Magic Fixup: Streamlining Photo Editing by Watching Dynamic Videos

Hadi Alzayer[1,2], Zhihao Xia[1], Xuaner Zhang[1], Eli Shechtman[1], Jia-Bin Huang[2], and Michael Gharbi[1]

[1] Adobe
[2] University of Maryland, College Park
https://magic-fixup.github.io

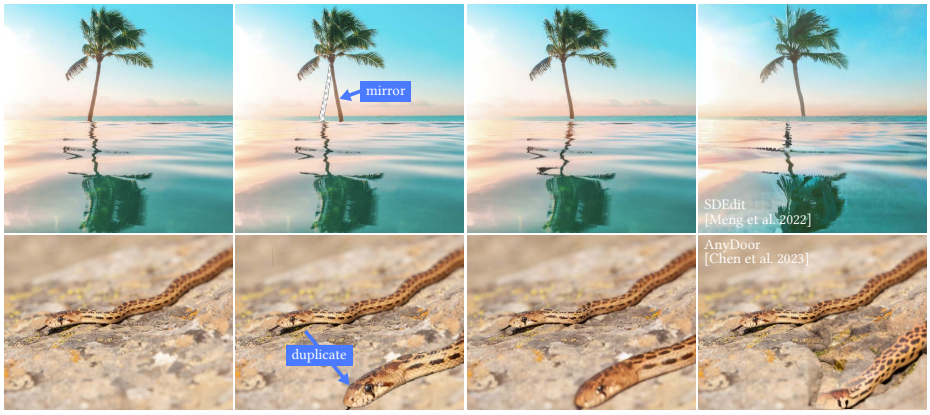

Fig. 1: Applications of Magic Fixup. We propose a diffusion model for image editing. Starting from an input image (a), a user specifies their desired changes by rearranging automatically segmented scene objects using simple 2D transforms to produce a coarse edit (b). Our model transforms this coarse edit into a realistic image (c), correctly accounting for secondary effects critical for realism, such as reflections on the water (top) or changes in depth-of-field (bottom), producing much more plausible edits than state-of-the-art methods (d).

Abstract. We propose a generative model that, given a coarsely edited image, synthesizes a photorealistic output that follows the prescribed layout. Our method transfers fine details from the original image and preserve the identity of its parts. Yet, it adapts it to the lighting and context defined by the new layout. Our key insight is that videos are a powerful source of supervision for this task: objects and camera motions provide many observations of how the world changes with viewpoint, lighting, and physical interactions. We construct an image dataset in which each sample is a pair of source and target frames extracted from the same video at randomly chosen time intervals. We warp the source frame toward the target using two motion models that mimic the expected test-time user edits. We supervise our model to translate the warped image into the ground truth, starting from a pretrained diffusion model. Our model design explicitly enables fine detail transfer from the

source frame to the generated image, while closely following the user-specified layout. We show that by using simple segmentations and coarse 2D manipulations, we can synthesize a photorealistic edit faithful to the user's input while addressing second-order effects like harmonizing the lighting and physical interactions between edited objects.

1 Introduction

Image editing is a labor-intensive process. Although humans can quickly and easily rearrange parts of an image to compose a new one, simple edits can easily look unrealistic, e.g., when the scene lighting and physical interactions between objects become inconsistent. Fixing these issues manually to make the edit plausible requires professional skills and careful modifications, sometimes down to the pixel level. The success of recent generative models [16, 18, 23, 42] paves the way for a new generation of automated tools that increase the realism of image edits while requiring much sparser user inputs [3, 14, 27, 45]. Generative methods providing explicit spatial keypoints control have been proposed but are either limited to certain domains [38] or modest changes [46]. State-of-the-art approaches, however, regenerate pixels based on a user-specified text prompt and a mask of the region to influence [9, 10, 52, 54]. This interface is not always natural. In particular, it does not allow spatial transformations of the existing scene content, as we show in Figure 2, and object identities are often not fully preserved by the re-synthesis step [12, 49].

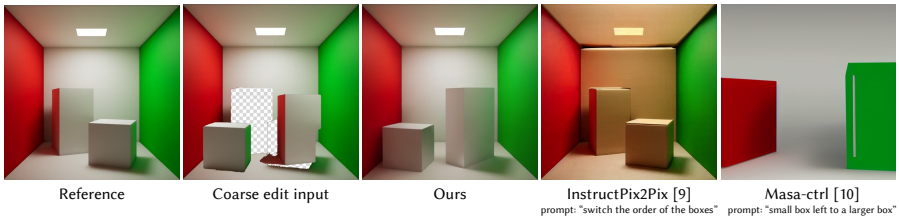

Fig. 2: Comparison with text based control. Our method directly takes a coarse user edit and makes it photorealistic. Our editing is both easy and precise, and our model can harmonize the global illumination appropriately. Text-based editing methods [9, 10] on the other hand, are not able to perform such edits, resulting in global appearance changes [9] or unrealistic image [10].

In this paper, we propose a new approach to image editing that offers the controls of conventional editing methods and the realism of the modern generative model (Figure 1). Our method uses human inputs where it shines: users can segment the image and rearrange its parts manually in a "cut-and-transform" approach, e.g., using simple 2D transforms, duplication, or deletion to construct their desired layout, just like a collage [45]. We call our collage-like editing interface the *Collage Transform*. We then train a diffusion model to take

care of the hard work of making the edit photorealistic. Our model "projects" the coarsely edited image onto the natural image manifold, fixing up all the low-level image cues that violate its image prior, such as tweaking poses, blending object boundaries, harmonizing colors, adding cast shadows, reflections and other second-order interactions between the object and the environment.

Crucially, we explicitly fine-tune a latent diffusion model [42] so its output deviates as little as possible from the user's specifications and the appearance of the original objects in the scene. This is essential for photographers, as they spend significant effort capturing their images and would like to retain the content identity as much as possible. When editing an image, there is a subtle balance between being faithful to the original image and harmonizing the edited image to preserve realism. This is the regime that our work focuses on. Our insight is that videos provide a rich signal of how an edited photo's appearance should change to preserve photorealism. From videos, we can learn how objects' appearances change in the real world as they deform and move under changing light. Camera motion and disocclusions give us priors about what hides behind other objects and how the same object looks under changing perspectives.

To exploit these cues, we build a paired image dataset from a large-scale video corpus. Each pair corresponds to two frames sampled from the same video: source and target frames. We then automatically segment [29], and transform objects in the source frame to match the pose of the corresponding objects in the target frame, using two motion models based on optical flow, designed to simulate the coarse edits a user would make using our Collage Transform interface. Since the images are now roughly aligned, we can train our model to convert the coarsely edited image into the ground truth target frame in an image-to-image [25,44] fashion. This alignment procedure encourages the model to follow the user-specified layout at test time closely. Additionally, our model is carefully designed to transfer fine details from the reference source frame to preserve the identity and appearance of objects in the scene.

