Generative Models for Image and Long Video Synthesis

Tim Brooks

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2023-100
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2023/EECS-2023-100.html

May 11, 2023

Generative Models for Image and Long Video Synthesis

By

Tim Brooks

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Alexei A. Efros, Chair
Professor Jitendra Malik
Professor Trevor Darrell
Professor Jaakko Lehtinen

Spring 2023

Generative Models for Image and Long Video Synthesis

Abstract

Generative Models for Image and Long Video Synthesis

by

Tim Brooks

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Alexei A. Efros, Chair

In this thesis, I present essential ingredients for making image and video generative models useful for general visual content creation through three contributions. First, I will present research on long video generation. This work proposes a network architecture and training paradigm that enables learning long-term temporal patterns from videos, a key challenge to advancing video generation from short clips to longer-form coherent videos. Next, I will present research on generating images of scenes conditioned on human poses. This work showcases the ability of generative models to represent relationships between humans and their environments, and emphasizes the importance of learning from large and complex datasets of daily human activity. Lastly, I will present a method for teaching generative models to follow image editing instructions by combining the abilities of large language models and text-to-image models to create supervised training data. Following instructions is an important step that will allow generative models of visual data to become more helpful to people. Together these works advance the capabilities of generative models for synthesizing images and long videos.

Contents

List of Figures

List of Tables

Acknowledgments

Thank you to many wonderful people who have been there for me throughout my PhD journey: my parents, mama and papa, and my siblings, Emily, Sarah and Simon, for their endless support; Sandy Campbell for always being by my side; my close friends and collaborators, Bill Peebles, Ilija Radoskavic and Tete Xiao, for teaching me what matters in a PhD; my advisor, Alyosha Efros, for guidance and for freedom to pursue my own research directions; the Berkeley vision faculty, Jitendra Malik, Trevor Darrell, and Angjoo Kanazawa, for sharing their research wisdom; Jaakko Lehtinen for discussing research ideas and welcoming me to his lab at Aalto University; my mentors at NVIDIA, Tero Karras, Timo Aila, and Ming-Yu Liu, for making me a stronger researcher; and the many folks at Berkeley and throughout my life who have encouraged and helped me along the way.

Chapter 1

Introduction

During my PhD studies, image and video generative models have evolved from niche demos into widely adopted creative tools. I am privileged to have pursued my doctorate in visual generative models during this pivotal time, and I am optimistic about the transformative potential and utility of future visual generative models. In this thesis, I present three works aimed at enhancing the capabilities of generative models for visual content creation. These works outline key elements necessary for making future image and video generative models more helpful to people at performing complex visual creation tasks.

In Chapter 2, I discuss the development of video generation models that can represent long-term patterns over time. Increasing the duration of generated videos is a vital aspect of improving visual generative models, which have previously concentrated on only brief video clips. Long video generation is essential for applications like AI-aided production of feature-length films. Additionally, learning from long videos contributes to a deeper understanding of the visual world, which is invaluable for general-purpose visual generative models. Increasing sequence length in other modalities, such as for language and speech modeling, has shown vast improvements in the emergent abilities of these models. Similarly, future visual generative models will likely handle extremely long videos, ultimately unlocking transformative visual understanding and generative capabilities.

The video generation approach I propose takes a step in this direction by expanding the time horizon modeled in videos compared to prior research. Long videos present particular challenges, such as modeling new objects and scenery that enter the video over time and maintaining physical consistencies expected of real environments. My work tackles these hard problems by introducing a new video generative adversarial network (GAN) capable of representing long-term patterns in an efficient temporal latent space, and capable of being efficiently trained on long videos

by decomposing the modeling problem into two complementary generative models that operate at different temporal and spatial scales.

In Chapter 3, I present research on learning from complex real-world data reflecting everyday human activity. The interactions between people, objects, and their surroundings offer a rich source of information about the world. I propose a method that learns these relationships via a conditional generative model. Earlier generative models primarily concentrated on specific content categories, such as faces or particular object classes. This work expands generative models into the domain of modeling complex scenes with humans. Provided an input skeletal pose of a person, the model is capable of generating a plausible scene that is compatible with that pose. The model can generate both empty scenes and scenes with a person in the input pose. Visual results demonstrate that the model emerges to learn a nuanced understanding of scene affordances and semantic relationships between environments and human actions. This research highlights the ability of generative models to understand complex relationships about the visual world by training on large visual datasets of everyday human activity.

In Chapter 4, I proposes a technique for making visual generative models more useful to people by teaching them to follow image editing instructions. It is crucial to consider the interface for how people will use generative models to create visual content, and I argue that an ideal interface, short of mind-reading, is to converse with an AI system as if talking with a creative human expert. We should be able to tell AI models exactly what we want them to do and receive a helpful output that adheres to our request. Building on this concept, the last work I will present teaches generative models to follow image editing instructions.

Instruction-based image editing is a particularly challenging task because, unlike other image prediction tasks, there does not exist a sizeable training dataset of examples. While there is a plethora of images including many images with corresponding text, there are no large datasets with editing instructions and corresponding before and after images, and collecting such data would be extremely expensive and challenging to scale. A key insight of the work I present is to combine capabilities of large language models and text-to-image models to generate the necessary training data. As generative models become increasingly powerful at producing realistic samples, they will also become increasingly useful at creating training data for other models or for specialized tasks. By combining the knowledge of two large generative models trained on different modalities—a large language model and a text-to-image model—it is possible to create training data for instruction-based image editing, which is a task that neither model can achieve on its own. While the training data is entirely generated, the resulting model generalizes to real inputs and produces compelling image edits for a wide variety of images and instructions. Teaching vi-

sual generative models to follow instructions is a key step toward making AI-based content creation more useful. In the future, it will be essential to extend these abilities beyond a single instruction and to enable full conversation between users and visual generative models.

Collectively, these works identify three critical components for future visual generative models: modeling long-term patterns over time, learning from complex visual data, and following visual generation instructions. All three elements will be essential in developing artificial superintelligence that performs complex visual creation tasks, aids human creativity, and brings our visual imaginations to life.

Chapter 2

Generating Long Videos

2.1 Introduction

Videos are data that change over time, with complex patterns of camera viewpoint, motion, deformation and occlusion. In certain respects, videos are unbounded — they may last arbitrarily long and there is no limit to the amount of new content that may become visible over time. Yet videos that depict the real world must also remain consistent with physical laws that dictate which changes over time are feasible. For example, the camera may only move through 3D space along a smooth path, objects cannot morph between each other, and time cannot go backward. Generating long videos thus requires the ability to produce endless new content while maintaining appropriate consistencies.

In this work, we focus on generating long videos with rich dynamics and new content that arises over time. While existing video generation models can produce "infinite" videos, the type and amount of change along the time axis is highly limited. For example, a synthesized infinite video of a person talking will only include small motions of the mouth and head. Moreover, common video generation datasets often contain short clips with little new content over time, which may inadvertently bias design choices toward training on short segments or pairs of frames, forcing content in videos to stay fixed, or using architectures with small temporal receptive fields.

We make the time axis a first-class citizen for video generation. To this end, we introduce two new datasets that contain motion, changing camera viewpoints, and entrances/exits of objects and scenery over time. We learn long-term consistencies

The work presented in this chapter was first published in Brooks *et al.* as *Generating Long Videos of Dynamic Scenes* at the Conference on Neural Information Processing Systems (NeurIPS), 2022 [1].

Figure 2.1: We aim to generate videos that accurately portray motion, changing camera viewpoint, and new content that arises over time. **Top:** Our horseback riding dataset exhibits these types of changes as the horse moves forward in the environment. **Middle:** StyleGAN-V, a state-of-the-art video generation baseline, is incapable of generating new content over time; the horse fails to move forward past the obstacle, the scene does not change, and the video morphs back and forth within a short window of motion. **Bottom:** Our novel video generation model prioritizes the time axis and generates realistic motion and scenery changes over long durations. The same videos can be viewed on the supplemental webpage.

by training on long videos and design a temporal latent representation that enables modeling complex temporal changes. Figure 2.1 illustrates the rich motion and scenery changes that our model is capable of generating. See our webpage[1] for video results, code, data and pretrained models.

Our main contribution is a hierarchical generator architecture that employs a vast temporal receptive field and a novel temporal embedding. We employ a multi-resolution strategy, where we first generate videos at low resolution and then refine them using a separate super-resolution network. Naively training on long videos at high spatial resolution is prohibitively expensive, but we find that the main aspects of a video persist at a low spatial resolution. This observation allows us to train with long videos at low resolution and short videos at high resolution, enabling us to prioritize the time axis and ensure that long-term changes are accurately portrayed. The low-resolution and super-resolution networks are trained independently with an RGB bottleneck in between. This modular design allows iterating on each network independently and leveraging the same super-resolution network for different low-

[1] https://www.timothybrooks.com/tech/long-videos

resolution network ablations.

We compare our results to several recent video generative models and demonstrate state-of-the-art performance in producing long videos with realistic motion and changes in content. Code, new datasets, and pre-trained models on these datasets will be made available.

2.2 Prior Work

Video generation is a challenging problem with a long history. The classic early works, Video Textures [2] and Dynamic Textures [3], model videos as textures by analogy with image textures. That is, they explicitly assume the content to be stationary over time, e.g., fire burning, smoke rising, foliage falling, pendulum swinging, etc., and use non-parametric [2] or parametric [3] approaches to model that stationary distribution. Although subsequent video synthesis works have dropped the "texture" moniker, much of the limitations remain similar — short training videos and models which produce little or no new objects entering the frame during the video. Below we summarize some of the more recent efforts on video generation.

Unconditional video generation. Many video generation works are based on GANs [4], including early models that output fixed-length videos [5–7] and approaches that use recurrent networks to produce a sequence of latent codes used to generate frames [8–11]. MoCoGAN [11] explicitly disentangles "motion" from "content" and keeps the latter fixed over the entire generated video. StyleGAN-V [12] is a recent state-of-the-art model we use as a primary baseline. Similar to MoCoGAN, StyleGAN-V employs a global latent code that controls content of an entire video. MoCoGAN-HD [10], which we also compare with, and StyleVideoGAN [9] attempt to generate videos by navigating the latent space of a pretrained StyleGAN2 model [13], but struggle to produce realistic motion. Unlike previous StyleGAN-based [14] video models, we prioritize the time axis in our generator through a new temporal latent representation and temporal convolutions. We also compare with DIGAN [15] that employs an implicit function to generate each video pixel.

Transformers are another class of models used for video generation [16–19]. We compare with TATS [16] that generates long unconditional videos with transformers, improving upon VideoGPT [19]. Both TATS and VideoGPT employ a GPT-like autoregressive transformer [20] that represents videos as sequences of tokens. However, the resulting videos tend to accumulate error over time and often diverge or change too rapidly. The models are also expensive to train and deploy due to their autoregressive nature over time and space. In concurrent work, promising results in

generating diverse videos have been demonstrated using diffusion-based models [21].

Conditional video prediction. A separate line of research focuses on predicting future video frames conditioned on one or more real video frames [22–27] or past frames accompanied by an action label [28–31]. Some video prediction methods focus specifically on generating infinite scenery by conditioning on camera trajectory [32, 33] and/or explicitly predicting depth [32,34] to then simulate a virtual camera flying through a 3D scene. Our goal, on the other hand, is to support camera movement as well as moving objects by having the scene structure emerge implicitly.

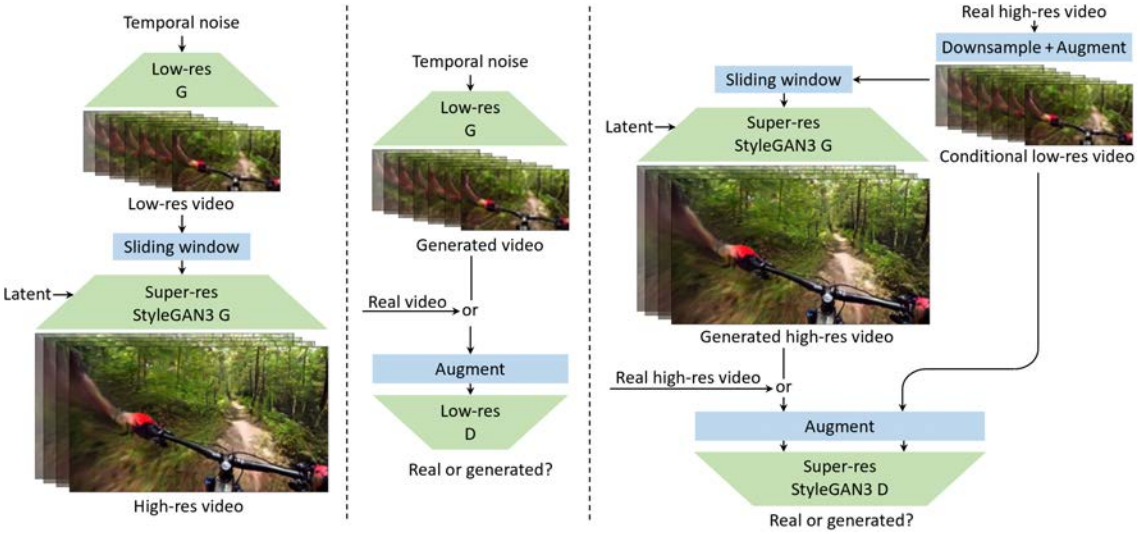
Multi-resolution training. Training at multiple scales is a common strategy for image generation models [35–39]. Transformer-based video generators also employ a related two-phase setup [16, 19]. Saito *et al.* [40] subsample frames at higher resolutions in their video generator architecture to improve efficiency. A similar idea is also used in SlowFast [41] networks where different network pathways are used for high and low frame rate video streams. Acharya *et al.* [5] propose a multi-scale GAN for video generation that increases both spatial resolution and sequence length during training to produce a fixed-length video. In contrast, our multi-resolution approach is designed to enable generating arbitrarily long videos with rich long-term dynamics by leveraging training of long sequences at low resolution.

2.3 Our Method

Modeling the long-term temporal behavior observed in real videos presents us with two main challenges. First, we must use long enough sequences during training to capture the relevant effects; using, e.g., pairs of consecutive frames fails to provide meaningful training signal for effects that occur over several seconds. Second, we must ensure that the networks themselves are capable of operating over long time scales; if, e.g., the receptive field of the generator spans only 8 adjacent frames, any two frames taken more than 8 frames apart will necessarily be uncorrelated with each other.

Figure 2.2a shows the overall design of our generator. We seed the generation process with a variable-length stream of temporal noise, consisting of 8 scalar components per frame drawn from i.i.d. Gaussian distribution. The temporal noise is first processed by a *low-resolution generator* to obtain a sequence of RGB frames at 64^2 resolution that are then refined by a separate *super-resolution network* to produce the final frames at 256^2 resolution.[2] The role of the low-resolution generator

[2]We handle datasets with non-square aspect ratio by shrinking all intermediate data accordingly.



(a) Generator overview (b) Low-res training (c) Super-res training

Figure 2.2: Overview of our method. **(a)** To achieve long temporal receptive field and high spatial resolution, we split our generator into two components: a low-resolution generator, responsible for modeling major aspects of the motion and scene composition, and a super-resolution network, responsible for hallucinating fine details. **(b)** The low-resolution generator (Section 2.3.1) employs a wide temporal receptive field and is trained with sequences of 128 frames at 64^2 resolution. **(c)** The super-resolution network (Section 2.3.2) is conditioned on short sequences of low-resolution frames and trained to produce their plausible counterparts at 256^2 resolution.

is to model major aspects of the motion and scene composition, which necessitates strong expressive power and a large receptive field over time, whereas the super-resolution network is responsible for the more fine-grained task of hallucinating the remaining details.

Our two-stage design provides maximum flexibility in terms of generating long videos. Specifically, the low-resolution generator is designed to be fully convolutional over time, so the duration and time offset of the generated video can be controlled by shifting and reshaping the temporal noise, respectively. The super-resolution network, on the other hand, operates on a frame-by-frame basis. It receives a short sequence of 9 consecutive low-resolution frames and outputs a single high-resolution frame; each output frame is processed independently using a sliding window. The combination of fully-convolutional and per-frame processing enables us to generate arbitrary frames in arbitrary order, which is highly desirable for, e.g., interactive

With 256×144 target resolution, for example, the low-resolution frames will have 64×36 resolution.

editing and real-time playback.

The low-resolution and super-resolution networks are modular with an RGB bottleneck in between. This greatly simplifies experimentation, since the networks are trained independently and can be used in different combinations during inference. We will first describe the training and architecture of the low-resolution generator in Section 2.3.1 and then discuss the super-resolution network in Section 2.3.2.

2.3.1 Low-resolution Generator

Figure 2.2b shows our training setup for the low-resolution generator. In each iteration, we provide the generator with a fresh set of temporal noise to produce sequences of 128 frames (4.3 seconds at 30 fps). To train the discriminator, we sample corresponding sequences from the training data by choosing a random video and a random interval of 128 frames within that video.

We have observed that training with long sequences tends to exacerbate the issue of overfitting [42]. As the sequence length increases, we suspect that it becomes harder for the generator to simultaneously model temporal dynamics at multiple time scales, but at the same time, easier for the discriminator to spot any mistakes. In practice, we have found strong discriminator augmentation [42, 43] to be necessary in order to stabilize the training. We employ DiffAug [43] using the same transformation for each frame in a sequence, as well as fractional time stretching between $\frac{1}{2}\times$ and $2\times$; see Section 2.4.1 for details.

Architecture. Figure 2.3 illustrates the architecture of our low-resolution generator. Our main goal is to make the time axis a first-class citizen, including careful design of a temporal latent representation, temporal style modulation, spatiotemporal convolutions, and temporal upsamples. Through these mechanisms, our generator spans a vast temporal receptive field (5k frames), allowing it to represent temporal correlations at multiple time scales.

We employ a style-based design, similar to Karras *et al.* [13, 44], that maps the input temporal noise into a sequence of *intermediate latents* $\{w_t\}$ used to modulate the behavior of each layer in the main synthesis path. Each intermediate latent is associated with a specific frame, but it can significantly influence the scene composition and temporal behavior of several frames through hierarchical 3D convolutions that appear in the main path.

In order to reap the full benefits of the style-based design, it is crucial for the intermediate latents to capture long-term temporal correlations, such as weather changes or persistent objects. To this end, we adopt a scheme where we first enrich the input temporal noise using a series of temporal lowpass filters and then pass it

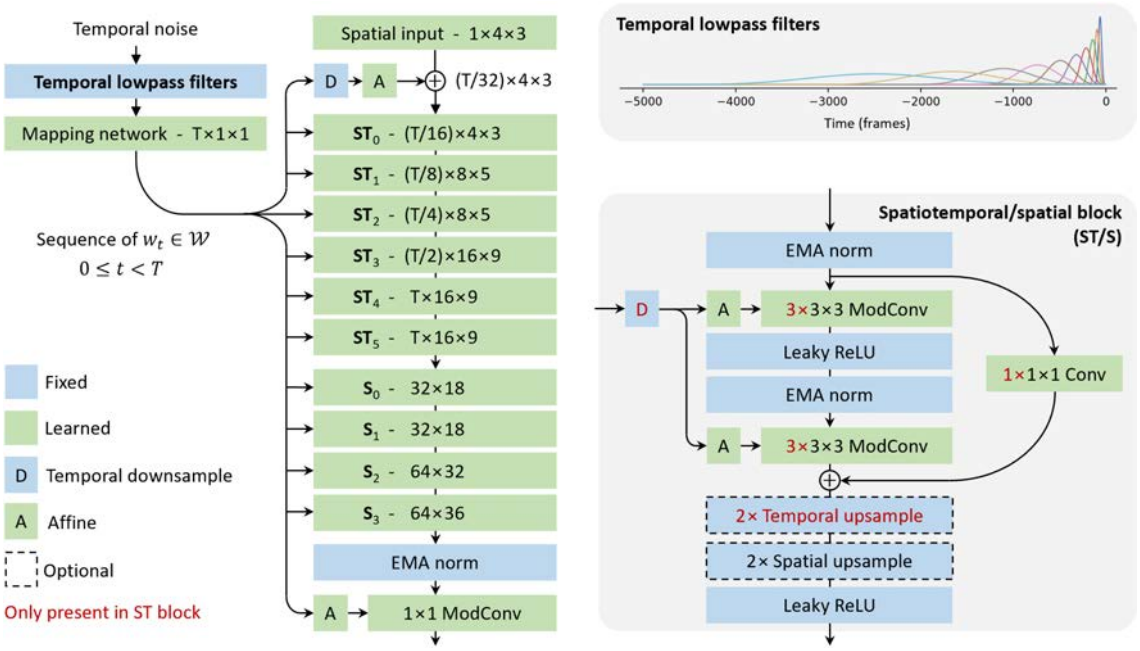


Figure 2.3: Low-resolution generator architecture, illustrated for 64×36 output. **Left:** The input temporal noise is mapped to a sequence of *intermediate latents* $\{w_t\}$ that modulate the intermediate activations of the main synthesis path. **Top right:** To facilitate the modeling of long-term dependencies, we enrich the temporal noise by passing it through a series of lowpass filters whose temporal footprints range all the way from 100 to 5000 frames. **Bottom right:** The main synthesis path consists of *spatiotemporal* (ST) and *spatial* (S) blocks that gradually increase the resolution over time and space.

through a fully-connected *mapping network* on a frame-by-frame basis. The goal of the lowpass filtering is to provide the mapping network with sufficient long-term context across a wide range of different time scales. Specifically, given a stream of temporal noise $z(t) \in \mathbb{R}^8$, we compute the corresponding enriched representation $z'(t) \in \mathbb{R}^{128\times 8}$ as $z'_{i,j} = f_i * z_j$, where $\{f_i\}$ is a set of 128 lowpass filters whose temporal footprint ranges from 100 to 5000 frames, and $*$ denotes convolution over time; see Section 2.4.2 for details.

The main synthesis path starts by downsampling the temporal resolution of $\{w_t\}$ by 32× and concatenating it with a learned constant at 4^2 resolution. It then gradually increases the temporal and spatial resolutions through a series of processing blocks, illustrated in Figure 2.3 (bottom right), focusing first on the time dimension (ST) and then the spatial dimensions (S). The first four blocks have 512 channels, followed by two blocks with 256, two with 128 and two with 64 channels. The processing blocks consist of the same basic building blocks as StyleGAN2 [13] and

StyleGAN3 [44] with the addition of a skip connection; the intermediate activations are normalized before each convolution [44] and modulated [13] according to an appropriately downsampled copy of $\{w_t\}$. In practice, we employ bilinear upsampling [14] and use padding [44] for the time axis to eliminate boundary effects. Through the combination of our temporal latent representation and spatiotemporal processing blocks, our architecture is able to model complex and long-term patterns across time.

For the discriminator, we employ an architecture that prioritizes the time axis via wide temporal receptive field, 3D spatiotemporal and 1D temporal convolutions, and spatial and temporal downsamples; see Section 2.4.3 for details.

2.3.2 Super-resolution Network

Figure 2.2c shows our training setup for the super-resolution network. Our video super-resolution network is a straightforward extension of StyleGAN3 [44] for conditional frame generation. Unlike the low-resolution network that outputs a sequence of frames and includes explicit temporal operations, the super-resolution generator outputs a single frame and only utilizes temporal information at the input, where the real low-resolution frame and 4 neighboring real low-resolution frames before and after in time are concatenated along the channel dimension to provide context. We remove the spatial Fourier feature inputs and resize and concatenate the stack of low-resolution frames to each layer throughout the generator. The generator architecture is otherwise unchanged from StyleGAN3, including the use of an intermediate latent code that is sampled per video. Low-resolution frames undergo augmentation prior to conditioning as part of the data pipeline, which helps ensure generalization to *generated* low-resolution images.

The super-res discriminator is a similar straightforward extension of the StyleGAN discriminator, with 4 low and high-resolution frames concatenated at the input. The only other change is the removal of the minibatch standard deviation layer that we found unnecessary in practice. Both low- and high-resolution segments of 4 frames undergo adaptive augmentation [42] where the same augmentation is applied to all frames at both resolutions. Low-resolution segments also undergo aggressive dropout ($p = 0.9$ probability of zeroing out the entire segment), which prevents the discriminator from relying too heavily on the conditioning signal; see Section 2.5.1 for details.

We find it remarkable that such a simple video super-resolution model appears sufficient for producing reasonably good high-resolution videos. We focus primarily on the low-resolution generator in our experiments, utilizing a single super-resolution network trained per dataset. We feel that replacing this simple network with a more

advanced model from the video super-resolution literature [45–48] is a promising avenue for future work.

2.4 Low-resolution Implementation Details

2.4.1 Augmentation

We find that overfitting of the discriminator network is particularly severe when training with long sequences. To alleviate the overfitting, we apply DiffAug [43] to real and generated videos prior to the discriminator. We use all categories of DiffAug augmentations — color, cutout, and translation — with default strengths for color and cutout augmentations, and maximum x- and y-translations of 32 pixels for the square SkyTimelapse dataset and 16 pixels for the non-square biking, horseback and ACID datasets. We also tried using the ADA [42] adaptive augmentation strategy, but it caused leakage of augmentations into the generated videos, even when augmentations were applied with low probability.

In addition to DiffAug, we employ fractional time stretching augmentation, where we resize the temporal axis by a factor of $s = 2^a$ for $a \sim \mathcal{U}(-1, 1)$ with linear interpolation and zero padding. If time stretching augmentation upsamples the time axis, the video is randomly cropped to fit within the original 128-frame window. Similarly, if time stretching augmentation downsamples the time axis, the video is zero padded with random amounts before and after to fit within the original 128-frame window. Fractional time stretching augmentation is related to subsampling augmentation that is commonly used by other methods [12], but supports a greater variety of augmentations since temporal scaling amounts are fractional. Further investigation into the best augmentation policies for video generation models is an important future area for investigation.

2.4.2 Temporal Lowpass Filters

To capture long-term temporal correlations in the intermediate latent codes, we enrich each of 8 channels of input temporal noise with a set of $N = 128$ lowpass filters $\{f_i\}$. Specifically, we use Kaiser lowpass filters [49], following the implementation of [44]. We space lowpass filter sizes exponentially, where each filter has temporal footprint:
$k_i = k_{\min}\left(\frac{k_{\max}}{k_{\min}}\right)^{\frac{i}{N-1}}$ where $0 \leq i < N$, $k_{\min} = 500$ and $k_{\max} = 10000$.

2.4.3 Discriminator Architecture

Our low-resolution discriminator architecture is heavily inspired by the StyleGAN [14] discriminators, with the addition of spatiotemporal and temporal processing in order to model realistic motions and changes over time (see Figure 2.4).

The video is first expanded from 3 RGB channels to 128 channels using a 1×1 convolutional layer. The first block only operates spatially, downsampling height and width by 2× and using 3×3 spatial convolutions. The remaining 3 blocks downsample both spatially and temporally and use 5×3×3 spatiotemporal convolutions. We omit temporal processing from the first block to save compute, since running 3D convolutions at the full resolution is substantially more expensive. We otherwise find the inclusion of temporal processing crucial for the model to learn temporal dynamics. In each block, the number of channels is doubled until reaching 512.

To further prioritize learning accurate motions and changes over time, we include 4× 1D temporal convolutions, each with a kernel size of 5 and followed by a LeakyReLU nonlinearity. Finally, following the StyleGAN discriminator, features are flattened and passed through 2 linear layers with a LeakyReLU nonlinearity in between to produce the final logits.

2.4.4 Training

We use a batch size of 64 videos, each of length 128 frames. We trained models with a variety of single- and multi-node jobs. We train each run for a maximum of 100,000 steps and cut training runs short if FVD begins increasing. Training the low-res generator takes 1.7 days for the maximum 100,000 steps using 4× nodes each containing 8× NVIDIA A100 GPUs. The low-res generator has 83.2M parameters and the low-res discriminator has 46.4M parameters. We use R1 regularization [50] with $\gamma = 1$ for non-square datasets, and $\gamma = 4$ for the square SkyTimelapse dataset. We train with the Adam optimizer [51] with generator learning rate of 0.003, discriminator learning rate of 0.002, and $\beta_1 = 0$ and $\beta_2 = 0.99$ for both generator and discriminator. (Note: Adam with $\beta_1 = 0$ is equivalent to RMSprop [52] with the bias correction term from Adam.) We use an exponential moving average of the generator weights, with $\beta_{\text{ema}} = 0.99985$. We select the checkpoint with best FVD_{128}.

