information and is easy to compute. Given an off-the-shelf segmentation network $E_S(\cdot)$, we obtain $S = E_S(x)$, with $S \in \mathbb{N}^{H\times W}$ which gives direct access to c_i, m_i.

Appearance. The appearance representation is designed to capture the visual aspects of the object. To represent the object faithfully, it needs to capture both the low-level features like color, texture, etc., as well as the high-level features in the case of complex objects. To capture such a wide range of information, we choose a combination of convolution and transformer-based image encoders [36], namely VGG [42] and DINOv2 [32].We use initial layers of VGG to capture low-level characteristics such as color, texture etc. [55, 57]. Conversely, DINOv2 has well-learned representations and has shown promising results for various downstream computer vision tasks. Hence, we use the middle layers of DINOv2 to capture the high-level characteristics of the object.

To compute per-object appearance representations, we first extract the feature maps from l^{th} block of an encoder $E_G(\cdot)$, *i.e.* $\tilde{G} = E^l_G(x)$, $\tilde{G} \in \mathbb{R}^{C\times h\times w}$, with $h \times w$ the spatial size and C the number of channels. Then, we parse object-level features, relying on m_i to pool over the spatial dimension and obtain the appearance vector $g^l_i \in \mathbb{R}^C$:

$$g^l_i = \frac{\sum_{j,k} E^l_G(x) \odot m_i}{\sum_{j,k} m_i} \quad (2)$$

In our framework, $E_G(\cdot)$ could be either DINOv2 or VGG. We use g^{Vl}_i and g^{Dl}_i to respectively denote the appearance vectors obtained using the features of VGG and DINOv2 extracted at the l^{th}-block. The appearance information

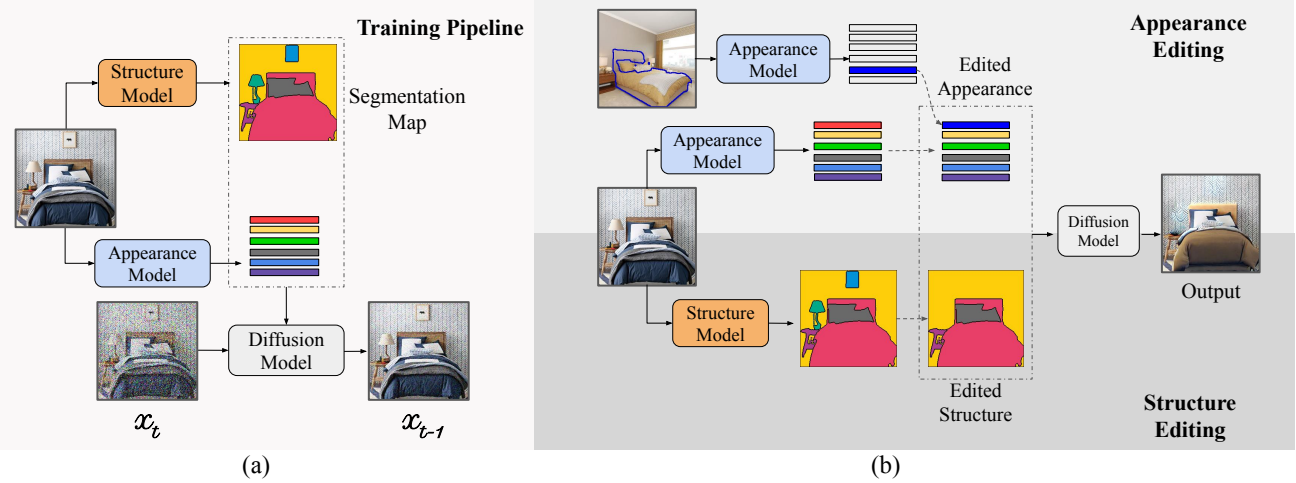

Figure 2. Overview of PAIR Diffusion. An image is seen as a composition of objects each defined by different properties like structure (shape and category), appearance, depth, etc. We focus on controlling structure and appearance. (a) During training, we extract structure and appearance information and train a diffusion model in a conditional manner. (b) At inference, the framework supports multiple editing operations by independently controlling the structure and appearance of any real image at the object level.

of i^{th} object is then given by a tuple $f_i = (g_i^{Vl_1}, g_i^{Dl_2}, g_i^{Dl_3})$ where $l_2 < l_3$. As a convention, we arrange the features in f_i in ascending order of abstraction, from low-level to high-level representations.