Our approach can produce plausible and realistic results from real user edits, and effectively projects coarse user edits into photorealistic images, confirming our insights on the advantages of using video data and a carefully designed motion model. Compared to the state-of-the-art, we show our outputs are preferred 89% of the time in a user study.

In short, our contributions are as follows:

- the Collage Transform, a natural interface for image editing that allows users to select and alter any part of an input image using simple transforms and that automatically turns the resulting edit into a realistic image,
- a new paired data generation approach to supervise the conversion from coarse edits to real images, which extracts pairs of video frames and aligns the input with the ground truth frame using simple motion models,
- a conditioning procedure that uses: 1. the warped image to guide layout in the diffusion generator, and 2. features from a second diffusion model to transfer fine image details and preserve object identity.

2 Related Work

Classical image editing. Classical image editing techniques offer various types of user controls to achieve diverse objectives. For instance, image retargeting aims to alter an image's size while preserving its key features and content [4, 43, 47, 53]. In contrast, image reshuffling rearranges an image's content based on user-provided rough layouts and imprecise mattes [7, 13, 47]. Image harmonization integrates objects from different images, adjusting their low-level statistics for a seamless blend [26, 50]. A common thread in these classical image editing applications is the crucial role of user interaction, which provides the necessary control for users to realize their vision. Our method aligns with this approach, allowing users to reconfigure a photograph based on their preliminary edits.

Controllable image generation. The rapid advancement in photorealistic image generation has inspired researchers to adapt generative models for image editing tasks. Early efforts focused on high-level edits, like altering age or style, by manipulating latent space of Generative Adversarial Networks (GANs) [1, 2, 11]. In a vein similar to our work, Generative Visual Manipulation [59] involves projecting user-edited images onto the natural image manifold as approximated by a pre-trained GAN. The recent introduction of CLIP embeddings [39] has further propelled image editing capabilities, particularly through text prompts [5, 9, 15, 19, 22, 27, 34]. DragGAN [38] introduces fine control in image editing by using key-handles to dictate object movement, and follow-up works extend the drag-control idea to diffusion models [32, 35, 46]. Image Sculpting [57] takes a different approach by directly reposing the reconstructed 3D model of an object and re-rendering it, providing high level of control, but time consuming editing process unlike our Collage Transform interface that is designed to increase editing efficiency. CollageDiffusion [45] guides text-to-image generation by using a collage as additional input. However, while CollageDiffusion focuses on controlling the generation of an image from scratch, we focus on using collage-like transformation to edit a reference image, and focus on preserving its identity.

Reference-based editing with generative models. To extend controllable image generation into editing real (non-generated images), one can invert the image back to noise [48], and then guide the iterative denoising process to control the image generation [6, 10, 33]. However, naively guiding the model without any grounding can lead to a loss in image identity. Prior work [12, 17, 56] preserves the image identity through a pretrained feature extractor like CLIP [39] or DINO [37], using a Control-Net like feature-injection [12, 58], a dual-network approach [10, 24], or a combination of those approaches [12, 55]. We adopt the dual-network approach, as it allows us to fully fine-tune the model and taylor it to our photorealistic editing task using our video-based dataset. AnyDoor [12] similarly uses video frames during training, but their focus is to recompose individual objects into the scene. On the other hand, we use video data to recompose the *entire scene* and use motion models designed for a convenient photo editing interface. Closest to our work is MotionGuidance [20] that uses optical flow to

guide editing the reference frame with diffusion guidance [6] for a highly user-controllable edit. However, dense optical flow is difficult to manually provide for a user, unlike simple cut-and-transform edits in our Collage Transform. Furthermore, they rely on a prohibitively time-consuming guidance that take as long as 70 minutes for a single sample. On the other hand, our approach takes less than 5 seconds to fix up the user edit, allowing for interactive editing process.

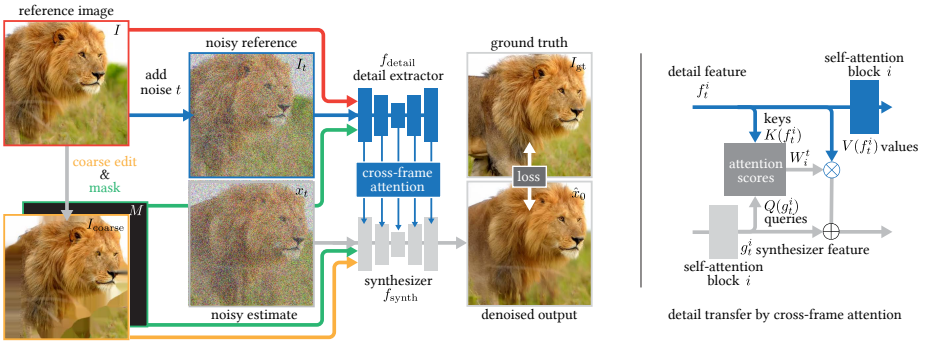

Fig. 3: Overview. Our pipeline (left panel) uses two diffusion models in parallel, a detail extractor (top) and a synthesizer (bottom), to generate a realistic image from a coarse user edit and a mask recording missing regions caused by the edit. The detail extractor processes the reference image, a noisy version of the reference and the mask, to produce a set of features that guide the synthesis and allow us to preserve the object appearance and fine details from the reference image. The synthesizer generates the output conditioned on the mask and coarse edit. The features from the detail extractor are injected via cross-attention at multiple stages in the synthesizer, in order to transfer details from the input. Both models are finetuned on our paired dataset. The right panel shows a detailed view of our cross-attention detail transfer operator.

3 Method

We aim to enable an image editing workflow in which users can select objects in a photograph, duplicate, delete or rearrange them using simple 2D transforms to produce a realistic new image (§ 3.1). We leverage image priors from pre-trained diffusion models to project the coarsely edited image onto the natural image manifold, so the user can focus on specifying high-level changes without worrying about making their edits plausible (§ 3.2). Existing diffusion models can produce impressive results but often do so at the expense of control and adherence to the user input [33]. In particular, they tend to "forget" the identity and appearance of the edited object [56], and often only loosely conform to the user-specified pose [12]. Our method addresses these issues using two mechanisms. First, our synthesis pipeline is a conditional diffusion model (§ 3.4) that follows the coarse layout defined by the user, and transfers fine details from the reference input image (§ 3.3) to best preserve the original image content. Second, we construct a supervised dataset exploiting object motion from videos to

finetune the pretrained model to explicitly encourage content preservation and faithfulness to the input edit (§ 3.5).

3.1 Specifying coarse structure with simple transforms

Starting from an image $I \in \mathbb{R}^{3hw}$, $h = w = 512$, we run an automatic segmentation algorithm [29] to split the image into non-overlapping semantic object segments. The user can edit this image by applying 2D transformations to the individual segments (e.g., translation, scaling, rotation, mirroring). Segments can also be duplicated or deleted. Figure 1 illustrates this workflow. We keep track of holes caused by disocclusions when moving the segment in a binary mask $M \in \{0,1\}^{hw}$, and inpaint them using a simple algorithm [8]. We denote the resulting, coarsely edited image by $I_{\text{coarse}} \in \mathbb{R}^{3hw}$.