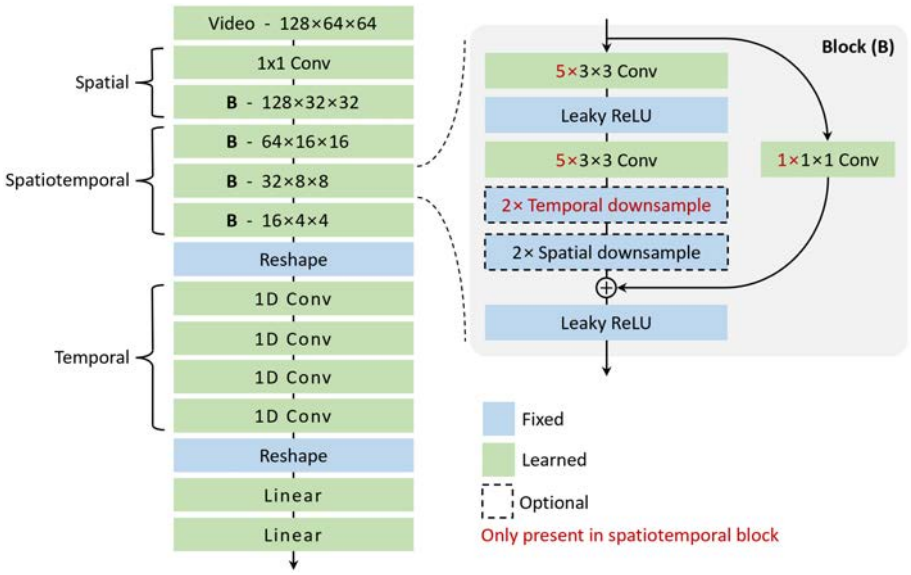


Figure 2.4: Low-resolution discriminator architecture. **Left:** The input video undergoes a single 1×1 convolutional layer, followed by 4 residual blocks. Features are then reshaped, combining spatial and channel dimensions, followed by 4 temporal 1D convolutional layers. Finally, features are flattened, followed by 2 linear layers to produce output logits. **Right:** The residual block follows the structure of discriminator blocks in StyleGAN [14] models, with optional temporal downsampling and 3D spatiotemporal convolutions used for all but the first block.

2.5 Super-resolution Implementation Details

2.5.1 Augmentation

The super-resolution network undergoes augmentation of two forms: (1) augmentation of real and generated videos applied prior to the discriminator to prevent overfitting, and (2) augmentation of conditional real low resolution videos during training to improve generalization to *generated* low-resolution videos.

Discriminator augmentation to prevent overfitting Augmentation to prevent discriminator overfitting uses ADA [42] with default settings, and applies the same augmentations to all frames from both high and low resolution videos. To additionally prevent overfitting and prevent the discriminator from focusing too much attention on the conditioning signal, we employ strong dropout augmentation with probability $p = 0.9$ of zeroing out the entire conditional low resolution video. This augmentation occurs before the discriminator only, and does not affect the inputs

to the super-resolution network.

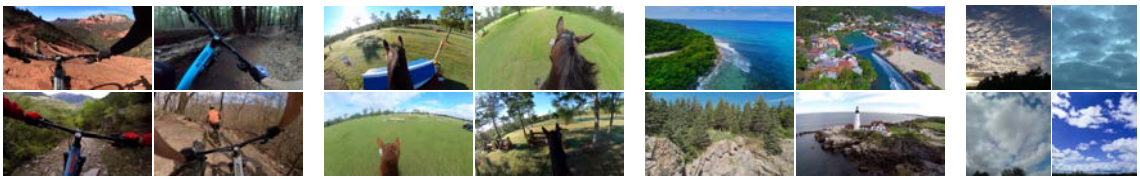
Low-resolution conditioning augmentation to improve generalization We train our super-resolution network with real low resolution videos as conditioning, but use generated low resolution videos at inference time. There exists a domain gap between the real and generated low resolution videos, and to ensure our super-resolution network is robust to the domain gap, we augment real low resolution videos during training. Similar strategies are used in image generators with super-resolution refinement [53], where corruption is added to real low resolution inputs during training. We modify the ADA [42] augmentation pipeline, only enabling additive Gaussian noise, isotropic and non-isotropic scaling, rotation, and fractional translation. Each augmentation is applied to the entire low resolution video with a fixed probability of 50%, and with relatively small strengths (`noise_std=0.08`, `scale_std=0.08`, `aniso_std=0.08`, `rotate_max=0.016`, `xfrac_std=0.016`). This is applied in the dataset pipeline and affects conditional inputs to the discriminator and super-resolution network only during training.

2.5.2 Prefiltering of Low-res Conditioning

The low resolution frame being upsampled is concatenated with 4 frames before and 4 frames after in the low resolution video sequence creating a stack of 9 low resolution frames. The stack is then resized and concatenated with features at each layer of the StyleGAN3 generator. We experimented with different prefiltering strengths when resizing the 9 conditioning frames, and found that strong prefiltering helps remove aliasing in the final video. This is related to the anti-aliasing properties of the StyleGAN3 generator that includes strong filtering of intermediate features [44]. Importantly, we do not prefilter the conditional frames when the input is the same resolution as the features (i.e., 64×64) since we found that negatively impacts the results. We only apply prefiltering when resizing, and we use the same prefiltering kernels as early layers of StyleGAN3.

2.5.3 Training

We use a batch size of 32 videos. The discriminator network inputs real and generated videos with a length of 4 frames, and for each generated frame the super-res network is provided 9 input frames (4 neighboring frames on either side of the primary frame) to provide temporal context. The network architectures share details with StyleGAN3 [44], except the differences mentioned in Section 2.3.2. We train for a maximum of 275,000 steps, which takes 6.8 days using one node of 8× 16GB NVIDIA V100 GPUs. The super-res network has 27.2M parameters, and the discriminator network has 24.0M parameters. We use R1 regularization with $\gamma = 1$ for all datasets. We train with the Adam optimizer with generator and discriminator learning rate of 0.003, $\beta_1 = 0$ and $\beta_2 = 0.99$. We use an



(a) Mountain biking (b) Horseback riding (c) ACID (d) Sky

Figure 2.5: Example real frames from training datasets. We introduce first-person datasets of **(a)** mountain biking and **(b)** horseback riding videos that contain complex motion and new content over time. We also evaluate on existing datasets of **(c)** nature drone footage and **(d)** sky timelapse videos.

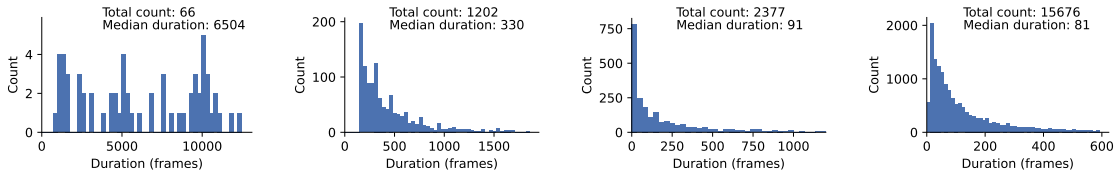
exponential moving average of the generator weights with $\beta_{\text{ema}} = 0.99985$. We select the checkpoint with best FVD_{16} when evaluated using real low resolution conditioning, and use the same super-resolution network for many low-resolution experiments.

2.6 Datasets

Most of the existing video datasets introduce little or no new content over time. For example, talking head datasets [54–57] show the same person for the duration of each video. UCF101 [58] portrays diverse human actions, but the videos are short and contain limited camera motion and little or no new objects that enter the videos over time.

To best evaluate our model, we introduce two new video datasets of first-person mountain biking and horseback riding (Figure 2.5a,b) that exhibit complex changes over time. Our new datasets include subject motion of the horse or biker, a first-person camera viewpoint that moves through space, and new scenery and objects over time. The videos are available in high definition and were manually trimmed to remove problematic segments, scene cuts, text overlays, obstructed views, etc. The mountain biking dataset has 1202 videos with a median duration of 330 frames at 30 fps, and the horseback dataset has 66 videos with a median duration of 6504 frames also at 30fps. We have permission from the content owners to publicly release the datasets for research purposes. We believe our new datasets will serve as important benchmarks for future work.

We also evaluate our model on the ACID dataset [59] (Figure 2.5c) that contains significant camera motion but lacks other types of motion, as well as the commonly used SkyTimelapse dataset [60] (Figure 2.5d) that exhibits new content over time as the clouds pass by, but the videos are relatively homogeneous and the camera remains fixed.



(a) Horseback (ours) (b) Biking (ours) (c) SkyTimelapse [61] (d) ACID [32]

Figure 2.6: Counts and durations of training videos. Training a model to prioritize the time axis requires training on long videos. Existing video datasets, such as (c) and (d), include relatively short videos with median durations of 91 and 81 frames respectively. We introduce two new datasets of longer videos, (a) and (b), with median durations of 6504 and 330 frames. We show results on all four of these datasets.

2.7 Dataset Preparation Details

We evaluate our model using two existing datasets, Aerial Coastline Imagery Dataset (ACID) [32] and SkyTimelapse [61], and two new datasets: horseback riding and mountain biking. We center crop videos to the desired aspect ratio if needed (16×9 for all datasets except SkyTimelapse, for which we use a square crop to match prior work), and then resize to the target resolution using the PIL library's Lanczos resampling method. For the ACID dataset we combine both train and test splits to maximize the amount of training data. For the SkyTimelapse dataset we use only the train split to ensure our model is comparable with prior work.

Figure 2.6 shows histograms of the durations and counts of training videos for all four datasets. Our new datasets both feature longer median clip lengths than the existing datasets. When training our model, we filter ACID and SkyTimelapse datasets for clips with at least 128 frames. We allow the StyleGAN-V baseline to train on all clips with at least 3 frames (the number needed by their method). Both datasets can be obtained from their respective project webpages. ACID: https://infinite-nature.github.io/, and SkyTimelapse: https://sites.google.com/site/whluoimperial/mdgan. The copyright status of both existing datasets is ambiguous, as neither specify a license or details about content ownership. We ensure to attain explicit licenses for our two new datasets below.

2.7.1 Horseback Riding

We introduce a new dataset of first-person horseback riding that we will release to the public for research purposes. The videos were created by Wallace Eventing and examples of the videos can be found on their YouTube channel: https://www.youtube.com/c/WallaceEventing. We reached out directly and received permission to create a dataset from their videos to use in our research and release as a dataset for non-commercial research purposes. We will release the filtered and processed video frames directly, which

	Horseback riding		Mountain biking	
	# Videos	Total duration	# Videos	Total duration
Videos considered	194	27h:29m:42s	48	38h:46m:56s
Videos selected	44	7h:21m:49s	28	9h:06m:50s
Clips extracted	66	4h:01m:41s	1202	5h:07m:55s

Table 2.1: We manually curate horseback riding and mountain biking datasets in two phases: first by selecting source videos containing sufficient first-person footage with stable motion and a consistent camera perspective, and then by extracting clips free from scene changes, text overlays, or other unwanted content. Here we report the number of videos and total duration of video content at each phase of curation.

avoids inconsistent versions of the dataset when videos become unavailable or are processed differently. The dataset will be released under a custom license agreed upon with Wallace Eventing that permits use for non-commercial research purposes but does not allow redistribution of the dataset.

The videos contain first-person helmet camera footage of horseback riding events, with little or no personally identifying information visible. They are high quality (1080p) at 60fps, although we subsample frames to attain 30fps. Statistics of our dataset filtering are presented in Table 2.1. The dataset was sourced from 194 original videos, which we then filtered down to 44 videos with stabilized motion and a consistent camera perspective. We manually extracted 66 clips from the selected videos, cutting out scene changes, text overlays, videos with obstructed views, and the beginnings and ends of videos.

2.7.2 Mountain Biking

We also introduce a new dataset of first-person mountain biking that we will release to the public. The videos were created by Brian Kennedy (BKXC) and examples of the videos can be found on their YouTube channel: https://www.youtube.com/c/bkxc. We reached out directly and received permission to create a dataset from their videos to use in our research and release as a dataset under a CC BY 4.0 license.

The videos contain first-person mountain biking. There is little personally identifying information visible, although there are occasional other bikers who pass by and whose faces can be seen. The videos are high quality (2160p) at 30fps. This dataset underwent much more extensive filtering and extraction of training clips since the source videos contain many cuts and abrupt changes. See Table 2.1 for statistics of our dataset curation. From 48 source videos we selected 28 videos with ample footage of stable mountain biking, and then manually filtered for contiguous segments of mountain biking that were at least 5 seconds long, resulting in 1202 total clips.

2.8 Results

We evaluate our model through qualitative examination of the generated videos (Section 2.8.1), analyzing color change over time (Section 2.8.2), computing the FVD metric (Section 2.8.3), and ablating the key design choices (Section 2.9.4). We compare with StyleGAN-V [12] on all datasets. Mountain biking, horseback riding and ACID [32] datasets contain videos with a 16×9 widescreen aspect ratio. We train at 256×144 resolution on these datasets to preserve the aspect ratio. Since StyleGAN-V is based on StyleGAN2 [13], we can easily extend it to support non-square aspect ratios by masking real and generated frames during training. We found it necessary to increase the R1 γ hyperparameter by 10× to produce good results with StyleGAN-V on our new datasets that exhibit complex changes over time. We compare with MoCoGAN-HD [11], TATS [16] and DIGAN [15] using pre-trained models for the SkyTimelapse dataset at 128^2 resolution. For these comparisons, we train a separate super-resolution network to output the frames at 128^2 resolution, but use the same low-resolution generator as in the 256^2 comparison.

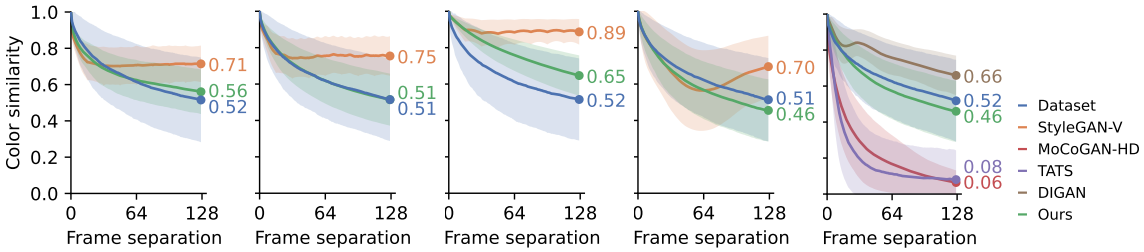
2.8.1 Qualitative Results

The major qualitative difference in results is that our model generates realistic new content over time, whereas StyleGAN-V continually repeats the same content. The effect is best observed by watching videos on the project webpage and is additionally illustrated in Figure 2.1. Scenery changes over time in real videos and our results as the horse moves forward through space. However, the videos generated by StyleGAN-V tend to morph back to the same scene at regular intervals. Similar repeated content from StyleGAN-V is apparent on all datasets. For example, results on the webpage for the SkyTimelapse dataset show that clouds generated by StyleGAN-V repeatedly move back and forth. MoCoGAN-HD and TATS suffer from unrealistic rapid changes over time that diverge, and DIGAN results contain periodic patterns visible in both space and time. Our model is capable of generating a constant stream of new clouds.

As a further validation of our observations, we conducted a preliminary user study on Amazon Mechanical Turk (Section 2.8.4). We created 50 pairs of videos for each of the 4 datasets. Each pair contained a random video generated by StyleGAN-V and one generated by our method, and we asked the participants which of them exhibited more realistic motion in a forced-choice response. Each pair was shown to 10 participants, resulting in a total of 50×4×10 responses. Our method was preferred over 80% of the time for every dataset.

2.8.2 Analyzing Color Change Over Time

To gain insight into how well different methods produce new content at appropriate rates, we analyze how the overall color scheme changes as a function of time. We measure color similarity as the intersection between RGB color histograms; this serves as a simple



(a) Biking (b) Horseback (c) ACID (d) Sky 256^2 (e) Sky 128^2

Figure 2.7: Color similarity (Eq. 2.1) of real and generated videos as a function of frame separation, reported as the mean (solid lines) and standard deviation (shaded regions) over 1000 random clips. TEST

proxy for actual content changes and helps reveal the biases of different models. Let $H(x, i)$ denote a 3D color histogram function that computes the value of histogram bin $i \in [1, \ldots, N^3]$ for the given image x, normalized so that $\sum_i H(x, i) = 1$. Given video clip $\boldsymbol{x} = \{x_t\}$ and frame separation t, we define the color similarity as

$$S(\boldsymbol{x}, t) = \sum\nolimits_i \min\left(H(x_0, i),\ H(x_t, i)\right), \tag{2.1}$$

where $S(\boldsymbol{x}, t) = 1$ indicates that the color histograms are identical between x_0 and x_t. In practice, we set $N = 20$ and report the mean and standard deviation of $S(\cdot, t)$, measured on 1000 random video clips containing 128 frames each.

Figure 2.7 shows $S(\cdot, t)$ as a function of t for real and generated videos on each dataset. The curves trend downward over time for real videos as content and scenery gradually change. StyleGAN-V and DIGAN are biased toward colors changing too slowly — both of these models include a global latent code that is fixed over the entire video. On the other extreme, MoCoGAN-HD and TATS are biased toward colors changing too quickly. These models use recurrent and autoregressive networks, respectively, both of which suffer from accumulating errors. Our model closely matches the shape of the target curve, indicating that colors in our generated videos change at appropriate rates.

Color change is a crude approximation of the complex changes over time in videos. In Section 2.9.2 we also consider LPIPS [62] perceptual distance instead of color similarly and observe the same trends in most cases.

2.8.3 Fréchet Video Distance (FVD)

The commonly used Fréchet video distance (FVD) [63] attempts to measure similarity between real and generated video distributions. We find that FVD is sensitive to the realism of individual frames and motion over short segments, but that it does not capture long-term realism. For example, FVD is essentially blind to unrealistic repetition of content over time, which is prominent in StyleGAN-V videos on all of our datasets. We

	Biking		Horseback		ACID		Sky 256^2	
	FVD_{128}	FVD_{16}	FVD_{128}	FVD_{16}	FVD_{128}	FVD_{16}	FVD_{128}	FVD_{16}
StyleGAN-V	533.3	353.7	427.0	319.2	112.4	91.5	151.2	48.4
with 10× R1 γ	224.6	99.2	196.2	159.0	–	–	–	–
Ours	113.7	83.8	95.9	113.5	166.6	127.3	152.7	116.5

	Sky 128^2	
	FVD_{128}	FVD_{16}
MoCoGAN-HD	635.6	224.9
TATS	435.0	97.0
DIGAN	228.6	153.4
Ours	142.6	107.5

Table 2.2: We compute FVD on segments of 128 and 16 frames (FVD_{128} and FVD_{16} respectively), where lower is better. **Left:** Our model outperforms StyleGAN-V on horseback riding and mountain biking datasets – both of which contain complex motion and new content over time. Our model underperforms StyleGAN-V on ACID and SkyTimelapse despite qualitative improvements and favorable user study ratings in Section 2.8.1. **Right:** Our model outperforms MoCoGAN-HD, TATS and DIGAN baselines on SkyTimelapse at 128^2 resolution on FVD_{128}.

found FVD to be most useful in ablations, i.e., when comparing slightly different variants of the same architecture.

FVD [63] computes the Wasserstein-2 distance [64] between sets of real and generated features extracted from a pre-trained I3D action classification model [65]. Skorokhodov *et al.* [12] note that FVD is highly sensitive to small implementation differences, down to the level of image compression settings, and that the reported results are not necessarily comparable between papers (Appendix C in [12]). We report all FVD results using consistent evaluation protocol, ensuring apples-to-apples comparison. We separately measure FVD using 128- and 16-frame segments, denoted by FVD_{128} and FVD_{16}, and sample 2048 random segments from both the dataset and generator in each case.

Table 2.2 (left) reports FVD on all datasets for StyleGAN-V and our model. We outperform StyleGAN-V on horseback riding and mountain biking datasets that contain more complex changes over time, but underperform on ACID and slightly underperform on SkyTimelapse in terms of FVD_{128}. However, this underperformance strongly disagrees with the conclusions from the qualitative user study in Section 2.8.4. We believe this discrepancy comes from StyleGAN-V producing better individual frames, and possibly better small-scale motion, but falling seriously short in recreating believable long-term realism – and the FVD being sensitive primarily to the former aspects. Table 2.2 (right) reports FVD metrics on MoCoGAN-HD, TATS, DIGAN and our model for SkyTimelapse at 128^2; we outperform all baselines in terms of FVD_{128} on this comparison.

2.8.4 User Study on Video Quality

We conducted a user study on Amazon Mechanical Turk to gauge realism of motion generated by our method in comparison to StyleGAN-V. While the user study is on a relatively small scale and does not measure all aspects of video quality, it provides an important signal about realism that is not captured by FVD. FVD does not favor our method on all datasets, but we observe a substantial qualitative improvement regarding generation

of motion and introduction of new content over time. The user study shows preference for videos generated by our method on all datasets, corroborating this observation.

For our user study we create 50 pairs of videos for each of the four datasets, where each pair has one random video from our method and one random video from StyleGAN-V. We instruct participants to select the favorable video in a forced-choice response: "Pick the video that is MORE realistic. For each comparison, you will be presented two videos. Please click each video to view it. Please pick the video that contains more realistic motions." See Figure 2.8 for a screenshot of instructions provided to participants and Table 2.3 for the portion of responses that favor our method compared to StyleGAN-V. Our method was preferred over 80% of the time for every dataset.

Each video pair was shown to 10 participants resulting in 500 responses per dataset. Each participant gave responses for 5 different video pairs. We select workers who have a past approval rating over 95% and who have completed over 1000 jobs. Our user study uses participants to complete a labeling task to measure video realism; humans are not the subjects and we do not study the participants themselves. Based on the average completion time, the hourly wage per participant ranged from $6 to $9.

	Mountain biking	Horseback riding	ACID	SkyTimelapse
StyleGAN-V	16.4%	13.4%	19.4%	18.4%
Ours	83.6%	86.6%	80.6%	81.6%

Table 2.3: Percent of responses that label motions more realistic in videos generated with our method compared with StyleGAN-V in a forced-choice user study with 500 responses per dataset.

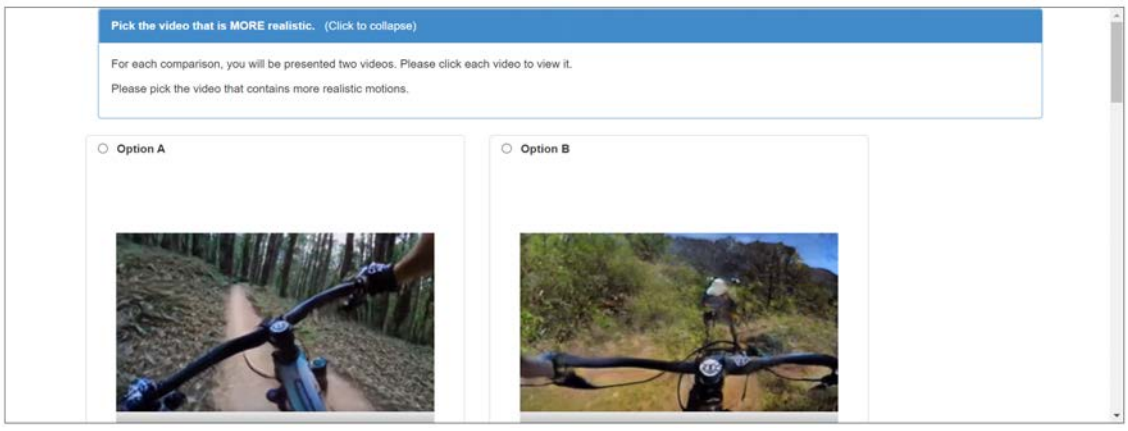


Figure 2.8: Screenshot of instructions provided to user study participants.

2.9 Additional Results

Figure 2.9: **Top:** Our mountain biking dataset exhibits complex motions and changes to the environment, such as transitioning between open areas and areas with tree coverage. **Middle:** StyleGAN-V is incapable of generating new content over time and the biker fails to move forward. **Bottom:** Our video generation method produces realistic motion and scenery changes. Over a 10s interval, the biker transitions out of the woods — a natural occurrence when mountain biking.

2.9.1 Additional Qualitative Results

See Figures 2.9, 2.10, 2.11, 2.12 for qualitative results of our videos compared with baseline methods. Please also see the project webpage to watch the same videos, as well as watch grids of randomly sampled videos for each dataset and method. In all videos, StyleGAN-V [12] fails to generate new content as the video progresses, and instead repeats the same content (e.g., clouds moving back and forth for the SkyTimelapse dataset).

Figure 2.10: **Top:** ACID [32] contains nature drone footage with large gradual changes in camera viewpoint. **Middle:** StyleGAN-V produces videos with pulsating camera motion, unable to create the illusion of a smooth camera trajectory. **Bottom:** Our model implicitly learns to generate changes in camera viewpoint over smooth trajectories, such as rotating while moving forward in 3D space.

Figure 2.11: **Top:** SkyTimelapse [61] (256^2 resolution) includes timelapse videos with a stream of new clouds and weather conditions. **Middle:** StyleGAN-V moves the same clouds back and forth. For example, compare the clouds at 1s, 2s and 5s marks: the clouds change between 1s and 2s, but then return back to the same clouds at 5s. **Bottom:** Our model generates new clouds over time.

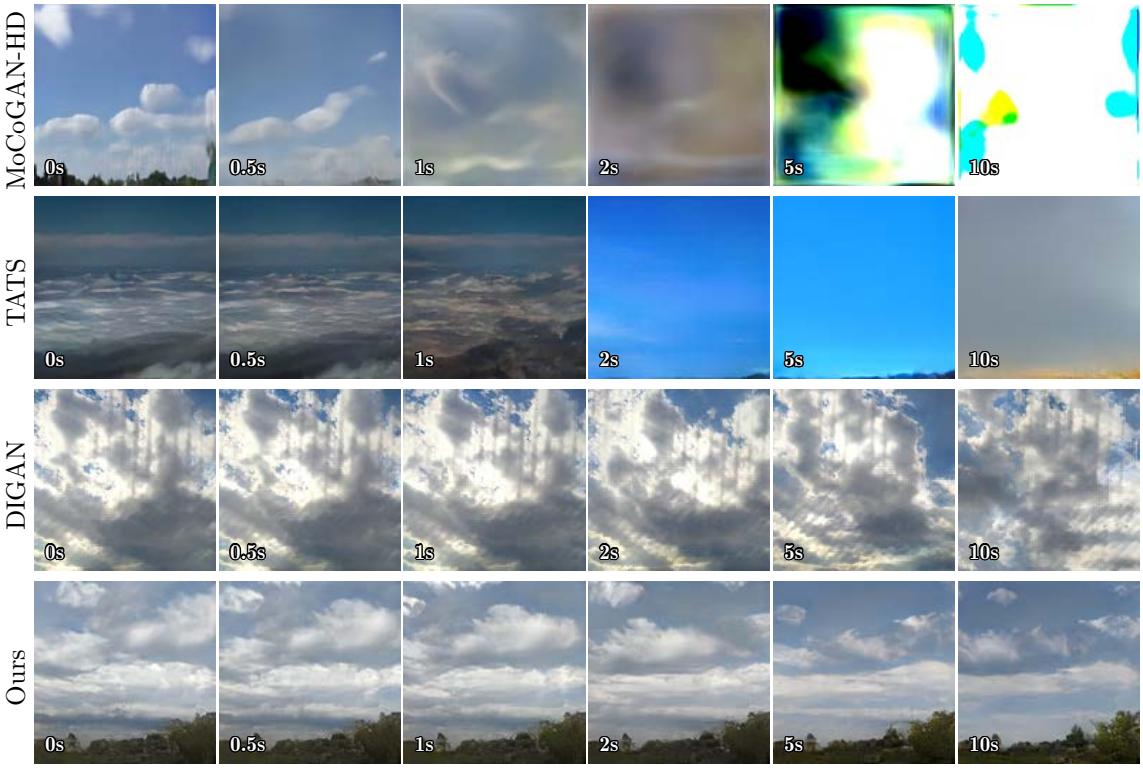
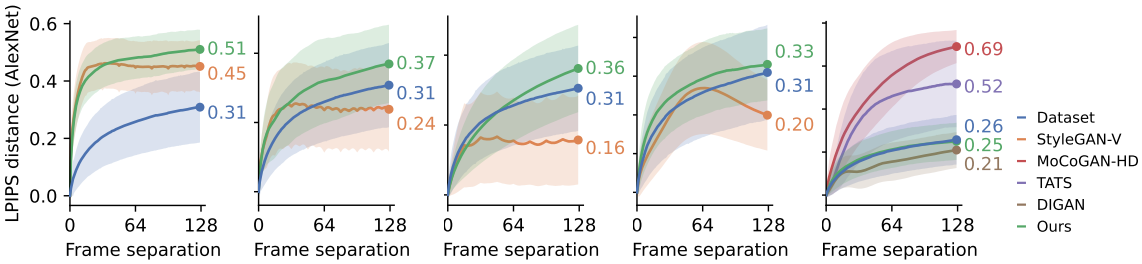
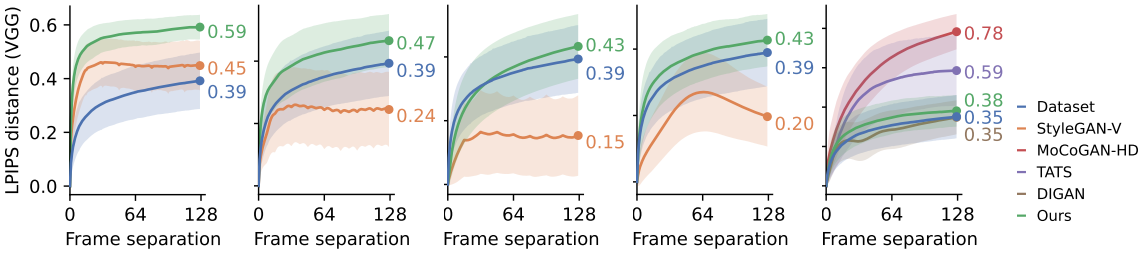


Figure 2.12: SkyTimelapse [61] (128^2 resolution). Real video omitted. **Top:** MoCoGAN-HD [10] is based on a recurrent network in latent space of a pretrained StyleGAN2 [13] model. It produces a realistic initial frame, but the video quickly explodes over a long duration. **2nd:** TATS [16] employs an autoregressive transformer to generate videos. While short segments produce plausible frames, videos change far too rapidly. **3rd:** DIGAN [15] uses an implicit representation to generate videos pixel by pixel. Strong periodic patterns are visible in space and time. **Bottom:** Our model generates videos that are consistent over time.