3.2. Image Editing Formulation

The proposed object-level design allows the definition of various image editing tasks within a single framework. Consider an image x with n objects $O = \{o_1, o_2, \ldots, o_n\}$, with each object o_i described by the structure s_i and the appearance f_i (see Sec. 3.1). Below we present the fundamental image editing operations that can be obtained with our framework. Importantly, they can be composed and applied to multiple objects, enabling *comprehensive* editing capabilities.

Appearance Editing $(s_i, f_i) \rightarrow (s_i, f_i')$. It is achieved by swapping appearance vector f_i with an edited appearance vector f_i'. Formally, $f_i' = a_0 f_i + a_1 f_j^R$ with f_j^R the appearance vector of the j^{th} object in the reference image.

Shape Editing $(s_i, f_i) \rightarrow (s_i', f_i)$. It is obtained by modifying the structure (c_i, m_i) to (c_i, m_i') i.e. the shape can be explicitly changed by the user while maintaining the appearance.

Object Addition $O \rightarrow O \cup \{o_{n+1}\}$. We can incorporate an object into an image by specifying both its structure and appearance. These attributes can be derived either entirely from a reference image or the user can provide a sketch of the structure alone, with the appearance being inferred from a reference image.

Object Appearance Variation. We can also get object-level appearance variations due to information loss in the pooling operation to calculate appearance vectors and the stochastic nature of the diffusion process.

Once we get object with edited properties O' and conditioning y we can sample a new image from the learned distribution $p(x|O', y)$. Our object-level design can easily incorporate various editing abilities and help us achieve a comprehensive image editor. In the next section, we will describe a way to implement $p(x|O, y)$ in practice, and present inference methods to sample and control the edited image.

3.3. Architecture Design and Inference

In practice, Eq. (1) represents a conditional generative model; building upon the recent success of diffusion models, we leverage them to implement it. Next, we describe a method to use the object-level representations outlined in Sec. 3.1 both in unconditional diffusion models and foundational text-to-image (T2I) diffusion models. In this way, we can transform any diffusion model into an object-level editor.

We start by representing structure and appearance in a spatial format to conveniently use them for conditioning. We represent the structure conditioning as $S \in \mathbb{N}^{2\times H\times W}$ where the first channel contains the category, while the second channel contains the shape information of each object. For appearance conditioning, we first $L2$-normalize each vector along channel dimension, splat them spatially using m_i, and combine them in a single tensor represented as $G \in \mathbb{R}^{C\times H\times W}$. The process is repeated for the features extracted through different encoders and at different layers, leading to the tuple $F = (G^{Vl_1}, G^{Dl_2}, G^{Dl_3})$. Lastly, we channel-wise concatenate S to every element of F which results in our final conditioning signals $F_s = (G_s^{Vl_1}, G_s^{Dl_2}, G_s^{Dl_3})$.

In the case of the foundational T2I diffusion model, we choose Stable Diffusion (SD) [38] as our base model. To condition it, we adopt ControlNet [59] because of its training

and data efficiency in conditioning SD model. The control module consists of encoder blocks and middle blocks that are replicated from SD UNet architecture. Various works show the tendency of the SD inner layers to focus more on high-level features, whereas the outer layers to focus more on low-level features [5, 24, 48]. Exploiting this finding, we use $G_s^{Vl_1}$ as input to the control module and add $G_s^{Dl_2}$, $G_s^{Dl_3}$ to the features after cross-attention in the first and second encoder blocks of the control module respectively. For the unconditional diffusion model, we use the unconditional latent diffusion model (LDM) [38] as our base model. Pertaining to the simplicity of the architecture and training of these models we simply concatenate the features in F_s to the input of LDM. The architecture is accordingly modified to incorporate the increased number of input channels. For further details please refer to *Supp. Mat.*.

For training both the models we follow standard practice [38] and use the simplified training objective $\mathcal{L} = ||\epsilon - \epsilon_\theta(z_t, S, F, y, t)||_2^2$, where z_t represents the noisy version of x in latent space at timestep t, ϵ is the noise used to get z_t, and ϵ_θ is the trainable model. In the case of Stable Diffusion, y represents the text prompts, while it is not present in the case of the unconditional diffusion model.

