GenHowTo: Learning to Generate Actions and State Transformations from Instructional Videos

Tomáš Souček[1]   Dima Damen[2]   Michael Wray[2]   Ivan Laptev[3]   Josef Sivic[1]

[1]CIIRC CTU   [2]University of Bristol   [3]MBZUAI

tomas.soucek@cvut.cz   https://soczech.github.io/genhowto/

Action prompt: ***a person slicing an avocado on a cutting board***
Final state prompt: ***slices of an avocado on a cutting board***

Action prompt: ***a man pouring beer into a glass***
Final state prompt: ***a man sitting at a table holding a glass of beer***

Action prompt: ***an electric mixer is being used to make whipped cream***
Final state prompt: ***whipped cream in a bowl***

Action prompt: ***a person is wrapping a tortilla on a plate***
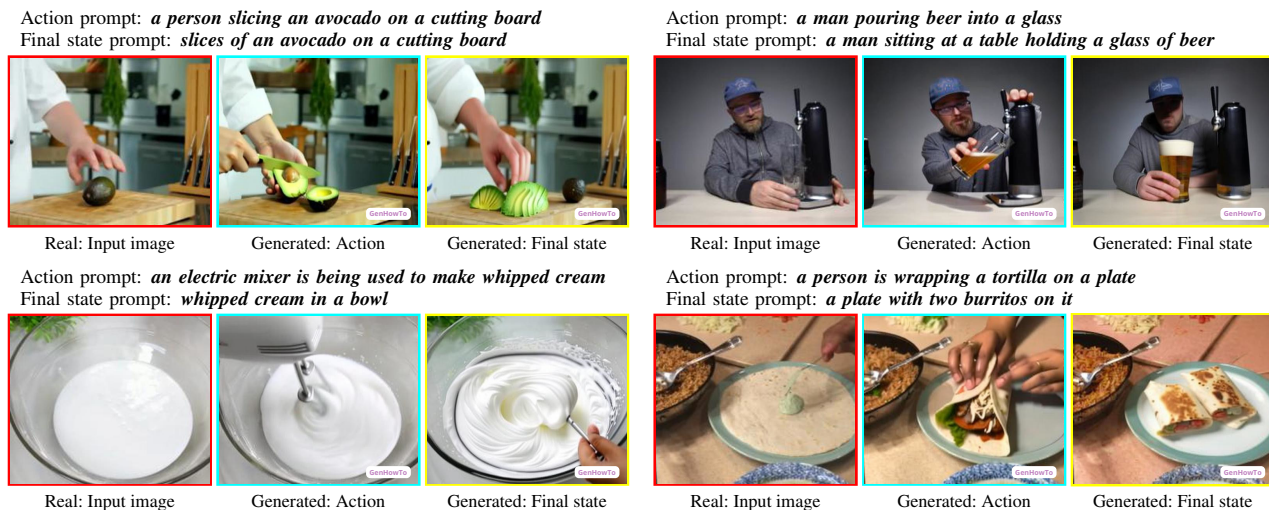Final state prompt: ***a plate with two burritos on it***

Figure 1. Given an image of an initial scene (red) and text prompts (**bold**), GenHowTo generates images corresponding to the action (blue) and the final state when action is completed (yellow). GenHowTo is learned from instructional videos and can generate new images of both seen and previously unseen object transformations. Importantly, GenHowTo learns to maintain the parts of the scene that showcase the action carried out in the same environment, in the spirit of HowTo examples, while introducing important objects (*e.g.*, hand and knife in the first example) and transforming the object according to the prompt.

Abstract

We address the task of generating temporally consistent and physically plausible images of actions and object state transformations. Given an input image and a text prompt describing the targeted transformation, our generated images preserve the environment and transform objects in the initial image. Our contributions are threefold. First, we leverage a large body of instructional videos and automatically mine a dataset of triplets of consecutive frames corresponding to initial object states, actions, and resulting object transformations. Second, equipped with this data, we develop and train a conditioned diffusion model dubbed GenHowTo. Third, we evaluate GenHowTo on a variety of objects and actions and show superior performance compared to existing methods. In particular, we introduce a quantitative evaluation where GenHowTo achieves 88% and 74% on seen and unseen interaction categories, respectively, outperforming prior work by a large margin.

1. Introduction

Imagine you want to cook dinner or prepare your favorite cocktail to spend an evening with friends. Your inspiration may come from an image of a dish you have recently seen in a restaurant or a cookbook. In a similar spirit, recent work in robotics aims to learn goal-conditioned policies where the tasks or intermediate goals are defined by images [12, 42–44, 50, 68]. Image-defined tasks have an advantage of providing detailed information such as the desired stiffness of the whipped cream or the thickness of avocado slices, which can be difficult to convey by language alone.

Moreover, policies can be further improved by image sequences illustrating the execution of a task in terms of actions [5, 6]. However, images of target states and actions are rarely available in advance for arbitrary tasks and environ-

---

[1]Czech Institute of Informatics, Robotics and Cybernetics at the Czech Technical University in Prague.
[3]Mohamed bin Zayed University of Artificial Intelligence.

ments. Motivated by this limitation, in this work, we aim to generate realistic images of actions and target states for a variety of tasks defined by text prompts, while *preserving the same environment* across the initial image and the generated images of the future, see Figure 1.

With the advent of powerful vision-language models, recent work excels in generating realistic and high-fidelity images from textual descriptions [16, 17, 33, 52, 55, 57]. Other recent methods show impressive results for text-conditioned image editing [4, 20, 24, 26, 40] where input images are modified according to provided text prompts. As these models are not trained to focus on action-induced transformations, they may fail to maintain a consistent environment while generating visually small yet semantically complex object changes. To address this challenge, we leverage image sequences from large-scale video data for training generative models. In particular, we exploit instructional videos with variation of tasks, scenes, and object state changes.

Instructional videos, however, lack annotation at scale. To this end, we build on the recent work [61] and automatically create a large-scale dataset with 200k image triplets and corresponding textual descriptions. Our triplets are temporally ordered video frames corresponding to (i) initial object states, (ii) state-modifying actions, and (iii) objects in new states. Equipped with this dataset, we propose a text- and image-conditioned generative model, GenHowTo, that transforms input images into images with actions and new object states according to textual prompts. We perform qualitative and quantitative evaluation of our method and demonstrate its superior performance.