We operate in an intermediate latent space for efficiency, but our approach also applies to pixel-space diffusion. With a slight abuse of notation, in the rest of the paper $I, I_{\text{coarse}} \in \mathbb{R}^{3hw}$, with $h = w = 64$ refer to the input and coarse edit after encoding with the latent encoder from Stable Diffusion [42], and M the mask downsampled to the corresponding size using nearest neighbor interpolation. The latent triplet $(I, I_{\text{coarse}}, M)$ forms the input to our algorithm.

3.2 From coarse edits to realistic images using diffusion

We want to generate a realistic image that (1) follows the large-scale structure defined by the coarse user edit, and (2) preserves the fine details and low-level object appearance from the unedited image, filling in the missing regions. Our pipeline, illustrated in Figure 3, uses 2 diffusion models.

The first, which we call *synthesizer* f_{synth}, generates our final output image. The second model, which we name *detail extractor* f_{detail}, transfers fine-grained details from the unedited reference image I to our synthesized output during the diffusion process. It modulates the synthesizer by cross-attention at each diffusion step, an approach similar to Masa-Ctrl [10] and AnimateAnyone [24]. Both models are initialized from a pretrained Stable Diffusion v1.4 model [42], and finetuned on our paired dataset (§ 3.5). Since we have a detailed reference image I to guide the synthesis, we do not need the coarse semantic guidance provided by CLIP, so we remove the CLIP cross-attention from the model.

Let $T \in \mathbb{N}^*$ be the number of sampling steps, and $\alpha_0, \ldots, \alpha_T \in \mathbb{R}^+$ be the alphas of the diffusion noise schedule [23]. Starting from an image $x_0 \in \mathbb{R}^{3hw}$, the forward diffusion process progressively adds Gaussian noise, yielding a sequence of increasingly noisy iterates:

$$x_t \sim \mathcal{N}\left(\sqrt{\alpha_t} x_{t-1}; (1 - \alpha_t)\mathbf{I}\right). \tag{1}$$

The base diffusion model f is trained to reverse this diffusion process and synthesize an image iteratively, starting from pure noise $x_T \sim \mathcal{N}(0, I)$. The synthesizer and detail extractor in our approach make a few modifications to this base model, which we describe next.

3.3 Extracting details from the reference image

During inference, at each time step t, we start by extracting a set of features F_t from the reference image using f_{detail} (Figure 3, top). These features will guide the synthesis model and help preserve realistic image details and object identity. Since we use a pretrained diffusion model as a feature extractor, we start by adding noise to the reference unedited image:

$$I_t = \sqrt{\bar{\alpha}_t} I + (1 - \bar{\alpha}_t)\epsilon, \tag{2}$$

with $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. We extract the feature tensors immediately before each of the $n = 11$ self-attention blocks in the model:

$$F_t := [f_t^1, \ldots, f_t^n] = f_{\text{detail}}([I_t, I, M]; t), \tag{3}$$

where $[\cdot]$ denotes concatenation along the channel dimension. Our feature extractor also takes as input the clean reference image since it is always available for detail transfer and mask, so the model knows which regions need inpainting. Since the pretrained model only takes I as an input, we modify the first layer at initialization by padding its weight with zeros to accept the additional channel inputs. Using a noisy version of the reference ensures the extracted features are comparable to those in the cross-attention operators of the synthesis model.

3.4 Image synthesis by detail transfer to the coarse edit

The synthesizer f_{synth} generates the final image, conditioned on the detail features F_t. Unlike standard diffusion sampling, we do not start from pure Gaussian noise. Instead, inspired by SDEDit [33], we start from an extremely noisy version of the coarsely edited image:

$$x_T = \sqrt{\bar{\alpha}_T} I_{\text{coarse}} + (1 - \bar{\alpha}_T)\epsilon. \tag{4}$$

This initialization circumvents a commonly observed issue where diffusion models struggle to generate images whose mean and variance deviate from the normal distribution. This is particularly important in our setup as the user input can have arbitrary color distribution, and we need the model to match the user input. This has been shown to stem from a domain gap between training and sampling [21, 31]: the model never sees pure noise during training, but a sample from the normal distribution is the starting point for inference. Our latent initialization addresses this issue by directly bridging the gap between training and inference. In Figure 4 we highlight that by starting from pure noise, we cannot synthesize images with deep dynamic range, while our intiailization does not suffer from such issues. For subsequent steps during inference, we update the current image estimate x_t at each time step t, using the following update rule:

$$x_{t-1} = f_{\text{synth}}([x_t, I_{\text{coarse}}, M]; t, F_t). \tag{5}$$

We provide the mask and coarse edit as conditions by simple concatenation, but because we need to extract fine details from the reference, we found passing

the reference information by cross-attention with the features F_t provided richer information. Again, we extend the weight tensor of the first convolution layer with zeros to accommodate the additional input channels.

Fig. 4: Effects of Latent Initialization. Starting from pure noise, as is standard practice, the model struggles to generate images with deep blacks and synthesizes nonsensical content to keep the image's mean and standard deviation close to the starting Gaussian noise. This is a known issue with current diffusion models [21, 31]. Instead, during inference, we initialize the latent to the warped image with a very large amount of additive Gaussian noise before running the diffusion. This simple change makes a drastic difference and lets the model preserve the image content.

Detail transfer via cross-attention We use the intermediate features $F_t = [f_t^1, \ldots, f_t^n]$, extracted *before* the detail extractor's self-attention layers to transfer fine image details from the reference image to our synthesis network by cross-attention with features $[g_t^1, \ldots, g_t^n]$ extracted *after* the corresponding self-attention layers in the synthesis model. See the right panel of Fig. 3 for an illustration, where Q, K, V are linear projection layers to compute the query, key, and value vectors, respectively, and W_i^t is the matrix of attention scores for layer i, at time step t. The feature tensors g_t^i, f_t^i are 2D matrices whose dimensions are the number of tokens and feature channels, which depend on the layer index i.

3.5 Training with paired supervision from video data

We jointly finetune the two diffusion models on a new dataset obtained by extracting image pairs from videos to reconstruct a ground truth frame given an input frame and a coarse edit automatically generated from it. Our insight is that motion provides useful information for the model to learn how objects change and deform. Videos let us observe the same object interact with diverse backgrounds, lights, and surfaces. For example, skin wrinkles as a person flexes their arm, their clothes crease in complex ways as they walk, and the grass under their feet reacts to each step. Even camera motion yields disocclusion cues and multiple observations of the same scene from various angles.

Concretely, each training sample is a tuple $(I, I_{\text{gt}}, I_{\text{coarse}}, M)$, where I and I_{gt} are the input and ground-truth frames, respectively, extracted from the video with a time interval sampled uniformly at random from $\{1, \ldots, 10\}$ seconds between them. However, if the computed flow between the two frames was too

large (at least 10 percent of the image has a flow magnitude of 350 pixels), we resample another pair. This is to ensure that the warping produces reasonable outputs. We construct the coarse edit I_{coarse} and corresponding mask M using an automated procedure that warps I to approximately match I_{gt}, in a way that mimics our Collage Transform interface. For this, we use one of 2 possible editing models: a flow-based model and a piecewise affine motion model (Fig 5).