Multimodal Inference. Once we have a trained model, we require an inference method that allows us to adjust the strengths of different conditioning signals and control the edited image accordingly. We consider the scenario where y represents text, with the unconditional diffusion models being a special case where y is null. Specifically, the structure S and appearance F come from a reference image and the information in y could be disjoint from F, we need a way to capture both in the final image. A well-trained diffusion model estimates the score function of the underlying data distribution [46], *i.e.* $\nabla_{z_t} p(z_t|O, y) = \nabla_{z_t} p(z_t|S, F, y)$, which in our case can be expanded as

$$\begin{aligned}\nabla_{z_t} \log p(z_t|S,F,y) =& \nabla_{z_t} \log p(z_t|S,F) \\ &+ \nabla_{z_t} \log p(z_t|y) \\ &- \nabla_{z_t} \log p(z_t)\end{aligned} \tag{3}$$

We use the concept of classifier-free guidance (CFG) [16] to represent all score functions in the above equation using a single model by dropping the conditioning with some probability during training. Using the CFG formulation we get the following update rule expanding Eq. (3):

$$\begin{aligned}\tilde{\epsilon_\theta}(z_t, S, F, y) = & \epsilon_\theta(z_t, \phi, \phi, \phi) \\ &+ s_S \cdot (\epsilon_\theta(z_t, S, \phi, \phi)) - \epsilon_\theta(z_t, \phi, \phi, \phi)) \\ &+ s_F \cdot (\epsilon_\theta(z_t, S, F, \phi) - \epsilon_\theta(z_t, S, \phi, \phi)) \\ &+ s_y \cdot (\epsilon_\theta(z_t, \phi, \phi, y) - \epsilon_\theta(z_t, \phi, \phi, \phi))\end{aligned} \tag{4}$$

For brevity, we did not include t in the equation above. A formal proof of the above equations is provided in *Supp. Mat.*.

Intuitively, F is more information-rich compared to y. For this reason, during training the network learns to give negligible importance to y in the presence of F, and we need to use y independently of F during inference to see its effect on the final image. In Eq. (4) s_S, s_F, s_y are guidance strengths for each conditioning signal. It provides PAIR Diffusion with an intuitive way to control and edit images using various conditions. For example, if a user wants to give more importance to a text prompt compared to the appearance from the reference image, it can set $s_y > s_F$ and vice-versa. For the unconditional diffusion models, we simply ignore the term corresponding to s_y in Eq. (4).

4. Experiments

In this section, we present qualitative and quantitative analysis that show the advantages of the PAIR diffusion framework introduced in Sec. 3. We refer to UC-PAIR Diffusion to denote our framework applied to unconditional diffusion models and reserve the name PAIR Diffusion when applying the framework to Stable Diffusion. Evaluating image editing models is hard, moreover, few works have comprehensive editing capabilities at the object level making a fair comparison even more challenging. For these reasons, we perform two main sets of experiments. Firstly, we train UC-PAIR Diffusion on widely used image-generation datasets such as the bedroom and church partitions of the LSUN Dataset [56], and the CelebA-HQ Dataset [21]. We conduct quantitative experiments on these datasets as they represent a well-study benchmark, with a clear distinction between training and testing sets, making it easier and fairer to perform evaluations. Secondly, we fine-tune PAIR Diffusion on the COCO [25] dataset. We use this model to perform in-the-wild editing and provide examples for the use cases described in Sec. 3.2, showing the comprehensive editing capabilities of our method. We refer the reader to the *Supp. Mat.* for the details regarding model training and implementations, along with additional results.

4.1. Editing Applications

In this section, we qualitatively validate that our model can achieve comprehensive object-level editing capabilities in practice. We primarily show results using PAIR Diffusion and refer to the *Supp. Mat.* for results on smaller datasets. We use different baselines according to the editing task. We adapt Prompt-Free-Diffusion (PFD) Xu et al. [53] as a baseline for localized appearance editing, by introducing masking and using the cropped reference image as input. Moreover, we adopt Paint-By-Example (PBE) Yang et al. [54] as a baseline for adding objects and shape editing. For further details regarding implementation please refer to *Supp. Mat.*. When we want the final output to be influenced by the text prompt as well we set $s_y > s_F$ else we set $s_y < s_F$. For the figures where there is no prompt provided below the image

