SinFusion: Training Diffusion Models on a Single Image or Video

Yaniv Nikankin [1 *] **Niv Haim** [1 *] **Michal Irani** [1]

Project Page: https://yanivnik.github.io/sinfusion

Abstract

Diffusion models exhibited tremendous progress in image and video generation, exceeding GANs in quality and diversity. However, they are usually trained on very large datasets and are not naturally adapted to manipulate a given input image or video. In this paper we show how this can be resolved by training a diffusion model on a single input image or video. Our image/video-specific diffusion model (SinFusion) learns the appearance and dynamics of the single image or video, while utilizing the conditioning capabilities of diffusion models. It can solve a wide array of image/video-specific manipulation tasks. In particular, our model can learn from few frames the motion and dynamics of a single input video. It can then generate diverse new video samples of the same dynamic scene, extrapolate short videos into long ones (both forward and backward in time) and perform video upsampling. Most of these tasks are not realizable by current video-specific generation methods.

1. Introduction

Until recently, generative adversarial networks (GANs) ruled the field of generative models, with seminal works like StyleGAN (Karras et al., 2017; 2019; 2020), BigGAN (Brock et al., 2018) etc. (Radford et al., 2015; Zhang et al., 2019). Diffusion models (**DMs**) (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020) have gained the lead in the last years, surpassing GANs by image quality and diversity (Dhariwal & Nichol, 2021) and becoming the leading method in many vision tasks like text-to-image generation, superresolution and many more (Jolicoeur-Martineau et al., 2020; Nichol & Dhariwal, 2021; Song et al., 2020; Saharia et al., 2022b; Ho et al., 2022b; Nichol et al., 2021; Saharia et al., 2022a; Rombach et al., 2022) (see surveys (Cao et al., 2022; Croitoru et al., 2022)). Recent works also demonstrate the effectiveness of DMs for video and text-to-video generation (Ho et al., 2022c; Singer et al., 2022; Ho et al., 2022a; Villegas et al., 2022).

DMs are trained on massive datasets and as such, these models are very large and resource demanding. Applying their capabilities to edit or manipulate a specific input provided by the user is non-trivial and requires careful manipulation and fine-tuning (Avrahami et al., 2022; Gal et al., 2022; Ruiz et al., 2022; Valevski et al., 2022; Kawar et al., 2022).

In this work we propose a framework for training diffusion models on a single input image or video - *"SinFusion"*. We harness the success and high-quality of DMs at image synthesis, to single-image/video tasks. Once trained, SinFusion can generate new image/video samples with similar appearance and dynamics to the original input and perform various editing and manipulation tasks. In the video case, SinFusion exhibits impressive generalization capabilities by coherently extrapolating an input video far into the future (or past). This is learned from very few frames (mostly 2-3 dozens, but is already apparent for fewer frames).

We demonstrate the applicability of SinFusion to a variety of single-video tasks, including: (i) diverse generation of new videos from a *single* input video (better than existing methods), (ii) video extrapolation (both forward and backward in time), (iii) video upsampling. Many of these taks (e.g., extrapolation/interpolation in time) are not realizable by current video-specific generation methods (Gur et al., 2020; Haim et al., 2021). Moreover, large-scale diffusion models for video generation (Yang et al., 2022; Ho et al., 2022c) trained on large video datasets are not designed to manipulate a real input video. When applied to a single input image, SinFusion can perform diverse image generation and manipulation tasks. However, the main focus in our paper is on *single-video* generation/manipulation tasks, as this is a more challenging and less explored domain.

Our framework builds on top of the commonly used DDPM architecture (Ho et al., 2020), but introduces several impor-

[*]Equal contribution [1]Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel. Correspondence to: Yaniv Nikankin <yaniv.nikankin@weizmann.ac.il>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

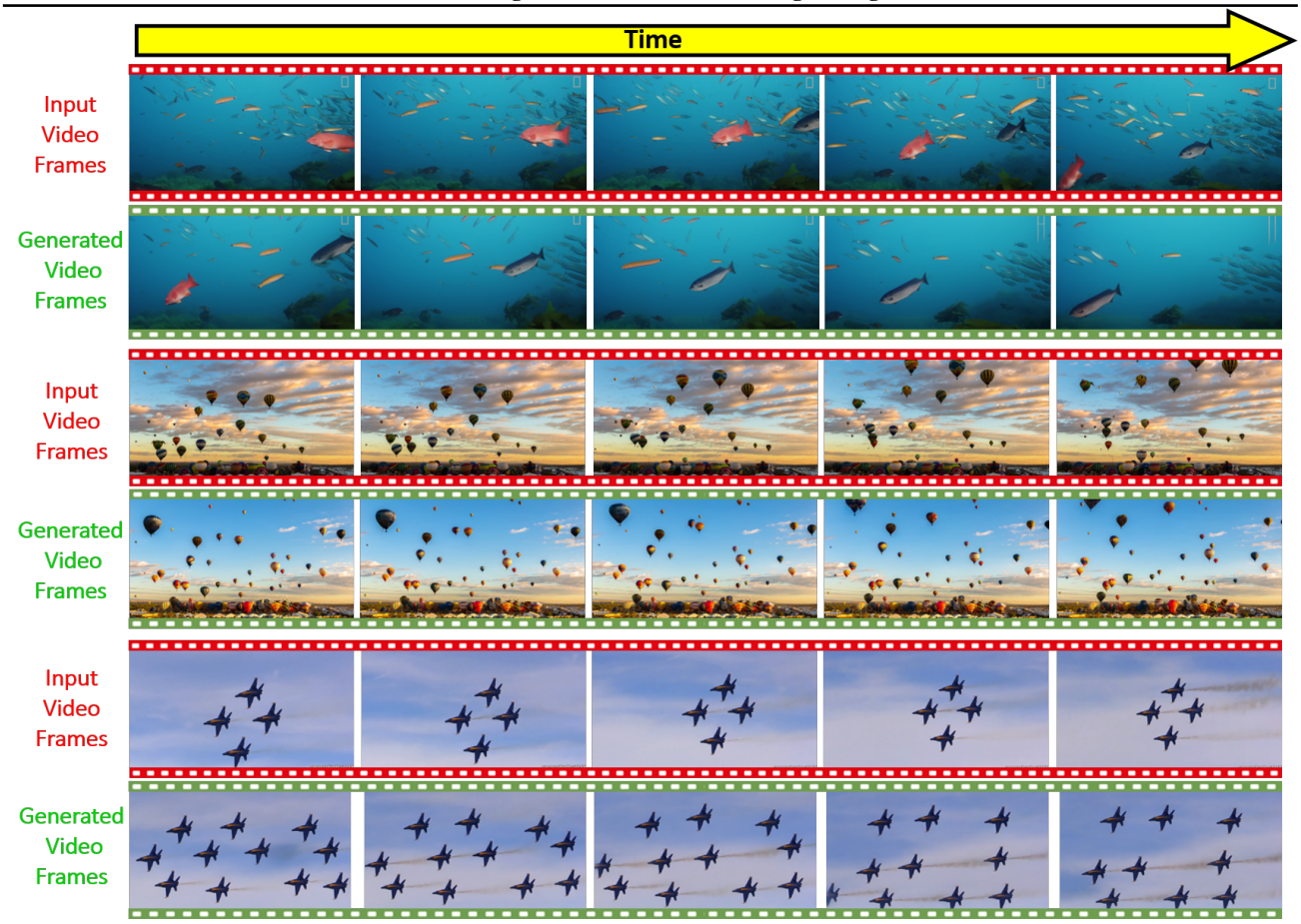


Figure 1. **Diverse video generation.** For each single training video, *red row* shows consecutive frames from the training video, whereas the *green row* show a set of consecutive frames generated by our single video DDPM. **Please see the videos in our project page**.

tant modifications that are essential for allowing it to train on a single image/video. Our backbone DDPM network is *fully convolutional*, hence can be used to generate images of any size by starting from a noisy image of the desired output size. Our single-*video* DDPM, consists of 3 single-*image* DDPMs, each trained to map noise to large crops of an image (a video frame), either unconditionally, or conditioned on other frames from the input video.

Our main contributions are as follows:

- First-ever diffusion model trained on a single image/video.
- Unlike general large-scale diffusion models, SinFusion can edit and manipulate a *real input video*. This includes: diverse video generation, video extrapolation (both forward and backward in time), and temporal upsampling.
- SinFusion provides new video capabilities and tasks not realizable by current single-video GANs (e.g., video extrapolation with impressive motion generalization capabilities).
- We propose a new set of evaluation metrics for diverse video generation from a single video.