Flow-based editing model We compute the optical flow using RAFT-Large [51] for each consecutive pair of frames between I and I_{gt} and compose the flow vectors by backward warping the flow to obtain the flow between the two frames. We then forward warp I using softmax-splatting [36], to obtain I_{coarse}, which roughly aligns with the ground truth frame. The forward warping process creates holes in the image. We record these holes in the mask M. Our model needs to learn to inpaint these regions and those we have no correspondence (e.g., an object appearing in the frame). Using flow-based warping helps the model learn to preserve the identity of the input, rather than always hallucinating new poses and content.

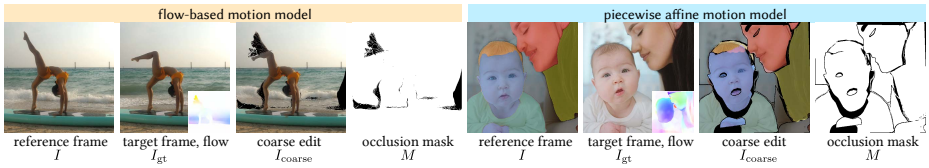

Fig. 5: Motion models. To generate aligned training pairs, we use 2 motion model to warp the reference frame towards the ground truth (target frame). The first model uses optical flow (left). It provides the most accurate alignment but does not correspond to what the user would provide during inference. This motion model encourages adherence of our model's output to the layout specified using the coarse edit. To generate training pairs closer to the collage-like user inputs, we use a second motion model (right). For this, we segment everything in the image [29] and apply similarity transforms to each segment, estimated from the flow within the segment. Figure 9 analyses the impact of these motion models on the final result.

Piecewise affine editing model Optical flow warping can sometimes match the ground truth too closely. As we discuss in Section 4 and Figure 9, training the flow-based editing model only can limit the diversity of our outputs, leading to images that do not deviate much from the coarse edit. Flow-warping is also reasonably distinct from our expected test-time user inputs (§ 3.1). Our second editing model addresses these issues by transforming the reference frame as a *collage*. We compute a depth map for the image using MiDaS [40, 41] and automatically segment the image using SegmentAnything [29].

We then transform each segment using the affine transformation that best matches the optical flow for this segment, compositing them back to front according to each segment's average depth. For the image regions that are not segmented, we use the optical flow warping scheme described above.

We use a dataset consisting of 12 million 5-10 second clips of stock videos, and we filter out keywords that indicate static scenes or synthetic/animated videos, as we are only interested in photo-realistic videos and also highly dynamic scenes where the motion is too large (like car racing). For each valid clip, we sample one pair and compute the warping using both motion models. After filtering for desired motion, we use 2.5 million clips, creating a dataset consists of 2.5 million samples for each motion model, making a total of 5 million training pairs.

3.6 Implementation details

We finetune both models jointly for 120,000 steps with a batch size of 32, using Adam [28], with a learning rate of 1×10^{-5} on 8 NVIDIA A100 GPUs, which takes approximately 48 hours. Note that this is considerably more efficient than recent compositing work [56] that uses 64 NVIDIA V100 GPUs for 7 days. We hypothesize that the stronger input signal helps the model converge faster. We use a linear diffusion noise schedule, with $\alpha_1 = 0.9999$ and $\alpha_T = 0.98$, with $T = 1000$. During inference, we sample using DDIM for 50 denoising steps.

4 Experimental Results

We evaluate our method qualitatively on a set of user edits to demonstrate real-world use cases, as well as on a held-out validation dataset created in the same way as our training set (§ 3.5) for quantitative evaluation.

Our model is trained on a synthetically-generated dataset. We validate that it generalizes to real user edits using a prototype interface illustrating our segment-based editing workflow. The user can segment any part of the image and transform, duplicate, or delete it. We provide a video demonstrating this editing interface in the supplementary materials. To the best of our knowledge, no previous work focuses exactly on our use case (photorealistic spatial edits), so we adapt closely related techniques to our problem setting for comparison. Specifically, we compare to the following baselines:

1. SDEdit [33]: a general text-based editing method that trades off the adherence to the input image and the faithfulness to the text. This is the most general method we compare against, as we can directly provide it with the coarse user edit and a generated caption.
2. AnyDoor [12]: an image compositing model that harmonizes objects from a source frame to a target frame. We follow the author's method of using it for spatially compositing an image by inpainting the object using an off-the-shelf inpainting algorithm and re-inserting the object into the desired location.
3. DragDiffusion [46]: a drag-based editing model that takes source-target key-handles to move parts of the object for re-posing.

Adapting the baselines. We convert our inputs to the interface expected by these baselines for comparison. SDEdit requires choosing a strength parameter