(a) Biking (b) Horseback (c) ACID (d) Sky 256^2 (e) Sky 128^2

Figure 2.13: LPIPS distance (AlexNet) over time.



(a) Biking (b) Horseback (c) ACID (d) Sky 256^2 (e) Sky 128^2

Figure 2.14: LPIPS distance (VGG) over time.

2.9.2 Analyzing Change Over Time in Feature Spaces

In Section 2.8.2, we measure color similarity at increasing frame spacings for different datasets and methods to uncover bias in how much change occurs over time. Intersection of color histograms (Equation 1) is a simple proxy for change over time, and is entirely agnostic to spatial patterns. It is reasonable to also consider other distance functions, such as perceptual similarity metrics [62, 66]. In Figure 2.13 and Figure 2.14 we show the LPIPS [62] metric based on AlexNet [67] and VGGNet [68] features respectively. (Note the opposite direction of change compared to Figure 2.7: color similarity decreases over time, whereas feature distance *increases* over time.)

In most cases, we observe the same trend as for color similarity — StyleGAN-V changes too slowly in horseback, ACID and SkyTimelapse, and our method does a relatively better job at matching the rate of change in real videos. The mountain biking dataset shows a different trend for perceptual similarity, where both our method and StyleGAN-V curves are shifted too high (too much change), and StyleGAN-V is closer to the dataset curve. One caveat of this use of perceptual metrics is that, even ignoring the temporal aspect, we observe substantial distributional shift of pretrained features between generated and real

frames (e.g., penultimate VGG features for both our model and StyleGAN-V have over 30% larger magnitudes than for real frames on the biking dataset). It is thus unclear to what extent the difference in curves between real and generated videos is due to different rates of change over time or the distributional shift independent of change over time.

We favor the color similarity measure as the simplest approximation for how quickly things change over time, and acknowledge that it is not intended as a standalone metric but a probe into the biases of videos generated with different methods.

2.9.3 Image Quality Tradeoff

In practice, there exists a tradeoff between per-frame image quality and the quality of motion and change over time. At one extreme, an image generator is optimized specifically for image quality. Image generators produce very high quality images, but have no inherent ability to produce realistic videos. Many video generation models prioritize frame quality, whereas our model prioritizes accurate changes over long durations. FVD_{128} and FVD_{16} metrics [63] measure unknown combinations of spatial and temporal patterns, and while they provide a useful signal, it is not clear where these metrics fall in terms of favoring per-frame image quality or accurate temporal changes.

We analyze color similarity over time in Section 2.8.2. Color similarity between frames is agnostic to spatial patterns, and provides insight on the rate of change over time in isolation from per-frame image quality. To gain a holistic picture of the priorities of our model, we also compute a per-frame image quality metric, video-balanced Fréchet inception distance (FID_V), which we describe below and report in Table 2.4. StyleGAN-V outperforms our model on three of the four datasets in terms of FID_V. This tradeoff is expected, since StyleGAN-V is heavily based on the StyleGAN2 [13] image generator. It produces high image quality but is unable to model complex motions or changes over time, whereas our model prioritizes the time axis.

Assessing quality of generated videos is multifaceted, and we believe all of the evaluation we provide — qualitative results, user study, color change over time, FVD, and FID — help expose gaps in the abilities of existing methods and the strengths and weaknesses of our new model.

Video-balanced Fréchet inception distance (FID_V) To correctly measure per-frame image quality, it is important to balance the computation of FID [69] such that very long videos in the dataset do not overpower results. (This is particularly important for the SkyTimelapse [61] dataset, which has an outlier video that is extremely long.) Skorokhodov *et al.* [12] point out that it is undesirable for these very long videos to bias training or computing FVD [63], and the same is true for computing FID [69] per-frame on video data.

To correctly balance FID to value each training video equally, we weight calculation of the covariance and mean by the inverse of the number of frames in each clip when

	Mountain biking	Horseback riding	ACID	SkyTimelapse
StyleGAN-V	33.9	51.6	11.3	12.6
with 10× R1 γ	12.5	17.7	–	–
Ours	18.9	12.2	18.2	26.6

Table 2.4: Video-balanced Fréchet inception distance (FID_V) measures per-frame image quality, where lower is better. While our emphasis is the time axis, we report image quality to gain insight on the priorities of StyleGAN-V and our model. StyleGAN-V outperforms our model in terms of per-frame image quality on three of the four datasets, which aligns with StyleGAN-V's focus on image quality and our focus on accurate change over time.

	FVD_{128}	FVD_{16}
Ours (128 frames)	113.7	83.8
16 frames	163.6	108.5
2 frames	396.8	169.4

(a) Ablation of training sequence length

	FVD_{128}	FVD_{16}
Ours	113.7	83.8
0.1× lowpass width	153.1	113.2
10× lowpass width	217.9	126.5

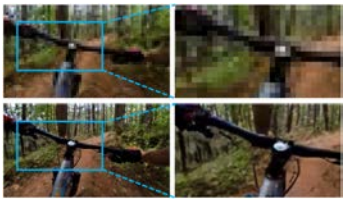
(b) Ablation of temporal lowpass footprint

Table 2.5: **(a)** Our model learns to generate realistic long videos by training on long videos; decreasing the sequence length used during training is consistently harmful. **(b)** The footprint of the temporal lowpass filters plays an important role in producing inputs to the low-resolution mapping network at appropriate temporal frequencies; changing the footprint by an order of magnitude hurts performance.

measuring the Wasserstein-2 distance [64] between sets of features. This has the effect of valuing each video equally, while still including contribution from all frames, which is important when there are a small number of long videos such as in our horseback riding dataset. A similar strategy to weight covariance and mean when computing FID is used by Kynkäänniemi *et al.* [70] to analyze the effect of balancing object class occurrences. When computing statistics for generated frames, we sample 50,000 videos of length 1 frame (at $t = 0$ for StyleGAN-V).

2.9.4 Ablations

Training on long videos improves generation of long videos. Observing long videos during training helps our model learn long-term consistency, which is illustrated in Table 2.5a that ablates the sequence length used during training of the low-resolution generator. We found that the benefits of training with long videos only became evident after designing a generator architecture with appropriate temporal receptive field to utilize the rich training signal. Note that even though we ablate aspects of the low-resolution generator, we still compute FVD using the final high-resolution videos produced by the super-resolution network.



(a) Mountain biking

(b) ACID

		FVD_{128}
Biking	Ours	113.7
	SR on reals	58.3
ACID	Ours	166.6
	SR on reals	68.8

(c) Ablation

Figure 2.15: Evaluation of the super-resolution network. **(a,b)** Generated low-resolution frames and the corresponding high-resolution frames produced by the super-resolution network. **(c)** The super-resolution network yields remarkably good FVD when provided with real low-resolution videos as input; the overall quality of our results is largely dictated by the low-resolution generator.

Footprint of the temporal lowpass filters. Our temporal latent representation serves a vital role in expanding the receptive field of our generator, modeling patterns over different time scales, and enabling the generation of new content over time. While we primarily leverage long training videos to learn long-term consistencies from data, the size of our temporal lowpass filters plays a role in encouraging the low-resolution mapping network to learn correlations at appropriate time scales. Table 2.5b demonstrates the negative impact of using inappropriately sized filters. We find that our model performs well with the same filter configuration for all datasets, although it is possible that the ideal settings may vary slightly between datasets.

Effectiveness of the super-resolution network. Figure 2.15a,b shows examples of low-resolution frames generated by our model along with the corresponding high-resolution frames produced by our super-resolution network; we find that the super-resolution network generally performs well. To ensure that the quality of our results is not disproportionately limited by the super-resolution network, we further measure FVD when providing the super-resolution network with *real* low-resolution videos as input in Figure 2.15c. Indeed, FVD greatly improves in this case, which indicates that there are still significant gains to be realized by further improving the low-resolution generator.

2.9.5 Failure Cases

Separate low- and super-resolution networks makes the problem computationally feasible, but it may somewhat compromise the quality of the final high-resolution frames. We observed that "swirly" artifacts are most prominent in the super-resolution output and not in the low-resolution output. Our model also struggles with long-term consistency of small details (e.g., distant jumps in generated horseback riding videos) that begin to appear before quickly fading out. We believe these issues are due to limitations of our super-resolution network, and that improving the super-resolution network would benefit

the model in this regard. Another failure case we observed is difficulty preserving 3D consistency for scenes with very little motion, such as in the ACID dataset. In cases where there is little motion, one may consider using an explicit 3D representation.

2.10 Conclusions

VVideo generation has historically focused on relatively short clips with little new content over time. We consider longer videos with complex temporal changes, and uncover several open questions and video generation practices worth reassessing — the temporal latent representation and generator architecture, the training sequence length and recipes for using long videos, and the right evaluation metrics for long-term dynamics.

We have shown that representations over many time scales serve as useful building blocks for modeling complex motions and the introduction of new content over time. We feel that the form of the latent space most suitable for video remains an open, almost philosophical question, leaving a large design space to explore. For example, what is the right latent representation to model persistent objects that exit from a video and re-enter later in the video while maintaining a consistent identity?

The benefits we find from training on longer sequences open up further questions. Would video generation benefit from even longer training sequences? Currently we train using segments of adjacent frames, but it might be beneficial to use larger frame spacings to cover longer time spans.

Quantitative evaluation of the results continues to be challenging. As we observed, FVD goes only a part of the way, being essentially blind to repetitive, even very implausible results. Our tests with how the colors and LPIPS distance change as a function of time partially bridge this gap, but we feel that this area deserves a thorough, targeted investigation of its own. We hope our work encourages further research into video generation that focuses on more complex and longer-term changes over time.

Negative societal impacts Our work falls within data-driven generative modeling, which, as a field, has well known potential for misuse with increasing quality improvements. The training of video generators is even more intensive computationally than training still image generators, increasing energy usage. Our project consumed 300MWh on an in-house cluster of V100 and A100 GPUs.

Chapter 3

Hallucinating Pose-Compatible Scenes

3.1 Introduction

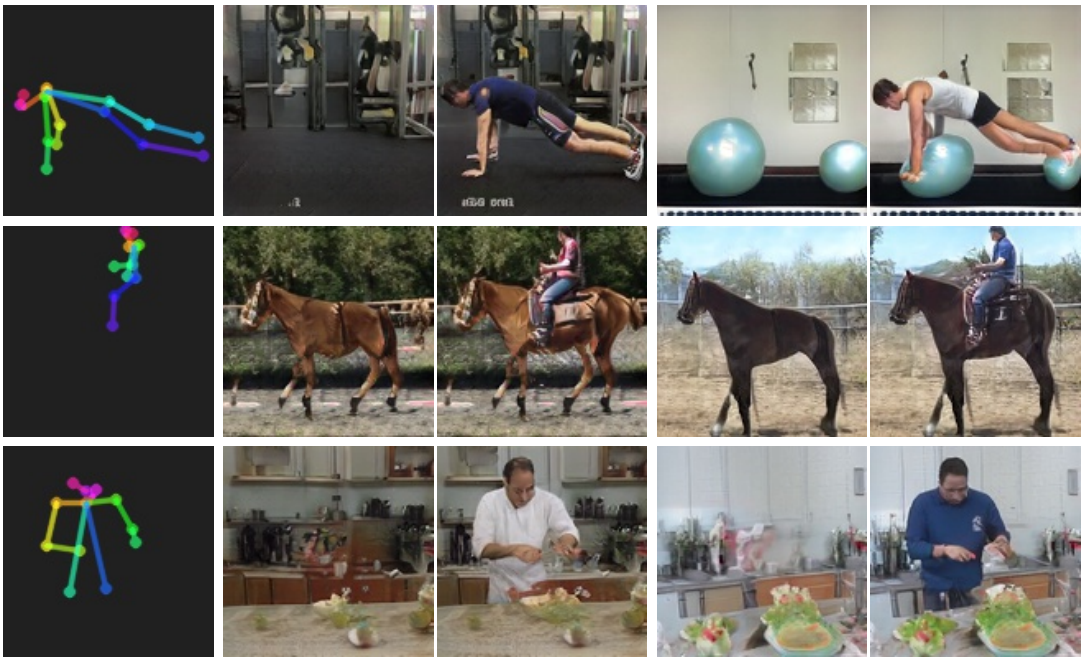
Human pose can reveal a lot about a scene. For example, mime artists[1] invoke vivid scenes in a viewer's mind through pose and movement alone, despite performing on a bare stage. The viewer is able to imagine the invisible objects and scene elements because of the strong relationship between human poses and scenes learned through a lifetime of daily observations.

Psychologists have long been interested in understanding this symbiotic relationship between human and scene [72, 73]. J.J. Gibson proposed the notion of *affordances* [72], which can be described as "opportunities for interactions" furnished by the environment. In computer vision, affordances have been used to provide a functional description of the scene. Given an image, a number of approaches try to predict likely human poses these scenes afford [74–77].

This work considers the opposite problem: given a human pose as input, the goal is to hallucinate scene(s) that are compatible with that pose. Consider Figure 3.1. A push-up pose (top) places severe constraints on the space of compatible scenes: they must not only be semantically compatible (e.g., gym, exercise room), but also have compatible spatial affordances (enough floor space or appropriate equipment). Objects in the scene can afford interaction with the human (e.g., squishing down an exercise ball). Other poses might not appear as constraining, but even a simple standing pose (bottom) — head looking down, hands reaching in, legs occluded — is actually a strong indicator of a cooking scene, and signals that an object (e.g., countertop) must be occluding the legs.

The work presented in this chapter was first published in Brooks *et al.* as *Hallucinating Pose-Compatible Scenes* at the European Conference on Computer Vision (ECCV), 2022 [71].

[1]For those unfamiliar with mime artists, here is a wonderful example performance: https://youtu.be/FPMBV3rd_hI



Input poses Sample output scenes

Figure 3.1: Given a human pose as input, the task presented in this chapter is to hallucinate scene(s) that are compatible with that pose. Our model can generate isolated scenes as well as scenes containing humans.

Rather than explicitly model scene affordances and contextual compatibility, we employ a modern large-scale generative model (based on a souped-up StyleGAN2 [13] architecture) to *discover* these relationships implicitly, from data. While GANs have performed well at capturing disentangled visual models in specialized scenarios (e.g., faces, churches, categories from ImageNet [78]), they have not been demonstrated *in situ*, on complex, real-world data across varying environments.

We curate a massive meta-dataset of humans interacting with everyday environments, containing over 19 million frames. The complexity and scale of data is much higher than common GAN datasets, such as FFHQ [14] (70,000 face images) and ImageNet [78] (1.3M object images). With an appropriate pose conditioning mechanism, increased model capacity, and removal of style mixing, we are able to successfully train a pose-conditioned GAN on this highly complex data. Our model and meta-dataset mark substantial progress leveraging GANs in real-world settings containing humans and diverse environments. Through numerous visual experiments, we demonstrate our model's emergent ability to capture affordances and contextual relationships between poses and scenes.

See our webpage[2] for our supplemental video and code release.

[2] https://www.timothybrooks.com/tech/hallucinating-scenes

3.2 Related Work

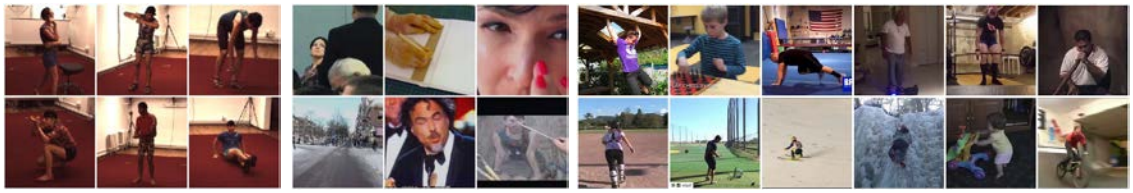
Scene and object affordances. Affordances [72] describe the possible uses of a given object or environment. A significant body of work learns scene affordances, such as where a person can stand or sit, from observing data of humans [74–76, 79–84]. Overlapping areas of work focus on human interactions with objects [85–89] or synthesize human pose conditioned on an input scene [90–92]. We propose the reverse task of hallucinating a scene conditioned on pose.

Pose-conditioned human synthesis. There are a plethora of methods that take a source image (or video) of a human plus a new pose and generate an image of the human in the new pose [93–99]. Although we too condition on pose, our goals are almost entirely opposite: we aim to generate novel scenes compatible with a given pose, whereas the above methods reuse the scene from the source image/video and only focus on reposing within that provided scene.

GANs for image synthesis. Introduced by Goodfellow *et al.* [4] a generative adversarial network (GAN) is an implicit generative model that learns to synthesize data samples by optimizing a minimax objective. The generator is tasked with fooling a discriminator, and the discriminator is tasked with differentiating real and generated samples. Modern GANs are capable of producing high quality images [13, 14, 36, 100]. Image translation [96, 101] utilizes conditional GANs [102] to translate from one domain to another. While our task is pose-conditional scene generation, we leverage benefits of modern unconditional GANs [13].

Visual disentanglement. Disentanglement methods attempt to separate out independent controllable attributes of images. This can be achieved with unsupervised methods [14, 103–105], or an auxiliary signal [106, 107]. Components of image samples can be added, removed and composed using pretrained GANs [108, 109]. Recent work has applied similar strategies to image translation models to compose style and content from different images [110]. The most related to us is the work of Ma *et al.* [111], who synthesize images of people, while independently controlling foreground, background, and pose. However, the focus is on generating humans in very tightly cropped images with simple backgrounds, rather than generating scenes with appropriate affordances. Many disentanglement methods assume all images or image attributes can be combined with all others [14, 103–107, 110, 111]. In this work, we seek disentangled representations of pose, human appearance and scene, yet it is essential our model understand which scenes can or cannot be composed with which poses.

Contextual relationships. Many works leverage contextual relationships among objects and scenes [73] to improve vision models such as object recognition and semantic



(a) Humans 3.6M [116] (b) Kinetics [117] (c) Humans in Context (ours)

Figure 3.2: **Dataset comparison.** (a) The largest human-centric video dataset with ground truth poses uses a fixed background, missing scene interactions. (b) Action recognition datasets include scenes, but contain videos without people or of close-up content. (c) Our dataset is a massive curation of humans in scenes.

segmentation [112–115]. Divvala *et al.* [114] explicitly enumerate (Table 1) a taxonomy of possible contextual information. In this chapter we are specifically interested in contextual relationships between humans and their environments, and aim to recover them implicitly, from data.

3.3 *Humans in Context* Dataset

To study the rich relationship between scenes and human poses requires large-scale data of people interacting with many different environments. Internet videos are a natural source, containing vast data of daily human activities. Unfortunately, large-scale action recognition datasets [117–119] include substantial content without humans, as well as close-up footage not of scenes. Most existing human-centric datasets are insufficiently small [120, 121], narrow in scene type [122, 123], or captured on a fixed background [116].

We therefore curate a meta-dataset of 229,595 video clips, each containing a single person in a scene, sourced from 10 existing human and action recognition video datasets [117–126], and supplemented with pseudo-ground truth pose obtained using OpenPose [127, 128]. Video offers a massive source of real-world data, and ensures all poses of human activity are represented, rather than only poses photographers choose to capture in still images.

Videos are extensively filtered for quality, ensuring satisfactory framerate, bitrate and resolution. 1,509,032 videos (75% of source videos) pass quality filtering. Frames are then filtered with pretrained Keypoint R-CNN [129, 130] person detection and OpenPose [127, 128] keypoint prediction models. The final dataset only includes clips of at least 30 frames where Keypoint R-CNN detects a single person and OpenPose predicts sufficient keypoints. This results in 19,503,700 frames (7.8% of high quality frames), with each clip averaging 85 frames long. While we train on images, we split data into partitions based on video clips, reserving 12,800 clips for testing and the remaining 216,795 for training.

Table 3.1: **Humans in Context source data.** Our dataset consists of video clips filtered from 10 existing human and action recognition datasets. High quality clips have sufficient bitrate, framerate and resolution. Person clips are those where pretrained person detection and pose prediction networks assert that a single person is present. In total we curate 229,595 clips and 19,503,700 frames.

	# Video Clips			# Frames	
	Source	High Quality	Person	High Quality	Person
HVU [118]	566,489	353,174	105,634	98,603,223	9,590,407
Moments [119]	757,804	653,368	54,156	56,074,418	3,374,112
Kinetics-700-2020 [117, 131]	620,119	432,502	26,911	78,037,500	2,428,079
Charades [120]	9,848	7,319	16,967	6,256,421	2,157,074
InstaVariety [124]	2,545	2,449	5,773	1,898,824	730,211
Oops [125]	29,940	27,953	8,360	5,738,042	596,488
MPII [121]	24,987	24,980	8,820	1,025,459	352,498
VLOG-people [122]	663	555	1,261	321,071	163,956
PennAction [123]	2,326	2,221	1,208	161,029	76,112
YouTube-VOS [126]	4,519	4,511	505	613,441	34,763
Total	2,019,240	1,509,032	**229,595**	248,729,428	**19,503,700**

3.3.1 Dataset Curation and Preprocessing Details

Our dataset contains diverse footage of humans immersed in everyday environments. Each image is supplemented with pseudo-ground truth human pose attained using OpenPose [127, 128]. The data is sourced from 10 existing human and action recognition datasets, with the numbers of clips and frames from each source dataset detailed in Table 3.1. Video footage provides a vast source of diverse human activity, and ensures all poses are represented, rather than only human poses photographers choose to capture in still images. For the MPII [121] dataset, which is primarily a still image dataset, we use short video clips of the frames preceding and following each image.

Each dataset contains unique biases, and combining data sources is less subject to the bias of any particular dataset. Different datasets also offer different types of scenes. For example, Moments [119] includes classes absent from HVU [118], and Oops [125] contains uncommon accidental actions. The number of useful examples from each source was only evident after extensive curation. Video offers a massive source of real-world data, and ensures all poses of human activity are represented, rather than only human poses photographers choose to capture in still images.

We filter out videos where either dimension is shorter than 256 pixels, and we resize remaining videos using Lanczos resampling [132] such that the smaller edge is exactly 256 pixels. We exclude videos with an average bitrate below 0.9 bits per pixel, or with a framerate that does not fall between (and cannot be subsempled to fall between) 23.9

fps and 30 fps. Videos are truncated to 3000 frames. Source datasets which provide pre-extracted frames only undergo quality filtering by spatial resolution.

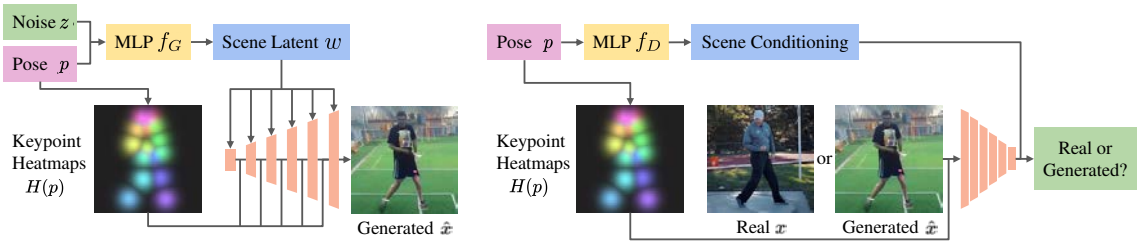
Frames are then filtered to contain a single person using pretrained Keypoint R-CNN [129, 130] person detection. Person bounding boxes are detected for each frame, with a minimum accuracy of 95%, a minimum bounding box area of 1% of the total frame area, and non-maximum suppression of overlapping bounding boxes with an intersection over union greater than 0.3. With these thresholds, any frame with more than a single person detected is removed. Stricter thresholds are then applied to the remaining frames with a single person bounding box: a minimum accuracy of 98%, a minimum bounding box area of 4% of the total frame area, and a maximum bounding box area of 80% of the total frame area. These thresholds ensure with high accuracy that there is a single person present in the frame at a reasonable size. Frames are then cropped to a 256×256 resolution toward the average bounding box center for each contiguous segment of frames.

Pseudo-ground truth pose labels are computed for each frame using OpenPose [127, 128] keypoint prediction. We use the single-scale OpenPose version to compute 18 body keypoints. Similar to person detection, we use a relaxed total score threshold of 2.5 when filtering for multiple people, and a strict total score threshold of 10.0 when ensuring there is a single person. Each individual keypoint has a score threshold of 0.3, and keypoints below this threshold are marked as not visible in the frame. To avoid frames of just legs or torso, we only include frames where the keypoint at the base of the neck is visible, and where a total of at least 8 of 14 keypoints (excluding eyes and ears) are visible.

The final dataset only includes clips of at least 30 adjacent frames where each frame passed filtering. Note that multiple clips may be sourced from the same video, and that duplicate videos from different source datasets are possible.

3.3.2 Dataset Licenses

The HVU dataset [118] is released for non-commercial research and educational purposes only, and was attained directly from the dataset authors. The Moments in Time [119] dataset is released for non-commercial research and educational purposes, and was attained from the dataset project website. The Kinetics dataset [117, 131] is licensed by Google Inc. under a Creative Commons Attribution 4.0 International License, and videos were downloaded directly from YouTube. The Charades dataset [120] is released under a non-commercial license detailed here: https://prior.allenai.org/projects/data/charades/license.txt; data was downloaded from the project webpage. The InstaVariety dataset [124] is released for non-commercial academic use, and was attained directly from the dataset authors. The Oops dataset [125] is released under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License, and was downloaded from the project webpage. The MPII [121] dataset is released under the Simplified BSD License detailed here: https://github.com/peiyunh/rg-mpii/blob/master/data/mpii_human/annotation/bsd.txt; data was downloaded from the project webpage.