In summary, we make the following contributions. (1) We address the problem of generating physically plausible state transformations of input images according to a variety of action and final state prompts and automatically collect a large-scale training dataset from narrated instructional videos. (2) We propose GenHowTo, a text-conditioned image generation model that modifies input images according to action or final state text prompts while preserving the scene from the input image. (3) We evaluate GenHowTo both qualitatively and quantitatively and show its superior performance compared to the state of the art. For quantitative results, we train classifiers on object states and actions to assess the method's ability to transform an image from its initial state to the action or final states. Our code, data, and pre-trained models are publicly available.

2. Related work

Object states. Localization, recognition, and generation of object transformations, and changes of object states, play a key role in applications such as procedural planning, robotics, and video action understanding [12, 68, 76, 77]. Most works focus on recognition, compositionality, and localization in videos [11, 13, 14, 41, 62]; however, the gen-

eration of object states has been underexplored. While recent powerful diffusion models have been able to generate rich, diverse, and high-quality images, generating objects in exact states (*e.g.*, carrot cut in a baton shape) is still a challenge [58]. Saini *et al.* [58] introduces a novel benchmark for the generation of object states, yet their focus is very limited to only the *cutting* action and a small dataset of objects recorded against a green screen. We focus on transforming a much wider range of actions and objects in realistic scenes by using in-the-wild instructional internet videos as the source of training data.

Conditional generation. GANs [39, 46, 71] and autoregressive models [15, 53] allowed conditional synthesis of images. Recently, large-scale diffusion models have enabled the synthesis of high-resolution realistic images with diverse and complex attributes [3, 10, 16, 17, 22, 28, 33, 52, 54–57, 60, 69, 73]. These models may be conditioned on images [56, 69, 73], textual descriptions [16, 17, 33, 52, 55, 57], or other modalities [3] to produce the desired output. While such models produce images that satisfy the text prompt or are similar to the input image, applying any targeted edits is difficult. To tackle that, new text-based methods and image-to-image models for image editing have emerged.

Text-based image editing. Using visual-text embeddings, such as CLIP [51], allowed editing images with only textual descriptions [2, 20, 24, 26, 27, 40, 45, 48, 49, 67]. Prompt-mixing [49] uses different text prompts during the denoising process to generate image variations. Alternatively, Prompt-to-Prompt [20] can apply various edits to generated images by tweaking activations of a trained diffusion model. Further, it can also be extended to editing real images [40]. Similarly, Imagic [26] also edits images using text prompts. However, these methods require hundreds of back-propagation steps to edit a single image, making them impractical. In contrast, we train a diffusion model that can manipulate any image by applying a standard diffusion schedule with the source image and the target prompt.

Image-to-image models. Image-to-Image translation [4, 8, 9, 15, 25, 47, 53, 70, 74, 75, 78–80] attempts to convert images between domains. Image-conditioned image generation has been attempted with adversarial networks using paired [25] or unpaired data with cycle-consistent loss [79]. Other methods explored conditional generation by inpainting a masked-out part of the input image [7, 18, 29, 66, 73]. Recently, Zhang *et al.* introduced ControlNet [74]—a scheme for training diffusion models with additional image conditioning using paired data such as automatically obtained edge images, thus allowing for fine-grained image generation. However, adapting diffusion models to image-to-image generation with unpaired data is not trivial. Brooks *et al.* [4] solves the lack of paired data by generating the pairs using diffusion models. In contrast, we find corresponding pairs of images by mining internet instructional videos.
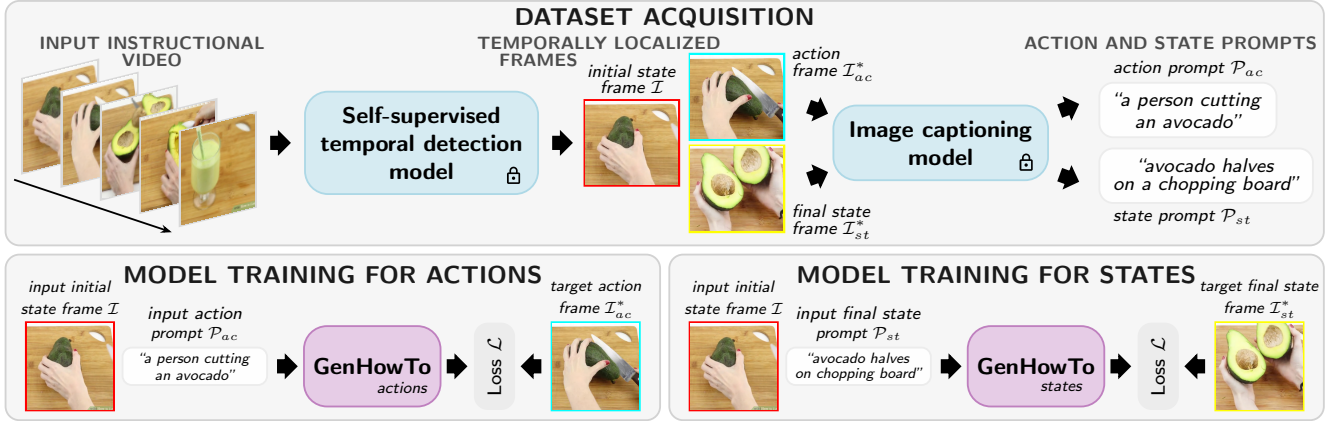
Figure 2. **Method overview.** We use a self-supervised model to detect objects before, during, and after they are manipulated in instructional videos (top left). Then, the detected frames are automatically annotated using an image captioning model (top right). Finally, the detected frames with the text annotations are used to train our two diffusion models for transforming objects in the images (bottom).

Video Frame Prediction. Another related topic is video frame prediction, in which future frames of a video are generated. This task has been addressed using recurrent networks [34, 59, 63], adversarial training [30, 32, 35, 36], or, more recently, diffusion models [72]. We differ from these approaches as we predict semantic change, such as the action and the final state, using a text prompt, not a continuous frame sequence immediately following the input frames.