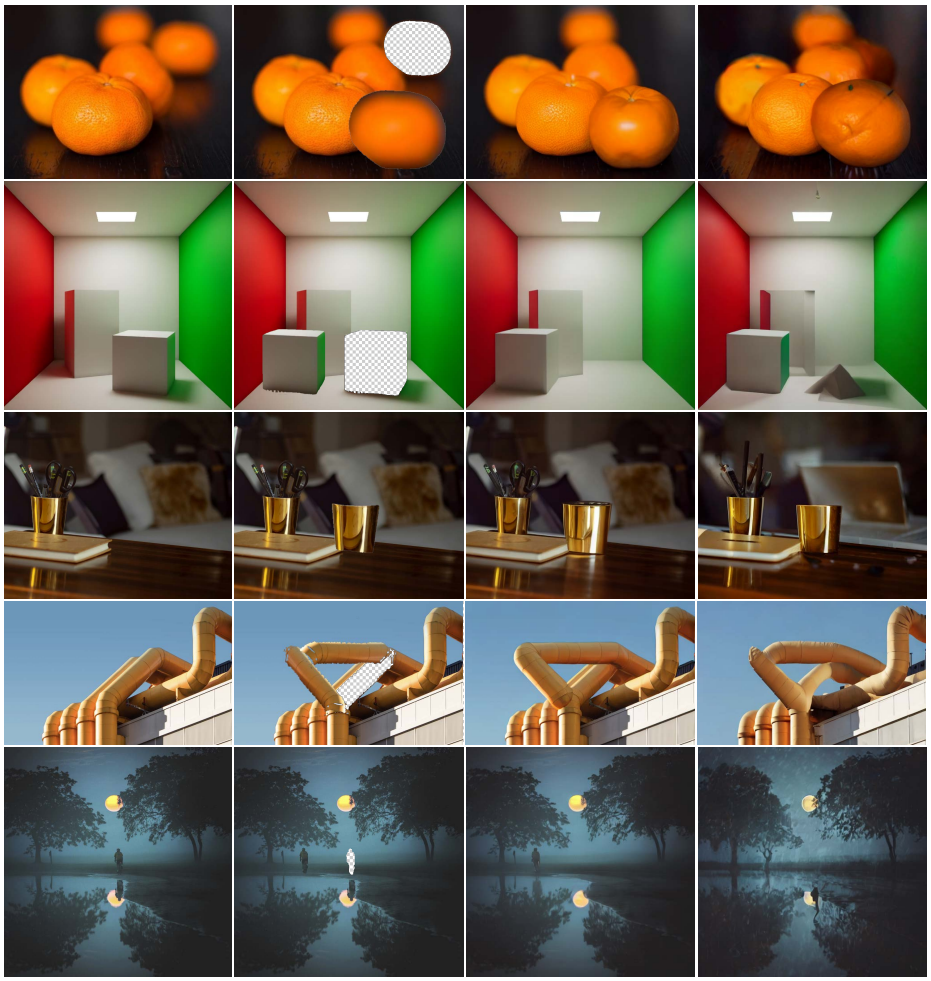

Fig. 6: Applications. We show example of scene recompositing. Our model is capable of synthesizing compelling effects that harmonize realistically with the rest of the image such as: changing the depth of field (row 1), adjusting the global illumination (green reflection on the cube, row 2), and removing or adding reflections (rows 3 and 5).

dictating the amount of noise added to the input and trades off between faithfulness and unconstrained synthesis. We set the strength to 0.4 in all experiments, i.e. we start at 40% of the way through the diffusion process, adding the corresponding level of noise to I_{coarse}. Unlike ours, their model expects a text input, which we automatically compute using BLIP [30].

To insert an object into a scene with AnyDoor, the user selects the object in a source image, and the destination region in a different target image. To adapt it to our use case, we follow the authors' suggestion of using the same image as source and target, using an off-the-shelf inpainting model to remove the selected object, then re-inserting it in a different image region. Their method offers limited control: the size of the insertion region is the only way to control the synthesized pose.

To compare with DragDiffusion [46], we record the segment motion in our user interface, compute the motion vectors for each pixel, and use this information to automatically create the keypoint-handles input needed by DragDiffusion.

Fig. 7: Comparison to Anydoor [12]. Anydoor was trained to insert objects from one image to another. We can repurpose their approach for our image editing task by using the same image as source and target. Their approach does not preserve the dog's identity in this example. AnyDoor also does not harmonize the lighting properly (the sun direction and shadows are wrong), the image is too bright, and some blending seams are visible. On the other hand, our output shows natural shadows and plausible contacts with the ground, adding realistic moving sand consistent with the pose.

4.1 Evaluation on user edits

Image recomposition. Figure 6 shows our model adds realistic details to objects moved to a region of sharper focus, snaps disconnected objects together, and resynthesizes shadows and reflections as needed. In Fig. 7, we used our model to delete the dog (and automatically remove the shadow), and then re-inserted the dog using AnyDoor. The dog's identity underwent significant changes, and AnyDoor does not harmonize the composite with the ground. It also does not completely remove the halo caused by the inpainting mask in the destination region. In contrast, our model synthesizes a coherent output without discontinuity artifacts.

Image reposing. Since we allow the user to edit the image by selecting segments of arbitrary size, the user can re-pose objects by selecting sub-parts and applying an affine transformation on them, effectively animating the object. In

Fig. 8 compares our method to DragDiffusion. DragDiffusion moves the lion's body higher up, which loosely aligns with the user edit, but is inconsistent with the user's intent of only moving the head. This example highlights how a non-interactive point-dragging interface can be at odds with the user's desired output, because it does not provide a good preview of what the model would generate before running it. Our Collage Transform interface is more immediate, and our coarse edit aligns with the final output. In the second example, DragDiffusion collapses, likely because the user input is complex and goes beyond a minimal displacement of the subject that it can handle.

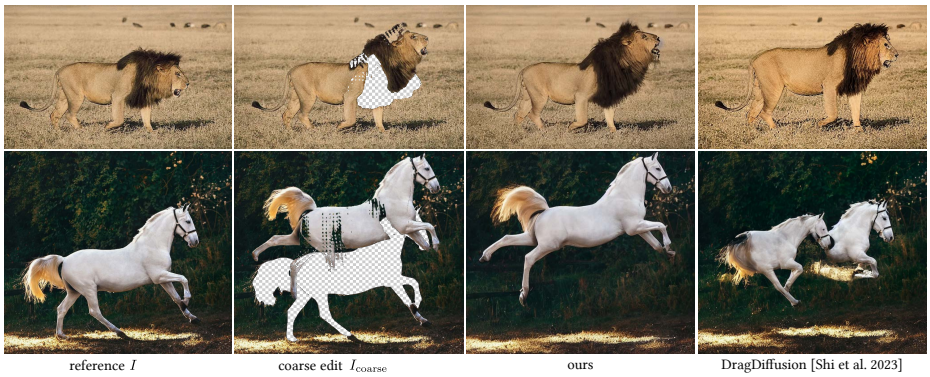

Fig. 8: Comparison with DragDiffusion. We use the Drag Diffusion [46] to generate the results in the right column. We seed dragging control points this method expects for each of the modified image segments, and displace them using the same affine transform used to produce our coarse edit (second column). DragDiffusion generates fairly conservative image edits, and collapses with more drastic reposing edits. However, our method successfully handles wide range of reposing levels.

Preceptual user study. To evaluate the realism of our editing, we conducted a user study to compare the quality of our edits against the edits with SDEdit [33]. We used 30 diverse photo edits, with 27 students participating and voting for all pairs of images. For each pair, we provided the users with the reference image as well as the *intended* user edit, and asked for each sample the following "For the following edit, which of those images do you find a more realistic result?" in a 2-alternative forced-choice (2AFC) format. For 80% of the edits, at least 75% of the users preferred our method. For the remaining images, except for one image, users preferred our method $65-80\%$ of the time. For one image in out of domain edit (editing a non-realistic artistic painting), users preferred both edits almost equally likely (52 % of users preferred SDEdit). We include a more detailed analysis as well as the visuals used in the supplementary material.

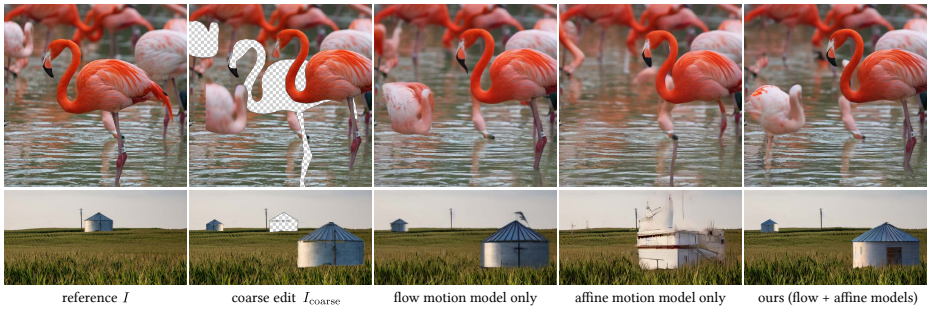

Fig. 9: Motion models ablation. We compare how the 2 motion models we use to create our coarse edits (column 2) during training affect the model's behavior. If we warp the reference frame (column 1) using the flow only (column 3), the model learns how to harmonize the edges of the edited regions, but remains very conservative and does not add much details to increase realism. On the other extreme, if we only use the piecewise affine motion model (column 4), the model learns to hallucinate excessively, losing its ability to preserve object identity. Our full solution trains with both motion models (column 5) to increase the model versatility, allowing the model to generate realistic details while still maintaining good adherence to the user input.