(a) Generator (b) Discriminator

Figure 3.3: **Network architectures.** Our networks are based on StyleGAN2, with simple modifications to ensure accurately placed humans and compatible scenes. In particular, the conditional generator and discriminator networks utilize pose p via two mechanisms: keypoint heatmaps and pose latent conditioning. Keypoint heatmaps correctly positions a human, and pose latent conditioning drives latent codes w to generate compatible scenes. Multiple plausible scenes can be produced for the same input pose by sampling different noise vectors z.

The VLOG dataset [122] is released for non-commercial research purposes only, and was downloaded from the project webpage. The PennAction dataset [123] is released without a license and was downloaded from the project webpage; dataset authors confirmed there are no terms of use and only ask the corresponding paper [123] be cited. The YouTube-VOS dataset [126] is released for non-commercial research purposes only and was downloaded from the project challenge webpage.

3.4 Pose-compatible Scene GAN

We design a conditional GAN [4, 102] to produce scenes compatible with human pose. Our network architectures are based on StyleGAN2 [13] and are depicted in Figure 3.3. Generating high quality pose-compatible scenes arises from simple yet important modifications: dual pose conditioning, removal of style mixing, and large-scale training. Our model can produce isolated scene images without any human by zeroing out keypoint heatmaps when generating images.

3.4.1 Dual Pose Conditioning

The conditional generator G and discriminator D both utilize input pose via two mechanisms: keypoint heatmap conditioning, which specifies spatial placement of a human subject, and pose latent conditioning, which infers compatible scenes. To succeed at our task, humans must be positioned correctly and generated scenes must be compatible. Dual pose conditioning drives strong performance in both respects, and outperforms conditioning on either alone in our ablation experiment (Table 3.4). Furthermore, dual

pose conditioning disentangles control of scene and human pose. We leverage these separate controls for numerous applications: generating scenes without humans, visualizing incompatible scenes and poses, placing a person in a new scene, and animating pose.

Keypoint heatmaps. Let pose $p = (p_1, ..., p_K)$ denote 2D locations of the $K = 18$ human keypoints detected by OpenPose [128], and let $v = (v_1, ..., v_K)$ indicate visibility of each keypoint. Following the works of [94, 95, 97], our keypoint heatmaps $H(p)$ consist of radial basis function kernels centered at each keypoint. For heatmap $k \in \{1, ..., K\}$, the intensity at location q is given by Equation 3.1. We concatenate heatmaps at each scale of the generator, and at the input of the discriminator. We set $\sigma^2 = \max(0.5, 0.005R^2)$ where R is the spatial resolution of the heatmaps. After training, we generate images of scenes without humans by simply zeroing out all keypoint heatmaps.

$$H_{k,q}(p) = \begin{cases} \exp\left(-\frac{||q-p_k||^2}{2\sigma^2}\right) & \text{if } v_k = 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

Pose latent conditioning. To generate compatible scenes, we condition scene latent codes on the input pose. Akin to intermediate latents in StyleGAN2 [13], the scene latent code w controls generation by modulating convolutional weights. To condition the latent code, pose locations and visibility are flattened and mapped to a 512-dimensional input via a learned linear projection. A noise sample $z \sim \mathcal{Z}$ is concatenated with the input vector and passed through a multi-layer perceptron (MLP) f_G to produce a scene latent code $w \in \mathcal{W}$. Multiple plausible scenes can be generated by sampling different noise vectors z for the same pose. The discriminator learns a separate linear projection and MLP f_D.

3.4.2 Removal of Style Mixing

Style mixing regularization [14, 110] encourages disentanglement by randomly mixing intermediate latent codes during training. The technique assumes image attributes at each layer are compatible with all other image attributes (e.g. any face could have any color hair). This assumption is not true when composing scenes and humans, which we visually demonstrate through the incompatible scenes and poses in Figure 3.8. This motivates removing style mixing regularization during training, which improves results in our ablation experiments (Table 3.3).

3.4.3 Large-scale GAN Training

Typical datasets used with StyleGAN2 (e.g. faces, bedrooms, churches [133, 134]) are relatively homogeneous. Increasing model capacity is a natural extension given the diversity and complexity of scene images in our dataset. We find that increasing the channel

width of convolutional layers by 2× significantly improves our model (see ablation in Table 3.3). Following prior work in scaling GANs [100], we also increase minibatch size (from 40 to 120). Concurrent work [135, 136] also explores scaling StyleGAN, and proposes strategies such as self-filtering the training dataset [135], progressive growing and leveraging pretrained classifiers [136].

3.5 Model Implementation Details

We train all models at 128×128 resolution. Many aspects of our model are borrowed directly from StyleGAN2 [13], including non-saturating logistic loss [4], equalized learning rates for all parameters [36], R_1 regularization [50], path length regularization [13], and exponential moving average of generator parameters [36].

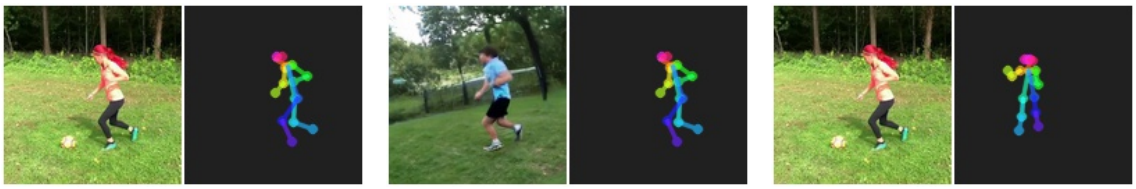
We use a learning rate of 2.5×10^{-3}, an exponential moving average rate of $\beta = 0.995$, a moving average warmup of 150,000 steps, and R_1 regularization strength of $\gamma = 0.05$. We remove spatial noise maps to isolate control over the scene to the latent code. We also remove style mixing regularization during training. Our final model was trained with a minibatch size of 120 on 10× NVIDIA Quadro RTX GPUs, and for 1,000,000 steps. Our ablations trained for 1 week, and we let the final large model continue for 3 weeks. The large generator has 85.4M parameters and discriminator has 98.2M. Ablations and the Pix2PixHD baseline were trained with a batch size of 40 on 5× NVIDIA GeForce RTX 2080 GPUs for 600,000 iterations. Multiple checkpoints were saved throughout training, and the checkpoint with the lowest FID score was used for all evaluation. The Pix2Pix baseline was trained for 10,000,000 iterations with a batch size of 40 on 5× NVIDIA GeForce RTX 2080 GPUs.

Code used from the public implementation of StyleGAN2-Ada is released under the NVIDIA code license found here: https://github.com/NVlabs/stylegan2-ada/blob/main/LICENSE.txt. Code used to run the Pix2Pix baseline is released under the BSD License license found here: https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix/blob/master/LICENSE.

3.5.1 Data Augmentation

Data augmentation of both generated and real images just prior to the discriminator can improve robustness and prevent the discriminator from overfitting to the train dataset [137, 138]. Our augmentation parameters are largely based on [138]. Brightness is augmented by randomly offsetting intensity by a value uniformly sampled from −25% to +25%. Saturation is augmented by interpolating red, green and blue channels toward or away from the mean of all three at each pixel, with interpolation weights uniformly sampled from 0.0 to 2.0. Contrast is augmented by interpolating color values toward or away from the mean of all color values in an entire frame sequence, with interpolation weights uniformly sampled from 0.5 to 1.5. Horizontal flipping is applied with a 50% chance to all

Figure 3.4: **Data augmentation.** We apply random spatial, cutout and color augmentations to frames and poses just prior to the discriminator network. Each pair above shows the original frame and pose on the left and the augmented output on the right.



(a) Real (b) Generated (c) Mismatched

Figure 3.5: **Mismatch discrimination.** The discriminator must classify (a) a real image and pose as real, (b) a generated frame and conditional pose as fake, and (c) a real frame and mismatched pose as fake.

frames and poses in a sequence. Frames are scaled by a factor uniformly sampled from 0.8 to 1.25 and translated by an offset sampled uniformly from -12.5% to $+12.5\%$. A random cutout, half the size of each dimension and randomly placed, is erased from each frame. Spatial transformations applied to frames are also applied to poses so that the frames and poses still correspond correctly. We briefly experimented with dropout augmentation of pose, but did not find it helpful. See Figure 3.4 for examples of our data augmentation.

3.5.2 Mismatch Discrimination

We force the discriminator to pay attention to pose conditioning by providing a mismatched real image with the incorrect pose conditioning as an additional fake example. For the mismatched fake example, the pose embedding and keypoint heatmaps both take pose from another sample in the minibatch. This training method was first introduced in text-to-image generation [139] but has not been widely used in the image or video translation literature; we found training with mismatch discrimination provides a slight improvement, forcing the discriminator to use conditioning.

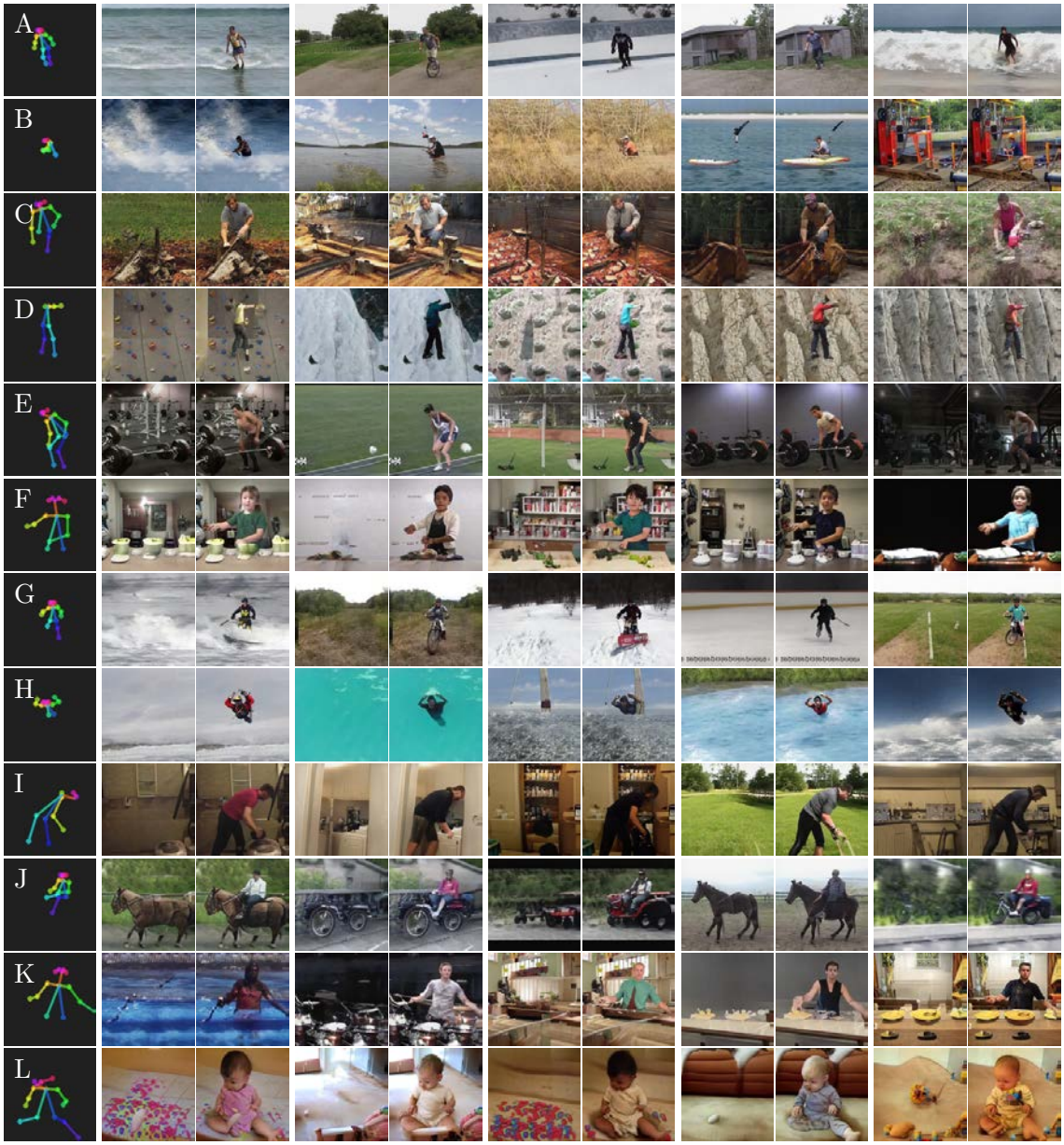


Figure 3.6: **Success cases.** Our model learns complex scene-pose relationships. For each input pose, we show many hallucinated scenes, with and without a human. Diverse outputs include a person paddling a kayak (B), lifting a barbell in their hand (E), and playing the drums (K). Our model provides insight into scenes with related affordances: in the same pose, a person may climb in an indoor gym or on a snowy ledge (D); a person can ride a horse, ride a bicycle, or ride a tractor (J). *See (I) for the world's first AI-generated image of a person cleaning a toilet.*

3.6 Experiments

Our model hallucinates diverse, high quality images of scenes compatible with input pose. We generate scenes in isolation as well as scenes containing humans, and analyze our model through several visual experiments. Generating scene images is challenging due to the high complexity of data, and our model outperforms Pix2Pix/Pix2PixHD [101, 140] and pose-conditioned StyleGAN2 [13] baselines in terms of image quality and accurate human placement. We present characteristic success and failure results in Figure 3.6 and Figure 3.7 respectively. See Section **??** for uncurated random samples from our model.

Figure 3.7: **Failure cases.** Causes for failure include: partially generating objects, such as a bike (A); poor overall image quality (B); missing limbs without proper occluders (C); difficulty placing objects, such as a golf club, in a person's hands (D); difficulty hallucinating an object on which to sit (E); overly repetitive textures (F); infeasible scenes, such as walking on water (G); and leaving behind a partial human when hallucinating the scene in isolation (H).

3.6.1 Not All Scenes and Poses Are Compatible

It is essential that we model which scenes are compatible with which poses. A person cannot do a push-up in the middle of a horse, ride atop a kitchen countertop, or be occluded by thin air. These scenarios sound obviously false, yet could occur if the scene and human pose are incompatible. We visualize images generated with correctly and incorrectly paired scenes and poses in Figure 3.8.

These examples of incompatible scenes and poses highlight an important difference between our scene data and other datasets commonly used for GAN training, such as cropped faces in the CelebA [133] and FFHQ [14] datasets. Any face can be given glasses, longer or shorter hair, or a darker or lighter skin tone and still remain a feasible image. This enables global disentanglement of attributes, and applications like style mixing, which combines different intermediate latent codes of any two samples (see Figure 3 of the original StyleGAN paper [14] for a wonderful example). The assumption of compatibility between all attribute pairs no longer holds for data of scenes with humans, which motivates conditioning scene latent codes on pose. Relatedly, we find that removing style mixing from training significantly improves performance (Table 3.3).

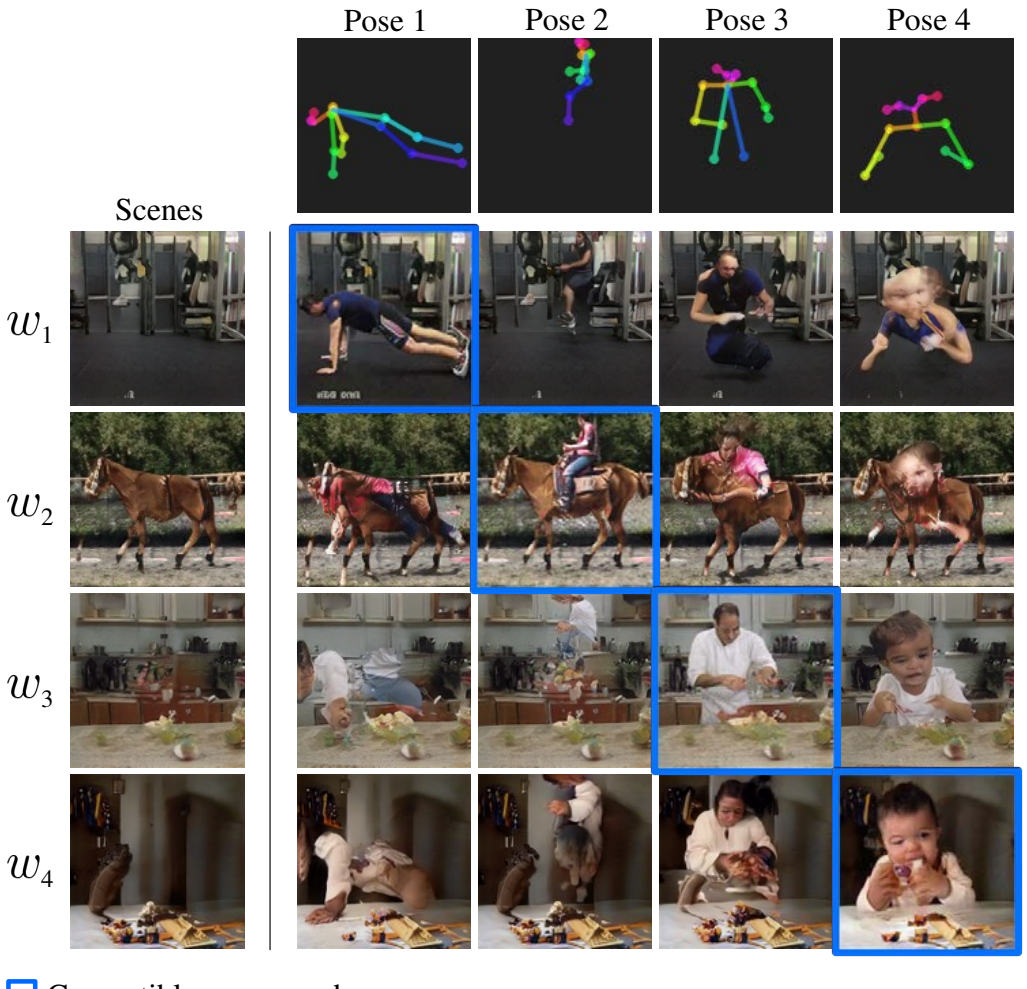
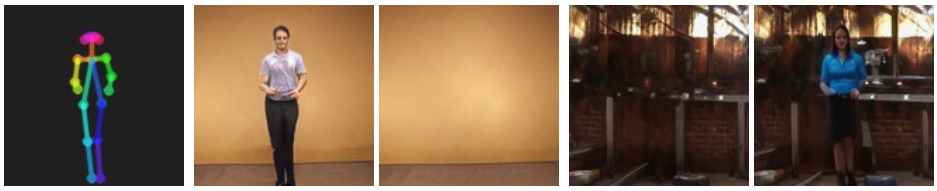
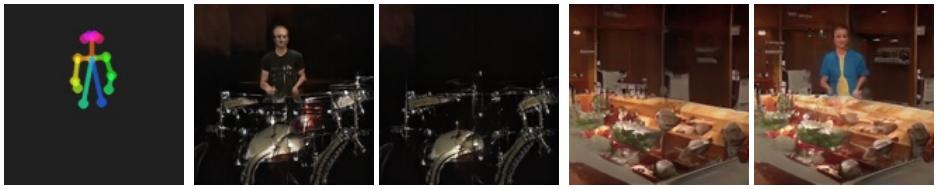


Figure 3.8: A central theme of this work is that scenes must be compatible with human poses to produce realistic images — here we visualize what happens when scenes and poses are *not* compatible. Correctly paired images are shown in blue on the diagonal — a person doing a pushup in a gym, riding a horse, cooking in a kitchen, and a baby leaning on a table. These exemplify interesting relationships between human pose and scene learned by our model. Other images mix scene latent codes with keypoint heatmaps from the wrong pose, often producing unrealistic images. Generating pose-compatible scenes is essential to avoid these incorrect pairings.



(a) Scenes generated with a full-body input pose.



(b) Scenes generated from the same input pose with legs not visible.

Figure 3.9: (a) A full-body input pose and corresponding scenes. (b) When the legs from an otherwise identical pose are hidden, our model hallucinates scenes with foreground objects, such as a drum kit or table, to occlude the missing legs.

3.6.2 Scene Occlusion Reasoning

Portions of a human pose may be occluded by foreground objects, such as pieces of furniture. Provided a partially visible human pose, our model hallucinates scenes with foreground objects to occlude portions of the pose not visible. Figure 3.9a shows an example full-body pose and output scenes. When the legs are not visible in the input pose in Figure 3.9b, our model produces scenes with occluders blocking the legs, demonstrating its emergent ability to reason about occlusions.

3.6.3 Human Appearance and Scene Disentanglement

Section 3.6.1 demonstrates why complete separation of pose and scene is undesirable. We can, however, disentangle human appearance from scene when both are conditioned on the same pose, as shown in Figure 3.10. We accomplish this by optimizing for a latent code which produces a scene matching one image and a subject matching another. To increase expressiveness of the latent space, we separately optimize the latent code used at each *scale* of our model, which is similar to the $\mathcal{W}^+$ space [141] used for inversion which has a separate latent for each *layer*. We minimize perceptual loss [62, 66] between a subject-only crop of the first generated image and the composition. When generating subject-only images, we zero out the learned constant input to the StyleGAN2 generator [13], which we found helps isolate the subject from the background. The crop region is attained from human pose. We also minimize perceptual loss between scene-only versions of the second generated image and composition image. We optimize for 1000 steps using the Adam optimizer [142] and a learning rate of 0.05.

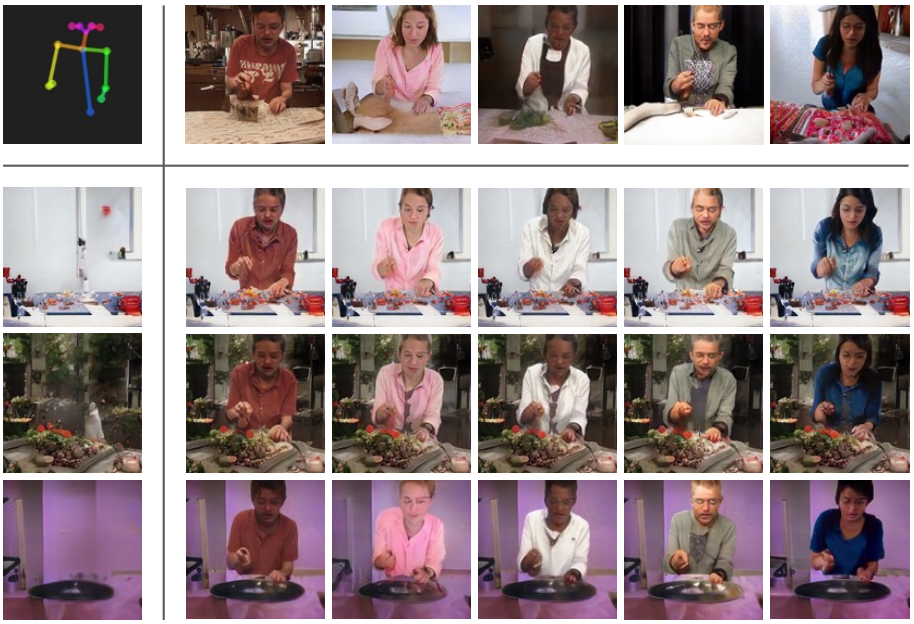


Figure 3.10: Given an input pose (top left), our method can compose human appearances (top row) and scenes (left column) from different generated images.

3.6.4 Animating Pose

After training, our model is capable of animating pose in a stationary scene. In Figure 3.11 we demonstrate a sequence of images generated by fixing the scene and animating the human pose. The scene is inferred from only the first pose, and is limited to small human motion and stationary backgrounds.

Figure 3.11: Provided an input pose sequence (top), we infer scenes based on the first pose, then generate animations (middle/bottom) by keeping the scene latent fixed and passing keypoint heatmaps for each subsequent pose.

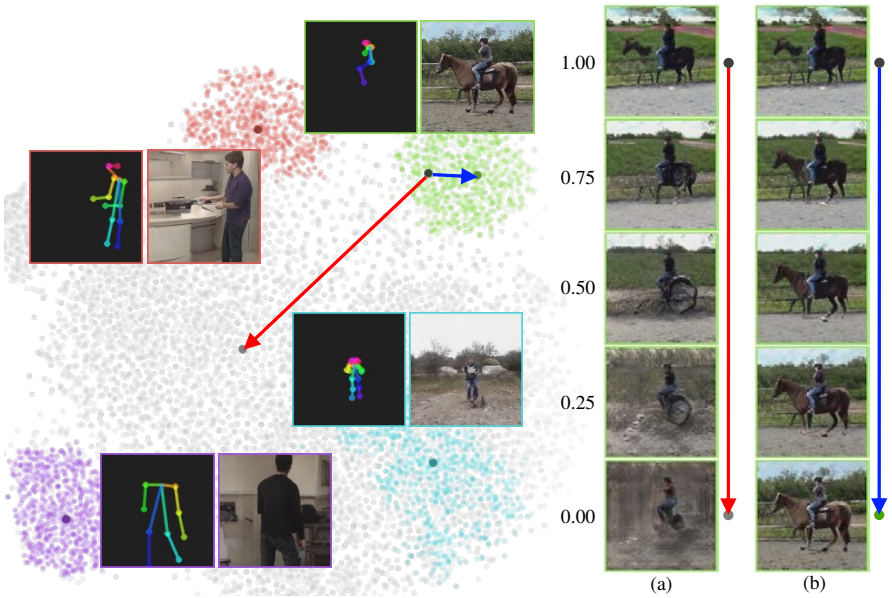


Figure 3.12: We contrast truncation via (a) interpolation toward the mean of random latents, and (b) interpolation toward the mean of conditional latent clusters. The left plot shows a t-SNE [143] visualization of latent codes. Gray points are 10,000 random latents. Colored sets of points are each 1000 latent samples conditioned on the same pose. The formation of clusters signifies that different scene latents conditioned on the same pose are close to each other in the intermediate latent space. The dark gray point in the center is the mean of all random latents, and dark colored points are the means for each pose. Beside each cluster is the input pose and image generated using the mean cluster latent. Conditional truncation (b) works significantly better than unconditional (a).

3.6.5 Scene Clustering and Truncation

Regions of low density in the data distribution are particularly challenging to model. Quality can be improved (at loss of some diversity) by sampling from a shrunk distribution [100, 144–148]. StyleGAN [14] interpolates intermediate latents w toward the mean $\bar{w} = \mathbb{E}_{z \sim \mathcal{Z}}[w]$ to shrink the distribution, which improves quality for models trained on data such as faces. However, on our more complex data, interpolating toward the mean scene latent produces a gray scene rather than improving quality (Figure 3.12a).

In visualizing a t-SNE [143] plot of scene latents in Figure 3.12, we observe that latents sampled from different noise vectors z yet conditioned on the same pose p form clusters. We apply conditional truncation by interpolating a latent w toward the conditional mean $\bar{w}_p = \mathbb{E}_{z \sim \mathcal{Z}}[w|p]$, shifting the sample toward the cluster center. Conditional truncation works significantly better for our model (Figure 3.12b). We apply conditional truncation $w' = \bar{w}_p + \psi(w - \bar{w}_p)$ of $\psi = 0.75$ to generated images throughout the chapter. Concurrent work [135] proposes a similar method truncation toward centers of perceptual clusters.

Pose Pix2Pix Pix2PixHD SG2 Ours

Figure 3.13: Baseline comparisons. Pix2Pix/Pix2PixHD struggle to produce realistic images; pose-conditioned StyleGAN2 (SG2) often generates humans in the wrong pose; our model generates realistic scenes with humans in the correct pose.

3.6.6 Baseline Comparisons

Please see Figure 3.13 for visual comparisons with baseline methods. Pix2Pix and Pix2PixHD were designed for image translation tasks with stronger conditioning, such as segmentation masks. These methods struggle to produce reasonable images on our more challenging task and dataset. StyleGAN2 (SG2) with latent pose conditioning provides a stronger baseline, but still has notable issues with image quality and often places humans in the incorrect pose. These observations are corroborated by metric performance in two respects: how accurately human subjects are positioned, and how realistic generated scenes look.

To succeed at our task, a model must both put a human in the correct pose and generate a compatible scene. Table 3.2 compares our model with Pix2Pix, Pix2PixHD and StyleGAN2 baselines on these metrics, demonstrating that our model achieves superior performance. Note that StyleGAN2 [13] is primarily an unconditional GAN. The public code release and follow-up work [137] support class-conditional generation. We refer to the version of our model with only pose latent conditioning as StyleGAN2, since it is the most straightforward extension of StyleGAN2 for our task.