2. Related Work

Our work lies in the intersection of several fields: generative models trained on a single image or video, manipulation of a real input image/video, diffusion models and methods for image/video generation in general. Here we briefly mention the main achievements in each field and their relation (and difference) from our proposed approach.

Video generation is a broad field of research including many areas such as video GANs (Vondrick et al., 2016; Tulyakov et al., 2018; Clark et al., 2019; Skorokhodov et al., 2022), video-to-video translation (Wang et al., 2018; Bansal et al., 2018) or autoregressive prediction models (Ballas et al., 2015; Villegas et al., 2017; Babaeizadeh et al., 2017; Denton & Fergus, 2018), to name a few. Diffusion models for video generation are fairly recent and mostly rely on DDPM (Ho et al., 2020) framework for image generation, extended to handle videos (Yang et al., 2022; Höppe et al., 2022; Voleti et al., 2022; Ho et al., 2022c;a;a; Harvey et al., 2022) (see Appendix D). These methods can synthesize beautiful videos, however, none of them can modify or manipulate an existing input video provided by the user, which is our goal.

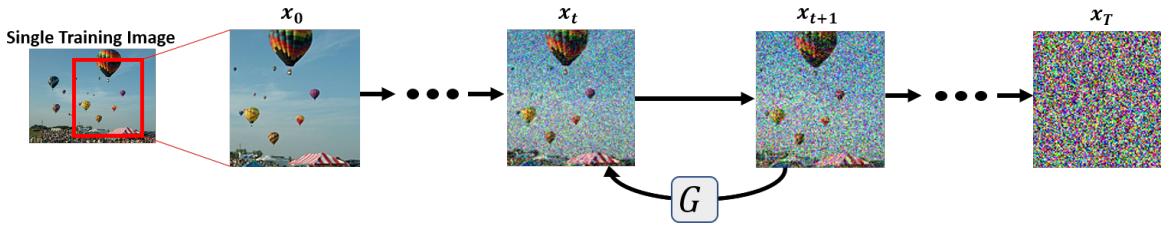


Figure 2. **Single Image DDPM.** Our single-image DDPM trains on large crops from a single image. It learns to remove noise from noisy crops, and, at inference, can generate diverse samples with similar structure and appearance to the training image.

Generative Models trained on a Single Image or Video aim to generate new diverse samples, similar in appearance and dynamics to the image/video on which they were trained. Most notably, SinGAN (Shaham et al., 2019) and InGAN (Shocher et al., 2018) trained multi-scale GANs to learn the distribution of patches in an image. They showed its applicability to diverse random generation from a single image, as well as a variety of other image synthesis applications (inpainting, style transfer, etc.). However, GPNN (Granot et al., 2022) showed that most image synthesis tasks proposed by single-image GAN-based models can be solved by classical non-parametric patch nearest-neighbour methods (Efros & Leung, 1999; Efros & Freeman, 2001; Simakov et al., 2008), and achieve outputs of higher quality while reducing generation time by orders of magnitude. Similarly, extensions of SinGAN (Shaham et al., 2019) to generation from a single *video* (Gur et al., 2020; Arora & Lee, 2021) were outperformed by patch nearest-neighbour methods (Haim et al., 2021). However, nearest-neighbour methods have a very limited notion of generalization and are therefore limited to tasks where it is natural to "copy" parts of the input. While generated samples are of high quality and look realistic, this is because the samples are essentially *copies* of parts of the original video stitched together. *They fail to exhibit motion generalization capabilities*. In contrast, our method generalizes well from just a few frames and can be easily trained on a long input video. Concurrently to our work, Kulikov et al. (2022); Wang et al. (2022) trained DMs on a single image and showed various capabilities. However, both works focused on generation from a single *image*, while we present applications on a single *video*.

Reference Image Manipulation with Large Generative Models. One of the practical application of generative models trained on large datasets is their strong generalization capabilities for semantic image editing, often obtained via latent space interpolation (Radford et al., 2015; Brock et al., 2018; Karras et al., 2019). Applying these capabilities to an existing reference image was mostly achieved by GAN "inversion" techniques (Xia et al., 2022), and very recently by fine-tuning large diffusion models (Gal et al., 2022; Kawar et al., 2022; Ruiz et al., 2022; Valevski et al., 2022; Avrahami et al., 2022). However, to the best of our knowledge, there are no existing large-scale models to-date which can manipulate an existing input reference video.

3. Perliminaries: Overview of DDPM

Denoising diffusion probabilistic models (DDPM) (Ho et al., 2020; Sohl-Dickstein et al., 2015) are a class of generative models that can learn to convert unstructured noise to samples from a given distribution, by performing an iterative process of removing small amounts of Gaussian noise at each step. Since our method heavily relies on DDPM, we provide here a very brief overview of DDPM and its basics. To train a DDPM, an input image x_0 is sampled, and small portions of gaussian noise ϵ are gradually added to it in a parameter-free *forward* process, resulting in a noisy image x_t. The forward process can be written as:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \tag{1}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$, $\beta_t \in (0,1)$ is a predefined parameter and $\epsilon \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ is the noise used to generate the noisy image x_t.

A neural network is then trained to perform the *reverse* process. In the reverse process, the noisy image x_t is given as input to the neural network, which predicts the noise ϵ that was used to generate the noisy image. The network is trained with an L_2 loss:

$$L(\theta) = \mathbb{E}_{\mathbf{x}_0,\epsilon}\left[\left\|\epsilon - \epsilon_\theta\left(x_t, t\right)\right\|^2\right] . \tag{2}$$

In existing DDPM-based methods, The network is typically trained on a large dataset of images, from which x_0 is sampled. Once trained, the generation process is initiated with a random noise image $x_T \sim \mathcal{N}(\mathbf{0},\mathbf{I})$. The image is passed through the model in a series of *reverse* steps. In each timestep $t = T, ..., 1$, the neural network predicts the noise ϵ_t. This noise is then used to generate a less noisy version of the image (x_{t-1}), and the process is repeated until a possible clean image x_0 is generated.

4. Single Image DDPM

Our goal is to leverage the powerful mechanism of diffusion models to generation from a single image/video. While the main contribution of this paper is in using DDPMs for generation from a single *video*, we first explain how a diffusion model can be trained on a single *image*. In Sec. 5 we show how this model can be extended to *video* generation. Some applications of single image DDPM are found in Sec. 6.

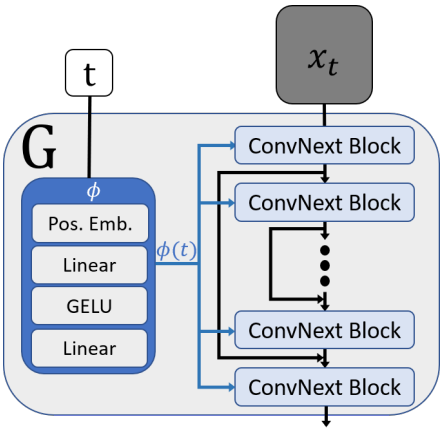


Figure 3. **Network Architecture**. Our backbone network is a fully convolutional chain of ConvNext (Liu et al., 2022) blocks with residual connections. Note that our network does not include any reduction in the spatial dimensions along the layers.

Given a single input image, we want our model to generate new diverse samples that are similar in appearance and structure to that of the input image, but also allow for semantically coherent variations. We build upon the common DDPM (Ho et al., 2020) framework (Section 3) and introduce several modifications to the training procedure and to the core network of DDPM. These are highlighted below:

Training on Large Crops. Instead of training on a large collection of images, we train a single diffusion model on many large random crops from the input image (typically, about 95% the size of the original image, Figure 2). We find that training on the original resolution of the image is sufficient for generating diverse image samples, even without the use of multi-scale pyramid (unlike most previous single image/video generative methods (Arora & Lee, 2021; Shocher et al., 2018; Shaham et al., 2019; Hinz et al., 2021; Gur et al., 2020; Granot et al., 2022; Haim et al., 2021)). By training on large crops our generated outputs retain the global structure of the input image.