4.2 Ablation studies

In this section, we evaluate the role that different motion models play, as well as the importance of cross-reference attention.

Qualitative comparison. Intuitively, training the model only on flow-warped images would prevent the model from learning to synthesize drastic changes, since flow-warping tends to be well-aligned around the edges. On the other hand, using the piecewise-affine motion model requires the model to adjust the pose of each segment (and learn to connect them together nicely), which forces the model to only use the input as a coarse conditioning. In Fig. 9, we show that the behavior of the model trained on different motion models is consistent with our intuition, where the model trained on flow-only preserves the content and refines the edges, while the model trained only on the piecewise-affine model struggles with preserving identity. On the other hand, the model trained on different motion models falls in the sweet-spot where it addresses user edits faithfully while adding content as needed.
On the architecture side, we compare using only the CLIP image embedding of the reference for the cross-attention as opposed to the cross-reference-attention. Since CLIP embeddings only carry semantics, we observe in Fig. 10 that the model struggles in harmonizing the edited regions, because of a limited awareness of what has changed in the image.

Quantitative comparison. We evaluate our ablations on a held-out validation dataset from our video dataset. In the table on the right, we show that the model trained with flow-data and affine-motion are the top performers on perceptual

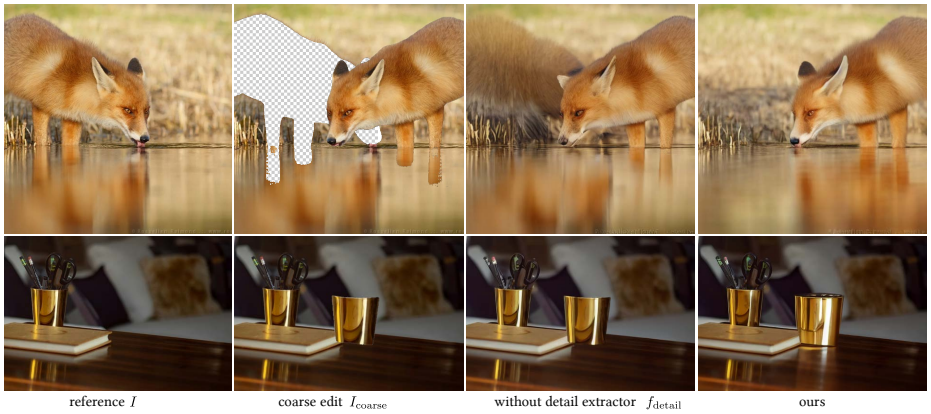

Fig. 10: Architecture ablation. Without the detail extractor branch (3rd column), the model struggles with spatial reasoning as it cannot access the grounding of the original reference image (1st column). This ablation's outputs are overly conservative, not steering too far away from the coarse edit (2nd column). Our full model produces much more realistic edits (4th column), with harmonious shadows and object-background contact. It refines object boundaries and synthesizes plausible reflections.

Model & Training Data	Test Data	LPIPS ↓
Piecewise affine	Piecewise affine	**0.231 ± 0.007**
	Flow-based	0.220 ± 0.007
Flow-based	Piecewise affine	**0.229 ± 0.007**
	Flow-based	**0.190 ± 0.007**
Both motion models (no cross-ref attn)	Piecewise affine	0.327 ± 0.007
	Flow-based	0.269 ± 0.008
Both motion models (Full method)	Piecewise affine	**0.231 ± 0.007**
	Flow-based	**0.196 ± 0.007**

loss on both types of test and that dropping the cross-reference attention and relying on the reference CLIP embedding causes a severe drop in performance.

5 Limitations and conclusions

We present a method of assisting artists in photo editing through generative models while retaining a large level of control that traditional editing pipelines provide. We observe that with the appropriate motion model, we can use videos to train a model that can serve as a direct plugin in the editing process. We hope that our work inspires future editing research that can simply remove the cumbersome last-mile work by the press of a button.

Our generative model is trained for spatial compositions using video data. It can spatially re-compose parts of the image but would struggle to insert objects

from a completely different image as opposed to image composition baselines. Furthermore, we inherit the limitations of Latent Diffusion Models, which we use as our base models, especially for generating hands, faces, and small objects.

Acknowledgment We would like to thank Sachin Shah for testing our user interface and creating several of the artwork used throughout the paper, and we graciously thank him for his feedback on the paper writing and project page.

References

1. Abdal, R., Qin, Y., Wonka, P.: Image2stylegan: How to embed images into the stylegan latent space? In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4432–4441 (2019) 4
2. Abdal, R., Qin, Y., Wonka, P.: Image2stylegan++: How to edit the embedded images? In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8296–8305 (2020) 4
3. Andonian, A., Osmany, S., Cui, A., Park, Y., Jahanian, A., Torralba, A., Bau, D.: Paint by word. arXiv preprint arXiv:2103.10951 (2021) 2
4. Avidan, S., Shamir, A.: Seam carving for content-aware image resizing. In: ACM SIGGRAPH 2007 Papers. SIGGRAPH '07, Association for Computing Machinery, New York, NY, USA (2007). https://doi.org/10.1145/1275808.1276390, https://doi.org/10.1145/1275808.1276390 4
5. Avrahami, O., Lischinski, D., Fried, O.: Blended diffusion for text-driven editing of natural images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18208–18218 (2022) 4
6. Bansal, A., Chu, H.M., Schwarzschild, A., Sengupta, S., Goldblum, M., Geiping, J., Goldstein, T.: Universal guidance for diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 843–852 (June 2023) 4, 5
7. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: A randomized correspondence algorithm for structural image editing. ACM Trans. Graph. **28**(3), 24 (2009) 4
8. Bertalmío, M., Bertozzi, A., Sapiro, G.: Navier-stokes, fluid dynamics, and image and video inpainting. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001 **1**, I–I (2001), https://api.semanticscholar.org/CorpusID:695955 6
9. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18392–18402 (2023) 2, 4
10. Cao, M., Wang, X., Qi, Z., Shan, Y., Qie, X., Zheng, Y.: Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 22560–22570 (October 2023) 2, 4, 6
11. Chai, L., Wulff, J., Isola, P.: Using latent space regression to analyze and leverage compositionality in gans. arXiv preprint arXiv:2103.10426 (2021) 4
12. Chen, X., Huang, L., Liu, Y., Shen, Y., Zhao, D., Zhao, H.: Anydoor: Zero-shot object-level image customization. arXiv preprint (2023) 2, 4, 5, 10, 12
13. Cho, T.S., Butman, M., Avidan, S., Freeman, W.T.: The patch transform and its applications to image editing. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2008) 4