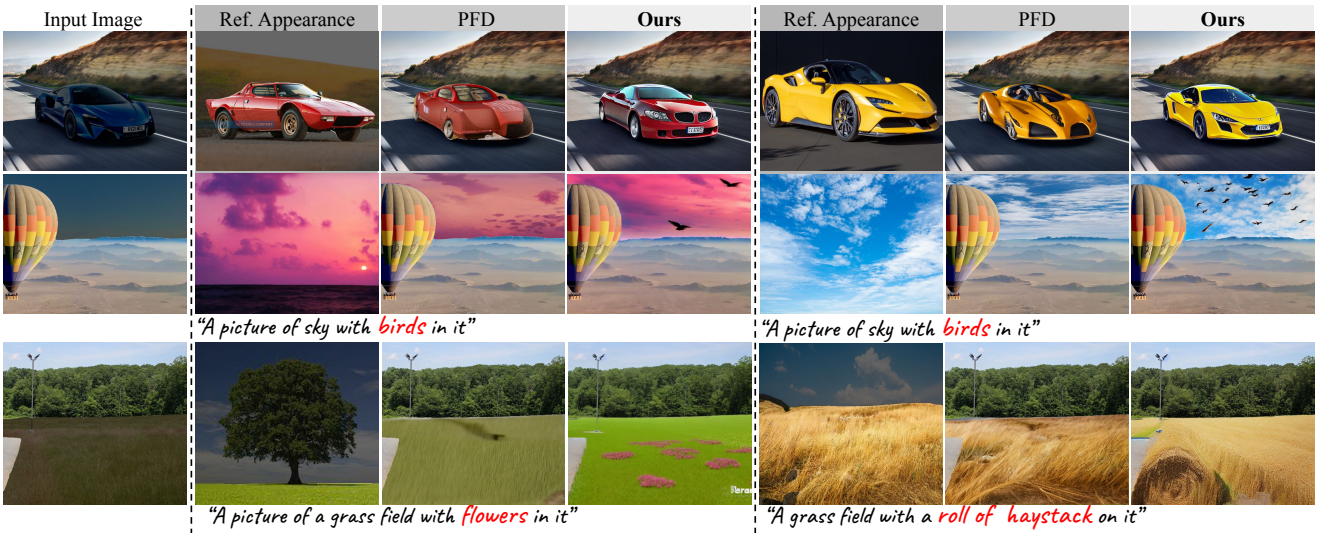

Figure 3. Qualitative results for appearance editing. We can drive the edit with reference images as well as with text prompts.

assume that prompt was auto-generated using the template: *'A picture of* {*category*}*'*, with the category inferred form the edited object. When editing a local region we used a masked sampling technique to only affect the selected region [38].

Appearance Editing. In Fig. 3, we report qualitative results for appearance editing driven by reference images and text. We observe that our multilevel appearance representation and object-level design help us edit the appearance of both simple objects such as the sky as well as complex objects like cars. On the other hand, PFD [53] gives poor results when editing the appearance of complex objects due to the missing object-level design. Furthermore, using our multimodal classifier free guidance, our model can seamlessly blend the information from the text and the reference images to get the final edited output whereas PFD [53] lacks this ability.

Add objects and Shape editing. We show the object addition and shape editing operations result together in Fig. 4. With PAIR Diffusion we can add complex objects with many details like a cake, as well as simpler objects like a lake. When changing the structure of the cake from a circle to a square, the model captures the sprinkles and dripping chocolate on the cake while rendering it in the new shape. In all the examples, we can see that the edges of the newly added object blend smoothly with the underlying image. On the other hand, PBE [54] completely fails to follow the desired shape and faces issues with large objects like lakes.

Object Variations. We can also achieve image variations at an object level as shown in Fig. 13 in *Supp. Mat.*. We note that our model can capture various details of the original object and still produce variations.