Table 3.2: **Baseline metric comparisons.** We report PCKh (higher is better) as a measure of how accurately humans are positioned, and FID (lower is better) as a measure of how realistic generated scenes look. Our model outperforms Pix2Pix, Pix2PixHD and pose-conditioned StyleGAN2 baselines on both metrics. While the poor performance of baselines may appear surprising, note that our task is much more challenging than standard conditional generation tasks: the dataset is diverse and complex, and conditioning on pose requires the network to infer scene contents and layout.

	PCKh ↑	FID ↓
Pix2Pix	48.4	71.2
Pix2PixHD	73.8	149.7
StyleGAN2 (with pose latent conditioning)	32.4	16.6
Ours	84.2	5.9

Accurate human positioning. PCKh [121] measures the percent of correct pose keypoints (within a radius relative to the head size), where a higher percent is better. We use OpenPose [128] to extract poses from generated images for comparison with input poses. PCKh is computed on a held out test set, ensuring accurate placement of new poses not seen during training.

Realistic scene images. FID — Fréchet inception distance [149] — measures realism by comparing distributions of Inception network [150] features between the training dataset and generated images. Lower FID scores are better and correlate with higher quality, more realistic images.

3.6.7 Ablations

We present two ablation experiments. Table 3.3 enumerates changes relative to a pose-conditioned StyleGAN2 baseline, demonstrating improvements gained by our simple yet important modifications. Table 3.4 compares three options for pose conditioning: latents only, keypoint heatmaps only, and dual conditioning of both. Keypoint heatmaps are necessary to accurately position a human in the scene, which is shown by a substantially higher PCKh. Latent conditioning improves quality, which is shown by a lower FID score. We condition with both mechanisms — in addition to offering the best trade-off in metric performance, dual conditioning enables applications of disentanglement, such as generating scenes without humans or visualizing incompatible scenes and poses.

Table 3.3: **StyleGAN2 ablation.** We enumerate modifications relative to a pose-conditioned StyleGAN2 baseline. In particular, removing style mixing, conditioning on keypoint heatmaps, augmenting discriminator inputs and passing a fake mismatched example to the discriminator, and increasing scale all contribute to our final model.

	PCKh ↑	FID ↓
StyleGAN2 (with pose latent conditioning)	32.4	16.6
– style mixing	36.4	11.6
+ keypoint heatmaps	79.8	12.2
+ augmentation, mismatch	80.7	12.1
+ large scale (Ours)	84.2	5.9

Table 3.4: **Pose conditioning ablation.** We contrast three options for pose conditioning: only conditioning the latent on pose, only conditioning on keypoint heatmaps, and dual conditioning of both latents and heatmaps. We conduct this ablation on the smaller version of our model. We find that keypoint heatmap conditioning is crucial for accurately placing a human (PCKh), whereas latent conditioning improves the quality of scene generation (FID). We condition with both mechanisms in our final model, which has the best metric trade-off, and enables separating control of human position and scene generation after training.

Conditioning method	PCKh ↑	FID ↓
Latent only	36.4	11.6
Heatmap only	79.7	15.1
Both	79.8	12.2

3.7 Discussion

Limitations. Our dataset and model only consider images with a single human subject. Dataset curation is limited by the performance of Keypoint R-CNN [129, 130] and OpenPose [127, 128] when filtering videos for humans. Training depends on OpenPose to correctly predict poses. Our model does not consider human movement when inferring scenes.

Societal impact. There is some risk of this or future generative models being used to create fake and misleading content. Our model also inherits any demographic bias present in the existing datasets used to source our training data.

Follow-up work. In *Putting People in Their Place: Affordance-Aware Human Insertion into Scenes* by Kulal *et al.* [151], we publish follow-up work that builds on ideas presented in this chapter and introduces new methods. In that work, we train a generative model to insert humans into scenes given an image of a person, a separate image of a scene, and a region specifying where to place the person in the new scene. Similar to the method in this chapter, the model implicitly learns about relationships between people and their environment to synthesize a realistic image of a scene containing a person in a compatible pose. The follow-up work uses the *Humans in Context* dataset we present in this chapter as well as additional training data of humans in scenes. We also introduce a new method that uses two separate frames from a video to improve training supervision, leverages a more powerful diffusion generative model, and conditions the model to support a variety of additional use cases. The results substantially improve over results in this chapter as well as newer baseline methods.

Conclusion. In this chapter, we present a new task: provided a human pose as input, hallucinate the possible scene(s) which are compatible with that input pose. Strong relationships between humans, objects and environments dictate which scenes afford a given pose. Many prior works study human affordances from the angle of predicting possible poses given an input scene — we study the other side of the same coin, and hallucinate scenes that afford an input pose.

We demonstrate the emergent ability of our model to capture affordance relationships between scenes and poses. This work marks a significant step toward using generative models to represent complex real-world environments. We hope it will motivate the broader research community to leverage modern generative approaches for scene understanding and to utilize more complex visual training data.

3.8 Random Samples

Figures 3.14 3.15 3.16 3.17 3.18 contain random samples from our model.

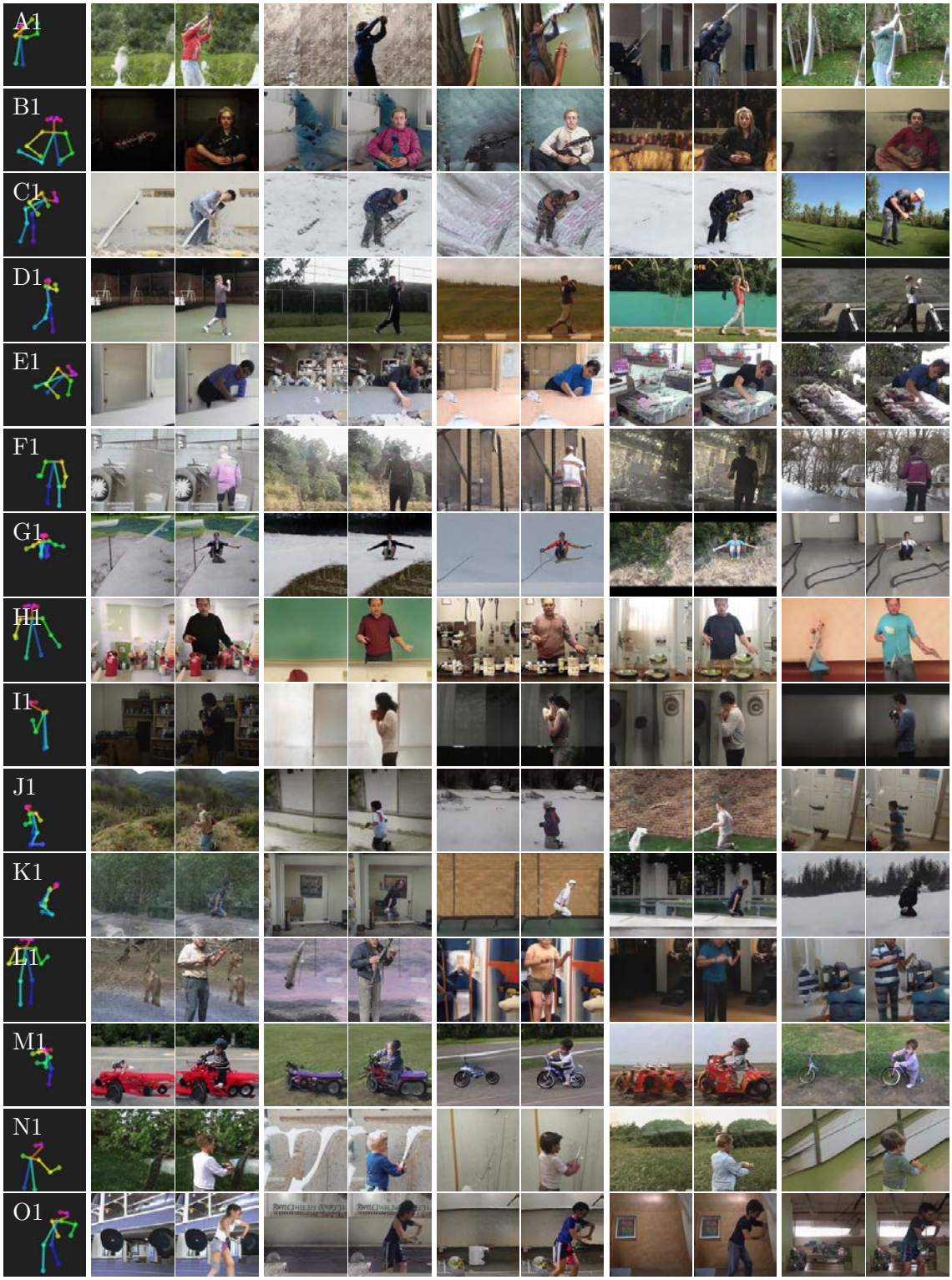


Figure 3.14: **Random samples.**

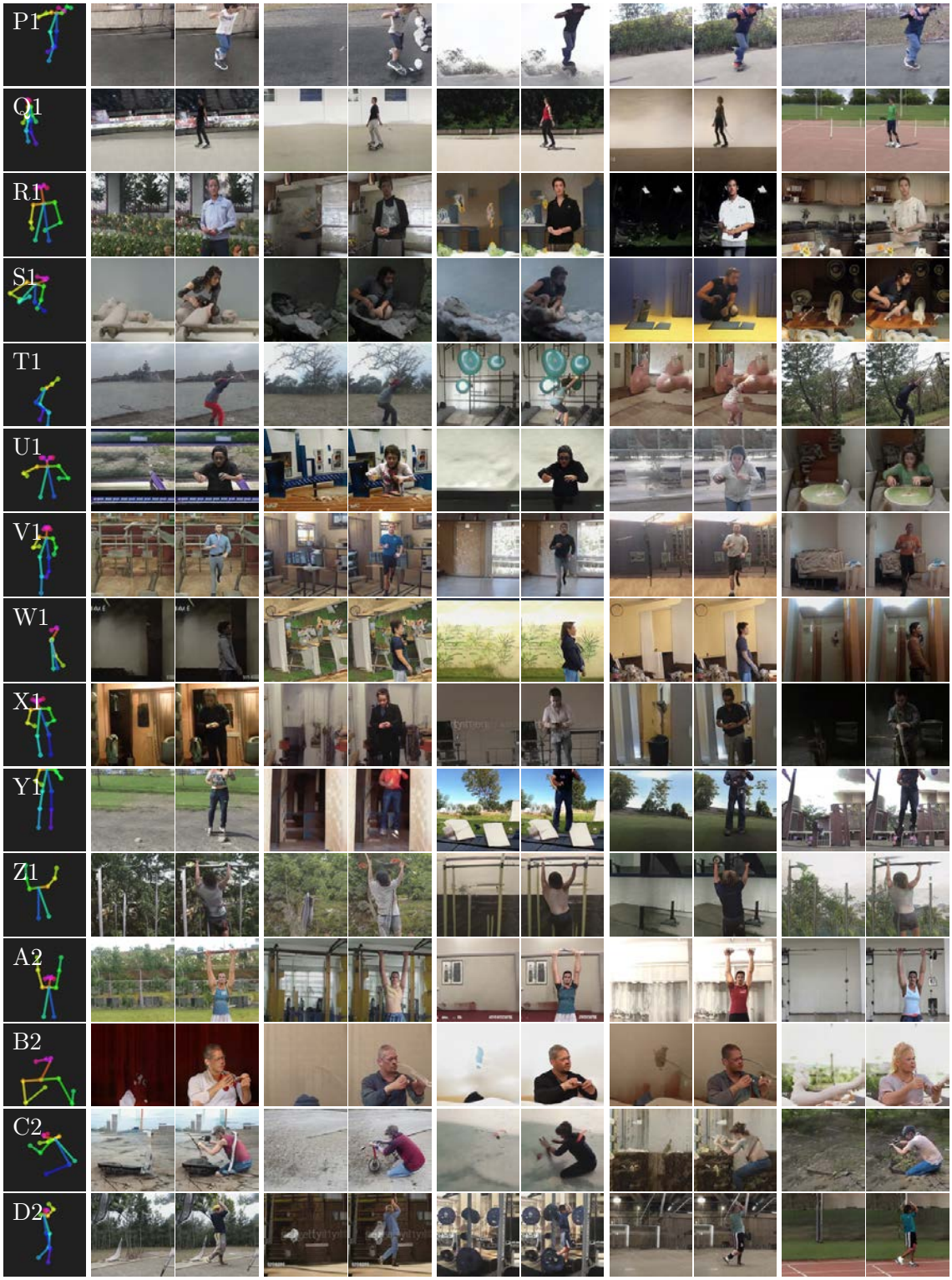


Figure 3.15: **Random samples (continued).**

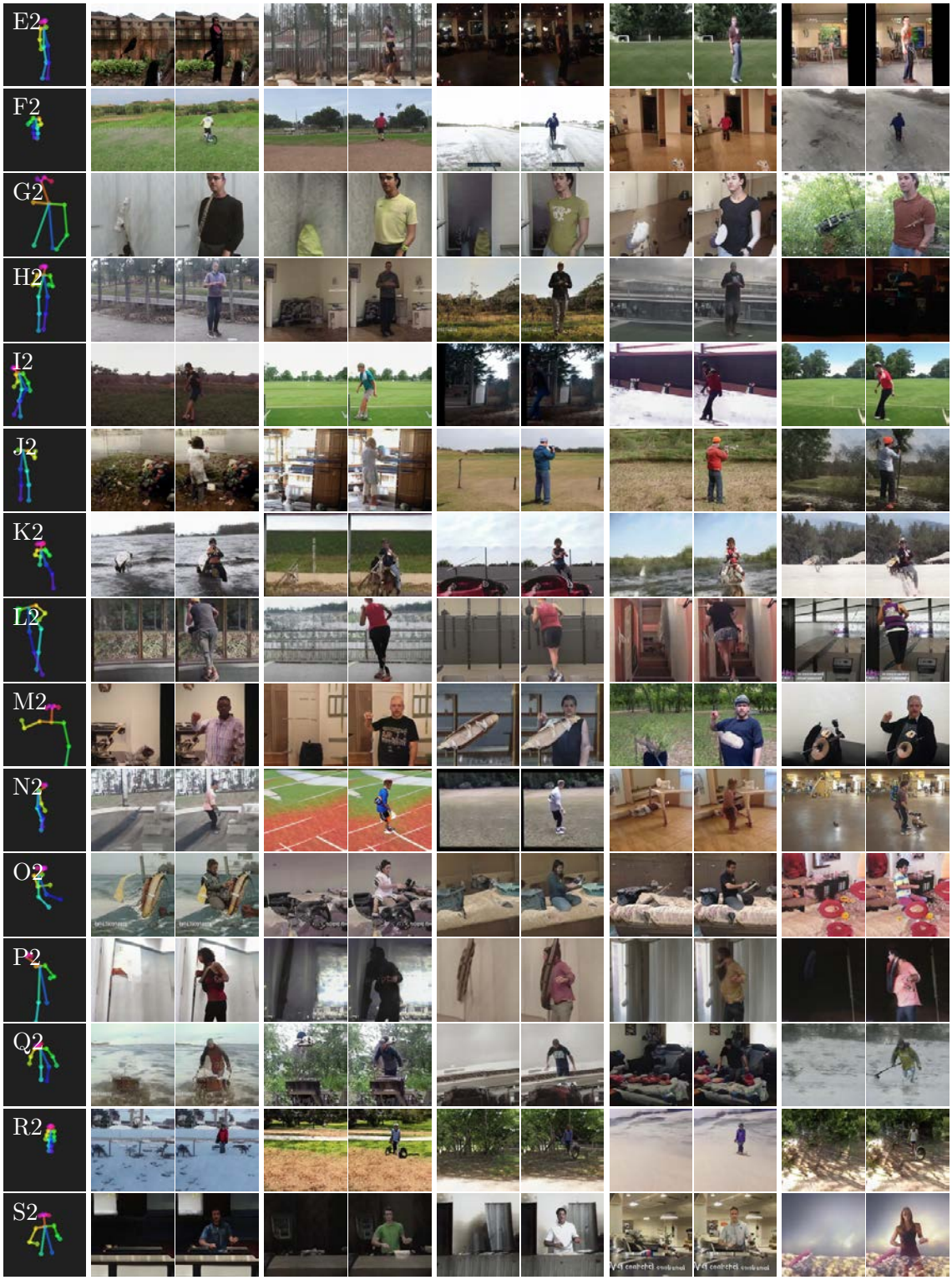


Figure 3.16: **Random samples (continued).**

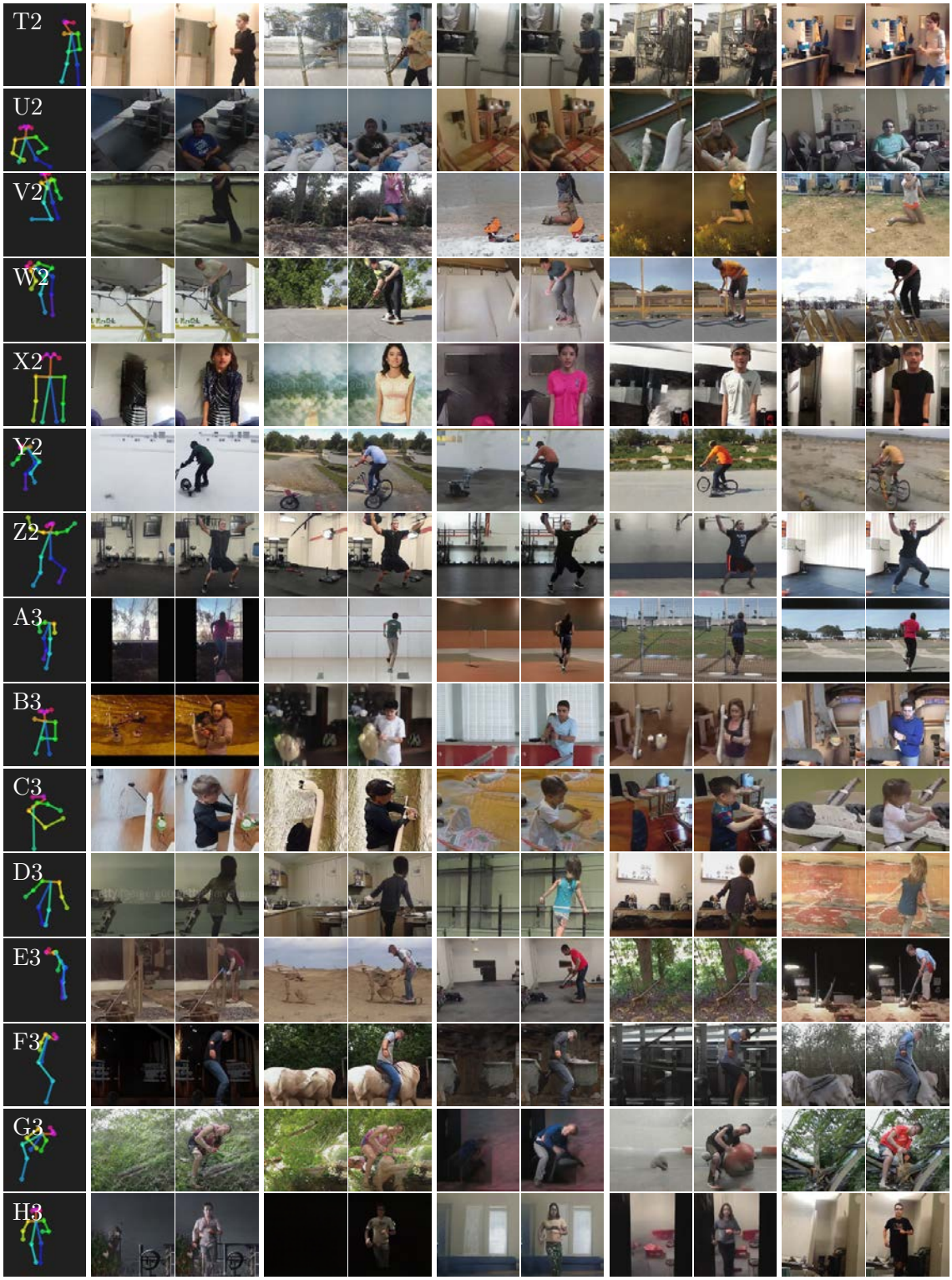


Figure 3.17: **Random samples (continued).**

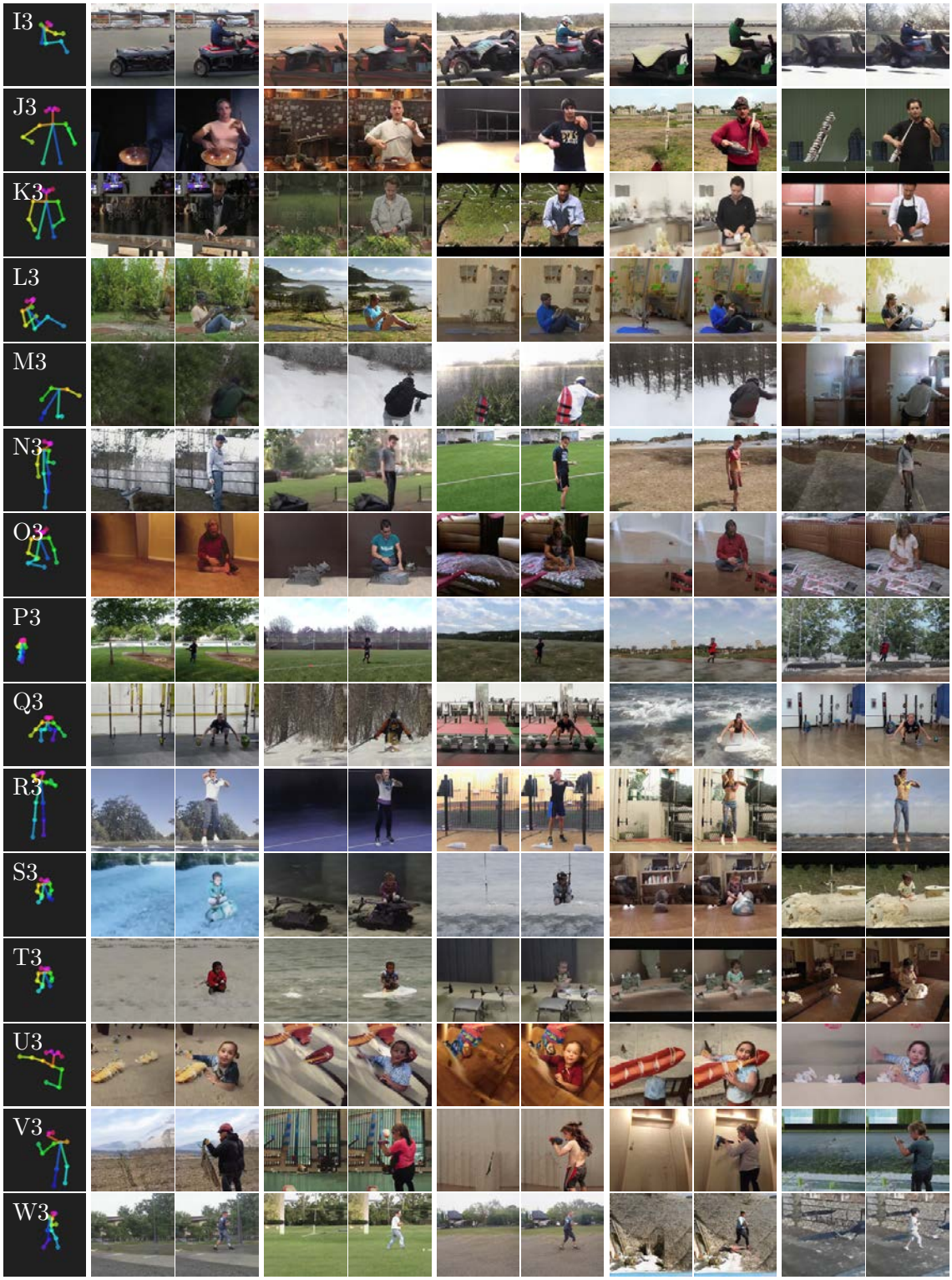


Figure 3.18: **Random samples (continued).**

Chapter 4

Learning to Follow Image Editing Instructions

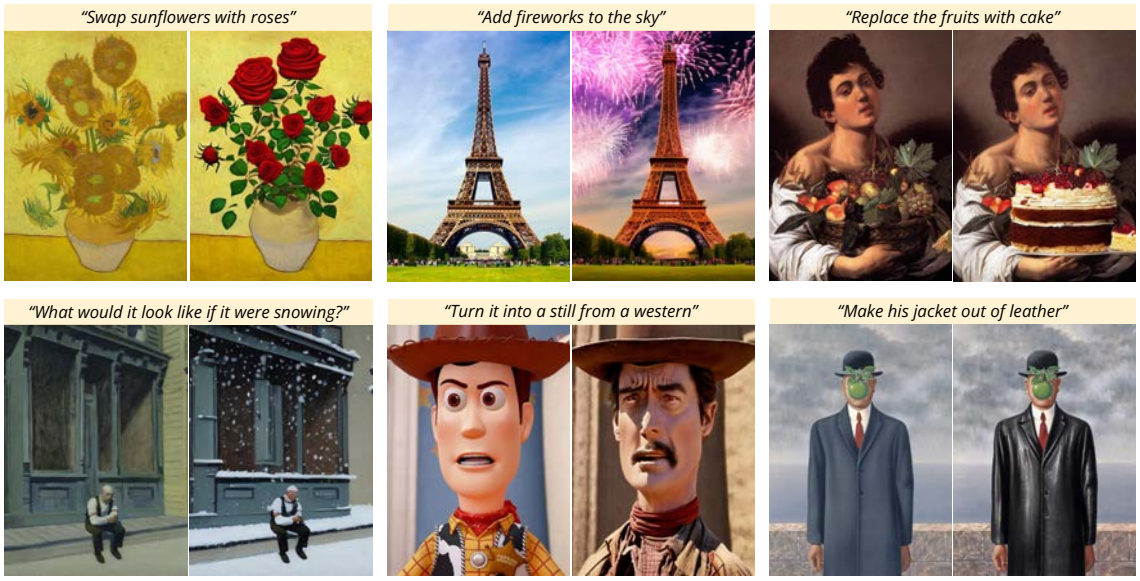


Figure 4.1: Given **an image** and **an instruction** for how to edit that image, our model performs the appropriate edit. Our model does not require full descriptions for the input or output image, and edits images in the forward pass without per-example inversion or fine-tuning.

The work presented in this chapter was first published in Brooks *et al.* as *InstructPix2Pix: Learning to Follow Image Editing Instructions* at the Conference on Computer Vision and Pattern Recognition (CVPR), 2023 [152].

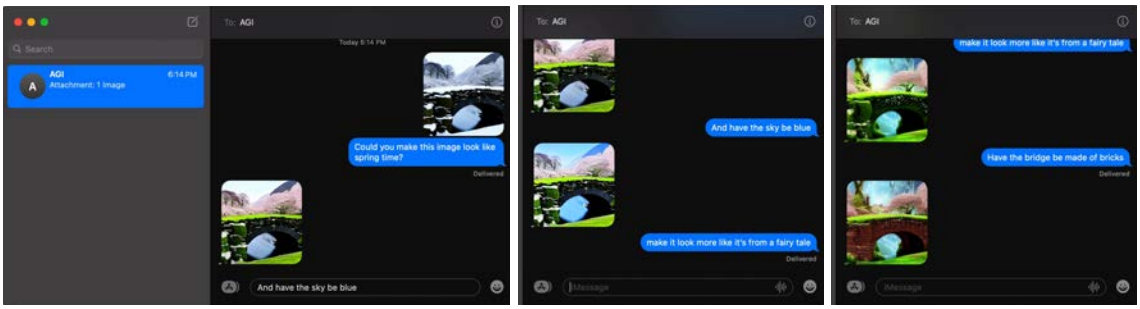


Figure 4.2: Instruction following is an important step toward more natural, conversational interaction with visual generative models. In this mock interface, we visualize using our image editing model in a text messaging conversation between a user and an AI assistant.

4.1 Introduction

This chapter presents a method for teaching a generative model to follow human-written instructions for image editing. See Figure 4.1 for example image edits performed by our model and Figure 4.2 for a mock interface using our model in a text messaging conversation. Teaching visual generative models to follow instructions is an important step toward making models that are more useful and easier to control.