14. Couairon, G., Verbeek, J., Schwenk, H., Cord, M.: Diffedit: Diffusion-based semantic image editing with mask guidance. arXiv preprint arXiv:2210.11427 (2022) 2
15. Crowson, K., Biderman, S., Kornis, D., Stander, D., Hallahan, E., Castricato, L., Raff, E.: Vqgan-clip: Open domain image generation and editing with natural language guidance. In: European Conference on Computer Vision. pp. 88–105. Springer (2022) 4
16. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in neural information processing systems **34**, 8780–8794 (2021) 2
17. Epstein, D., Jabri, A., Poole, B., Efros, A.A., Holynski, A.: Diffusion self-guidance for controllable image generation. Advances in Neural Information Processing Systems (2023) 4
18. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12873–12883 (2021) 2
19. Gal, R., Patashnik, O., Maron, H., Bermano, A.H., Chechik, G., Cohen-Or, D.: Stylegan-nada: Clip-guided domain adaptation of image generators. ACM Transactions on Graphics (TOG) **41**(4), 1–13 (2022) 4
20. Geng, D., Owens, A.: Motion guidance: Diffusion-based image editing with differentiable motion estimators. International Conference on Learning Representations (2024) 4
21. Guttenberg, N.: Diffusion with offset noise (jan 2023), https://www.crosslabs.org/blog/diffusion-with-offset-noise 7, 8
22. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control (2022) 4
23. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020) 2, 6
24. Hu, L., Gao, X., Zhang, P., Sun, K., Zhang, B., Bo, L.: Animate anyone: Consistent and controllable image-to-video synthesis for character animation. arXiv preprint arXiv:2311.17117 (2023) 4, 6
25. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017) 3
26. Jia, J., Sun, J., Tang, C.K., Shum, H.Y.: Drag-and-drop pasting. ACM Trans. Graph. **25**(3), 631–637 (jul 2006). https://doi.org/10.1145/1141911.1141934, https://doi.org/10.1145/1141911.1141934 4
27. Kim, G., Kwon, T., Ye, J.C.: Diffusionclip: Text-guided diffusion models for robust image manipulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2426–2435 (2022) 2, 4
28. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014), https://api.semanticscholar.org/CorpusID:6628106 10
29. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2023) 3, 6, 9
30. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: ICML (2022) 12
31. Lin, S., Liu, B., Li, J., Yang, X.: Common diffusion noise schedules and sample steps are flawed. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 5404–5411 (2024) 7, 8

32. Luo, G., Darrell, T., Wang, O., Goldman, D.B., Holynski, A.: Readout guidance: Learning control from diffusion features. arXiv preprint arXiv:2312.02150 (2023) 4
33. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: SDEdit: Guided image synthesis and editing with stochastic differential equations. In: International Conference on Learning Representations (2022) 4, 5, 7, 10, 13
34. Mokady, R., Hertz, A., Aberman, K., Pritch, Y., Cohen-Or, D.: Null-text inversion for editing real images using guided diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6038–6047 (2023) 4
35. Mou, C., Wang, X., Song, J., Shan, Y., Zhang, J.: Dragondiffusion: Enabling drag-style manipulation on diffusion models (2023) 4
36. Niklaus, S., Liu, F.: Softmax splatting for video frame interpolation. In: IEEE Conference on Computer Vision and Pattern Recognition (2020) 9
37. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.Y., Xu, H., Sharma, V., Li, S.W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: Dinov2: Learning robust visual features without supervision (2023) 4
38. Pan, X., Tewari, A., Leimkühler, T., Liu, L., Meka, A., Theobalt, C.: Drag your gan: Interactive point-based manipulation on the generative image manifold. In: ACM SIGGRAPH 2023 Conference Proceedings. SIGGRAPH '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3588432.3591500, https://doi.org/10.1145/3588432.3591500 2, 4
39. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021) 4
40. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. ArXiv preprint (2021) 9
41. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2020) 9
42. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models (2021) 2, 3, 6
43. Rubinstein, M., Shamir, A., Avidan, S.: Improved seam carving for video retargeting. ACM transactions on graphics (TOG) **27**(3), 1–9 (2008) 4
44. Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., Norouzi, M.: Palette: Image-to-image diffusion models. In: ACM SIGGRAPH 2022 Conference Proceedings. pp. 1–10 (2022) 3
45. Sarukkai, V., Li, L., Ma, A., Ré, C., Fatahalian, K.: Collage diffusion. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 4208–4217 (2024) 2, 4
46. Shi, Y., Xue, C., Pan, J., Zhang, W., Tan, V.Y., Bai, S.: Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. arXiv preprint arXiv:2306.14435 (2023) 2, 4, 10, 12, 13
47. Simakov, D., Caspi, Y., Shechtman, E., Irani, M.: Summarizing visual data using bidirectional similarity. In: 2008 IEEE conference on computer vision and pattern recognition. pp. 1–8. IEEE (2008) 4
48. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020) 4

49. Song, Y., Zhang, Z., Lin, Z., Cohen, S., Price, B., Zhang, J., Kim, S.Y., Aliaga, D.: Objectstitch: Object compositing with diffusion model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18310–18319 (2023) 2
50. Sunkavalli, K., Johnson, M.K., Matusik, W., Pfister, H.: Multi-scale image harmonization. ACM Trans. Graph. **29**(4) (jul 2010). https://doi.org/10.1145/1778765.1778862, https://doi.org/10.1145/1778765.1778862 4
51. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. pp. 402–419. Springer (2020) 9
52. Wang, S., Saharia, C., Montgomery, C., Pont-Tuset, J., Noy, S., Pellegrini, S., Onoe, Y., Laszlo, S., Fleet, D.J., Soricut, R., et al.: Imagen editor and editbench: Advancing and evaluating text-guided image inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18359–18369 (2023) 2
53. Wang, Y.S., Tai, C.L., Sorkine, O., Lee, T.Y.: Optimized scale-and-stretch for image resizing. In: ACM SIGGRAPH Asia 2008 papers, pp. 1–8. Association for Computing Machinery (2008) 4
54. Xie, S., Zhang, Z., Lin, Z., Hinz, T., Zhang, K.: Smartbrush: Text and shape guided object inpainting with diffusion model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22428–22437 (2023) 2
55. Xu, Z., Zhang, J., Liew, J.H., Yan, H., Liu, J.W., Zhang, C., Feng, J., Shou, M.Z.: Magicanimate: Temporally consistent human image animation using diffusion model (2023) 4
56. Yang, B., Gu, S., Zhang, B., Zhang, T., Chen, X., Sun, X., Chen, D., Wen, F.: Paint by example: Exemplar-based image editing with diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18381–18391 (2023) 4, 5, 10
57. Yenphraphai, J., Pan, X., Liu, S., Panozzo, D., Xie, S.: Image sculpting: Precise object editing with 3d geometry control. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024) 4
58. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models (2023) 4
59. Zhu, J.Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. In: European Conference on Computer Vision (2016), https://api.semanticscholar.org/CorpusID:14924561 4