4.2. Quantitative Results

As described in Sec. 3, the backbone of our design is the ability to control two major properties of the objects, the appearance and the structure. The aim of the quantitative evaluation is to verify that we can control the mentioned properties and not to push the state-of-the-art results. We start by evaluating our model on appearance control: the task

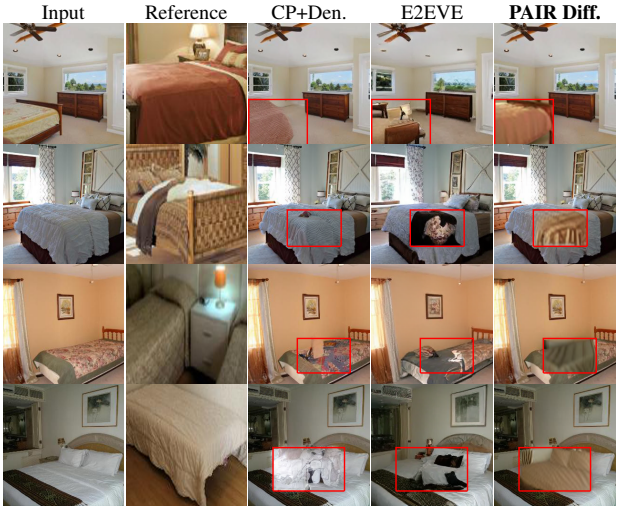

Figure 5. Visual results for appearance control on LSUN bedroom. We show the results obtained with relevant baselines for editing the red area in the input image using the reference as a driver.

Table 1. Quantitative results for appearance control on the LSUN Bedroom validation set.

Model	**FID** (↓)	**L1** (↓)	**SSIM** (↑)
Copy-Paste (CP)	21.37	0.0	0.87
Inpainting [38]	8.25	0.02	0.17
CP + Denoise	9.15	0.02	0.32
E2EVE [3]	13.59	0.05	0.34
PAIR Diffusion	12.81	0.02	0.51

Figure 4. Qualitative results for adding objects and shape editing.

Table 2. Quantitative results for structure control on CelebA-HQ validation dataset.

Model	mIoU	SSIM
SEAN [62]	0.64	0.32
PAIR Diff.	**0.67**	**0.52**

Table 3. Quantitative results of ablation study on appearance representation

Model	L1	LPIPS
M_{VGG}	0.1893	0.555
M_{DINO}	0.1953	0.549
Full	**0.1891**	**0.545**

consists of modifying a specific region of the input image using a reference image to drive the edit. We compare our method with the recent work of [3] (E2EVE), and follow their evaluation procedure. In particular, different models are compared based on: (i) Naturalness: we expect the edited image to look realistic and rely on FID between input and edited images to assess it, (ii) Locality: we expect the edit to be limited to the specific region where the edit is performed and use L1 distance to measure it, (iii) Faithfulness: we expect the edited region and the target image to be similar and we use SSIM to evaluate it. As discussed in E2EVE, all the above-mentioned criteria should hold at the same time, and the best-performing method is the one giving good results in the three metrics at the same time. We compare our method with four baselines: (1) Copy-Paste: the driver image is simply copied in the edit region of the input image, (2) Inpainting: we use LDM Rombach et al. [38] to inpaint the target edit region, (3) Copy-Paste + Denoise: starting form copy-paste edit, we invert the image with DDIM, and denoise it with LDM, (4) E2EVE. In Tab. 1 we report the quantitative results on the validation set of LSUN Bedroom [56] and visual comparisons are shown in Fig. 5. The copy-paste baseline provides an upper bound to the faithfulness and locality but produces images that are unrealistic (high FID score). Vice-versa, Inpainting and CP+Denoise produce natural results (low FID score) but are not faithful to the driver image (low SSIM score). Only our method performs well w.r.t. all the aspects and outperforms E2EVE in all metrics showing that we can control the appearance of a region. We refer the reader to *Supp. Mat.* for a detailed description of the evaluation procedure and baseline implementation.

Secondly, we evaluate the structure-controlling ability of our method. We adopt the validation set of CelebA-HQ (5000 samples) and compare with SEAN [62]. We generate images conditioning the model on the ground truth structure maps from the validation set and then segment the generated images with a pre-trained model [63]. We report the mIoU score, calculated using the ground truth segmentation map as the reference, as well as the SSIM score in Tab. 2. The proposed method outperforms [62] in terms of both mIoU and SSIM, demonstrating that our method can precisely follow the guidance of structure and retain the appearance.