Network Architecture. Directly training the standard DDPM (Ho et al., 2020) on the single image or its large crops results in "overfitting", namely the model only generates the same image crops. We postulate that this phenomenon occurs because of the receptive field of the core backbone network in DDPM, which is the entire input image. To this end we modify the backbone UNet (Ronneberger et al., 2015) network of DDPM, in order to reduce the size of its receptive field. We remove the attention layers as they have global receptive field. We also remove the downsampling and upsampling layers which cause the receptive field to grow too rapidly. Removing the attention layers has an unwanted side-effect - harming the performance of the diffusion model. Liu et al. (2022) proposed a fully convolutional network that matches the attention mechanism on many vision tasks. Inspired by this idea, we replace the ResNet (He et al., 2016) blocks in the network with ConvNext (Liu et al., 2022) blocks. This architectural choice is meant to replace the functionality of the attention layers, while keeping a non-global receptive field. It also has the advantage of reducing computation time. The overall receptive field of our network is then determined by the number of ConvNext blocks in the network. Changing the number of ConvNext blocks allows us to control the diversity of the output samples. Please see further analysis and hyperparameter choice in Appendix A. The rest of our backbone network is similar to DDPM, as well as the embedding network (ϕ) which is used to incorporate the diffusion timestep t into the model (and will be later used to embed the video frame difference, see Section 5). See Figure 3 for details.

Loss. At each training step, the model is given a noisy image crop x_t. However, in contrast to DDPM (Ho et al., 2020), whose model predicts the added noise (as in Equation (2)), our model predicts the clean image crop $\tilde{x}_{0,\theta}$. The loss in our single-image DDPM is:

$$L(\theta) = \mathbb{E}_{\mathbf{x}_0,\epsilon}\left[\left\|x_0 - \tilde{x}_{0,\theta}\left(x_t, t\right)\right\|^2\right] \tag{3}$$

We find that predicting the image instead of the noise leads to better results when training on a single image, both in terms of quality and training time. We attribute this difference to the simplicity of the data distribution in a single image compared to the data distribution of a large dataset of images. The full training algorithm is as follows:

Algorithm 1 Training on a single image x

1: **repeat**
2: $x_0 \leftarrow Crop(x)$
3: $t \sim \text{Uniform}(1, ..., T = 50)$
4: $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
5: Take gradient descent step on:
$\nabla_\theta \left\|x_0 - \tilde{x}_{0,\theta}\left(\sqrt{\bar{\alpha}_t}\mathrm{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t\right)\right\|^2$
6: **until** converged

Our single-image DDPM can be used for various image synthesis tasks like diverse generation (Figure 6), generation from sketch and image editing.

5. Single Video DDPM

Our video generation framework consists of 3 single-image-DDPM models (Fig. 4), whose combination gives rise to a variety of different video-related applications (Sec. 6). Our framework is essentially an autoregressive video generator. Namely, we train the models on a given input video with frames $\{x_0^1, x_0^2, ..., x_0^N\}$, and generate new videos with frames $\{\tilde{x}_0^1, \tilde{x}_0^2, ..., \tilde{x}_0^M\}$ such that each generated new frame $\tilde{x}_0^{n+1}$ is conditioned on its previous frame $\tilde{x}_0^n$. The three models that constitute our framework are all single-image

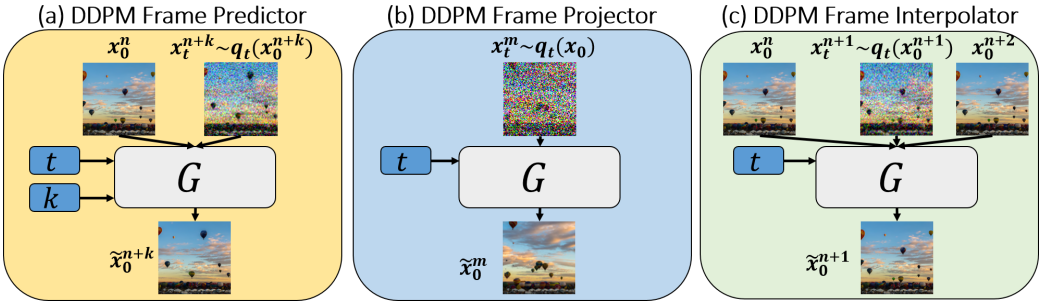


Figure 4. **Single Video DDPM** Our video framework consists of three models. The *Predictor* (left) generates new frames, conditioned on previous frames. The *Projector* (middle) generates frames from noise, and corrects small artifacts in predicted frames. The *Interpolator* (right) interpolates between adjacent frames (conditioned on them), to upsample the video temporally. These models are used together at inference to perform various video related applications.

DDPM models with the same network architecture as described in Sec. 4. The models are trained *separately* and differ by the type of inputs they are given, and by their role in the overall generation framework. The inference is application-dependant and is discussed in Sec. 6. Here we describe the training procedure of each model:

DDPM Frame Predictor (Fig. 4a). The role of the *Predictor* model is to generate new frames, each conditioned on its previous frame. At each training iteration we sample a condition frame from the video x_0^n and a noisy version of the $(n+k)$'th frame (x_t^{n+k}), which is to be denoised. The two frames are concatenated along the channels axis before being passed to the model (as in Saharia et al. (2022b)). The model is also given an embedding of the temporal difference (i.e frame index difference) between the two frames ($\phi(k)$). This embedding is concatenated to the timestep embedding ($\phi(t)$) of the DDPM. At early training k=1, and in following iterations it is gradually increased to be sampled at random from $k = [-3, 3]$. We find that such a curriculum learning approach improves outputs quality (even when at inference k=$\pm$1).

DDPM Frame Projector (Fig. 4b). The role of the *Projector* model is to "correct" frames that were generated by the *Predictor*. The Projector is a straightforward single-image-DDPM as described in Section 4, only it is trained on image crops from *all* the frames in the video. After learning the image structure and appearance of the video frames it is used to correct small artifacts in the generated frames, that may otherwise accumulate and destroy the video generation process. Intuitively, it "projects" patches from the generated frames back unto the original patch distribution, hence its name. The Projector is also used to generate the first frame. Frame correction is done at inference via a truncated diffusion process on the predicted frame.

DDPM Frame Interpolator (Fig. 4c). Our video-specific DDPM framework can be further trained to increase the temporal resolution of our generated videos, known also as "video upsampling" or "frame interpolation". Our DDPM frame *Interpolator* receives as input a pair of clean frames (x_0^n, x_0^{n+2}) as conditioning, and a noised version of the frame between them (x_t^{n+1}). The frames are concatenated along the channels axis, and the model is trained to predict the clean version of the interpolated frame ($\tilde{x}_0^{n+1}$). We find that this interpolation generalizes well to small motions in the video, and can be used to interpolate between every two consecutive frames, thus increasing the temporal resolution of generated videos as well as the input video.

Losses. We find that some models work better with different losses. The Projector and the Interpolator are trained with the loss in Eq. (3), while the Predictor is trained with Eq. (2), i.e., the noise is predicted instead of the output.

6. Applications

In this section we show how combinations of our single image/video DDPMs (Sections 4 and 5) provide a variety of video synthesis tasks. *We refer the reader to our project page*, especially to view our video results.

Diverse Video Generation: We can generate diverse videos from a single input video, to any length, such that the output samples have similar appearance, structure and motions as the original input video. This is done by combining our Predictor and Projector models. The first frame is either some frame from the original video, or a generated output image from the unconditional Projector. The Predictor is then used to generate the next frame, conditioned on the previous generated frame. Next, the predicted frame is corrected by the Projector (to remove small artifacts that may have been created, thus preventing error accumulation over time). This process is repeated until the desired number of frames has been generated. Repeating this autoregressive generation process creates a new video of arbitrary length. Note that the process is inherently stochastic – even if the initial frame is the same, different generated outputs will quickly diverge and create different videos. See Fig. 1 and *our project page for live videos and many more examples*.