3. Learning to generate actions and state transformations

Given an image of an object in a scene, our goal is to generate images that showcase an *action exerted* upon the object as well as the *transformation of the object* as a result of the outcome of that action. Importantly, we wish to produce consistent images, where the action is exerted on the same object in an unchanged environment. For this conditional generation, three challenges need to be tackled:

First, from the same input image, many possible actions or object transformations can be generated. It is essential to condition the generation process on this sought-after transformation. We address this challenge by specifying the action and the desired final state of the object through given text prompts. For example, given an image of an avocado on a chopping board, we specify the action through the prompt: *"a person cutting an avocado"* and the desired final state via the prompt: *"avocado halves on a chopping board."*

Second, to generate consistently paired images showcasing the action or object transformation, the generation should maintain parts of the scene unrelated to the action, including the background and other objects (*e.g.*, the chopping board in the example above) but also introduce new objects consistent with the action (*e.g.*, a knife and a person's hand). We address this challenge with an image-conditioned diffusion model fine-tuned for maintaining a

consistent background between pairs of images.

Third, to train an image-conditioned model, data of such paired images are required. Manually collecting such data at scale is very challenging due to the nature of the problem. We address this challenge by mining 200k triplets of images from instructional HowTo videos.

We define our problem formally in Section 3.1. Then, we address the challenge of acquiring paired image data by extracting images from instructional videos in Section 3.2. In the same section, we address the issue of many possible object transformations by automatically collecting text prompts describing the transformations. Lastly, in Section 3.3, we describe our image- and text-conditioned model to modify the object but maintain the scene.

3.1. Problem definition

We define the problem as follows. Let $\mathcal{I}$ be an image depicting the object in its initial state and $\mathcal{P}$ be a text prompt describing an action exerted upon an object or a transformation (the final state) of the object in the image $\mathcal{I}$. Then, we define the target image $\mathcal{I}^*$ as the modification of $\mathcal{I}$ that satisfies the prompt $\mathcal{P}$. Our goal is to produce an image similar to $\mathcal{I}^*$ given the original image $\mathcal{I}$ and the text prompt $\mathcal{P}$. We differentiate between two target images $\mathcal{I}^*$—one that shows the ongoing action, denoted as $\mathcal{I}^*_{ac}$, and another that demonstrates the final state after the action has concluded, denoted as $\mathcal{I}^*_{st}$. Similarly, we differentiate the prompts describing the action $\mathcal{P}_{ac}$ and the final state $\mathcal{P}_{st}$. We omit the subscripts if we talk about both actions and the final states. We use the input image $\mathcal{I}$ and one of the prompts $\mathcal{P}$ as input. The corresponding target image $\mathcal{I}^*$ to the text prompt serves as the ground truth in the loss during training.

3.2. Dataset acquisition

To form the dataset, our goal is to acquire 5-tuples $(\mathcal{I}, \mathcal{I}^*_{ac}, \mathcal{P}_{ac}, \mathcal{I}^*_{st}, \mathcal{P}_{st})$ which include the input image, the

target action image, the action text prompt, the target final-state image, and the final-state text prompt, respectively. For example, the input image can be an empty glass on a countertop. For the action of filling the glass with orange juice, we aim to find two images, one showcasing the action of filling the glass and a second image showcasing the end state after the action is finished. Importantly the two images should showcase the same glass on the same countertop.

While it is possible to obtain images of glasses, both empty and full, on kitchen countertops, obtaining triplets of images showing (i) the initial state, (ii) the action, and (iii) the final state with the same glass and background in sufficient quantity is difficult. This correspondence, while missing from arbitrary images, is present in instructional videos. Such videos often capture visual changes of objects in static and continuous shots, *i.e.*, keep the same background. However, the challenge in using instructional videos is to find the right frames that correspond to such triplets. We address this challenge next.

Image triplets from instructional videos. We seek instructional videos as a source for our corresponding image triplets. Instructional videos have the advantage of typically capturing a procedure with minimal camera and scene changes, thus showcasing object changes while preserving the background. To get image triplets from these videos, we use the self-supervised model of Souček *et al.* [61], which localizes actions and objects in various states from videos automatically. The model is trained with sets of videos from various interaction categories such as *chopping onions* or *assembling a PC*. For each video frame, the model predicts the likelihood of the frame containing the object (*e.g.*, onion or PC) in its initial state (before the action), during the action, or in its final state (after the action). The model is trained in a self-supervised iterative process where the target labels are computed from the model's own predictions subject to ordering and other constraints. We use this method to train the model on the COIN [65] and the ChangeIt [62] datasets. Once trained, we use this model to localize the image triplets in the input videos.

In detail, for every video from the two datasets, we apply the trained model to select the frames with the highest initial state, action, and final state scores. From each video, we use these three frames to form the triplet. In the next paragraphs, these are extended with text prompts $\mathcal{P}_{ac}$ and $\mathcal{P}_{st}$ to form the dataset 5-tuple.

Enriching images with text prompts. For each input image $\mathcal{I}$, there are multiple possible outcomes. For example, if the input image contains an avocado, the outcome may be the avocado in halves or slices, peeled or smashed. As we wish to generate the image of the action or object transformation, we need a way to specify which of the various options the model should target. Building on the success of text-based conditioning, we use a text description of the

target image to guide the image generation. We refer to this text description as the text prompt $\mathcal{P}$.

A straightforward method of obtaining the text prompt could utilize the corresponding narration from the instructional video where the images were obtained. However, it has been shown that automatic narrations do not always correspond to the ongoing action [19, 38]. Additionally, even when aligned, we observe the narrations do not emphasize the objects and actions shown in videos (see examples in the appendix [1]). Therefore, instead of the narrations, we use the image captioning model BLIP2 [31] applied to both target images $\mathcal{I}_{ac}^*$ and $\mathcal{I}_{st}^*$ to produce accurate text prompts $\mathcal{P}_{ac}, \mathcal{P}_{st}$ describing the desired outcome of the transformation, completing the dataset 5-tuple $(\mathcal{I}, \mathcal{I}_{ac}^*, \mathcal{P}_{ac}, \mathcal{I}_{st}^*, \mathcal{P}_{st})$.