4.3. Ablation Study

Multimodal Classifier Free Guidance. We validate the effectiveness of the proposed multimodal classifier-free guidance. Instead of factorizing, which results in Eq. (3), we directly expand the conditional score function $\nabla_{z_t} \log p(z_t|S,F,y)$ and apply classifier free guidance formulation on it and get the following equation:

$$\begin{aligned}\tilde{\epsilon_\theta}(z_t,S,F,y) = \epsilon_\theta(z_t,\phi,\phi,\phi) \\ + s_S \cdot (\epsilon_\theta(z_t,S,\phi,\phi) - \epsilon_\theta(z_t,\phi,\phi,\phi)) \\ + s_F \cdot (\epsilon_\theta(z_t,S,F,\phi) - \epsilon_\theta(z_t,S,\phi,\phi)) \\ + s_y \cdot (\epsilon_\theta(z_t,S,F,y) - \epsilon_\theta(z_t,S,F,\phi))\end{aligned} \tag{5}$$

We highlight the difference between Eq. (4) and Eq. (5) using the blue color. In Fig. 6, we compare the results sampled from Eq. (4) (column (a)) and Eq. (5) (columns (b)-(d)). We use the same seed to generate all the images,

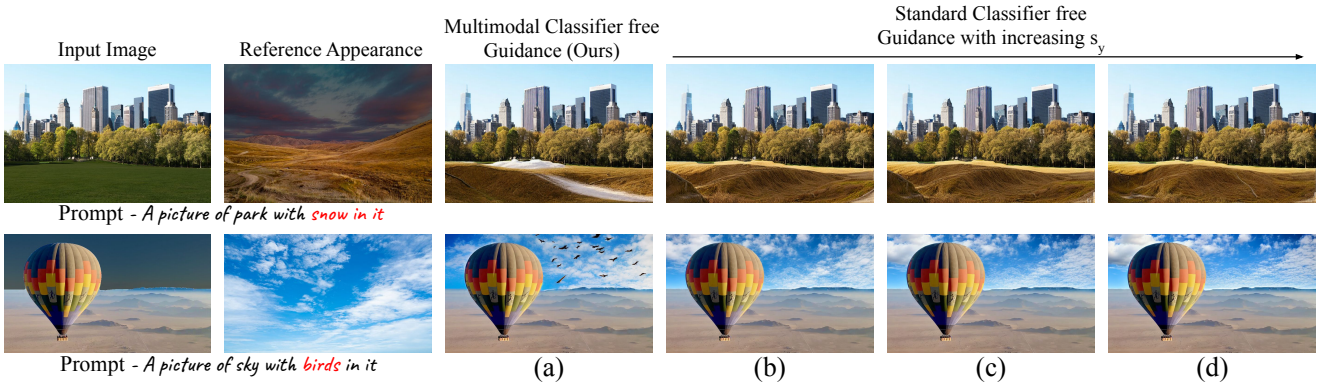

Figure 6. Ablation study for multimodal classifier-free guidance. We can see that if we use standard classifier-free guidance Eq. (5) the model completely ignores the text when sampling the image.

further the values of s_S, s_F, s_y are the same in columns (a) and (b). For the first row we set $s_S = 8, s_F = 3, s_y = 8$ and for second row it is $s_S = 6, s_F = 4, s_y = 8$. The values of s_y for (b)-(d) are $8, 15, 20$ respectively. We can clearly see that sampling results using Eq. (5) completely fail to consider text prompt even after increasing the value of s_y. This shows the effectiveness of the proposed classifier-free guidance Eq. (4) for controlling the image in a multimodal manner. Lastly, we conduct an ablation study on control parameters of CFG, namely s_S, s_F, s_y, to better understand the relationship between them. Please refer to *Supp. Mat.* for discussion and the visual results in Fig. 7-Fig. 8.

Appearance representation. We ablate the importance of using VGG and DINOv2 for representing the appearance of an object. We train two models, one using only VGG features (M_{VGG}) and the second using only DINOv2 features (M_{DINO}) to capture the appearance of an object. We train both models using identical hyperparameters to our original model. We assess the performance of each model using pairwise image similarity metrics on the COCO [4] validation set. We use L1 as our low-level metric and LPIPS [60] as our high-level metric and report the results Tab. 3. While M_{VGG} has a better L1 score compared to M_{DINO}, the LPIPS score indicates that M_{DINO} outperforms M_{VGG}. This experiment confirms the intuition that VGG features are good at capturing low-level details, while DINO features excel at capturing high-level details in our framework. In our final design, we found that combining both VGG and DINOv2 features for appearance vectors yielded the best L1 and LPIPS scores, leveraging the strengths of both representations. Supporting visuals illustrating this can be found in Fig. 9.