A User study

We asked 27 users to evaluate 30 pairs of our output against the baseline, with the question "For the following edit, which of those images do you find a more realistic result?" So that the user considers the realism of the output as well as the faithfulness of the output to the edit. Out of total of 810 votes, 722 votes were for Magic Fixup edit. In 8 out of 30 images, 100% of the users preferred our method over the baseline. The output with lowest votes for Magic Fixup had 13 out of 27 votes, so the least preferred edit was on par with SDEdit. In Figure 11, we plot the preferences of the users for Magic Fixup, in a sorted order. We note that there is a significant preference for our model against SDEdit in the majority of edits. The pair with the least votes for Magic Fixup is the edit

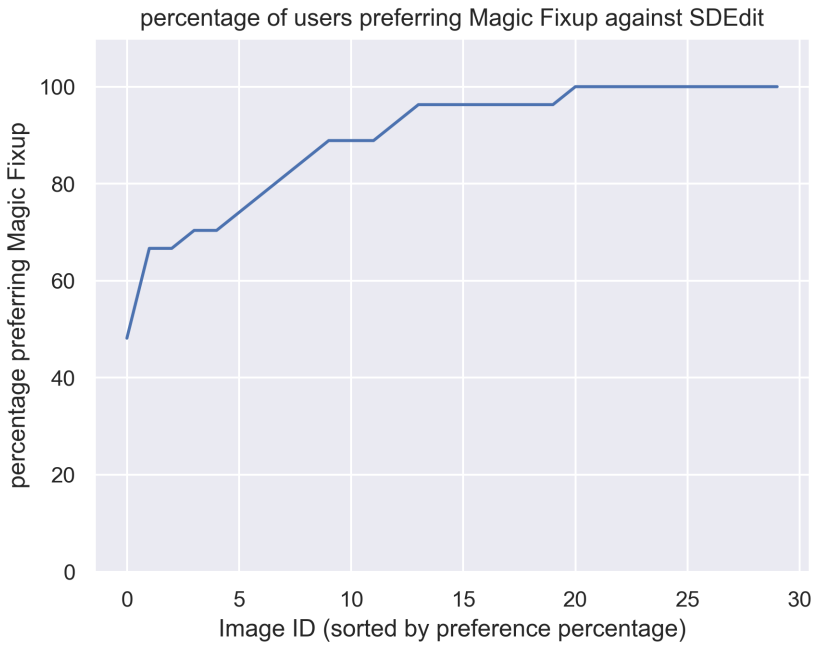

Fig. 11: User study comparisons. Here we show the percentage amount of users that preferred our editing output against SDEdit, in a sorted order in terms of the percentage preference. Note that users heavily prefer our images in majority of images, with 8 out of 30 edits, all users unanimously preferred our edit.

of the Monet painting shown on the last row of Figure 12. Note that paintings are out of the domain for our model since we cannot have videos of dynamic painting to train on. In the output on the painting edit, we can notice that the brush strokes associated with impressionist painting style is less apparent in the output, as the model is increasing the realism of the output.

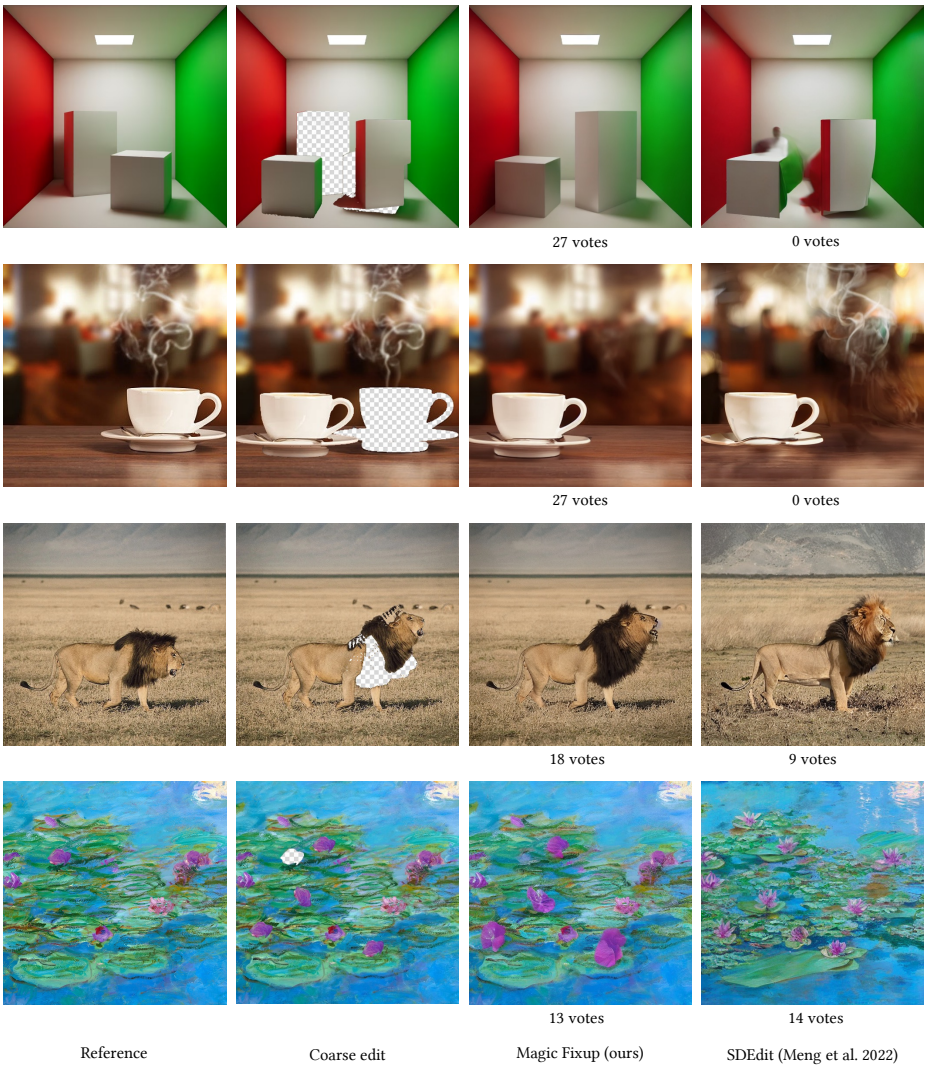

Fig. 12: Visual comparisons for the user study. We show sample pairs from the user study that compare our method against SDEdit. The top two rows are examples where users unanimously preffered our method. The last row (the painting example), is the example with the least votes for our method, where the number of votes is on par with the votes for SDEdit.