3.3. GenHowTo model

We propose a model for generating our target images using the input image and text prompts. We refer to our model as **GenHowTo**, given its capability of generating actions or final state images from an initial image and the text prompt, as well as its training from instructional videos.

To train our GenHowTo model, we utilize the dataset described in Sec. 3.2. Critically, we train two distinct models with different sets of weights, one for generating action images and one for generating the final state images. We ablate this decision against training jointly in Section 4. We thus instantiate two subsets from our dataset: $(\mathcal{I}, \mathcal{I}_{ac}^*, \mathcal{P}_{ac})$ for actions and $(\mathcal{I}, \mathcal{I}_{st}^*, \mathcal{P}_{st})$ for final states. For the rest of this section, we refer to the input to our model in a general manner as $(\mathcal{I}, \mathcal{I}^*, \mathcal{P})$. The notation is applicable to any of the two cases noted above.

Our model aims to generate images that maintain parts of the input scene irrelevant to the action or object transformation, such as the background. At the same time, the model needs to modify objects in the scene while preserving their identity or even introduce new objects in accordance with the text prompt. For example, the new objects can be hands, if a prompt calls for some action performed. To solve these challenges, we build on the state-of-the-art ControlNet architecture [74], which we adapt to our setting. In particular, in order to preserve the input scene yet allow for major semantic edits (introducing new objects or changes of the object state), we (i) introduce semantic conditioning on the input image at training time and (ii) initialize the noise sampling from the input image at inference time. Details of the model, including our modifications, are given in the remainder of this section.

Model components. The model ϵ_θ, shown in Figure 3, is a pretrained text-conditioned latent Stable Diffusion [54] formed by latent U-Net encoder $\mathcal{U}_\mathcal{E}$ and latent U-Net decoder $\mathcal{U}_\mathcal{D}$. The latent representation is then decoded into the output image using a VAE decoder $\mathcal{A}_\mathcal{D}$. To incorporate the conditioning on the input RGB image, the latent U-Net
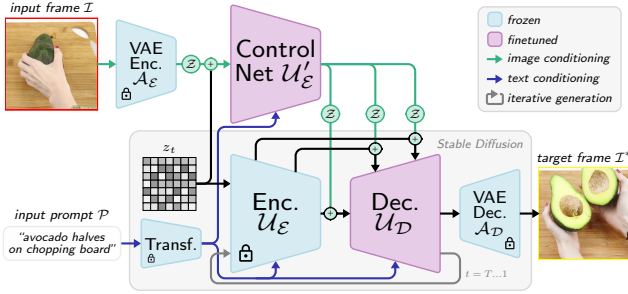
Figure 3. **GenHowTo model overview.** The model ϵ_θ takes as input (left) a frame depicting the object in its initial state $\mathcal{I}$ and a text prompt $\mathcal{P}$ describing an action or the desired final state. The output of the model is an image $\mathcal{I}^*$ of the same scene but depicting the action or the desired final state.

encoder $\mathcal{U}_\mathcal{E}$ is duplicated to form the ControlNet encoder $\mathcal{U}'_\mathcal{E}$. The outputs of both encoders are summed together before being passed into the UNet decoder $\mathcal{U}_\mathcal{D}$.

Semantic image conditioning. While ControlNet has mainly focused on spatial edits by conditioning, *e.g.*, on edge maps or human poses, our goal is to enable semantic edits of the input scene given by an RGB image. Hence, in contrast to ControlNet [74], which projects the input image into the latent space by a handful of randomly initialized layers, our model directly projects the input image into the latent "semantic" space of the U-Net using the encoder $\mathcal{A}_\mathcal{E}$ of the pre-trained VAE. This semantic representation of the input image is then summed with the standard noise input and passed into the ControlNet encoder. To produce meaningful gradients at the start of the training, independent learnable convolution layers $\mathcal{Z}_i$ are added to the outputs of the ControlNet encoder $\mathcal{U}'_\mathcal{E}$ with their parameters initialized to zeros.

Model training. All parameters, including the duplicated ControlNet encoder $\mathcal{U}'_\mathcal{E}$, are initialized by the pretrained Stable Diffusion weights except for the zeroed-out convolutional layers $\mathcal{Z}_i$. During training, we keep the original Stable Diffusion encoder $\mathcal{U}_\mathcal{E}$ frozen and only finetune the decoder $\mathcal{U}_\mathcal{D}$ and the ControlNet encoder $\mathcal{U}'_\mathcal{E}$. For training, the target image $\mathcal{I}^*$ is encoded by the pretrained VAE encoder to produce the latent encoding z. Further, a timestep $t \in [1, T]$ is sampled, and noise ϵ is added to z according to the diffusion schedule, producing the noised latent z_t. The model ϵ_θ is trained to minimize the squared L_2 loss (Eq. (1)) between the predicted noise and the actual noise ϵ:

$$\mathcal{L} = \mathbb{E}_{(\mathcal{I}, \mathcal{I}^*, \mathcal{P}),\, t,\, \epsilon \sim \mathcal{N}(0,1)} \left[\| \epsilon - \epsilon_\theta \left(z_t, t, \mathcal{I}, \mathcal{P} \right) \|_2^2 \right], \quad (1)$$

where the expectation is over the input dataset (input $\mathcal{I}$ and target image $\mathcal{I}^*$, input prompt $\mathcal{P}$) and diffusion timesteps t. ϵ is the actual noise, and $\epsilon_\theta(z_t, t, \mathcal{I}, \mathcal{P})$ is the predicted noise by the model with parameters θ for input image $\mathcal{I}$ with input prompt $\mathcal{P}$ at timestep t with noised latent z_t.

Model inference and sampling initialization. To generate images, we use DDIM sampling [60] with classifier-free guidance applied to the text prompt [23]. In contrast to the standard DDIM sampler, we initialize the sampling process from a noise-perturbed latent representation z_t of the input image $\mathcal{I}$ instead of Gaussian noise $\mathcal{N}(0, 1)$ [37]. This improves the quality of the results and makes the output image more consistent with the input scene, which is important in our case as we wish to preserve the background and the identity of the objects in the scene, yet transform their states. This is especially true for out-of-distribution scenes when the sampled noise represents an image far from the input image in the latent space. We demonstrate benefits of this initialization in Section 4.2.