Since training data for instruction-based image editing is difficult to acquire at scale, we propose an approach for generating a paired dataset that combines multiple large models pretrained on different modalities: a large language model (GPT-3 [20]) and a text-to-image model (Stable Diffusion [153]). These two models capture complementary knowledge about language and images that can be combined to create paired training data for a task spanning both modalities that neither model is capable of alone.

Using our generated paired data, we train a conditional diffusion model that, given an input image and a text instruction for how to edit it, generates the edited image. Our model directly performs the image edit in the forward pass, and does not require any additional example images, full descriptions of the input/output images, or per-example finetuning. Despite being trained entirely on synthetic examples (i.e., both generated written instructions and generated imagery), our model achieves zero-shot generalization to both arbitrary *real* images and natural human-written instructions. Our model enables intuitive image editing that can follow human instructions to perform a diverse collection of edits: replacing objects, changing the style of an image, changing the setting, the artistic medium, among others.

4.2 Prior Work

Composing large pretrained models Recent work has shown that large pretrained models can be combined to solve multimodal tasks that no one model can perform alone,

such as image captioning and visual question answering (tasks that require the knowledge of both a large language model and a text-image model). Techniques for combining pretrained models include joint finetuning on a new task [154–157], communication through prompting [158, 159], composing probability distributions of energy-based models [160, 161], guiding one model with feedback from another [162], and iterative optimization [163]. Our method is similar to prior work in that it leverages the complementary abilities of two pretrained models—GPT-3 [20]) and Stable Diffusion [153]—but differs in that we use these models to generate paired multi-modal training data.

Diffusion-based generative models Recent advances in diffusion models [164] have enabled state-of-the-art image synthesis [53, 165–169] as well as generative models of other modalities such as video [21, 170], audio [171], text [172] and network parameters [173]. Recent text-to-image diffusion models [153, 174–176] have shown to generate realistic images from arbitrary text captions.

Generative models for image editing Image editing models traditionally targeted a single editing task such as style transfer [177, 178] or translation between image domains [179–183]. Numerous editing approaches invert [184–187] or encode [109, 188, 189] images into a latent space (e.g., StyleGAN [13, 14]) where they can be edited by manipulating latent vectors. Recent models have leveraged CLIP [190] embeddings to guide image editing using text [174, 191–197]. We compare with one of these methods, Text2Live [198], an editing method that optimizes for an additive image layer that maximizes a CLIP similarity objective.

Recent works have used pretrained text-to-image diffusion models for image editing [175, 196, 199–201]. While some text-to-image models natively have the ability to edit images (e.g., DALLE-2 can create variations of images, inpaint regions, and manipulate the CLIP embedding [175]), using these models for *targeted* editing is non-trivial, because in most cases they offer no guarantees that similar text prompts will yield similar images. Recent work by Hertz *et al.* [201] tackles this issue with Prompt-to-Prompt, a method for assimilating the generated images for similar text prompts, such that isolated edits can be made to a generated image. We use this method in generating training data. To edit non-generated (i.e., real) imagery, SDEdit [199] uses a pretrained model to noise and denoise an input image with a new target prompt. We compare with SDEdit as a baseline. Other recent works perform local inpainting given a caption and user-drawn mask [175, 196], generate new images of a specific object or concept learned from a small collection of images [202, 203], or perform editing by inverting (and fine-tuning) a single image, and subsequently regenerating with a new text description [200]. In contrast to these approaches, our model takes only a single image and an instruction for how to edit that image (i.e., not a full description of any image), and performs the edit directly in the forward pass without need for a user-drawn mask, additional images, or per-example inversion or finetuning.

Learning to follow instructions Our method differs from existing text-based image editing works [198–203] in that it enables editing from *instructions* that tell the model what action to perform, as opposed to text labels, captions or descriptions of input/output images. A key benefit of following editing instructions is that the user can just tell the model exactly what to do in natural written text. There is no need for the user to provide extra information, such as example images or descriptions of visual content that remains constant between the input and output images. Instructions are expressive, precise, and intuitive to write, allowing the user to easily isolate specific objects or visual attributes to change. Our goal to follow written image editing instructions is inspired by recent work teaching large language models to better follow human instructions for language tasks [204–206].

Training data generation with generative models Deep models typically require large amounts of training data. Internet data collections are often suitable, but may not exist in the form necessary for supervision, e.g., paired data of particular modalities. As generative models continue to improve, there is growing interest in their use as a source of cheap and plentiful training data for downstream tasks [207–212]. In this paper, we use two different off-the-shelf generative models (language, text-to-image) to produce training data for our editing model.

4.3 Method

We treat instruction-based image editing as a supervised learning problem: (1) first, we generate a paired training dataset of text editing instructions and images before/after the edit (Sec. 4.3.1, Fig. 4.3a-c), then (2) we train an image editing diffusion model on this generated dataset (Sec. 4.3.2, Fig 4.3d). Despite being trained with generated images and editing instructions, our model is able to generalize to editing *real* images using arbitrary human-written instructions. See Fig. 4.3 for an overview of our method.

4.3.1 Generating a Multi-modal Training Dataset

We combine the abilities of two large-scale pretrained models that operate on different modalities—a large language model [20] and a text-to-image model [153]—to generate a multi-modal training dataset containing text editing instructions and the corresponding images before and after the edit. We describe in detail the two steps of this process. First, we describe the process of fine-tuning GPT-3 [20] to generate a collection of text edits: given a prompt describing an image, produce a text instruction describing a change to be made and a prompt describing the image after that change (Figure 4.3a). Then, we describe the process of converting the two text prompts (i.e., before and after the edit) into a pair of corresponding images using a text-to-image model [153] (Figure 4.3b).

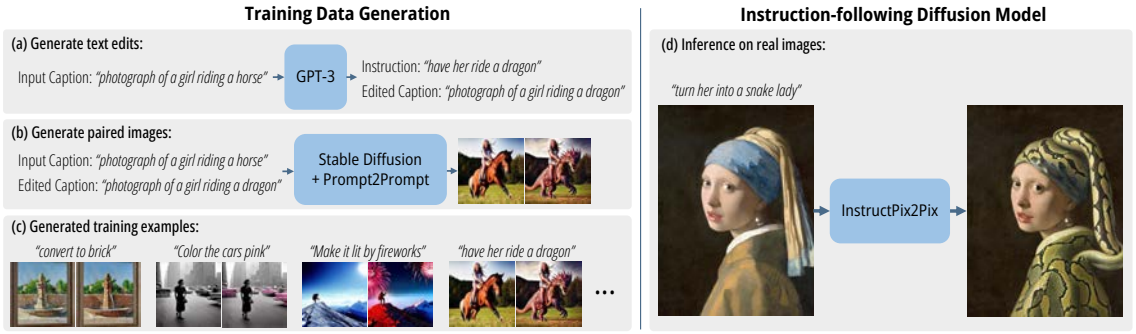


Figure 4.3: Our method consists of two parts: generating an image editing dataset, and training a diffusion model on that dataset. (a) We first use a finetuned GPT-3 to generate instructions and edited captions. (b) We then use StableDiffusion [153] in combination with Prompt-to-Prompt [201] to generate pairs of images from pairs of captions. We use this procedure to create a dataset (c) of over 450,000 training examples. (d) Finally, our InstructPix2Pix diffusion model is trained on our generated data to edit images from instructions. At inference time, our model generalizes to edit real images from human-written instructions.

Generating Instructions and Paired Captions

We first operate entirely in the text domain, where we leverage a large language model to take in image captions and produce editing instructions and the resulting text captions after the edit. For example, as shown in Figure 4.3a, provided the input caption *"photograph of a girl riding a horse"*, our language model can generate both a plausible edit instruction *"have her ride a dragon"* and an appropriately modified output caption *"photograph of a girl riding a dragon"*. Operating in the text domain enables us to generate a large and diverse collection of edits, while maintaining correspondence between the image changes and text instructions.

Our model is trained by finetuning GPT-3 on a relatively small human-written dataset of editing triplets: (1) input captions, (2) edit instructions, (3) output captions. To produce the fine-tuning dataset, we sampled 700 input captions from the LAION-Aesthetics V2 6.5+ [213] dataset and manually wrote instructions and output captions. See Table 4.1a for examples of our written instructions and output captions. Using this data, we fine-tuned the GPT-3 Davinci model for a single epoch using the default training parameters.

Benefiting from GPT-3's immense knowledge and ability to generalize, our finetuned model is able to generate creative yet sensible instructions and captions. See Table 4.1b for example GPT-3 generated data. Our dataset is created by generating a large number of edits and output captions using this trained model, where the input captions are real image captions from LAION-Aesthetics (excluding samples with duplicate captions or duplicate image URLs). We chose the LAION dataset due to its large size, diversity of

	Input LAION caption	Edit instruction	Edited caption
Human-written (700 edits)	*Yefim Volkov, Misty Morning*	*make it afternoon*	*Yefim Volkov, Misty Afternoon*
	girl with horse at sunset	*change the background to a city*	*girl with horse at sunset in front of city*
	painting-of-forest-and-pond	*Without the water.*	*painting-of-forest*
	...	...	...
GPT-3 generated (>450,000 edits)	*Alex Hill, Original oil painting on canvas, Moonlight Bay*	*in the style of a coloring book*	*Alex Hill, Original coloring book illustration, Moonlight Bay*
	The great elf city of Rivendell, sitting atop a waterfall as cascades of water spill around it	*Add a giant red dragon*	*The great elf city of Rivendell, sitting atop a waterfall as cascades of water spill around it with a giant red dragon flying overhead*
	Kate Hudson arriving at the Golden Globes 2015	*make her look like a zombie*	*Zombie Kate Hudson arriving at the Golden Globes 2015*
	...	...	...

Table 4.1: We label a small text dataset, finetune GPT-3, and use that finetuned model to generate a large dataset of text triplets. As the input caption for both the labeled and generated examples, we use real image captions from LAION. Highlighted text is generated by GPT-3.

content (including references to proper nouns and popular culture), and variety of mediums (photographs, paintings, digital artwork). A potential drawback of LAION is that it is quite noisy and contains a number of nonsensical or undescriptive captions—however, we found that dataset noise is mitigated through a combination of dataset filtering and classifier-free guidance (Section 4.3.2). Before filtering, our corpus of generated instructions and captions consists of $454,445$ examples.

Generating Paired Images from Paired Captions

Next, we use a pretrained text-to-image model to transform a pair of captions (referring to the image before and after the edit) into a pair of images. One challenge in turning a pair of captions into a pair of corresponding images is that text-to-image models provide no guarantees about image consistency, even under very minor changes of the conditioning prompt. For example, two very similar prompts: *"a picture of a cat"* and *"a picture of a black cat"* may produce wildly different images of cats. This is unsuitable for our purposes, where we intend to use this paired data as supervision for training a model to edit images (and not produce a different random image). We therefore use Prompt-to-Prompt [201], a recent method aimed at encouraging multiple generations from a text-to-image diffusion model to be similar. This is done through borrowed cross attention weights in some number of denoising steps. Figure 4.4 shows a comparison of sampled images with and without Prompt-to-Prompt.

While this greatly helps assimilate generated images, different edits may require different amounts of change in image-space. For instance, changes of larger magnitude, such

(a) Without Prompt-to-Prompt. (b) With Prompt-to-Prompt.

Figure 4.4: Pair of images generated using StableDiffusion [153] with and without Prompt-to-Prompt [201]. For both, the corresponding captions are *"photograph of a girl riding a horse"* and *"photograph of a girl riding a dragon"*.

as those which change large-scale image structure (e.g., moving objects around, replacing with objects of different shapes), may require less similarity in the generated image pair. Fortunately, Prompt-to-Prompt has as a parameter that can control the similarity between the two images: the fraction of denoising steps p with shared attention weights. Unfortunately, identifying an optimal value of p from only the captions and edit text is difficult. We therefore generate 100 sample pairs of images per caption-pair, each with a random $p \sim \mathcal{U}(0.1, 0.9)$, and filter these samples by using a CLIP-based metric: the directional similarity in CLIP space as introduced by Gal *et al.* [194]. This metric measures the consistency of the change between the two images (in CLIP space) with the change between the two image captions. Performing this filtering not only helps maximize the diversity and quality of our image pairs, but also makes our data generation more robust to failures of Prompt-to-Prompt and Stable Diffusion.

4.3.2 InstructPix2Pix

We use our generated training data to train a conditional diffusion model that edits images from written instructions. We base our model on Stable Diffusion, a large-scale text-to-image latent diffusion model.

Diffusion models [164] learn to generate data samples through a sequence of denoising autoencoders that estimate the score [214] of a data distribution (a direction pointing toward higher density data). Latent diffusion [153] improves the efficiency and quality of diffusion models by operating in the latent space of a pretrained variational autoencoder [215] with encoder $\mathcal{E}$ and decoder $\mathcal{D}$. For an image x, the diffusion process adds noise to the encoded latent $z = \mathcal{E}(x)$ producing a noisy latent z_t where the noise level increases over timesteps $t \in T$. We learn a network ϵ_θ that predicts the noise added to the noisy latent z_t given image conditioning c_I and text instruction conditioning c_T. We minimize the latent diffusion objective in Equation 4.1.

$$L = \mathbb{E}_{\mathcal{E}(x),\mathcal{E}(c_I),c_T,\epsilon\sim\mathcal{N}(0,1),t}\Big[\|\epsilon - \epsilon_\theta(z_t, t, \mathcal{E}(c_I), c_T))\|_2^2\Big] \tag{4.1}$$

Wang *et al.* [216] show that fine-tuning a large image diffusion models outperforms training a model from scratch for image translation tasks, especially when paired training data is limited. We therefore initialize the weights of our model with a pretrained Stable Diffusion checkpoint, leveraging its vast text-to-image generation capabilities. To support image conditioning, we add additional input channels to the first convolutional layer, concatenating z_t and $\mathcal{E}(c_I)$. All available weights of the diffusion model are initialized from the pretrained checkpoints, and weights that operate on the newly added input channels are initialized to zero. We reuse the same text conditioning mechanism that was originally intended for captions to instead take as input the text edit instruction c_T. Additional training details are provided in the supplemental material.

Classifier-free Guidance for Two Conditionings

Classifier-free diffusion guidance [217] is a method for trading off the quality and diversity of samples generated by a diffusion model. It is commonly used in class-conditional and text-conditional image generation to improve the visual quality of generated images and to make sampled images better correspond with their conditioning. Classifier-free guidance effectively shifts probability mass toward data where an implicit classifier $p_\theta(c|z_t)$ assigns high likelihood to the conditioning c. The implementation of classifier-free guidance involves jointly training the diffusion model for conditional and unconditional denoising, and combining the two score estimates at inference time. Training for unconditional denoising is done by simply setting the conditioning to a fixed null value $c = \varnothing$ at some frequency during training. At inference time, with a guidance scale $s \geq 1$, the modified score estimate $\tilde{e_\theta}(z_t, c)$ is extrapolated in the direction toward the conditional $e_\theta(z_t, c)$ and away from the unconditional $e_\theta(z_t, \varnothing)$.

$$\tilde{e_\theta}(z_t, c) = e_\theta(z_t, \varnothing) + s \cdot (e_\theta(z_t, c) - e_\theta(z_t, \varnothing)) \tag{4.2}$$

For our task, the score network $e_\theta(z_t, c_I, c_T)$ has two conditionings: the input image c_I and text instruction c_T. We find if beneficial to leverage classifier-free guidance with respect to both conditionings. Liu *et al.* [161] demonstrate that a conditional diffusion model can compose score estimates from multiple different conditioning values. We apply the same concept to our model with two separate conditioning inputs. During training, we randomly set only $c_I = \varnothing_I$ for 5% of examples, only $c_T = \varnothing_T$ for 5% of examples, and both $c_I = \varnothing_I$ and $c_T = \varnothing_T$ for 5% of examples. Our model is therefore capable of conditional or unconditional denoising with respect to both or either conditional inputs. We introduce two guidance scales, s_I and s_T, which can be adjusted to trade off how strongly the generated samples correspond with the input image and how strongly they correspond

with the edit instruction. In Figure 4.5, we show the effects of these two parameters on generated samples. Our modified score estimate is expressed by Equation 4.3.

$$\begin{aligned}\tilde{e_\theta}(z_t, c_I, c_T) = e_\theta(z_t, \varnothing, \varnothing) \\ + s_I \cdot (e_\theta(z_t, c_I, \varnothing) - e_\theta(z_t, \varnothing, \varnothing)) \\ + s_T \cdot (e_\theta(z_t, c_I, c_T) - e_\theta(z_t, c_I, \varnothing))\end{aligned} \tag{4.3}$$

Explanation of Equation 4.3:

Our generative model learns $P(z|c_I, c_T)$, the probability distribution of image latents $z = \mathcal{E}(x)$ conditioned on an input image c_I and a text instruction c_T. We arrive at the classifier-free guidance formulation in Equation 4.3 by expressing the conditional probability as follows:

$$P(z|c_T, c_I) = \frac{P(z, c_T, c_I)}{P(c_T, c_I)} = \frac{P(c_T|c_I, z)P(c_I|z)P(z)}{P(c_T, c_I)}$$

Diffusion models estimate the score [214] of the data distribution, i.e., the derivative of the log probability. Taking the logarithm gives us the following expression:

$$\begin{aligned}\log(P(z|c_T, c_I)) = \log(P(c_T|c_I, z)) + \log(P(c_I|z)) \\ + \log(P(z)) - \log(P(c_T, c_I))\end{aligned}$$

Taking the derivative and rearranging we attain:

$$\begin{aligned}\nabla_z \log(P(z|c_T, c_I)) = \nabla_z \log(P(z)) \\ + \nabla_z \log(P(c_I|z)) \\ + \nabla_z \log(P(c_T|c_I, z))\end{aligned}$$

This corresponds with the terms in our classifier-free guidance formulation in Equation 4.3. Our guidance scale s_I effectively shifts probability mass toward data where an implicit classifier $p_\theta(c_I|z_t)$ assigns high likelihood to the image conditioning c_I, and our guidance scale s_T effectively shifts probability mass toward data where an implicit classifier $p_\theta(c_T|c_I, z_t)$ assigns high likelihood to the text instruction conditioning c_T. Our model is capable of learning these implicit classifiers by taking the differences between estimates with and without the respective conditional input. Note there are multiple possible formulations such as switching the positions of c_T and c_I variables. We found that our particular decomposition works better for our use case in practice.

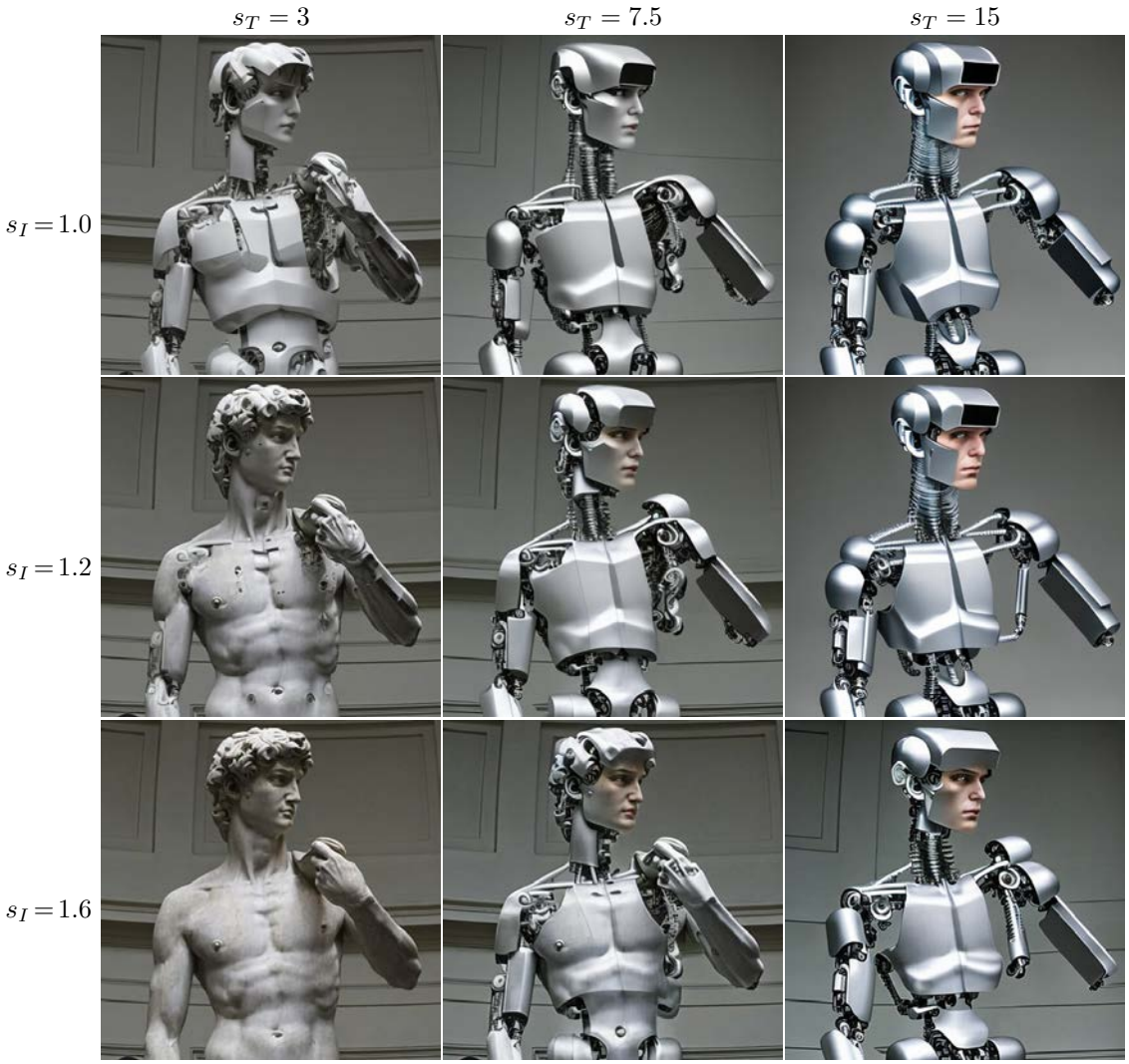


Figure 4.5: Michelangelo's *David* with classifier-free guidance weights over two conditional inputs. s_I controls similarity with the input image, while s_T controls consistency with the edit instruction. In practice, to attain the best results, it is necessary to tune these guidance weights for specific images and instructions. Results throughout this chapter tune s_I and s_T accordingly.

4.4 Implementation Details

4.4.1 Instruction and Caption Generation

We finetune GPT3 to generate edit instructions and edited captions. The text prompt used during fine-tuning is the input caption concatenated with `"\n##\n"` as a separator token. The text completion is a concatenation of the instruction and edited caption with `"\n%%\n"` as a separator token in between the two and `"\nEND"` appended to the end as the stop token. During inference, we sample text completions given new input captions using `temperature=0.7` and `frequency_penalty=0.1`. We exclude generations where the input and output captions are the same.

4.4.2 Paired Image Generation

We generate paired before/after training images from paired before/after captions using Stable Diffusion [153] in combination with Prompt-to-Prompt [201]. We use exponential moving average (EMA) weights of the Stable Diffusion v1.5 checkpoint and the improved ft-MSE autoencoder weights. We generate images with 100 denoising steps using an Euler ancestral sampler with denoising variance schedule proposed by Kerras *et al.* [218]. We ensure the same latent noise is used for both images in each generated pair (for initial noise as well as noise introduced during stochastic sampling).

Prompt-to-Prompt replaces cross-attention weights in the second generated image differently based on the specific edit type: word swap, adding a phrase, increasing or decreasing weight of a word. We instead replaced *self*-attention weights of the second image for the first p fraction of steps, and use the same attention weight replacement strategy for all edits.

We generate 100 pairs of images for each pair of captions. We filter training data for an image-image CLIP threshold of 0.75 to ensure images are related, an image-caption CLIP threshold of 0.2 to ensure images correspond with their captions, and a directional CLIP similarity of 0.2 to ensure the change in before/after captions correspond with the change in before/after images. For each pair of captions, we sort any image pairs that pass all filters by the directional CLIP similarity and keep up to 4 examples.

4.4.3 Training InstructPix2Pix

We train our image editing model for 10,000 steps on 8× 40GB NVIDIA A100 GPUs over 25.5 hours. We train at 256×256 resolution with a total batch size of 1024. We apply random horizontal flip augmentation and crop augmentation where images are first resized randomly between 256 and 288 pixels and then cropped to 256. We use a learning rate of 10^{-4} (without any learning rate warm up). We initialize our model from EMA weights of the Stable Diffusion v1.5 checkpoint, and adopt other training settings from the public Stable Diffusion code base.

Input | *"Make it a Modigliani painting"* | *"Make it a Miro painting"*

"Make it a Matisse painting" | *"Make it an Egyptian sculpture"* | *"Make it a marble roman sculpture"*

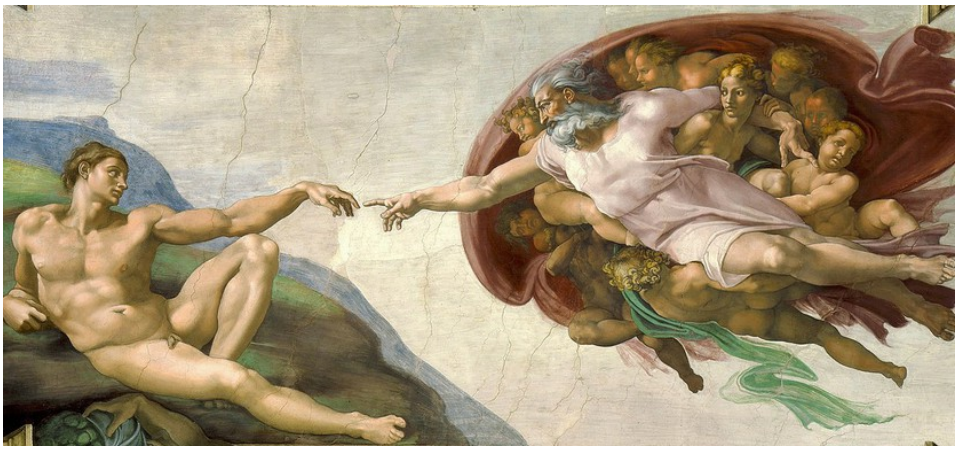
Figure 4.6: Leonardo da Vinci's *Mona Lisa* transformed into various styles and mediums.

While our model was trained at 256 resolution, we find it generalizes well to 512 resolution at inference time, and generate results in this paper at 512 resolution with 100 denoising steps using an Euler ancestral sampler with noise schedule proposed by Kerras *et al.* [218]. Editing an image with our model takes roughly 9 seconds on an A100 GPU.

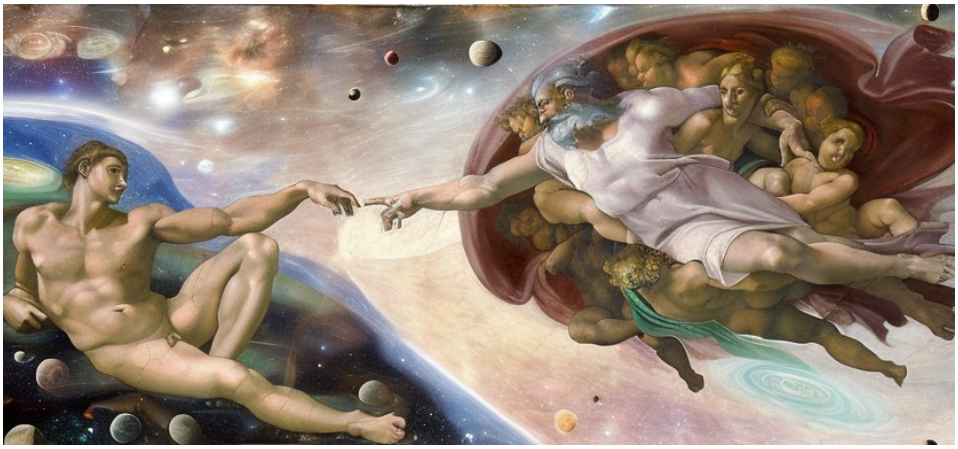
4.5 Results

We show instruction-based image editing results on a diverse set of real photographs and artwork, for many edit types and instruction wordings. See Figures 4.1 and 4.6–4.15 for selected results. Our model successfully performs many challenging edits, including replacing objects, changing seasons and weather, replacing backgrounds, modifying material attributes, converting artistic medium, and a variety of others.