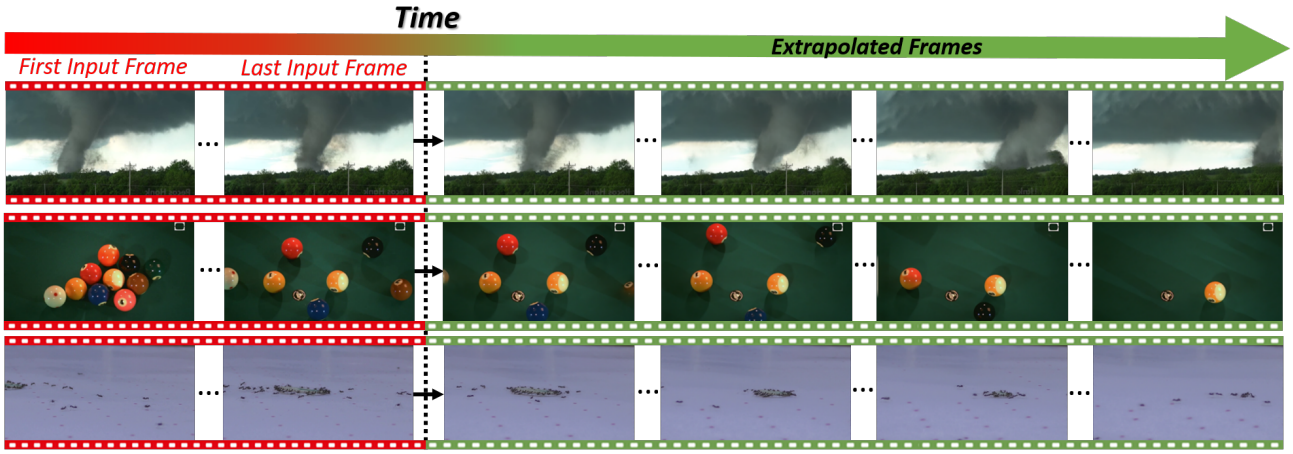
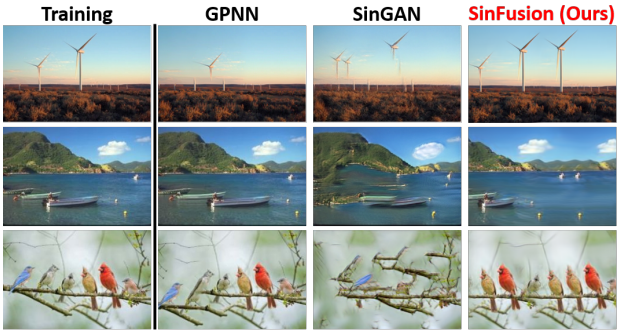


Figure 5. **Video Extrapolation (into the Future):** *SinFusion* trains on a single input video (*red*) - exemplified on video frames of Tornado, Balls, Ants. At inference, the auto-regressive generation process starts from the *last* frame of the input video, and generates a frame sequence of any desired length. The extrapolated frames (*green*) were never seen in the original video. *See full videos in our project page.*

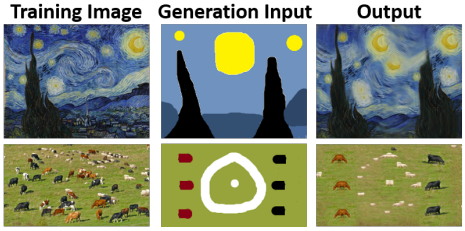
Video Extrapolation (*into the Future and into the Past*): Given an input video, we can *"predict the future"* (i.e., predict its future frames) by initializing the generation process described above with the last frame of the input video. Fig. 5 shows a few such examples. Note how our method extrapolates the motion in a realistic way, preserving the appearance and dynamics of the original video. **To the best of our knowledge, no existing single-video generation method can extrapolate a video in time.** Since our Predictor is also trained backward in time (predicting the previous frame using negative k), it can also *extrapolate videos backwards in time* (*"predict the past"*) by starting from the first frame of the video. This e.g. causes flying balloons to "land" (see video in our project page), even though these motions were never observed in the original video. This is a straightforward manifestation of the generalization capabilities of our framework. See Sec. 7 for evaluations of the generalization capabilities, and *full videos in our project page*.

Temporal Upsampling: Not only can SinFusion *extrapolate* input videos, it can also *interpolate* them – generate new frames in-between the original ones. This is done by training the DDPM Frame Interpolator (Fig. 4c) to predict each frame from its 2 *neighboring* frames, and at inference applying it to interpolate between *successive* frames. The appearance of the interpolated frames is corrected by the DDPM Frame Projector. *See example videos in our project page*.

Single-Image Applications When training our single-image DDPM (Sec. 4) on a single input image, our framework reduces to standard single-image generation and manipulation tasks, including: Diverse image generation, Sketch-guided image generation and Image editing. Diverse image generation is done by sampling a noisy image $x_T \sim \mathcal{N}(0, \mathbb{I})$ and iteratively denoising using our trained model such that $x_{t-1}=G(x_t)$. Since our backbone DDPM network is fully convolutional, it can be used to generate



(a) Diverse generation from a single image.



(b) Sketch-guided image generation.

Figure 6. **Single Image Applications**: (a) Images generated by SinFusion are comparable in visual quality to the patch nearest-neighbour based method GPNN (Granot et al., 2022), and outperforms SinGAN (Shaham et al., 2019). (b) SinFusion can generate new images from a single image, conditioned on input sketches.

images of any size by starting from a noisy image of the desired size. Fig. 6a shows such results (visually compared to SinGAN (Shaham et al., 2019) and GPNN (Granot et al., 2022)). *See more results in our project page*. SinFusion can also edit an input image by coarsely moving crops between locations in the image, and then let the model "correct" the image. We can similarly draw a sketch and let the model "fill in" the sketch with similar details from the input image (see Fig. 6b). The model is applied to the edited image/sketch by adding noise to the image, and then denoising the input image until a coherent image is obtained.

7. Evaluations & Comparisons

This section presents quantitative evaluations to support our main claim for the motion generalization capabilities of SinFusion. We measure the performance of our framework by training a model on a small portion of the original video, and test it on unseen frames from a different portion of the same video (Sec. 7.1). We further propose new useful evaluation metrics for diverse video generation from a single video (Sec. 7.2), and compare our diverse video generation from a single video to other methods for this task.

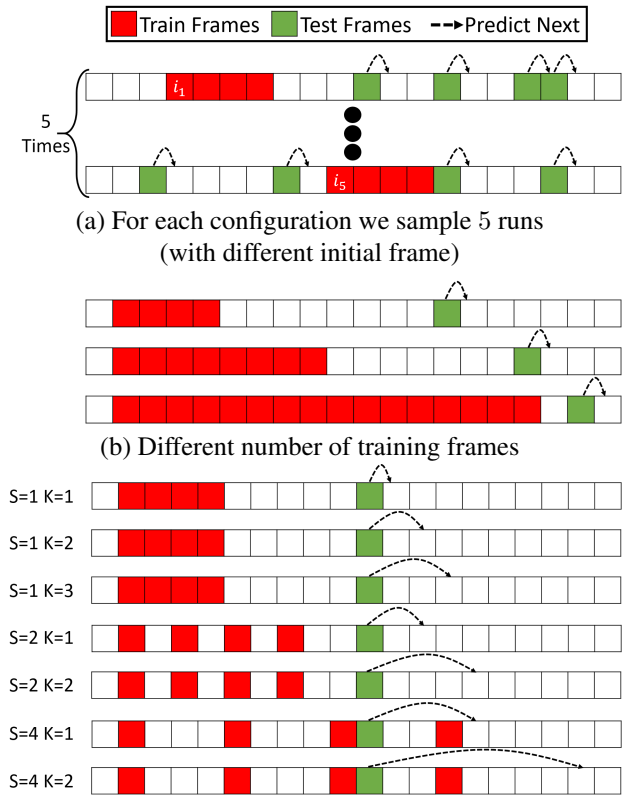
7.1. Future-Frame Prediction from a Single Video

Given a video with N frames, we train a model on $n < N$ frames. At inference, we sample 100 frames from the rest of the $N - n$ frames (not seen during training), and for each of them, use the trained model to predict its next (or a more distant) frame. We use PSNR to compare a predicted and real frame, and use the average PSNR as the overall score.

Baseline. Since no other methods exist for frame-prediction from a single video, we use a simple but strong baseline: Given a frame $f(i)$, we predict its next frame to be identical, namely, $f(i+1) = f(i)$. This is a strong baseline, since most videos have large static backgrounds, hence there is little change between consecutive frames.

Evaluating w.r.t. Different Training Set Sizes (Fig. 8a): We repeat this experiment for varying number of training frames n ($n = [4, 8, 16, 32, 64]$). For each choice of n, we choose a random location in the video, and take the n frames starting at that random location to be the "training frames". This is depicted in Fig. 7a – training frames (red) and test frames (green), where each test-frame is used to predict its next frame. In Fig. 7b we depict runs trained with different number of training frames n. The results are shown in Fig. 8a where each dot corresponds to averaging the score of 5 different runs (each time selecting the n training frames at a different random video location). As seen from Fig. 8a, our framework (red) is consistently better than the baseline (blue), exhibiting the motion prediction/generalization capabilities of SinFusion. Note that generalization increases (higher PSNR) with the size of the training set n, while the naive baseline does not improve. Note also that our framework generalizes quite well to next-frame prediction with as few as $n = 4$ frames in the training set.