4. Experiments

We introduce our dataset and evaluation protocol in Sec. 4.1. Then, in Sec. 4.2, we show our qualitative results and ablations. Finally, we compare our model with related methods both qualitatively and quantitatively in Sec. 4.3.

4.1. Dataset and experimental setup

Dataset. Our dataset consists of 5-tuples extracted from videos of the ChangeIt and the COIN datasets (Section 3.2) using the self-supervised model [61]. Specifically, from each video, we select one 5-tuple of images and corresponding text descriptions, $(\mathcal{I}, \mathcal{I}^*_{ac}, \mathcal{P}_{ac}, \mathcal{I}^*_{st}, \mathcal{P}_{st})$. We remove images predicted with low confidence. As a form of augmentation, we train the self-supervised model multiple times with different seeds to select multiple different tuples from each video. In total, from 35k videos of the COIN and the ChangeIt datasets, we selected approx. 200k tuples.

We select five interaction categories of the ChangeIt dataset as held-out categories and use all other examples as the training set $\mathcal{D}_{train}$. From those five held-out categories, we manually select and verify 233 tuples as the test set $\mathcal{D}_{test}$. Additionally, we use all other tuples from the held-out categories as the held-out set $\mathcal{D}_{heldout}$, containing 14k tuples, for quantitative evaluation detailed in the next paragraphs. We make sure that tuples from the same video are not both in $\mathcal{D}_{test}$ and $\mathcal{D}_{heldout}$.

Evaluation details. For quantitative evaluation, we report standard FID score [21]. However, we acknowledge its limitations [64], and introduce a novel classification-based evaluation. We report accuracy on the fixed test set of real images $\mathcal{D}_{test}$ from five held-out categories of the ChangeIt dataset. For each category, we consider the initial and final states as separate classes, and thus have 10 classes in total (5 × 2). As the initial $\mathcal{I}$ and the final state $\mathcal{I}^*_{st}$ test images come from the same video, they contain the same background scene but only differ in the object state. We train the classifier using real images for initial states ($\mathcal{I}$ from $\mathcal{D}_{heldout}$) but *generated images for final states* (the output
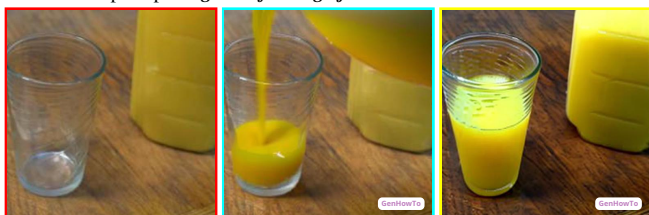
Action prompt: *a person folding an origami paper on a table*
Final state prompt: *a person holding an origami paper airplane*

Action prompt: *a person is painting a painting on a canvas*
Final state prompt: *a piece of blue and purple paint on a canvas*
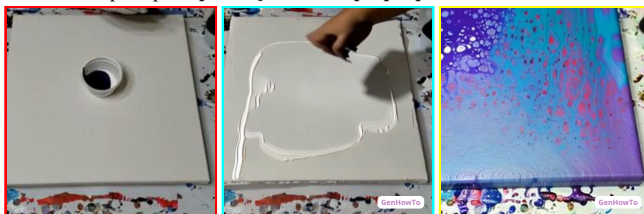
Action prompt: *a person pouring orange juice into a glass*
Final state prompt: *a glass of orange juice next to a bottle*

Action prompt: *a woman slicing an apple on a cutting board*
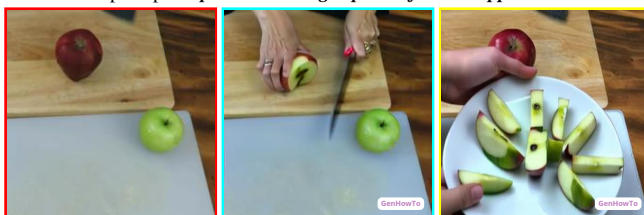Final state prompt: *a person holding a plate of sliced apples*

Figure 4. **Various model predictions for moments from instructional videos unseen during training.** We generate the action (blue) and the final state of the object (yellow) given the initial state image (red) and the corresponding text prompt (**bold**) as the input. Our method correctly models hands interacting with objects (top left) and preserves scene elements such as the cutting board (bottom right). The method can also introduce tools, such as a knife, into the scene to fulfill the prompts, *e.g.*, the slicing action (bottom right).

of a generative method conditioned on the initial state images and prompts $\mathcal{I}, \mathcal{P}_{st}$ from $\mathcal{D}_{heldout}$). Our intuition is that if the generated images (used for training) indeed properly represent the semantics of the final states, then the fixed test set (of real images) will be correctly classified. We also run the same evaluation for actions with the generated action images $\mathcal{I}_{ac}^*$. Additionally, we perform a user study (see the appendix [1]) to further verify the semantics and consistency of the generated images.

4.2. Results

Qualitative results. We show qualitative results of our method for various tasks in Figures 1, 4, and 5. The results show our model can make complex semantic changes to the images by (i) faithfully modifying the objects according to the prompts, (ii) correctly modeling hands interacting with the objects, (iii) introducing new objects such as knives, and (iv) preserving the background and other static parts of the scene from the input image. We show we not only correctly capture object transformations, but also actions, which are more difficult due to the involvement of other objects such as tools and human body parts. While our method is trained on instructional videos only, we also show the model retains generalization ability stemming from the original Stable Diffusion weighs used for initialization. Figures in the appendix [1] show our method generating objects not available in our datasets, such as a toy, a parrot, or northern lights in a landscape photo.

Ablations. We ablate our decision to train two separate models for generating actions and for generating the final states. We show the quantitative results of training one joint

Model training data	Acc$_{ac}$ ↑	Acc$_{st}$ ↑
Only actions	0.77	-
Only final states	-	0.88
Joint model	0.76	0.79

Table 1. **Ablation of joint training.** Test set accuracy of a linear classifier trained on generated images and evaluated on a set of real images, measured for actions Acc$_{ac}$ and final states Acc$_{st}$ separately. Training separate models for actions and final states is better than a single joint model.

or two separate models in Table 1. While we do not observe almost any drop in performance for the actions when using the joint model, the performance of the classifier for object states decreases significantly, coinciding with the drop in visual quality of the generated images. We hypothesize that the additional visual complexity (*e.g.*, correct placement of hands, *etc.*) necessary for correct action generation harms the visual fidelity of the generated object states.