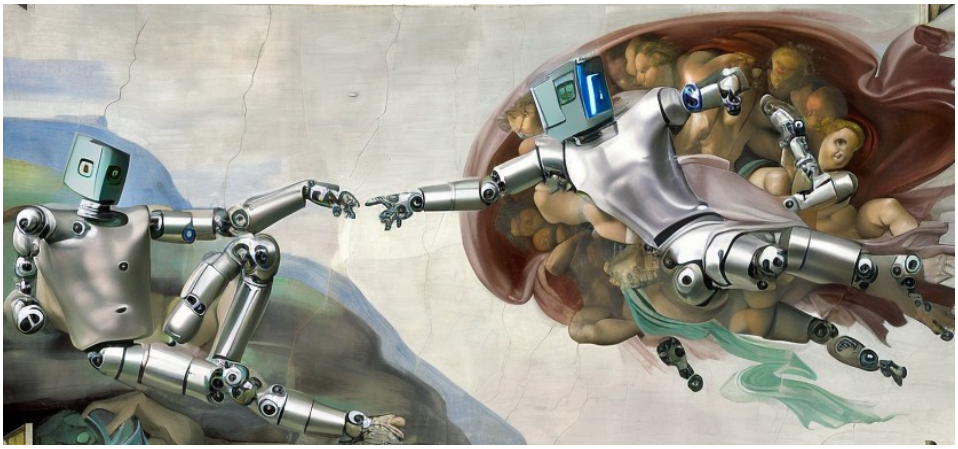
We compare our method qualitatively with recent works SDEdit [199], Text2Live [198], and Prompt-to-Prompt [201]. Our model follows instructions for how to edit the image, but prior works (including these baseline methods) expect descriptions of the image (or edit layer). Therefore, we provide them with the "after-edit" text caption instead of the edit instruction. We also compare our method quantitatively with SDEdit and Prompt-to-Prompt, using two metrics measuring image consistency and edit quality, further described in Section **??**. Finally, we show ablations on how the size and quality of generated training data affect our model's performance in Section 4.5.3.



Input



"Put them in outer space"



"Turn the humans into robots"

Figure 4.7: Michelangelo's *The Creation of Adam* with new context and subjects (768 resolution).

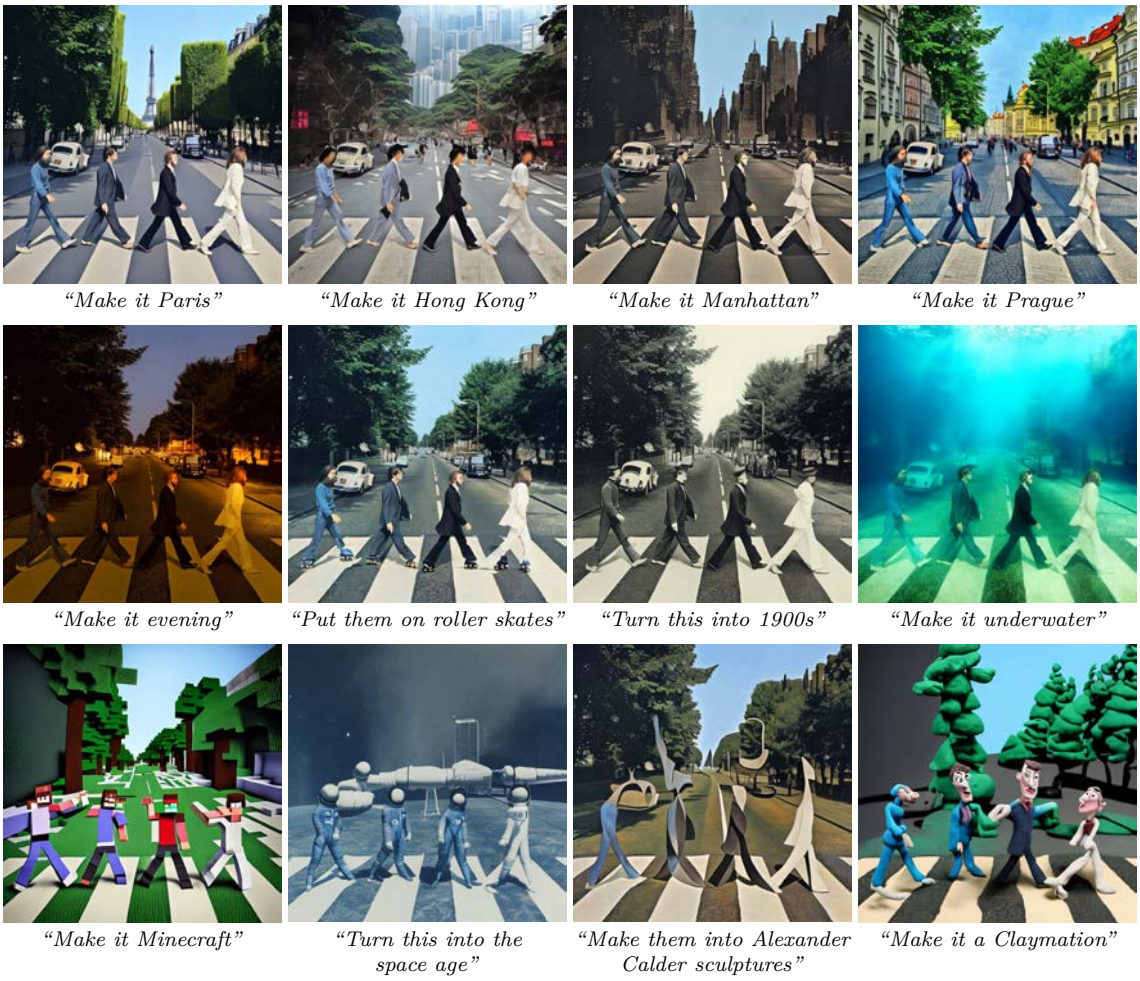


Figure 4.8: The iconic Beatles *Abbey Road* album cover transformed in a variety of ways.

Figure 4.9: Leighton's *Lady in a Garden* moved to a new setting.

Input *"Convert to a realistic photo"* *"Turn into a 3D model"*

Figure 4.10: Van Gogh's *Self-Portrait with a Straw Hat* in different mediums.

Input *"It is now midnight"* *"Add a beautiful sunset"*

Figure 4.11: A cityscape photograph changed to different times of day. *Photograph by Michael Pewny (edited version).*

Input *"Add boats on the water"* *"Replace the mountains with a city skyline"*

Figure 4.12: A landscape photograph edited to show accompanying contextual effects: the addition of boats also adds wind ripples in the water, and the added city skyline is reflected on the lake. *Photograph by Kordula Vahle.*

Input | *"Apply face paint"* | *"What would she look like as a bearded man?"* | *"Put on a pair of sunglasses"* | *"She should look 100 years old"*

"What if she were in an anime?" | *"Make her terrifying"* | *"Make her more sad"* | *"Make her James Bond"* | *"Turn her into Dwayne The Rock Johnson"*

Figure 4.13: Vermeer's *Girl with a Pearl Earring* with a variety of edits.

Figure 4.14: Applying our model iteratively produces compounded edits.

Input | *"in a race car video game"*

Figure 4.15: By varying the latent noise, our model can produce many possible image edits for the same input image and instruction.

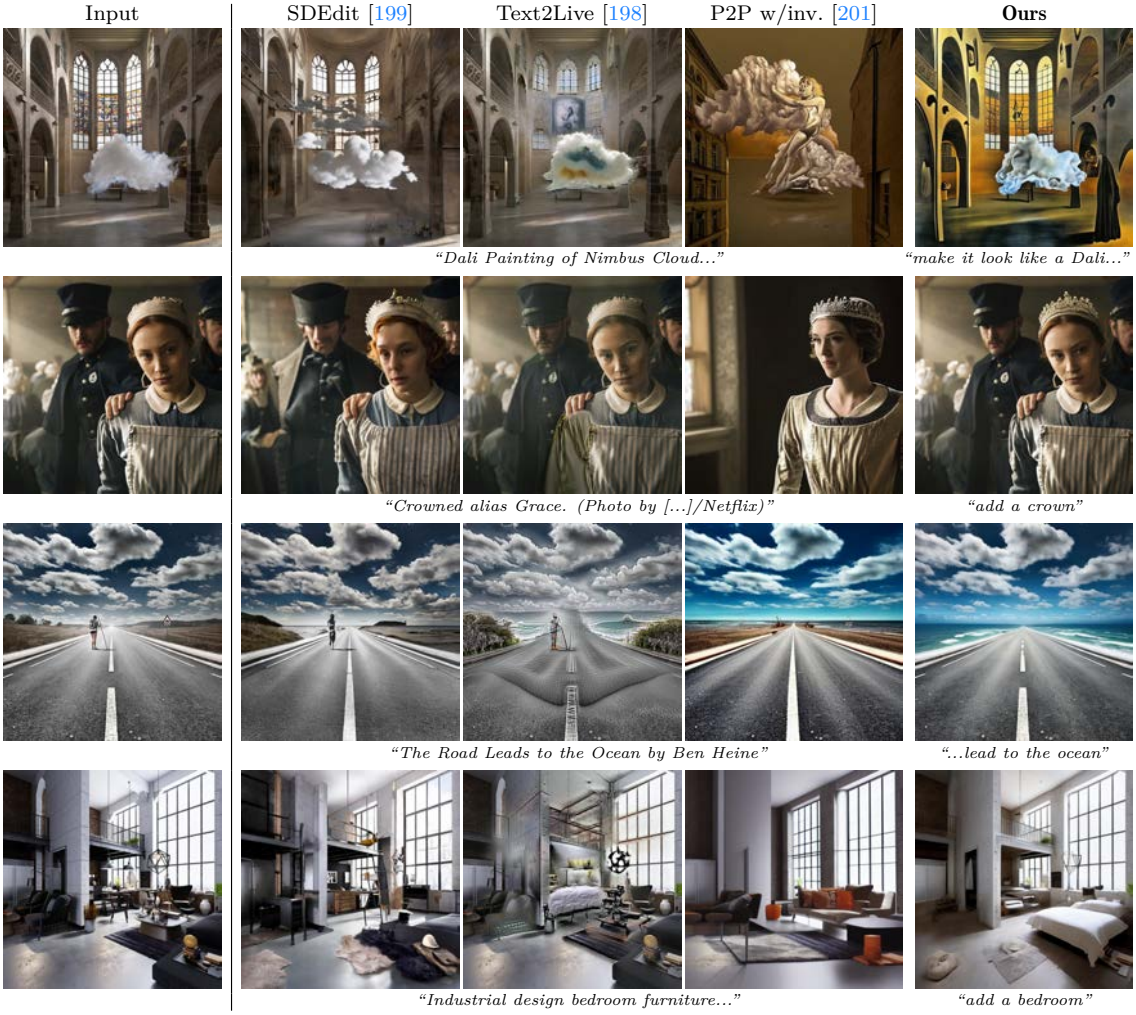


Figure 4.16: Comparisons with image editing approaches SDEdit [199], Text2Live [198], and Prompt-to-Prompt [201] with DDIM inversion [167, 219]. Unlike these methods, our model takes an editing instruction as text input.

4.5.1 Qualitative Baseline Comparisons

We provide qualitative comparisons with SDEdit [199], Text2Live [198], and Prompt-to-Prompt [201]. SDEdit [199] is a technique for editing images with a pretrained diffusion model, where a partially noised input image is denoised using a new caption to produce an edited image. Text2Live [198] edits images by generating an overlay layer, conditioned on text prompts. Prompt-to-Prompt [201] copies attention maps from one generated image to another during the denoising process; we use Prompt-to-Prompt when generating training data (Section 4.3.1), and while it is primarily designed to generate both before and after images, the method can be combined with DDIM inversion [167, 219] to edit real images.

See Figure 4.16 for qualitative comparisons. SDEdit struggles to preserve the identity of subjects or isolate individual objects. Text2Live works for edits that can be achieved by adding an overlay layer, but its formulation can not handle other categories of edits. Prompt-to-Prompt with DDIM inversion often makes additional unwanted changes to the image. Our model outperforms these baselines, and only needs an instruction as text input, whereas other methods require entire image captions (or specific edit layer prompts for Text2Live). We experimented with giving SDEdit edit instructions (instead of output captions) and giving Text2Live various combinations of input/output captions and edit instructions, however we did not observe any clear improvements in qualitative results.

Figure 4.17 compares with images taken directly from the Prompt-to-Prompt paper. In this case, Prompt-to-Prompt takes a pair of captions and generates both before and after images, sharing intermediate attention maps between them. Prompt-to-Prompt performs well when it generates both images. Our method performs comparably given only the "before" image and an instruction, and performs better at editing real images.

Figure 4.18 compares with images taken directly from the Text2Live paper. We prepend "make it" to prompts to use them as instructions. Text2Live performs well at these edits, yet is limited at other categories of edits. Our model is more general and can handle edits designed for the Text2Live method as well as other categories of edits.

Figure 4.17: Comparison on images from the Prompt-to-Prompt paper [201], where both before and after images are generated. Our model can perform comparably given only the before image and an edit instruction.

Figure 4.18: Comparison on images from the Text2Live paper [198].

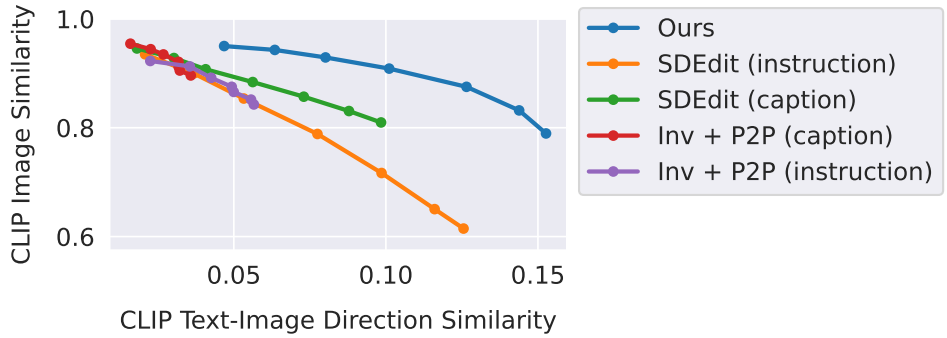


Figure 4.19: Plot of the tradeoff between CLIP similarity with input image (Y-axis) and directional CLIP similarity of edit (X-axis) using CLIP ViT-L/14. For both metrics, higher is better. We fix text guidance to 7.5, and vary: our method's $s_I \in [1.0, 2.2]$, SDEdit's strength (the amount of denoising) in $[0.3, 0.9]$, and Prompt-to-Prompt's cross-attention period in $[0, 1]$. We experiment with two variants of SDEdit and Prompt-to-Prompt, using either the output caption or edit instruction.

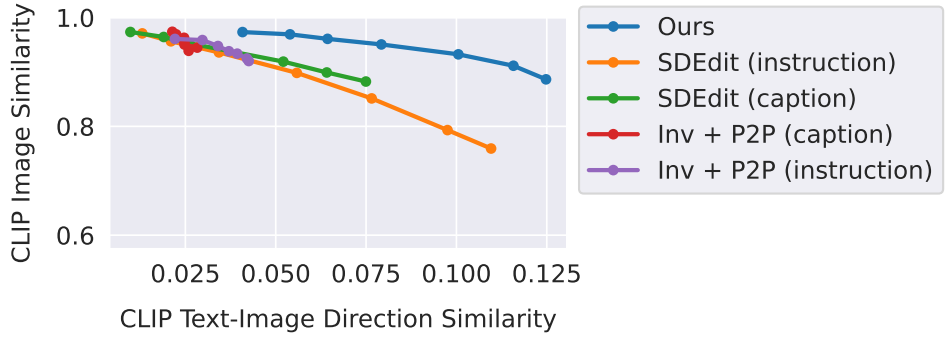


Figure 4.20: The same study as Figure 4.19, but using CLIP ViT-B/32.

4.5.2 Quantitative Baseline Comparisons

Quantitative comparisons with SDEdit and Prompt-to-Prompt are shown in Figures 4.19 and 4.20. We plot the tradeoff between two metrics, cosine similarity of CLIP image embeddings (how much the edited image agrees with the input image) and the directional CLIP similarity introduced by [194] (how much the change in text captions agrees with the change in the images). These are competing metrics—increasing the degree to which the output correspond to a desired edit will reduce its similarity with the input image—and we are interested in which method achieves the best tradeoff (highest curve).

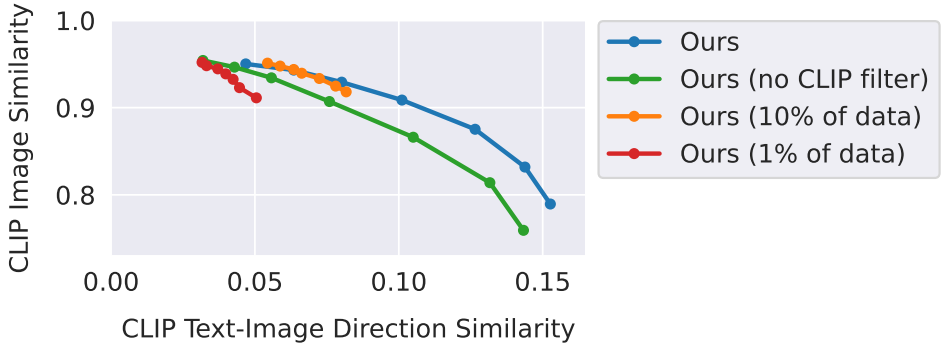


Figure 4.21: Comparison of models trained on ablated variants of our dataset (smaller subsets of dataset and no CLIP filtering) by fixing s_T and sweeping values of $s_I \in [1.0, 2.2]$.

We find that compared to SDEdit and Prompt-to-Prompt, our results achieve higher directional similarity for the same image similarity values, indicating it better performs the desired edit. Values are averaged across 2000 edits.

One potential source of bias in our evaluation protocol is the use of the same CLIP model both in evaluation and in our method (to filter training data, and to embed text conditioning in our model). To help assuage concern, we compare using two different CLIP models: ViT-L/14 in Figure 4.19 (same as used in our dataset and model) and ViT-B/32 in Figure 4.20. We find that results are consistent across these different CLIP models.

Outperforming Prompt-to-Prompt (both in quantitative metrics and qualitative results on real images) may seem counter-intuitive, since Prompt-to-Prompt is used to generate training data, however this may be for a number of reasons: (1) we train on CLIP-filtered examples, improving the quality of training data; (2) our method does not need DDIM inversion, which can cause errors; (3) we use a different classifier-free guidance formulation; (4) our model potentially benefits from training on a large dataset with many different image editing examples.

Our model takes roughly 9 seconds per edit on an A100 GPU. This is the same speed as SDEdit (although varying with number of diffusion steps) and twice as fast as Prompt-to-Prompt, since it requires DDIM inversion for real images. Text2Live takes ∼5min.

4.5.3 Ablations

In Figure 4.21, we provide ablations for both our dataset size and our dataset filtering approach described in Section 4.3.1. Decreasing the size of the dataset typically results in decreased ability to perform more significant image edits, instead only performing subtle or stylistic image adjustments (and thus, maintaining a high image similarity score, but a low directional score). Removing the CLIP filtering from our dataset generation reduces the overall consistency with the input image. We use CLIP ViT-L/14 for this plot.

Input | *"Make them look like flight attendants"* | *"Make them look like doctors"*

Figure 4.22: Example edit performed by our model that exhibits gender biases. Our model inherits biases from the data and models it is based upon. It is also possible that our model introduces additional biases.

"Zoom into the image" | *"Move it to Mars"* | *"Color the tie blue"* | *"Have the people swap places"*

Figure 4.23: Failure cases. Left to right: our model is not capable of performing viewpoint changes, can make undesired excessive changes to the image, can sometimes fail to isolate the specified object, and has difficulty reorganizing or swapping objects with each other.

4.6 Discussion

In this chapter, I present an approach that combines two large pretrained models, a large language model and a text-to-image model, to generate a dataset for training a diffusion model to follow written image editing instructions. While our method is able to produce a wide variety of compelling edits to images, including style, medium, and other contextual changes, there still remain a number of limitations.

For the best results, including those in this chapter, our model requires tuning classifier-free guidance weights for each example. Reducing this need is an important area for improvement. Our model is limited by the visual quality of training data, and therefore by the text-to-image model used to generate that data (in this case, Stable Diffusion [153]). Our method's ability to generalize to new edits and make correct associations between visual changes and instructions is limited by the human-written instructions used to fine-tune GPT-3 [20], by the ability of GPT-3 to write instructions and modify captions, and by the ability of Prompt-to-Prompt [201] to generated corresponding pairs of images.

Our model struggles in particular with counting numbers of objects and with spatial reasoning (e.g., *"move it to the left of the image"*, *"swap their positions"*, or *"put two cups on the table and one on the chair"*), just as in Stable Diffusion and Prompt-to-Prompt. Examples of failures can be found in Figure 4.23. We additionally find that performing many sequential edits sometimes causes accumulating artifacts. Furthermore, there are well-documented biases in the data and the pretrained models that our method uses. Images edited with our method may contain these biases or introduce others. See Figure 4.22 for an example edit performed by our model that exhibits gender biases.

Aside from mitigating the above limitations, our work also opens up questions, such as: how to follow instructions for spatial reasoning, how to combine instructions with other conditioning modalities like user interaction, how to enable edits that include context of a conversation with multiple rounds of instructions or multiple images, and how to evaluate instruction-based editing. Incorporating human feedback, such as with the use of reinforcement learning, is another important direction for future work and could improve alignment between our model and human intentions.

Chapter 5

Conclusions

In this thesis, I presented key components for improving the abilities and usefulness of visual generative models: capturing long-term patterns over time, learning from complex visual data, and teaching models to follow instructions. I proposed new methodologies that advanced visual generative models on each of these axes, as well as opened up several new research questions. Combining these components in future models will unlock new capabilities, such as performing the example task in Figure 5.1.

"Generate a video of a car chase scene, in the style of a James Bond movie, where the people in the first two images are escaping together while driving the car in the third image."

Figure 5.1: Example task for a future generative model that combines all components discussed in this thesis: generating long videos, modeling complex real-world visual data, and following written instructions.

In addition to the three components I focused on in this thesis, creating artificial superintelligence that outperforms human experts at arbitrary visual generation tasks will require other advancements. Scaling model size, training compute, and the amount of training data are crucial. This trend is well established for language models, and better understandings of scale will be paramount for visual generative models as well. It will be important to not only increase scale, but to study how the performance and tradeoffs among different models and design choices change at increasing scales.

Training data was an important aspect in all three contributions: training on long video data (Chapter 2), training on complex real-world image data (Chapter 3), and training on multimodal visual/language data (Chapter 4). It will be valuable to train on massive datasets that combine all of these and other modalities together – videos, images, language, audio, and likely other forms of information. In Chapter 4, I also generate training data using generative models. In that case, a large language model and a text-to-image model produced training data for an image editing task that neither model could perform alone. The success of this method suggests that using generative models to create training data will be an important source of data for us to better utilize in future models.

The method in Chapter 4 enables visual generative models to follow image editing instructions. In future models, it will be important to support more general conversation with models that generate visual data. In Figure 4.2, I visualized a mock interface for interacting with InstructPix2Pix via a text messaging conversation, hinting at the future uses of conversational visual generative models. Rather than following a single instruction, it will be valuable for future models to converse back-and-forth with a user, ask follow up questions, and understand arbitrary context of text and visual data when generating responses. Furthermore, just as alignment and learning from human feedback are essential for language models, it will be crucial to develop mechanisms for aligning visual generative models and for using human feedback to improve visual outputs.

Together the contributions I presented advanced the capabilities and usefulness of visual generative models for image and long video synthesis, as well as highlighted important directions for future research. Modeling long-term patterns over time, learning from complex visual data, and teaching models to follow instructions are key ingredients for building transformative generative models that will perform challenging visual tasks, be easy and intuitive to use, and enable anyone to make creative visual content.