Evaluating w.r.t. Video "speed" & Frame-gap k (Fig. 8b): We evaluate how well our framework generalizes on videos with faster motions. To this end, we sub-sample the original video in intervals of increasing size, resulting in faster motions in the sub-sampled videos. This way we can synthesize videos with larger speeds from the same video, making the results consistent with the first experiment (Fig. 8a). In this experiment we uses a fixed $n = 32$.



(a) For each configuration we sample 5 runs (with different initial frame)

(b) Different number of training frames

(c) Different choices of video-speed (S) and k

Figure 7. **Frame Prediction from a Single-Video**. Depicting evaluations experiments from Section 7.1 and Figure 8

A video with "speed" S is defined as the original video subsampled at $1/S$. After subsampling the video, the rest of the experiment is carried out as described above. For example, if the starting frame is frame number 17, then the training frames will be frames number $17, 19, 21, ..., 79$ for $S = 2$, and $17, 21, 25, ..., 141$ for $S = 4$.

We further evaluate w.r.t. k, which is the frame-gap between the current frame and the predicted frame (in the subsampled video) as in Fig. 4a. Recall that our model trains on $k = [-3, 3]$. Several setups for S and k are depicted in Fig. 7c.

Results are shown in Fig. 8b (note that for S=1,k=1, the result is the same as in Fig. 8a for n=32). Our framework is consistently better than the baseline. Larger speeds increase the performance gap between our framework and the baseline, further validating our claim for motion generalization.

7.2. A New Diversity Metric for Single-Video Methods

We devise a metric to quantify the diversity of our generated samples from a given input video. SinGAN (Shaham et al., 2019) proposed the following diversity metric (adapted in a straightforwad manner from images to videos): calculate the standard deviation of the intensity values of each voxel over all generated samples, then average this over all the

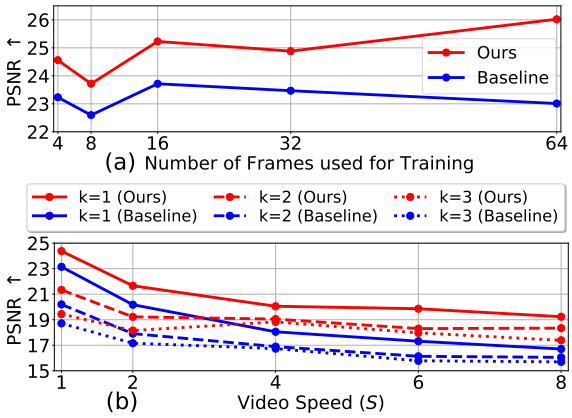


Figure 8. **Next-Frame Prediction from a Single Video.** SinFusion consistently beats the baseline on this task (see Sec. 7.1).

Table 1. **Diverse Video Generation – Comparison.**

Dataset	Method	NNFDIV ↑	NNFDIST ↓	SVFID↓
SinGAN-GIF	VGPNN	0.20	0.28	0.0058
	SinGAN-GIF	0.40	1.10	0.0119
	SinFusion (Ours)	0.30	0.45	0.0090
HP-VAE-GAN	VGPNN	0.22	0.14	0.0072
	HP-VAE-GAN	0.31	0.39	0.0081
	SinFusion (Ours)	0.35	0.26	0.0107

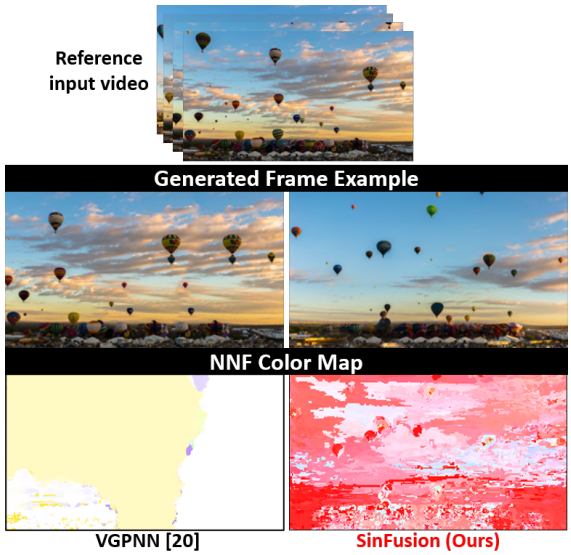


Figure 9. **Nearest-Neighbour Field (NNF) Color Map.** Patch-NNF between the generated video and the input video shows that VGPNN (Haim et al., 2021) tends to copy large chunks of the input video, whereas SinFusion generates new spatio-temporal compositions.

voxels, and then divide the result by the standard deviation of the intensity values of the original video.

This metric fails on a simple example: given an input video, one could generate "new" samples by just applying random translations to the video. With enough such "samples" this will converge to a high diversity score of 1. Rewarding for such global translations (or "copies" of large chunks of the input video) is an unwanted artifact of this metric. We introduce a nearest-neighbor-field (**NNF**) based diversity measure that captures the diversity of generated samples while penalizing for such unnecessary global translations.

The NNF is computed by searching for each $(3, 3, 3)$ spatio-temporal patch in a generated video, its nearest-neighbour (**n.n**) patch in the original video (with MSE). Each voxel is then associated with vector pointing to its n.n. Simple generated videos (e.g. a simple translation of the input) will have a rather constant NNF, while more complex generated videos will have complex NNFs. A visualized example for such NNF is shown in Fig.9 (a vector is converted to RGB using a color wheel (Baker et al., 2011)). See how the NNF of a VGPNN output is simple (corresponds to copying large chunks from a single input frame) whereas ours is more complex (*see full videos of these in our project page*).

We quantify the "complexity" of an NNF as follows: we use ZLIB (Gailly & Adler, 2004) to compress the NNF, and record the compression ratio. This gives a diversity measure in $[0, 1]$ that we term ***NNFDIV***. (The inspiration comes from *Kolmogorov complexity* (Kolmogorov, 1963) – simpler objects have simpler "description", which can be easily bounded by any compression algorithm). We also measure the RGB-similarity (termed ***NNFDIST***) by averaging the MSE distance of all generated patch to their n.n's.

In Table 1 we report the results of these metrics, as well as SVFID (Gur et al., 2020) score, on 2 diverse video generation datasets (see details in Appendix B.1). We compare our diverse video generation samples to existing single-video methods – HP-VAE-GAN (Gur et al., 2020), SinGAN-GIF (Arora & Lee, 2021) and VGPNN (Haim et al., 2021).

VGPNN is expected to have better quality (low NNFDIST / SVFID) because it is *copying chunks of frames from the original video*. However, its diversity (NNFDIV) is very low. On HP-VAE-GAN dataset, we outperform HP-VAE-GAN in both quality and diversity. On SinGAN-GIF dataset, SinGAN-GIF has higher diversity, however this may be attributed to its very low quality (NNFDIST). For both datasets, SinFusion has the best trade-off in terms of diversity and quality. **Further Experiments and Ablations** can be found in Appendices. B and C (e.g., comparison to VDM (Ho et al., 2022c)).

8. Limitations

As in all single-video generation methods, our method is also limited to videos with relatively small camera motion. Moreover, in videos with large objects of highly non-rigid motions (e.g., with many moving parts), SinFusion may break the object (or remove parts of it). This is because SinFusion has no notion of semantics. Some of the these limitations may be mitigated by incorporating suitable priors, and is part of our future work.

9. Conclusions

We propose SinFusion, a diffusion-based framework trained on a single video or image. Our unified framework can be applied for a variety of tasks. Our main application – generation and extrapolation of an input video, exhibits unprecedented generalization capabilities, that were not shown either by previous single-video methods, nor by large-scale video diffusion models.

Acknowledgements

We thank Assaf Shocher and Barak Zackay for useful discussions. This project received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 788535) and from the D. Dan and Betty Kahn Foundation.

References

Arora, R. and Lee, Y. J. Singan-gif: Learning a generative video model from a single gif. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1310–1319, 2021.

Avrahami, O., Lischinski, D., and Fried, O. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18208–18218, 2022.

Babaeizadeh, M., Finn, C., Erhan, D., Campbell, R. H., and Levine, S. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017.

Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M. J., and Szeliski, R. A database and evaluation methodology for optical flow. *International journal of computer vision*, 92(1):1–31, 2011.

Ballas, N., Yao, L., Pal, C., and Courville, A. Delving deeper into convolutional networks for learning video representations. *arXiv preprint arXiv:1511.06432*, 2015.

Bansal, A., Ma, S., Ramanan, D., and Sheikh, Y. Recycle-gan: Unsupervised video retargeting. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 119–135, 2018.

Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Cao, H., Tan, C., Gao, Z., Chen, G., Heng, P.-A., and Li, S. Z. A survey on generative diffusion model. *arXiv preprint arXiv:2209.02646*, 2022.

Clark, A., Donahue, J., and Simonyan, K. Adversarial video generation on complex datasets. *arXiv preprint arXiv:1907.06571*, 2019.

Croitoru, F.-A., Hondru, V., Ionescu, R. T., and Shah, M. Diffusion models in vision: A survey. *arXiv preprint arXiv:2209.04747*, 2022.

Denton, E. and Fergus, R. Stochastic video generation with a learned prior. In *International Conference on Machine Learning*, pp. 1174–1183. PMLR, 2018.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

Efros, A. A. and Freeman, W. T. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 341–346, 2001.

Efros, A. A. and Leung, T. K. Texture synthesis by non-parametric sampling. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pp. 1033–1038. IEEE, 1999.

Gailly, J.-l. and Adler, M. Zlib compression library. 2004.

Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A. H., Chechik, G., and Cohen-Or, D. An image is worth one word: Personalizing text-to-image generation using textual inversion, 2022. URL https://arxiv.org/abs/2208.01618.

Granot, N., Feinstein, B., Shocher, A., Bagon, S., and Irani, M. Drop the gan: In defense of patches nearest neighbors as single image generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13460–13469, 2022.

Gur, S., Benaim, S., and Wolf, L. Hierarchical patch vae-gan: Generating diverse videos from a single sample. *arXiv preprint arXiv:2006.12226*, 2020.

Haim, N., Feinstein, B., Granot, N., Shocher, A., Bagon, S., Dekel, T., and Irani, M. Diverse generation from a single video made possible. *arXiv preprint arXiv:2109.08591*, 2021.

Harvey, W., Naderiparizi, S., Masrani, V., Weilbach, C., and Wood, F. Flexible diffusion modeling of long videos. *arXiv preprint arXiv:2205.11495*, 2022.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hinz, T., Fisher, M., Wang, O., and Wermter, S. Improved techniques for training single-image gans. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1300–1309, 2021.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D. P., Poole, B., Norouzi, M., Fleet, D. J., et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022a.

Ho, J., Saharia, C., Chan, W., Fleet, D. J., Norouzi, M., and Salimans, T. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23:47–1, 2022b.

Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022c.

Höppe, T., Mehrjou, A., Bauer, S., Nielsen, D., and Dittadi, A. Diffusion models for video prediction and infilling. *arXiv preprint arXiv:2206.07696*, 2022.

Jacobs, N., Bies, B., and Pless, R. Using cloud shadows to infer scene structure and camera calibration. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1102–1109. IEEE, 2010.

Jacobs, N., Abrams, A., and Pless, R. Two cloud-based cues for estimating scene structure and camera calibration. *IEEE transactions on pattern analysis and machine intelligence*, 35(10):2526–2538, 2013.

Jolicoeur-Martineau, A., Piché-Taillefer, R., Combes, R. T. d., and Mitliagkas, I. Adversarial score matching and improved sampling for image generation. *arXiv preprint arXiv:2009.05475*, 2020.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Karras, T., Laine, S., and Aila, T. 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.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. 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.

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

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

Kolmogorov, A. N. On tables of random numbers. *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 369–376, 1963.

Kulikov, V., Yadin, S., Kleiner, M., and Michaeli, T. Sinddm: A single image denoising diffusion model. *arXiv preprint arXiv:2211.16582*, 2022.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986, 2022.

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

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. 2017.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. 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.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., and Aberman, K. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. 2022.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S.,

Lopes, R. G., et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022a.

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

Shaham, T. R., Dekel, T., and Michaeli, T. Singan: Learning a generative model from a single natural image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4570–4580, 2019.

Shocher, A., Bagon, S., Isola, P., and Irani, M. Ingan: Capturing and remapping the" dna" of a natural image. *arXiv preprint arXiv:1812.00231*, 2018.

Simakov, D., Caspi, Y., Shechtman, E., and Irani, M. Summarizing visual data using bidirectional similarity. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8. IEEE, 2008.

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

Skorokhodov, I., Tulyakov, S., and Elhoseiny, M. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3626–3636, 2022.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

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

Tulyakov, S., Liu, M.-Y., Yang, X., and Kautz, J. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1526–1535, 2018.

Valevski, D., Kalman, M., Matias, Y., and Leviathan, Y. Unitune: Text-driven image editing by fine tuning an image generation model on a single image. *arXiv preprint arXiv:2210.09477*, 2022.

Villegas, R., Yang, J., Hong, S., Lin, X., and Lee, H. Decomposing motion and content for natural video sequence prediction. *arXiv preprint arXiv:1706.08033*, 2017.

Villegas, R., Babaeizadeh, M., Kindermans, P.-J., Moraldo, H., Zhang, H., Saffar, M. T., Castro, S., Kunze, J., and Erhan, D. Phenaki: Variable length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399*, 2022.

Voleti, V., Jolicoeur-Martineau, A., and Pal, C. Mcvd: Masked conditional video diffusion for prediction, generation, and interpolation. *arXiv preprint arXiv:2205.09853*, 2022.

Vondrick, C., Pirsiavash, H., and Torralba, A. Generating videos with scene dynamics. *arXiv preprint arXiv:1609.02612*, 2016.

Wang, K., Wu, Q., Song, L., Yang, Z., Wu, W., Qian, C., He, R., Qiao, Y., and Loy, C. C. Mead: A large-scale audio-visual dataset for emotional talking-face generation. In *European Conference on Computer Vision*, pp. 700–717. Springer, 2020.

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Liu, G., Tao, A., Kautz, J., and Catanzaro, B. Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*, 2018.

Wang, W., Bao, J., Zhou, W., Chen, D., Chen, D., Yuan, L., and Li, H. Sindiffusion: Learning a diffusion model from a single natural image. *arXiv preprint arXiv:2211.12445*, 2022.

Xia, W., Zhang, Y., Yang, Y., Xue, J.-H., Zhou, B., and Yang, M.-H. Gan inversion: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Xu, R., Wang, X., Chen, K., Zhou, B., and Loy, C. C. Positional encoding as spatial inductive bias in GANs. *arXiv preprint arXiv:2012.05217*, 2020.

Yang, R., Srivastava, P., and Mandt, S. Diffusion probabilistic modeling for video generation. *arXiv preprint arXiv:2203.09481*, 2022.

Zhang, H., Goodfellow, I., Metaxas, D., and Odena, A. Self-attention generative adversarial networks. In *International conference on machine learning*, pp. 7354–7363. PMLR, 2019.

Appendix

A. Further Evaluation - Effect of Crop-Size and Receptive Field

Our goal is to generate outputs (image / video) that preserve global structure, are of high quality, and with large diversity. These are affected by the choice of the crop-size on which the model is trained, and the effective receptive field of the model (determined by the depth of the convolutional model and controlled via the number of ConvNext blocks in the network). As seen in Figure A2b, the largest diversity is achieved for small crop-size and small receptive field. However, small networks fail to learn the underlying image structure and result in blurry outputs (Figure A2a). We therefore use more blocks for the model. This reduces the diversity, but dramatically improves outputs quality (as is evident from Figure A2a). We choose the crop-size as a trade-off to preserve global-structure but also high diversity, which means using crop-size of about 95% of the image, with network depth of 16 blocks.

B. Comparisons

B.1. Generation from Single-Video (Table 1)

We run our comparisons on the data provided by the previous works on video generation from a single video: VGPNN (Haim et al., 2021), HP-VAE-GAN (Gur et al., 2020) and SinGAN-GIF (Arora & Lee, 2021). We follow the same methodology used in VGPNN (Haim et al., 2021).