We also provide a qualitative analysis of the DDIM sampling strategy. As discussed in Sec. 3.3, we initialize the sampling process from a noise-perturbed latent representation z_t of the input image $\mathcal{I}$ instead of pure random noise $\mathcal{N}(0, 1)$. In Figure 6, we show that starting purely from noise (the rightmost image) can lead to generated images that do not preserve well the original scene. More examples as well as additional ablations on dataset acquisition are in the appendix [1].
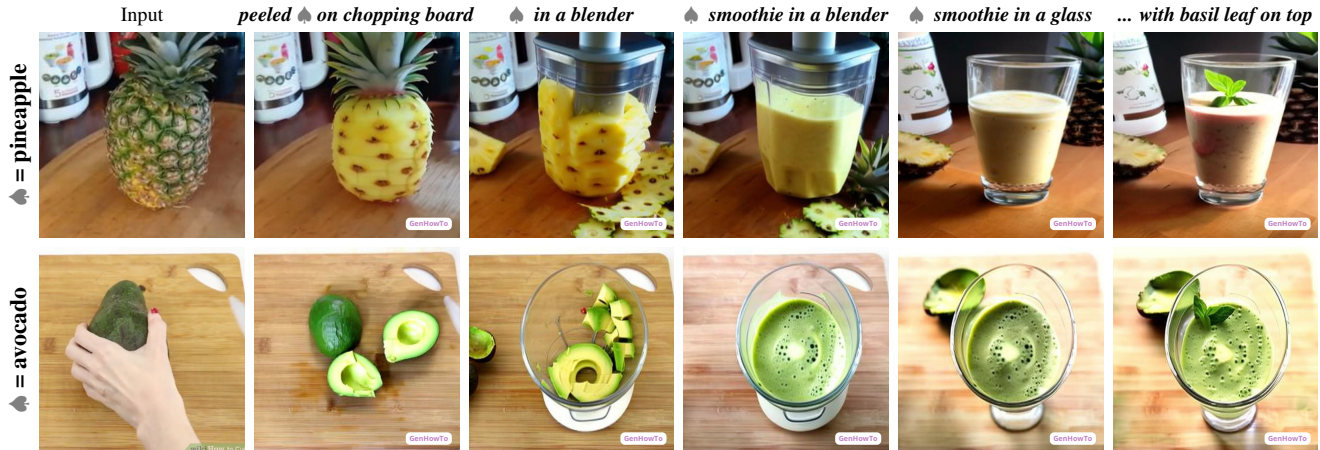
Figure 5. **Long Term Generation.** Each image is generated recurrently using the image to the left and the prompt (top). The leftmost image is a real photo. Even though there are some compounding artifacts, especially when large changes to the scene are required (*e.g.*, a blender → a glass), our model can generate plausible chains of transformations while preserving the scene.
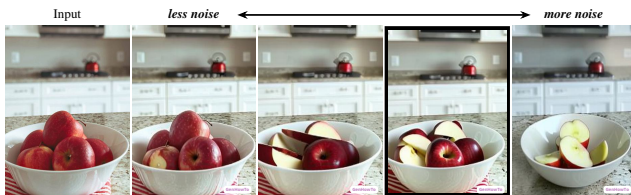

Figure 6. **Varying levels of noise used to initialize image generation process.** The rightmost image shows vanilla DDIM sampling initialized from Gaussian noise. Other images show when the sampling process is initialized from a noise-perturbed latent representation of the input image (left) with varying amounts of noise. Less noise generates images closer to the input scene. An initialization with the right amount of noise (black frame) better preserves the scene while changing the state of the object.

4.3. Comparison with the state-of-the-art

Compared methods. We compare our method to various baselines as well as state-of-the-art methods for editing images based on text prompts: **(a) Stable Diffusion** [54] is a text-conditioned model that only considers the text input without the input image. Our GenHowTo model is derived from stable diffusion hence we consider stable diffusion as an important baseline in our experiments. **(b) Edit Friendly DDPM** [24] uses an off-the-shelf diffusion model (in our case, Stable Diffusion) but alters the sampling process to allow for more natural image editing. **(c) InstructPix2Pix** [4] trains a conditional diffusion model that takes the input image $\mathcal{I}$ and the edit prompt $\mathcal{P}$ as an input to produce the modified image. As the method is trained to follow instructions, we prepend the prompts with the phrase "make it ...". We do not consider methods such as Imagic [26] as they are not suited for large-scale image editing due to their immense time requirements just to edit a single image.

Classification results. Table 2 shows the comparison of different methods using the classification accuracy metric.

	Method	Acc$_{ac}$ ↑	Acc$_{st}$ ↑
	test set categories unseen during training		
(a)	Stable Diffusion [54]	0.51	0.50
(b)	Edit Friendly DDPM [24]	0.60	0.61
(c)	InstructPix2Pix [4]	0.55	0.63
(d)	*CLIP (manual prompts)* [51]	0.52	0.62
(e)	**GenHowTo**	**0.66**	**0.74**
	test set categories seen during training		
(f)	Edit Friendly DDPM† [24]	0.69	0.80
(g)	**GenHowTo**†	**0.77**	**0.88**
(h)	*Real images*	0.96	0.97

† Models trained also on the test set *categories*.

Table 2. **Comparison with state-of-the-art using classification accuracy.** Test set accuracy of a linear classifier trained on generated images from various methods, measured for actions and final states separately.