Bibliography

[1] T. Brooks, J. Hellsten, M. Aittala, T.-C. Wang, T. Aila, J. Lehtinen, M.-Y. Liu, A. A. Efros, and T. Karras, "Generating long videos of dynamic scenes," *arXiv preprint arXiv:2206.03429*, 2022. 4

[2] A. Schödl, R. Szeliski, D. H. Salesin, and I. Essa, "Video textures," in *Proc. SIGGRAPH*, p. 489–498, 2000. 6

[3] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic textures," *International Journal of Computer Vision*, vol. 51, no. 2, pp. 91–109, 2003. 6

[4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014. 6, 33, 37, 39

[5] D. Acharya, Z. Huang, D. P. Paudel, and L. Van Gool, "Towards high resolution video generation with progressive growing of sliced wasserstein gans," *CoRR*, vol. abs/1810.02419, 2018. 6, 7

[6] M. Saito, E. Matsumoto, and S. Saito, "Temporal generative adversarial nets with singular value clipping," in *Proc. ICCV*, pp. 2830–2839, 2017. 6

[7] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *Proc. NIPS*, 2016. 6

[8] A. Clark, J. Donahue, and K. Simonyan, "Adversarial video generation on complex datasets," *CoRR*, vol. abs/1907.06571, 2019. 6

[9] G. Fox, A. Tewari, M. Elgharib, and C. Theobalt, "Stylevideogan: A temporal generative model using a pretrained stylegan," *CoRR*, vol. abs/2107.07224, 2021. 6

[10] Y. Tian, J. Ren, M. Chai, K. Olszewski, X. Peng, D. N. Metaxas, and S. Tulyakov, "A good image generator is what you need for high-resolution video synthesis," in *Proc. ICLR*, 2021. 6, 25

[11] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, "Mocogan: Decomposing motion and content for video generation," in *Proc. CVPR*, pp. 1526–1535, 2018. 6, 19

[12] I. Skorokhodov, S. Tulyakov, and M. Elhoseiny, "Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2," *CoRR*, vol. abs/2112.14683, 2021. 6, 12, 19, 21, 23, 27

[13] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8110–8119, 2020. 6, 9, 10, 11, 19, 25, 27, 32, 33, 37, 38, 39, 42, 44, 47, 58

[14] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019. 6, 11, 13, 14, 32, 33, 38, 42, 46, 58

[15] S. Yu, J. Tack, S. Mo, H. Kim, J. Kim, J.-W. Ha, and J. Shin, "Generating videos with dynamics-aware implicit generative adversarial networks," in *International Conference on Learning Representations (ICLR)*, 2022. 6, 19, 25

[16] S. Ge, T. Hayes, H. Yang, X. Yin, G. Pang, D. Jacobs, J.-B. Huang, and D. Parikh, "Long video generation with time-agnostic vqgan and time-sensitive transformer," *CoRR*, vol. abs/2204.03638, 2022. 6, 7, 19, 25

[17] R. Rakhimov, D. Volkhonskiy, A. Artemov, D. Zorin, and E. Burnaev, "Latent video transformer," *CoRR*, vol. abs/2006.10704, 2020. 6

[18] J. Walker, A. Razavi, and A. v. d. Oord, "Predicting video with vqvae," *CoRR*, vol. abs/2103.01950, 2021. 6

[19] W. Yan, Y. Zhang, P. Abbeel, and A. Srinivas, "Videogpt: Video generation using vq-vae and transformers," *CoRR*, vol. abs/2104.10157, 2021. 6, 7

[20] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020. 6, 57, 58, 59, 76

[21] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, "Video diffusion models," *arXiv preprint arXiv:2204.03458*, 2022. 7, 58

[22] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine, "Stochastic variational video prediction," in *Proc. ICLR*, 2018. 7

[23] N. Kalchbrenner, A. Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu, "Video pixel networks," in *Proc. ICML*, pp. 1771–1779, 2017. 7

[24] M. Kumar, M. Babaeizadeh, D. Erhan, C. Finn, S. Levine, L. Dinh, and D. Kingma, "Videoflow: A conditional flow-based model for stochastic video generation," in *Proc. ICLR*, 2020. 7

[25] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine, "Stochastic adversarial video prediction," *CoRR*, vol. abs/1804.01523, 2018. 7

[26] P. Luc, A. Clark, S. Dieleman, D. d. L. Casas, Y. Doron, A. Cassirer, and K. Simonyan, "Transformation-based adversarial video prediction on large-scale data," *CoRR*, vol. abs/2003.04035, 2020. 7

[27] C. Nash, J. Carreira, J. Walker, I. Barr, A. Jaegle, M. Malinowski, and P. Battaglia, "Transframer: Arbitrary frame prediction with generative models," *CoRR*, vol. abs/2203.09494, 2022. 7

[28] S. Chiappa, S. Racaniere, D. Wierstra, and S. Mohamed, "Recurrent environment simulators," in *Proc. ICLR*, 2017. 7

[29] D. Ha and J. Schmidhuber, "World models," *CoRR*, vol. abs/1803.10122, 2018. 7

[30] S. W. Kim, J. Philion, A. Torralba, and S. Fidler, "Drivegan: Towards a controllable high-quality neural simulation," in *Proc. CVPR*, pp. 5820–5829, 2021. 7

[31] S. W. Kim, Y. Zhou, J. Philion, A. Torralba, and S. Fidler, "Learning to Simulate Dynamic Environments with GameGAN," in *Proc. CVPR*, Jun. 2020. 7

[32] A. Liu, R. Tucker, V. Jampani, A. Makadia, N. Snavely, and A. Kanazawa, "Infinite nature: Perpetual view generation of natural scenes from a single image," in *Proc. ICCV*, 2021. 7, 17, 19, 24

[33] X. Ren and X. Wang, "Look outside the room: Synthesizing a consistent long-term 3d scene video from a single image," in *Proc. CVPR*, 2022. 7

[34] A. K. Akan, S. Safadoust, E. Erdem, A. Erdem, and F. Güney, "Stochastic video prediction with structure and motion," *CoRR*, vol. abs/2203.10528, 2022. 7

[35] R. Child, "Very deep vaes generalize autoregressive models and can outperform them on images," *arXiv preprint arXiv:2011.10650*, 2020. 7

[36] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *International Conference on Learning Representations*, 2018. 7, 33, 39

[37] A. Razavi, A. Van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with vq-vae-2," *Proc. NeurIPS*, vol. 32, 2019. 7

[38] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, "Image super-resolution via iterative refinement," *arXiv:2104.07636*, 2021. 7

[39] A. Vahdat and J. Kautz, "NVAE: A deep hierarchical variational autoencoder," in *Proc. NeurIPS*, 2020. 7

[40] M. Saito, S. Saito, M. Koyama, and S. Kobayashi, "Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal gan," *International Journal of Computer Vision*, vol. 128, no. 10, pp. 2586–2606, 2020. 7

[41] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6202–6211, 2019. 7

[42] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," in *Proc. NeurIPS*, 2020. 9, 11, 12, 14, 15

[43] S. Zhao, Z. Liu, J. Lin, J.-Y. Zhu, and S. Han, "Differentiable augmentation for data-efficient gan training," in *Proc. NeurIPS*, 2020. 9, 12

[44] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Alias-free generative adversarial networks," in *Proc. NeurIPS*, 2021. 9, 11, 12, 15

[45] M. Haris, G. Shakhnarovich, and N. Ukita, "Recurrent back-projection network for video super-resolution," in *Proc. CVPR*, pp. 3897–3906, 2019. 12

[46] A. Kappeler, S. Yoo, Q. Dai, and A. K. Katsaggelos, "Video super-resolution with convolutional neural networks," *IEEE transactions on computational imaging*, vol. 2, no. 2, pp. 109–122, 2016. 12

[47] M. S. M. Sajjadi, R. Vemulapalli, and M. Brown, "Frame-recurrent video super-resolution," in *Proc. CVPR*, 2018. 12

[48] X. Tao, H. Gao, R. Liao, J. Wang, and J. Jia, "Detail-revealing deep video super-resolution," in *Proc. ICCV*, 2017. 12

[49] J. F. Kaiser, "Nonrecursive digital filter design using the i_0-sinh window function," in *Proc. 1974 IEEE International Symposium on Circuits & Systems, San Francisco DA, April*, pp. 20–23, 1974. 12

[50] L. Mescheder, S. Nowozin, and A. Geiger, "Which training methods for gans do actually converge?," in *International Conference on Machine Learning (ICML)*, 2018. 13, 39

[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 13

[52] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *Cited on*, vol. 14, no. 8, p. 2, 2012. 13

[53] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans, "Cascaded diffusion models for high fidelity image generation.," *J. Mach. Learn. Res.*, vol. 23, pp. 47–1, 2022. 15, 58

[54] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Interspeech*, 2018. 16

[55] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics: A large-scale video dataset for forgery detection in human faces," *arXiv preprint arXiv:1803.09179*, 2018. 16

[56] K. Wang, Q. Wu, L. Song, Z. Yang, W. Wu, C. Qian, R. He, Y. Qiao, and C. C. Loy, "Mead: A large-scale audio-visual dataset for emotional talking-face generation," in *European Conference on Computer Vision (ECCV)*, 2020. 16

[57] T.-C. Wang, A. Mallya, and M.-Y. Liu, "One-shot free-view neural talking-head synthesis for video conferencing," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 16

[58] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," in *IEEE International Conference on Computer Vision (ICCV)*, 2013. 16

[59] A. Liu, R. Tucker, V. Jampani, A. Makadia, N. Snavely, and A. Kanazawa, "Infinite nature: Perpetual view generation of natural scenes from a single image," in *Proc. CVPR*, 2021. 16

[60] J. Zhang, C. Xu, L. Liu, M. Wang, X. Wu, Y. Liu, and Y. Jiang, "Dtvnet: Dynamic time-lapse video generation via single still image," in *European Conference on Computer Vision (ECCV)*, 2020. 16

[61] W. Xiong, W. Luo, L. Ma, W. Liu, and J. Luo, "Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks," in *Proc. CVPR*, pp. 2364–2373, 2018. 17, 24, 25, 27

[62] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018. 20, 26, 44

[63] T. Unterthiner, S. van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly, "Towards accurate generative models of video: A new metric & challenges," *arXiv preprint arXiv:1812.01717*, 2018. 20, 21, 27

[64] L. N. Vaserstein, "Markov processes over denumerable products of spaces, describing large systems of automata," *Problemy Peredachi Informatsii*, vol. 5, no. 3, pp. 64–72, 1969. 21, 28

[65] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *Proc. CVPR*, pp. 4724–4733, 2017. 21

[66] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*, pp. 694–711, Springer, 2016. 26, 44

[67] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012. 26

[68] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. 26

[69] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017. 27

[70] T. Kynkäänniemi, T. Karras, M. Aittala, T. Aila, and J. Lehtinen, "The role of imagenet classes in fréchet inception distance," *arXiv preprint arXiv:2203.06026*, 2022. 28

[71] T. Brooks and A. A. Efros, "Hallucinating pose-compatible scenes," in *European Conference on Computer Vision*, pp. 510–528, Springer, 2022. 31

[72] J. J. Gibson, *The Ecological Approach to Visual Perception*. Boston: Houghton Mifflin, 1979. 31, 33

[73] I. Biederman, "On the semantics of a glance at a scene.," in *Perceptual Organization*, 1981. 31, 33

[74] A. Gupta, S. Satkin, A. A. Efros, and M. Hebert, "From 3d scene geometry to human workspace," in *Computer Vision and Pattern Recognition(CVPR)*, 2011. 31, 33

[75] V. Delaitre, D. Fouhey, I. Laptev, J. Sivic, A. Gupta, and A. Efros, "Scene semantics from long-term observation of people," in *Proc. 12th European Conference on Computer Vision*, 2012. 31, 33

[76] D. F. Fouhey, X. Wang, and A. Gupta, "In defense of the direct perception of affordances," 2015. 31, 33

[77] M. Hassan, P. Ghosh, J. Tesch, D. Tzionas, and M. J. Black, "Populating 3D scenes by learning human-scene interaction," in *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 31

[78] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009. 32

[79] H. Grabner, J. Gall, and L. Van Gool, "What makes a chair a chair?," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1529–1536, 06 2011. 33

[80] D. F. Fouhey, V. Delaitre, A. Gupta, A. A. Efros, I. Laptev, and J. Sivic, "People watching: Human actions as a cue for single view geometry," in *European Conference on Computer Vision*, pp. 732–745, Springer, 2012. 33

[81] Y. Jiang, H. Koppula, and A. Saxena, "Hallucinated humans as the hidden context for labeling 3d scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013. 33

[82] X. Wang, R. Girdhar, and A. Gupta, "Binge watching: Scaling affordance learning from sitcoms," in *CVPR*, 2017. 33

[83] C.-Y. Chuang, J. Li, A. Torralba, and S. Fidler, "Learning to act properly: Predicting and explaining affordances from images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 975–983, 2018. 33

[84] X. Li, S. Liu, K. Kim, X. Wang, M.-H. Yang, and J. Kautz, "Putting humans in a scene: Learning affordance in 3d indoor environments," in *CVPR*, 2019. 33

[85] B. Yao and L. Fei-Fei, "Modeling mutual context of object and human pose in human-object interaction activities," *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 17–24, 2010. 33

[86] H. S. Koppula, R. Gupta, and A. Saxena, "Learning human activities and object affordances from rgb-d videos," *The International Journal of Robotics Research*, vol. 32, no. 8, pp. 951–970, 2013. 33

[87] Y. Zhu, A. Fathi, and L. Fei-Fei, "Reasoning about object affordances in a knowledge base representation," in *European conference on computer vision*, pp. 408–424, Springer, 2014. 33

[88] G. Gkioxari, R. Girshick, P. Dollár, and K. He, "Detecting and recognizing human-object interactions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8359–8367, 2018. 33

[89] Z. Cao, I. Radosavovic, A. Kanazawa, and J. Malik, "Reconstructing hand-object interactions in the wild," *arXiv e-prints*, pp. arXiv–2012, 2020. 33

[90] J. Lee, J. Chai, P. S. Reitsma, J. K. Hodgins, and N. S. Pollard, "Interactive control of avatars animated with human motion data," in *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pp. 491–500, 2002. 33

[91] Z. Cao, H. Gao, K. Mangalam, Q. Cai, M. Vo, and J. Malik, "Long-term human motion prediction with scene context," in *ECCV*, 2020. 33

[92] J. Wang, H. Xu, J. Xu, S. Liu, and X. Wang, "Synthesizing long-term 3d human motion and interaction in 3d scenes," 2020. 33

[93] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool, "Pose guided person image generation," in *Advances in Neural Information Processing Systems*, pp. 405–415, 2017. 33

[94] A. Siarohin, E. Sangineto, S. Lathuiliere, and N. Sebe, "Deformable gans for pose-based human image generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3408–3416, 2018. 33, 38

[95] G. Balakrishnan, A. Zhao, A. V. Dalca, F. Durand, and J. Guttag, "Synthesizing images of humans in unseen poses," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8340–8348, 2018. 33, 38

[96] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, "Video-to-video synthesis," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 33

[97] K. Aberman, M. Shi, J. Liao, D. Lischinski, B. Chen, and D. Cohen-Or, "Deep video-based performance cloning," in *Computer Graphics Forum*, vol. 38, pp. 219–233, Wiley Online Library, 2019. 33, 38

[98] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, "Everybody dance now," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5933–5942, 2019. 33

[99] Y. Li, C. Huang, and C. C. Loy, "Dense intrinsic appearance flow for human pose transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 33

[100] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *International Conference on Learning Representations*, 2019. 33, 39, 46

[101] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 2017. 33, 42

[102] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014. 33, 37

[103] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris, "Ganspace: Discovering interpretable gan controls," in *Proc. NeurIPS*, 2020. 33

[104] A. Jahanian, L. Chai, and P. Isola, "On the "steerability" of generative adversarial networks," in *International Conference on Learning Representations*, 2020. 33

[105] W. Peebles, J. Peebles, J.-Y. Zhu, A. A. Efros, and A. Torralba, "The hessian penalty: A weak prior for unsupervised disentanglement," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 33

[106] L. Goetschalckx, A. Andonian, A. Oliva, and P. Isola, "Ganalyze: Toward visual definitions of cognitive image properties," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5744–5753, 2019. 33

[107] Y. Li, K. K. Singh, U. Ojha, and Y. J. Lee, "Mixnmatch: Multifactor disentanglement and encoding for conditional image generation," in *CVPR*, 2020. 33

[108] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba, "Gan dissection: Visualizing and understanding generative adversarial networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 33

[109] L. Chai, J. Wulff, and P. Isola, "Using latent space regression to analyze and leverage compositionality in gans.," in *International Conference on Learning Representations*, 2021. 33, 58

[110] T. Park, J.-Y. Zhu, O. Wang, J. Lu, E. Shechtman, A. A. Efros, and R. Zhang, "Swapping autoencoder for deep image manipulation," in *Advances in Neural Information Processing Systems*, 2020. 33, 38

[111] L. Ma, Q. Sun, S. Georgoulis, L. Van Gool, B. Schiele, and M. Fritz, "Disentangled person image generation," in *The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 33

[112] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin, "Context-based vision system for place and object recognition," in *Computer Vision, IEEE International Conference on*, vol. 2, pp. 273–273, IEEE Computer Society, 2003. 34

[113] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, "Objects in context," in *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8, IEEE, 2007. 34

[114] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert, "An empirical study of context in object detection," in *2009 IEEE Conference on computer vision and Pattern Recognition*, pp. 1271–1278, IEEE, 2009. 34

[115] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 891–898, 2014. 34

[116] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1325–1339, 2013. 34

[117] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017. 34, 35, 36

[118] A. Diba, M. Fayyaz, V. Sharma, M. Paluri, J. Gall, R. Stiefelhagen, and L. Van Gool, "Large scale holistic video understanding," in *European Conference on Computer Vision*, pp. 593–610, Springer, 2020. 34, 35, 36

[119] M. Monfort, A. Andonian, B. Zhou, K. Ramakrishnan, S. A. Bargal, T. Yan, L. Brown, Q. Fan, D. Gutfreund, C. Vondrick, *et al.*, "Moments in time dataset: one million videos for event understanding," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 2, pp. 502–508, 2019. 34, 35, 36

[120] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, "Hollywood in homes: Crowdsourcing data collection for activity understanding," in *European Conference on Computer Vision*, pp. 510–526, Springer, 2016. 34, 35, 36

[121] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pp. 3686–3693, 2014. 34, 35, 36, 48

[122] D. F. Fouhey, W.-c. Kuo, A. A. Efros, and J. Malik, "From lifestyle vlogs to everyday interactions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4991–5000, 2018. 34, 35, 37

[123] W. Zhang, M. Zhu, and K. G. Derpanis, "From actemes to action: A strongly-supervised representation for detailed action understanding," in *Proceedings of the*

IEEE International Conference on Computer Vision, pp. 2248–2255, 2013. 34, 35, 37

[124] A. Kanazawa, J. Y. Zhang, P. Felsen, and J. Malik, "Learning 3d human dynamics from video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5614–5623, 2019. 34, 35, 36

[125] D. Epstein, B. Chen, and C. Vondrick, "Oops! predicting unintentional action in video," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 919–929, 2020. 34, 35, 36

[126] N. Xu, L. Yang, Y. Fan, J. Yang, D. Yue, Y. Liang, B. Price, S. Cohen, and T. Huang, "Youtube-vos: Sequence-to-sequence video object segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 585–601, 2018. 34, 35, 37

[127] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7291–7299, 2017. 34, 35, 36, 49

[128] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: realtime multi-person 2d pose estimation using part affinity fields," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 1, pp. 172–186, 2019. 34, 35, 36, 38, 48, 49

[129] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017. 34, 36, 49

[130] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2." https://github.com/facebookresearch/detectron2, 2019. 34, 36, 49

[131] J. Carreira, E. Noland, C. Hillier, and A. Zisserman, "A short note on the kinetics-700 human action dataset," *arXiv preprint arXiv:1907.06987*, 2019. 35, 36

[132] C. E. Duchon, "Lanczos filtering in one and two dimensions," *Journal of Applied Meteorology and Climatology*, vol. 18, no. 8, pp. 1016 – 1022, 1979. 35

[133] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 38, 42

[134] F. Yu, Y. Zhang, S. Song, A. Seff, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop.," *CoRR*, vol. abs/1506.03365, 2015. 38

[135] R. Mokady, M. Yarom, O. Tov, O. Lang, D. Cohen-Or, T. Dekel, M. Irani, and I. Mosseri, "Self-distilled stylegan: Towards generation from internet photos," *arXiv preprint arXiv:2202.12211*, 2022. 39, 46

[136] A. Sauer, K. Schwarz, and A. Geiger, "Stylegan-xl: Scaling stylegan to large diverse datasets," *arXiv preprint arXiv:2202.00273*, 2022. 39

[137] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," *arXiv preprint arXiv:2006.06676*, 2020. 39, 47

[138] S. Zhao, Z. Liu, J. Lin, J.-Y. Zhu, and S. Han, "Differentiable augmentation for data-efficient gan training," *arXiv preprint arXiv:2006.10738*, 2020. 39

[139] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *International Conference on Machine Learning*, pp. 1060–1069, PMLR, 2016. 40

[140] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 42

[141] R. Abdal, Y. Qin, and P. Wonka, "Image2stylegan: How to embed images into the stylegan latent space?," in *ICCV*, pp. 4431–4440, 2019. 44

[142] D. P. Kingma and J. Ba, "Adam: A methodfor stochastic optimization," in *International Conference onLearning Representations (ICLR)*, 2015. 44

[143] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008. 46

[144] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," *Cognitive science*, vol. 9, no. 1, pp. 147–169, 1985. 46

[145] M. Marchesi, "Megapixel size image creation using generative adversarial networks," 2017. 46

[146] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *NeurIPS*, pp. 10236–10245, 2018. 46

[147] A. Fan, M. Lewis, and Y. Dauphin, "Hierarchical neural story generation," *arXiv preprint arXiv:1805.04833*, 2018. 46

[148] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," *arXiv preprint arXiv:1904.09751*, 2019. 46

[149] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, p. 6629–6640, 2017. 48

[150] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015. 48

[151] S. Kulal, T. Brooks, A. Aiken, J. Wu, J. Yang, J. Lu, A. A. Efros, and K. K. Singh, "Putting people in their place: Affordance-aware human insertion into scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 50

[152] T. Brooks, A. Holynski, and A. A. Efros, "Instructpix2pix: Learning to follow image editing instructions," *arXiv preprint arXiv:2211.09800*, 2022. 56

[153] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022. 57, 58, 59, 60, 62, 66, 76

[154] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, *et al.*, "Flamingo: a visual language model for few-shot learning," *arXiv preprint arXiv:2204.14198*, 2022. 58

[155] R. Mokady, A. Hertz, and A. H. Bermano, "Clipcap: Clip prefix for image captioning," *arXiv preprint arXiv:2111.09734*, 2021. 58

[156] Z. Wang, J. Yu, A. W. Yu, Z. Dai, Y. Tsvetkov, and Y. Cao, "Simvlm: Simple visual language model pretraining with weak supervision," *arXiv preprint arXiv:2108.10904*, 2021. 58

[157] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, "Visualbert: A simple and performant baseline for vision and language," *arXiv preprint arXiv:1908.03557*, 2019. 58

[158] A. Zeng, A. Wong, S. Welker, K. Choromanski, F. Tombari, A. Purohit, M. Ryoo, V. Sindhwani, J. Lee, V. Vanhoucke, *et al.*, "Socratic models: Composing zero-shot multimodal reasoning with language," *arXiv preprint arXiv:2204.00598*, 2022. 58

[159] A. M. H. Tiong, J. Li, B. Li, S. Savarese, and S. C. Hoi, "Plug-and-play vqa: Zero-shot vqa by conjoining large pretrained models with zero training," *arXiv preprint arXiv:2210.08773*, 2022. 58

[160] Y. Du, S. Li, and I. Mordatch, "Compositional visual generation with energy based models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6637–6647, 2020. 58

[161] N. Liu, S. Li, Y. Du, A. Torralba, and J. B. Tenenbaum, "Compositional visual generation with composable diffusion models," *arXiv preprint arXiv:2206.01714*, 2022. 58, 63

[162] Y. Tewel, Y. Shalev, I. Schwartz, and L. Wolf, "Zero-shot image-to-text generation for visual-semantic arithmetic," *arXiv preprint arXiv:2111.14447*, 2021. 58

[163] S. Li, Y. Du, J. B. Tenenbaum, A. Torralba, and I. Mordatch, "Composing ensembles of pre-trained models via iterative consensus," *arXiv preprint arXiv:2210.11522*, 2022. 58

[164] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proceedings of the 32nd International Conference on Machine Learning* (F. Bach and D. Blei, eds.), vol. 37 of *Proceedings of Machine Learning Research*, (Lille, France), pp. 2256–2265, PMLR, 07–09 Jul 2015. 58, 62

[165] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," *Advances in Neural Information Processing Systems*, vol. 32, 2019. 58

[166] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020. 58

[167] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794, 2021. 58, 72

[168] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, "Image super-resolution via iterative refinement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 58

[169] C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, and M. Norouzi, "Palette: Image-to-image diffusion models," in *ACM SIGGRAPH 2022 Conference Proceedings*, pp. 1–10, 2022. 58

[170] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni, *et al.*, "Make-a-video: Text-to-video generation without text-video data," *arXiv preprint arXiv:2209.14792*, 2022. 58

[171] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," *arXiv preprint arXiv:2009.09761*, 2020. 58

[172] X. L. Li, J. Thickstun, I. Gulrajani, P. Liang, and T. B. Hashimoto, "Diffusion-lm improves controllable text generation," *arXiv preprint arXiv:2205.14217*, 2022. 58

[173] W. Peebles, I. Radosavovic, T. Brooks, A. A. Efros, and J. Malik, "Learning to learn with generative models of neural network checkpoints," *arXiv preprint arXiv:2209.12892*, 2022. 58

[174] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, "Glide: Towards photorealistic image generation and editing with text-guided diffusion models," *arXiv preprint arXiv:2112.10741*, 2021. 58

[175] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, 2022. 58

[176] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *arXiv preprint arXiv:2205.11487*, 2022. 58

[177] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *arXiv preprint arXiv:1508.06576*, 2015. 58

[178] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2414–2423, 2016. 58

[179] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CVPR*, 2017. 58

[180] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017. 58

[181] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *ECCV*, 2018. 58

[182] M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz, "Few-shot unsupervised image-to-image translation," in *IEEE International Conference on Computer Vision (ICCV)*, 2019. 58

[183] U. Ojha, Y. Li, C. Lu, A. A. Efros, Y. J. Lee, E. Shechtman, and R. Zhang, "Few-shot image generation via cross-domain correspondence," in *CVPR*, 2021. 58

[184] R. Abdal, Y. Qin, and P. Wonka, "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. 58

[185] R. Abdal, Y. Qin, and P. Wonka, "Image2stylegan++: How to edit the embedded images?," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8296–8305, 2020. 58

[186] Y. Alaluf, O. Tov, R. Mokady, R. Gal, and A. Bermano, "Hyperstyle: Stylegan inversion with hypernetworks for real image editing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18511–18521, 2022. 58

[187] D. Epstein, T. Park, R. Zhang, E. Shechtman, and A. A. Efros, "Blobgan: Spatially disentangled scene representations," in *European Conference on Computer Vision*, pp. 616–635, Springer, 2022. 58

[188] O. Tov, Y. Alaluf, Y. Nitzan, O. Patashnik, and D. Cohen-Or, "Designing an encoder for stylegan image manipulation," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–14, 2021. 58

[189] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or, "Encoding in style: a stylegan encoder for image-to-image translation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2287–2296, 2021. 58

[190] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*, pp. 8748–8763, PMLR, 2021. 58

[191] G. Kwon and J. C. Ye, "Clipstyler: Image style transfer with a single text condition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18062–18071, 2022. 58

[192] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski, "Styleclip: Text-driven manipulation of stylegan imagery," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2085–2094, October 2021. 58

[193] W. Zheng, Q. Li, X. Guo, P. Wan, and Z. Wang, "Bridging clip and stylegan through latent alignment for image editing," *arXiv preprint arXiv:2210.04506*, 2022. 58

[194] R. Gal, O. Patashnik, H. Maron, A. H. Bermano, G. Chechik, and D. Cohen-Or, "Stylegan-nada: Clip-guided domain adaptation of image generators," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 1–13, 2022. 58, 62, 74

[195] G. Kim, T. Kwon, and J. C. Ye, "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. 58

[196] O. Avrahami, D. Lischinski, and O. Fried, "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. 58

[197] K. Crowson, S. Biderman, D. Kornis, D. Stander, E. Hallahan, L. Castricato, and E. Raff, "Vqgan-clip: Open domain image generation and editing with natural language guidance," in *European Conference on Computer Vision*, pp. 88–105, Springer, 2022. 58

[198] O. Bar-Tal, D. Ofri-Amar, R. Fridman, Y. Kasten, and T. Dekel, "Text2live: Text-driven layered image and video editing," in *European Conference on Computer Vision*, pp. 707–723, Springer, 2022. 58, 59, 67, 72, 73

[199] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, "Sdedit: Guided image synthesis and editing with stochastic differential equations," in *International Conference on Learning Representations*, 2021. 58, 59, 67, 72

[200] B. Kawar, S. Zada, O. Lang, O. Tov, H. Chang, T. Dekel, I. Mosseri, and M. Irani, "Imagic: Text-based real image editing with diffusion models," *arXiv preprint arXiv:2210.09276*, 2022. 58, 59

[201] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-Or, "Prompt-to-prompt image editing with cross attention control," *arXiv preprint arXiv:2208.01626*, 2022. 58, 59, 60, 61, 62, 66, 67, 72, 73, 76

[202] R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-Or, "An image is worth one word: Personalizing text-to-image generation using textual inversion," *arXiv preprint arXiv:2208.01618*, 2022. 58, 59

[203] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," *arXiv preprint arXiv:2208.12242*, 2022. 58, 59

[204] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, *et al.*, "Training language models to follow instructions with human feedback," *arXiv preprint arXiv:2203.02155*, 2022. 59

[205] S. Mishra, D. Khashabi, C. Baral, and H. Hajishirzi, "Cross-task generalization via natural language crowdsourcing instructions," *arXiv preprint arXiv:2104.08773*, 2021. 59

[206] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned language models are zero-shot learners," *arXiv preprint arXiv:2109.01652*, 2021. 59

[207] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2107–2116, 2017. 59

[208] S. Ravuri and O. Vinyals, "Classification accuracy score for conditional generative models," *Advances in neural information processing systems*, vol. 32, 2019. 59

[209] Y. Viazovetskyi, V. Ivashkin, and E. Kashin, "Stylegan2 distillation for feed-forward image manipulation," in *European conference on computer vision*, pp. 170–186, Springer, 2020. 59

[210] N. Tritrong, P. Rewatbowornwong, and S. Suwajanakorn, "Repurposing gans for one-shot semantic part segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4475–4485, 2021. 59

[211] D. Li, H. Ling, S. W. Kim, K. Kreis, S. Fidler, and A. Torralba, "Bigdatasetgan: Synthesizing imagenet with pixel-wise annotations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21330–21340, 2022. 59

[212] W. Peebles, J.-Y. Zhu, R. Zhang, A. Torralba, A. A. Efros, and E. Shechtman, "Gan-supervised dense visual alignment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13470–13481, 2022. 59

[213] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, *et al.*, "Laion-5b: An open large-scale dataset for training next generation image-text models," *arXiv preprint arXiv:2210.08402*, 2022. 60

[214] A. Hyvärinen and P. Dayan, "Estimation of non-normalized statistical models by score matching.," *Journal of Machine Learning Research*, vol. 6, no. 4, 2005. 62, 64

[215] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013. 62

[216] T. Wang, T. Zhang, B. Zhang, H. Ouyang, D. Chen, Q. Chen, and F. Wen, "Pretraining is all you need for image-to-image translation," *arXiv preprint arXiv:2205.12952*, 2022. 63

[217] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022. 63

[218] T. Karras, M. Aittala, T. Aila, and S. Laine, "Elucidating the design space of diffusion-based generative models," *arXiv preprint arXiv:2206.00364*, 2022. 66, 67

[219] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2021. 72