We compare to two datasets of videos. One provided by SinGAN-GIF (Arora & Lee, 2021) and the other by HP-VAE-GAN (Gur et al., 2020). In SinGAN-GIF there are 5 videos with 8 to 15 frames, each of maximal spatial resolution 168×298. For each of the 5 input videos, each method generates 6 samples. In HP-VAE-GAN there are 10 videos each of spatial resolution 144×256. For each of the 10 input videos, each method generates 10 samples. HP-VAE-GAN and VGPNN only use the first 13 frames since their methods are limited by runtime and memory. Since learning on small amounts of data is not a goal for the task of diverse generation from a single video, and since our framework can easily learn from much more data, we train our framework on longer sequences of frames from the given input videos.

B.2. Comparison to VDM (Ho et al., 2022c)

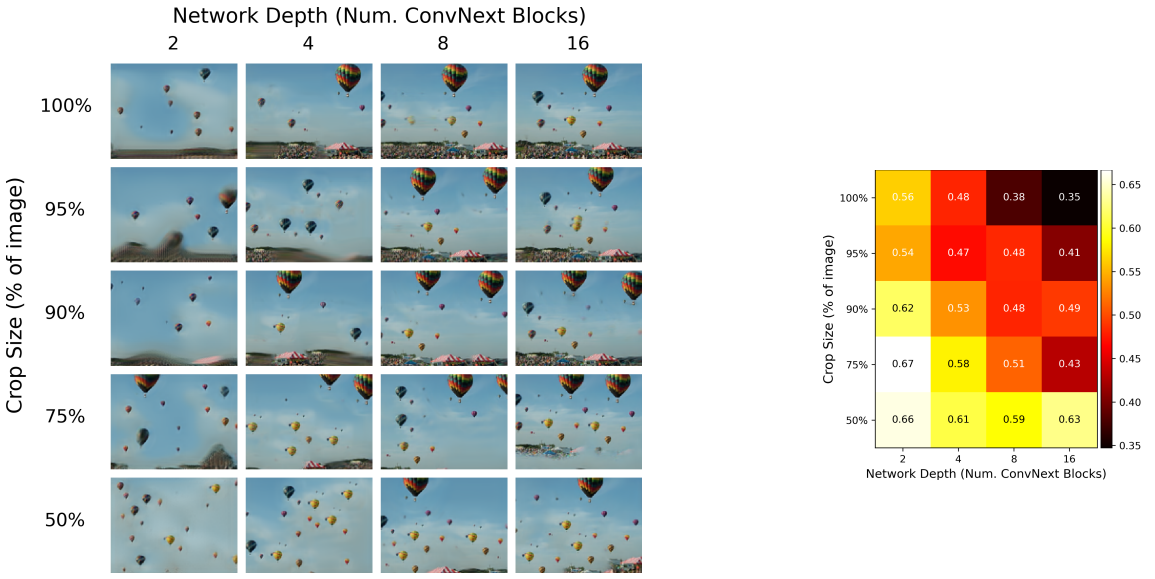
In the project page we show the results of VDM (Ho et al., 2022c) trained on a single video. Since the official implementation was not published at the time of writing, we use a third-party implementation [1]). Since VDM expects a dataset of videos, we slice a long video of 420 frames into 42 short videos of 10 frames each and let VDM train on those videos. We could only train the model on a resolution of 64×64 pixels before exceeding the memory of our GPU. We trained the model for about 150 epochs using two different learning rates (total run time was about a day on a V100 GPU). The results seems to capture the motions of the original videos, but it is difficult to evaluate since the resolution is too low compared to the original videos. The results also contain artifacts that may result from the low amount of data usually needed to train such models (without including our proposed modifications). It is qualitatively evident from the results that our framework, SinFusion, generates outputs of much higher quality when trained on a single video. For the full video results please see our project page.

B.3. Comparison to Single Image GANs

While the main focus our work is on single-video generation/manipulation tasks, we also measure the performance of our single-image DDPM on diverse image generation, in comparison to existing single-image GAN works (Shaham et al., 2019; Hinz et al., 2021). Such a quantitative comparison is presented in Table A1. We use the established Single Image FID (SIFID) metric, as well as our NNFDIV metric. We perform the comparison on the Places50 benchmark dataset. The quantitative comparison shows that our single-image DDPM achieves good performance compared to existing single image generation methods (better in one measure; worse in the other measure). While the Single Image FID achieved by our approach is slightly worse than the competing methods, we attribute that to the fact that our model generalizes beyond the internal patch distribution of the single training image (as evident in our better diversity score NNFDIV).

An additional advantage of our single-image DDPM, when compared to single-image GANs, stems from the boundary bias that exists in SinGAN and ConSinGAN (Xu et al., 2020). This induced bias causes fixed content in corner regions of the generated images, which hurts diversity. This boundary bias occurs because the discriminator (in each scale) of single-image GANs sees the same ground-truth image in all training iterations. Thus, the generator "learns" to output the same boundary as in the original image, by relying on the padding of the training image. In contrast, our single image DDPM trains on different crops of the input image, hence it does not suffer from this bias, and produces more diverse outputs, as is seen the higher diversity score in Table A1.

[1] https://github.com/lucidrains/video-diffusion-pytorch



(a) Quality

(b) NNF Diversity (See Section 7)

Figure A2. Analysing the effect of Crop-Size and Network-Depth on the diversity and quality of the generated outputs

Method	SIFID↓	NNFDIV ↑
SinGAN	0.085	0.280
ConSinGAN	0.072	0.315
SinFusion (Ours)	0.110	0.341

Table A1. **Diverse Image Generation – Comparison.**

Model	SIFID↓	NNFDIV ↑
Our single-image DDPM	0.181	0.480
Noise Prediction	0.473	0.506
w/ Up/Down sampling layers	0.145	0.258
w/ Attention layers	0.256	0.396
w/ ResNet blocks	0.246	0.463

Table A2. **Architectural changes and noise prediction ablations.** We ablate our design choices by measuring the quality (via SIFID) and diversity (via NNFDIV) generated images. The results show that our final single-image DDPM achieves the best tradeoff between generation quality and the diversity of the generated samples.

C. Ablations

Predicting Image Instead of Noise. As opposed to the standard DDPM (Ho et al., 2020) training, our single-image-DDPM model outputs a prediction for the un-noised input crop. In Figure A3 we show examples for our generated outputs for predicting the crop/image (top) against generated outputs for predicting noise (bottom). As shown, predicting the un-noised crop instead of the noise generates higher quality images. This is also evident in Table A2, where noise prediction leads to a worse SIFID score. In addition, predicting the image instead of noise also converges with much fewer training iterations, a feat we attribute to the lower complexity of the patch distribution of the training image compared to the patch distribution of random noise.

Architectural changes. We check the importance of our proposed architectural modifications to the original DDPM (Ho et al., 2020) by reverting each change and generating several images. We compare these generated images to the images generated by our final model. An example for these comparisons can be seen in Figure A4. The quantitative results for these ablations can be found in Table A2.
The first comparison shows outputs of our model with upsampling and downsampling layers. The generated outputs completely overfit the training image, and have no diversity. This is also evident in the low NNFDIV score in Table A2.
The second comparison shows outputs of our model with attention layers. Other than significantly increasing training time (to almost 2 hours per training image), the added attention decreases the quality and diversity of the generated samples, as evident in Table A2.
The third comparison shows outputs of our model with all ConvNext (Liu et al., 2022) blocks replaced with ResNet (He et al., 2016) blocks. The generated outputs suffer from smearing artifacts and are of lesser quality than our generated outputs, as also evident by the lower SIFID in Table A2.

Importance of DDPM frame Projector in diverse video generation. We show the necessity of our DDPM frame Projector model as part of the diverse video generation framework. In this ablation, we generate videos from several input videos using only the DDPM frame Predictor

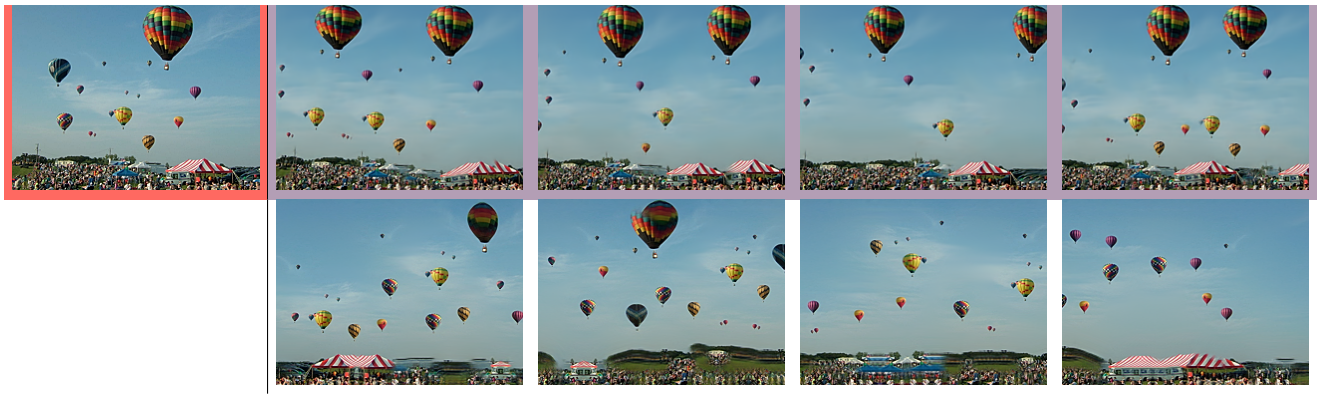


Figure A3. **Noise vs Image prediction ablation**. *Top Row*: Input image (left;red) and generated outputs of our final model (right;purple) – predicts the un-noised image crop. *Bottom Row*: Generated outputs using the standard DDPM noise prediction.