First, we compare the zero-shot capability of the methods by evaluating on categories unseen during training. As expected, a text-only diffusion model **(a)** performs poorly in the benchmark. It generates images visually distant from the input images as it does not have an image as input. Therefore, the learned classifier cannot distinguish correctly between the classes. The learned classifier does not differentiate between the object state classes (*e.g.*, empty vs. full glass), instead, it differentiates between the real images and generated ones (all *empty glass* training images are real, and all *full glass* images are generated). The same issue largely persists for Edit Friendly DDPM **(b)** or InstructPix2Pix **(c)**. We hypothesize the root cause is the aesthetic-oriented still images the methods have been trained with are very different from the scenes in instructional videos. In contrast, images generated by our method **(e)** better imitate the frames from real videos, yielding much-improved accuracy. As a

Method	FID$_{ac}$ ↓	FID$_{st}$ ↓
Initial state images $\mathcal{I}$ from test set	192	186
Stable Diffusion [54]	208	177
Edit Friendly DDPM [24]	210	169
InstructPix2Pix [4]	223	173
GenHowTo	**181**	**158**

Table 3. **Comparison with state-of-the-art using the FID score.** Images generated using the initial state test set images $\mathcal{I}$ are contrasted with the real target action $\mathcal{I}^*_{ac}$ or final state $\mathcal{I}^*_{st}$ test set images. GenHowTo is the only method consistently generating images more similar to the target test set images than the input images (first row).

baseline, we also measure the accuracy of the CLIP [51] zero-shot classifier on the test set **(d)**—we conclude the CLIP baseline is not sufficient to distinguish and accurately classify small changes in the object appearance.

We also train our method as well as fine-tune Stable Diffusion on our dataset, including the held-out categories from $\mathcal{D}_{heldout}$ **(f–g)**. We observe a significant boost in performance when the models are tasked with generating types of objects and actions seen during training, significantly closing the gap with respect to the real images **(h)**, that serve as an upper bound.

FID results. The results in Table 3 show that all methods generate final state images (column FID$_{st}$) semantically more similar (lower FID score) to the real final state images than the real initial state images (Table 3, first row). However, in the case of actions (column FID$_{ac}$), only our method has a lower FID score than the first row, *i.e.*, it generates action images semantically more similar to the real action images than the real initial state images. This confirms the benefits of our approach that trains the generation model on real object transformations in instructional videos.

Qualitative comparison with other methods. We qualitatively compare our method to other methods in Figure 7. Both InstructPix2Pix and EF-DDPM often struggle to preserve background correctly. This can be seen especially in the first row, where both methods cannot preserve the chopping board from the input image. Also, the generated images of the related methods often look too artistic and unrealistic—we hypothesize it is due to the nature of the original "artistic" datasets those methods have been trained on. In contrast, our method both correctly preserves the background and realistically models the transformation of the object in the image.

Limitations and societal impact. While our method can make complex semantic edits, it is based on trained models inheriting their limitations. Image triplets selected for training (using [61]) can have weak spatiotemporal alignment. As a result, our method can struggle with some scenes and objects that move quickly, such as preserving the identity of
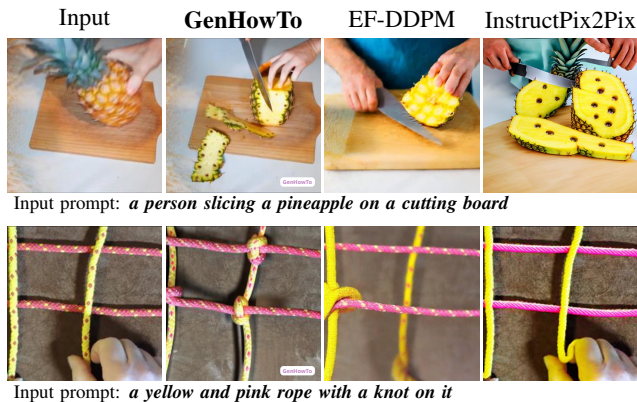
Input	**GenHowTo**	EF-DDPM	InstructPix2Pix

Input prompt: *a person slicing a pineapple on a cutting board*

Input prompt: *a yellow and pink rope with a knot on it*

Figure 7. **Qualitative comparison with related work.** We compare our method with Edit Friendly DDPM [24] and Instruct-Pix2Pix [4] on input images from videos unseen during training (first column) and edit prompts (below). The related methods struggle to preserve the background correctly. See the appendix [1] for additional examples.

people's faces or objects not represented well in the training data (see the appendix [1] for examples). Additionally, our training preserves the qualities of the original Stable Diffusion weights, including its biases. However, this allows for manipulations, such as painting style changes, when paired data is not available during our training.

5. Conclusion

In this work, we have introduced GenHowTo, an image- and text-conditioned generational model, that can generate images of actions and object state transformations while preserving the input image scene. We trained GenHowTo on frames from instructional HowTo videos, automatically gathered by a recent self-supervised method, and prompts from an image captioning model. We show our method is able to generate images that are more realistic with fewer hallucinations than recent competitors. We believe this work can open up new avenues in generating intermediate goals for robotics and fine-grained action generation.

References

[1] Supplementary material (appendix) for the paper. http://arxiv.org/abs/2312.07322. 4, 6, 8

[2] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *CVPR*, 2022. 2

[3] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *CVPR*, 2023. 2

[4] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023. 2, 7, 8

[5] Elliot Chane-Sane, Cordelia Schmid, and Ivan Laptev. Learning video-conditioned policies for unseen manipulation tasks. In *ICRA*, 2023. 1

[6] Annie S Chen, Suraj Nair, and Chelsea Finn. Learning generalizable robotic reward functions from "in-the-wild" human videos. *arXiv preprint arXiv:2103.16817*, 2021. 1

[7] Bor-Chun Chen and Andrew Kae. Toward realistic image compositing with adversarial learning. In *CVPR*, 2019. 2

[8] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *CVPR*, 2021. 2

[9] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, 2018. 2

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

[11] Hazel Doughty, Ivan Laptev, Walterio Mayol-Cuevas, and Dima Damen. Action modifiers: Learning from adverbs in instructional videos. In *CVPR*, 2020. 2

[12] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *NeurIPS*, 2023. 1, 2

[13] Nikita Dvornik, Isma Hadji, Ran Zhang, Konstantinos G Derpanis, Richard P Wildes, and Allan D Jepson. Stepformer: Self-supervised step discovery and localization in instructional videos. In *CVPR*, 2023. 2

[14] Dave Epstein, Jiajun Wu, Cordelia Schmid, and Chen Sun. Learning temporal dynamics from cycles in narrated video. In *CVPR*, 2021. 2

[15] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 2

[16] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *ECCV*. Springer, 2022. 2

[17] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022. 2

[18] Zonghui Guo, Haiyong Zheng, Yufeng Jiang, Zhaorui Gu, and Bing Zheng. Intrinsic image harmonization. In *CVPR*, 2021. 2