We define appearance as the visual characteristics of an object (see Sec. 1), and do not aim to maintain the exact identity of the object in the reference image. This is in contrast with recent research on personalization [39]. However, this formulation allows us to employ reference images in a versatile manner, contributing to our comprehensive editing capabilities. We use reference images that depict texture-only images (Fig. 1(a) second row), perform image variations (Fig. 13), realistic edits (Fig. 3), style transfer (Fig. 17) as well as semantically complex edits (Fig. 11).

5. Conclusion

In this paper, we show that we can build a comprehensive image editor by interpreting images as the amalgamations of various objects. We propose a generic framework, dubbed PAIR Diffusion, that enables structure and appearance editing at the object-level in any diffusion model. Our framework enables various object-level editing operations on real images without the need for inversion including appearance editing, structure editing, adding objects, and object variations. All the operations can be obtained with a single model trained once. Furthermore, we propose multimodal classifier-free guidance which enables precise control in the editing operations. We demonstrate its effectiveness with extensive editing examples on real image across different domains.

Limitations and future work. Currently, the architecture modifications present a simple formulation of the appearance vectors and the structure conditioning. While offering advantages by seamlessly integrating into existing Diffusion Models with minimal modification, in the future we plan to explore more sophisticated designs while maintaining the core object-level formulation. We plan to extend the explicit control over other aspects of the objects, such as the illumination, pose, etc., and improve the identity preservation of the edited object. The proposed object-level formulation can also help in devising standardizing metrics for image editing tasks in a unified manner which is lacking in the field.

Acknowledgement. This work was partly supported by the MUR PNRR project FAIR (PE00000013) funded by the NextGenerationEU and by the EU Horizon project AI4Trust (No. 101070190), NSF CAREER Award #2239840, and the National AI Institute for Exceptional Education (Award #2229873) by National Science Foundation and the Institute of Education Sciences, U.S. Department of Education.

References

[1] Omri Avrahami, Thomas Hayes, Oran Gafni, Sonal Gupta, Yaniv Taigman, Devi Parikh, Dani Lischinski, Ohad Fried, and Xi Yin. Spatext: Spatio-textual representation for controllable image generation. In *CVPR*, 2023. 2, 3

[2] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023. 2, 3

[3] Andrew Brown, Cheng-Yang Fu, Omkar Parkhi, Tamara L. Berg, and Andrea Vedaldi. End-to-end visual editing with a generatively pre-trained artist. In *ECCV*, 2022. 6, 7, 4

[4] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *CVPR*, 2018. 8, 2

[5] Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *CVPR*, 2023. 2, 5

[6] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. In *CVPR*, 2023. 3

[7] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 2

[8] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint*, 2022. 2, 3

[9] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NIPS*, 2021. 2

[10] Dave Epstein, Taesung Park, Richard Zhang, Eli Shechtman, and Alexei A Efros. Blobgan: Spatially disentangled scene representations. In *ECCV*, 2022. 3

[11] Dave Epstein, Allan Jabri, Ben Poole, Alexei A Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. *arXiv preprint*, 2023. 3

[12] Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. Training-free structured diffusion guidance for compositional text-to-image synthesis. *arXiv preprint*, 2022. 3

[13] Xinyu Gong, Shiyu Chang, Yifan Jiang, and Zhangyang Wang. Autogan: Neural architecture search for generative adversarial networks. In *CVPR*, 2019. 3

[14] Jiatao Gu, Alex Trevithick, Kai-En Lin, Joshua M Susskind, Christian Theobalt, Lingjie Liu, and Ravi Ramamoorthi. Nerfdiff: Single-image view synthesis with nerf-guided distillation from 3d-aware diffusion. In *ICML*, 2023. 3

[15] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint*, 2022. 2, 3

[16] Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance. *arXiv preprint*, 2022. 5, 1

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NIPS*, 2020. 2

[18] Minghui Hu, Chuanxia Zheng, Heliang Zheng, Tat-Jen Cham, Chaoyue Wang, Zuopeng Yang, Dacheng Tao, and Ponnuthurai N Suganthan. Unified discrete diffusion for simultaneous vision-language generation. *arXiv preprint*, 2022. 2