Figure A4. **Architectural ablations**. *Top Row*: Input image (left;red) and Generated outputs of our final model (right;purple).
2nd Row: Generated outputs of our model with upsampling and downsampling layers.
3rd Row: Generated outputs of our model with added attention layers (similar to the standard DDPM (Ho et al., 2020) Unet(Ronneberger et al., 2015) network).
4th Row: Generated outputs of our model where each ConvNext (Liu et al., 2022) block is replaced with a ResNet (He et al., 2016) block.

Dataset	SVFID↓	
	With Projector	No Projector
All videos in project page	**0.0066**	0.0081
HP-VAE-GAN	**0.0107**	0.0129
SinGAN-GIF	**0.0090**	0.0136

Table A3. **DDPM frame Projector ablation.** The DDPM frame Projector consistently improves the quality of the generated videos, as evident by the lower SVFID scores.

to generate frames, without using the Projector model to correct small artifacts in the generated frames. In all examples, it can be seen that the small artifacts, which remain uncorrected, accumulate over time and severely degrade the generation quality. The quantitative results can be seen in Table A3, where we measure the SVFID score of the generated videos. For qualitative video results please see the project page.

Effect of training with $k = [-3, 3]$. As written in Section 5, at inference time we always use either $k = 1$ (for forward prediction) or $k = -1$ (for backward prediction). However, we found that training the predictor with $k \in [-3, 3]$ improves the prediction for $k = \pm 1$. For example, training the predictor with only results in SVFID = 0.0112 (averaged on all videos in the supplementary), whereas training it with results in SVFID = 0.0095 (lower SVFID is better).

D. Further Explanations on Related Works

In this section we elaborate further on existing methods.

Diffusion models for Videos:

- RVD (Yang et al., 2022) tackles video prediction by conditioning the generative process on recurrent neural networks.
- RaMViD (Höppe et al., 2022) and MCVD (Voleti et al., 2022) train an autoregressive model conditioned on previous frames for video prediction and infilling using masking mechanisms.
- VDM (Ho et al., 2022c) introduces unconditional video generation by modifying the Conv2D layers in the basic DDPM UNet to Conv3D, as well as autoregressive generation.
- Imagen-Video (Ho et al., 2022a) extends VDM to text-to-video and also include spatio-temporal superresolution conditioned on upsampled versions of smaller scales.
- FDM (Harvey et al., 2022) modifies DDPM to include temporal attention mechanism and can be conditioned on any number of previous frames.

Generation from a Single Image. Generative models trained on a single image aim to generate new diverse samples, similar in appearance to the image/video on which they were trained. Most notably, SinGAN (Shaham et al., 2019) and InGAN (Shocher et al., 2018) trained multi-scale GANs to learn the distribution of patches in an image. They showed its applicability to diverse random generation from a single image, as well as a variety of other image synthesis applications (inpainting, style transfer, etc.). Their results are usually better suited to synthesis from a single image than models trained on large collection of data. More recently, GPNN (Granot et al., 2022) showed that most image synthesis tasks proposed by single-image GAN-based models (Hinz et al., 2021; Shaham et al., 2019; Shocher et al., 2018) can be solved by classical non-parametric patch nearest-neighbour methods (Efros & Leung, 1999; Efros & Freeman, 2001; Simakov et al., 2008), and achieve outputs of higher quality while reducing generation time by orders of magnitude. However, nearest-neighbour methods have a very limited notion of generalization, and are therefore limited to tasks where it is natural to "copy" parts of the input. In this respect, learning based methods like SinGAN (Shaham et al., 2019) still offer applicability like shown in the tasks of harmonization or animation.

Generation from a Single Video. Similar to the image domain, extensions of SinGAN (Shaham et al., 2019) to generation from a single *video* were proposed (Gur et al., 2020; Arora & Lee, 2021), generating diverse new videos of similar appearance and dynamics to the input video. These too, were outperformed by patch nearest-neighbour methods (Haim et al., 2021) in both output quality and speed. However, these video-based nearest-neighbour methods suffer from drawbacks similar to the image case. While the generated samples are of high quality and look realistic, the main reason for this is that the samples are essentially copies of parts of the original video stitched together. They fail to exhibit motion generalization capabilities. None of the above-mentioned methods can handle input videos longer than a few dozens frames. Single-video GAN based methods are limited in compute time (e.g., HP-VAE-GAN (Gur et al., 2020) takes 8 days to train on a short video of 13 frames), whereas VGPNN (Haim et al., 2021) is limited in memory (since each space-time patch in the output video searches for its nearest-neighbor space-time patch in the entire input video, at each iteration). In contrast, our method can handle any length of input video. While it can generalize well from just a few frames, it can also easily train on a long input video at a fixed and very small memory print, and at reasonable compute time (a few hours per video).

E. Implementation Details

Our code is implemented with PyTorch (Paszke et al., 2017). We make the following hyper-parameters choices:

- We use a batch size of 1. Each large crop contains many large "patches". Since our network is a fully convolutional network, each large "patch" is a single training example.
- We use ADAM optimizer (Kingma & Ba, 2014) with a learning rate of 2×10^{-4}, reduced to 2×10^{-5} after $100K$ iterations.
- We set the diffusion timesteps $T = 50$. This allows for fast sampling, without sacrificing image/video quality (This trade-off is simpler in our case because of the simplicity of our learned data distribution).
- When generating diverse videos, we use the DDPM frame Projector to correct predicted frames by noising and denoising $T_{corr} = 3$ steps.
- We compared several noise schedules for the diffusion models and ended up using linear noise schedule ($\beta_0 = 2 \times 10^{-3}, \beta_T = 0.4$) for single-image DDPM and cosine noise schedule (Nichol & Dhariwal, 2021) for single-video DDPM.
- Our standard network architecture consists of 16 ConvNext (Liu et al., 2022) blocks, each block with a base dimension of 64 channels.

E.1. Runtimes

On a Tesla V100-PCIE-16GB, for images/videos of resolution 144×256, our model trains for about 1.5 minutes per 1000 iterations, where each iteration is running one diffusion step on a large image crop. The total amount of iterations and total runtime for each of our models are:

- Single-Image DDPM - $50K$ iterations, total runtime of 80 minutes (good results are already seen after $15K$ iterations).
- Single-Video DDPM Frame Predictor - $200K$ iterations, total runtime of 5.5 hours.
- Single-Video DDPM Frame Projector - $100K$ iterations, total runtime of 2.5 hours
- Single-Video DDPM Frame Interpolator - $50K$ iterations, total runtime of 1.5 hours.

F. Videos Sources

In our project page we show results for video generation and and extrapolation for videos excerpts from the following YouTube videos (YouTube video IDs):
- LkrnpO5v0z8
- hj6EG7x-BT8
- nRxSUkZYeOE
- 9ePic3dtykk
- pB6XSixrCC8
- ZO5lV0gh5i4
- tmPqO_TGa-U
- bsSypB9gI0s
- RZ1kK-X3QwM
- FR5l48_h5Eo
- 4i6VSrIYRYY
- m_e7jUfvt-I
- DniKM5SKe6c
- rbzxxbuk3sk
- W_yWqFYSggc
- WA5fqO6LUUQ
- -ydgKb5K_kc

We also use several videos from MEAD Faces Dataset (Wang et al., 2020), and Timlapse Clouds Dataset (Jacobs et al., 2010; 2013).