[19] Tengda Han, Weidi Xie, and Andrew Zisserman. Temporal alignment networks for long-term video. In *CVPR*, 2022. 4

[20] 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 arXiv:2208.01626*, 2022. 2

[21] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 2017. 5

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

[23] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshops*, 2021. 5

[24] Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly ddpm noise space: Inversion and manipulations. *arXiv preprint arXiv:2304.06140*, 2023. 2, 7, 8

[25] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 2

[26] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *CVPR*, 2023. 2, 7

[27] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *CVPR*, 2022. 2

[28] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *NeurIPS*, 2021. 2

[29] Sumith Kulal, Tim Brooks, Alex Aiken, Jiajun Wu, Jimei Yang, Jingwan Lu, Alexei A Efros, and Krishna Kumar Singh. Putting people in their place: Affordance-aware human insertion into scenes. In *CVPR*, 2023. 2

[30] Yong-Hoon Kwon and Min-Gyu Park. Predicting future frames using retrospective cycle gan. In *CVPR*, 2019. 3

[31] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 4

[32] Xiaodan Liang, Lisa Lee, Wei Dai, and Eric P Xing. Dual motion gan for future-flow embedded video prediction. In *ICCV*, 2017. 3

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

[34] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. *ICLR*, 2017. 3

[35] Weixin Luo, Wen Liu, Dongze Lian, and Shenghua Gao. Future frame prediction network for video anomaly detection. *TPAMI*, 2021. 3

[36] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015. 3

[37] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *ICLR*, 2021. 5

[38] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, October 2019. 4

[39] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2

[40] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. *arXiv preprint arXiv:2211.09794*, 2022. 2

[41] Muhammad Ferjad Naeem, Yongqin Xian, Federico Tombari, and Zeynep Akata. Learning graph embeddings for compositional zero-shot learning. In *CVPR*, 2021. 2

[42] Ashvin Nair, Shikhar Bahl, Alexander Khazatsky, Vitchyr Pong, Glen Berseth, and Sergey Levine. Contextual imagined goals for self-supervised robotic learning. In *CoRL*, 2020. 1

[43] Ashvin V Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. *NeurIPS*, 2018. 1

[44] Soroush Nasiriany, Vitchyr Pong, Steven Lin, and Sergey Levine. Planning with goal-conditioned policies. *NeurIPS*, 2019. 1

[45] 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 arXiv:2112.10741*, 2021. 2

[46] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*, 2017. 2

[47] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019. 2

[48] Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *SIGGRAPH*, 2023. 2

[49] 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 *ICCV*, 2023. 2

[50] Deepak Pathak, Parsa Mahmoudieh, Guanghao Luo, Pulkit Agrawal, Dian Chen, Yide Shentu, Evan Shelhamer, Jitendra Malik, Alexei A. Efros, and Trevor Darrell. Zero-shot visual imitation. In *ICLR*, 2018. 1

[51] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2, 7, 8

[52] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2

[53] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021. 2

[54] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2, 4, 7, 8

[55] 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. In *CVPR*, 2023. 2

[56] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *SIGGRAPH*, 2022. 2

[57] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 2022. 2

[58] Nirat Saini, Hanyu Wang, Archana Swaminathan, Vinoj Jayasundara, Bo He, Kamal Gupta, and Abhinav Shrivastava. Chop & learn: Recognizing and generating object-state compositions. In *ICCV*, 2023. 2

[59] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *NeurIPS*, 2015. 3

[60] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2, 5

[61] Tomáš Souček, Jean-Baptiste Alayrac, Antoine Miech, Ivan Laptev, and Josef Sivic. Multi-task learning of object states and state-modifying actions from web videos. *TPAMI*, 2024. 2, 4, 5, 8

[62] Tomáš Souček, Jean-Baptiste Alayrac, Antoine Miech, Ivan Laptev, and Josef Sivic. Look for the change: Learning object states and state-modifying actions from untrimmed web videos. In *CVPR*, 2022. 2, 4

[63] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015. 3

[64] George Stein et al. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. *NeurIPS*, 2023. 5

[65] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *CVPR*, 2019. 4

[66] Michael W Tao, Micah K Johnson, and Sylvain Paris. Error-tolerant image compositing. *IJCV*, 2013. 2

[67] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *CVPR*, 2023. 2

[68] Jianren Wang, Sudeep Dasari, Mohan Kumar Srirama, Shubham Tulsiani, and Abhinav Gupta. Manipulate by seeing: Creating manipulation controllers from pre-trained representations. In *ICCV*, 2023. 1, 2

[69] Tengfei Wang, Ting Zhang, Bo Zhang, Hao Ouyang, Dong Chen, Qifeng Chen, and Fang Wen. Pretraining is all you need for image-to-image translation. *arXiv preprint arXiv:2205.12952*, 2022. 2

[70] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018. 2

[71] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*, 2018. 2

[72] Cheng Yan, Shiyu Zhang, Yang Liu, Guansong Pang, and Wenjun Wang. Feature prediction diffusion model for video anomaly detection. In *ICCV*, 2023. 3

[73] 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

[74] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 2, 4, 5

[75] Pan Zhang, Bo Zhang, Dong Chen, Lu Yuan, and Fang Wen. Cross-domain correspondence learning for exemplar-based image translation. In *CVPR*, 2020. 2

[76] He Zhao, Isma Hadji, Nikita Dvornik, Konstantinos G Derpanis, Richard P Wildes, and Allan D Jepson. P3iv: Probabilistic procedure planning from instructional videos with weak supervision. In *CVPR*, 2022. 2

[77] Yiwu Zhong, Licheng Yu, Yang Bai, Shangwen Li, Xueting Yan, and Yin Li. Learning procedure-aware video representation from instructional videos and their narrations. In *CVPR*, 2023. 2

[78] Xingran Zhou, Bo Zhang, Ting Zhang, Pan Zhang, Jianmin Bao, Dong Chen, Zhongfei Zhang, and Fang Wen. Cocosnet v2: Full-resolution correspondence learning for image translation. In *CVPR*, 2021. 2

[79] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 2

[80] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. *NeurIPS*, 2017. 2