[19] Jitesh Jain, Anukriti Singh, Nikita Orlov, Zilong Huang, Jiachen Li, Steven Walton, and Humphrey Shi. Semask: Semantically masking transformer backbones for effective semantic segmentation. *arXiv preprint*, 2021. 2

[20] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two transformers can make one strong gan. *arXiv preprint*, 2021. 3

[21] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint*, 2017. 5

[22] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 3

[23] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *arXiv*, 2022. 3

[24] Jun Hao Liew, Hanshu Yan, Daquan Zhou, and Jiashi Feng. MagicMix: Semantic Mixing with Diffusion Models. *arXiv preprint*, 2022. 2, 3, 5

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 5

[26] Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. Editgan: High-precision semantic image editing. *NIPS*, 2021. 3

[27] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *ECCV*, 2022. 2, 3

[28] Gautam Mittal, Jesse Engel, Curtis Hawthorne, and Ian Simon. Symbolic music generation with diffusion models. *arXiv preprint*, 2021. 3

[29] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *CVPR*, 2023. 3

[30] Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. Dragondiffusion: Enabling drag-style manipulation on diffusion models. *arXiv preprint*, 2023. 3

[31] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint*, 2021. 3

[32] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint*, 2023. 3

[33] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. *arXiv preprint*, 2023. 3

[34] Or Patashnik, Daniel Garibi, Idan Azuri, Hadar Averbuch-Elor, and Daniel Cohen-Or. Localizing object-level shape variations with text-to-image diffusion models. In *CVPR*, 2023. 2

[35] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint*, 2022. 3

[36] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *NIPS*, 2021. 3

[37] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint*, 2022. 3

[38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 3, 4, 5, 6, 7, 2

[39] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation, 2022. 2, 8

[40] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint*, 2022. 3

[41] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint*, 2022. 3

[42] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 3

[43] Uriel Singer, Shelly Sheynin, Adam Polyak, Oron Ashual, Iurii Makarov, Filippos Kokkinos, Naman Goyal, Andrea Vedaldi, Devi Parikh, Justin Johnson, et al. Text-to-4d dynamic scene generation. *arXiv preprint*, 2023. 3

[44] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 2

[45] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *NIPS*, 2019. 2, 4

[46] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint*, 2020. 5

[47] Yizhi Song, Zhifei Zhang, Zhe Lin, Scott Cohen, Brian Price, Jianming Zhang, Soo Ye Kim, and Daniel Aliaga. ObjectStitch: Generative Object Compositing. *arXiv preprint*, 2022. 2

[48] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *CVPR*, 2023. 3, 5

[49] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *NIPS*, 2021. 3

[50] Andrey Voynov, Kfir Aberman, and Daniel Cohen-Or. Sketch-guided text-to-image diffusion models. In *ACM SIGGRAPH*, 2023. 2

[51] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Yi Wang, and Zhangyang Wang. Neurallift-360: Lifting an in-the-wild 2d photo to a 3d object with 360° views. *arXiv preprint*, 2022. 3

[52] Xingqian Xu, Zhangyang Wang, Eric Zhang, Kai Wang, and Humphrey Shi. Versatile diffusion: Text, images and variations all in one diffusion model. *arXiv preprint*, 2022. 3

[53] Xingqian Xu, Jiayi Guo, Zhangyang Wang, Gao Huang, Irfan Essa, and Humphrey Shi. Prompt-free diffusion: Taking" text" out of text-to-image diffusion models. *arXiv preprint*, 2023. 5, 6, 4

[54] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *CVPR*, 2023. 2, 5, 6, 4

[55] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint*, 2015. 3

[56] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint*, 2015. 5, 7, 2

[57] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 3

[58] Yu Zeng, Zhe Lin, Jianming Zhang, Qing Liu, John Collomosse, Jason Kuen, and Vishal M Patel. Scenecomposer: Any-level semantic image synthesis. In *CVPR*, 2023. 2, 3

[59] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *CVPR*, 2023. 4, 2

[60] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 8

[61] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 2

[62] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *CVPR*, 2020. 7, 4, 10

[63] zllrunning. https://github.com/zllrunning/face-parsing.pytorch. *github*, 2019. 7