Modelling Latent Dynamics of StyleGAN using Neural ODEs

Weihao Xia[1] Yujiu Yang[2] Jing-Hao Xue[1]
[1]Department of Statistical Science, University College London, UK
[2]Tsinghua Shenzhen International Graduate School, Tsinghua University, China

Abstract

In this paper, we propose to model the video dynamics by learning the trajectory of independently inverted latent codes from GANs. The entire sequence is seen as discrete-time observations of a continuous trajectory of the initial latent code, by considering each latent code as a moving particle and the latent space as a high-dimensional dynamic system. The latent codes representing different frames are therefore reformulated as state transitions of the initial frame, which can be modeled by neural ordinary differential equations. The learned continuous trajectory allows us to perform infinite frame interpolation and consistent video manipulation. The latter task is reintroduced for video editing with the advantage of requiring the core operations to be applied to the first frame only while maintaining temporal consistency across all frames. Extensive experiments demonstrate that our method achieves state-of-the-art performance but with much less computation. Code is available at https://github.com/weihaox/dynode_released.

1. Introduction

GAN inversion [1, 23, 37] have recently been developed, allowing us to invert various kinds of images back into the latent space of pretrained GAN models. We can then manipulate the attributes of images using latent editing methods [10, 27, 40]. But these pretrained models, inversion methods and latent editing techniques are all being trained on and developed for images, limiting their applications in video processing. Recent work [28] suggests training a GAN for video synthesis. Due to the lack of high-quality video datasets and the challenges of integrating an additional data dimension, video GANs have not been able to match their single-image counterparts. Recent methods [30, 40] have shown that even equipped with an off-the-shelf and non-temporal StyleGAN, the elusive temporal coherency, an essential requirement for video tasks to meet, can be achieved by maintaining that of the original video during inversion and latent editing processes. These methods, however, require frame-by-frame operations. The operations are almost the same across all frames, which makes us wonder: could these operations be applied just to the first frame, or specifically, could we change attributes for the entire video by only applying a latent editing method to the initial latent code?

Given a video, most existing GAN inversion methods invert consecutive frames separately without considering the temporal correlation between them. This leads to non-temporal latent codes, meaning that the independently-inverted latent codes become known but remain isolated in the latent space and their temporal relationships are not exploited. This paper approaches the question from a new perspective: we model the trajectory of (or equivalently, the temporal correlation among) the latent codes inverted independently in a GAN's latent space. More specifically, we propose a method called `DynODE`, to model the video `Dyn`amics by learning the trajectory of latent codes with neural ordinary differential equations (`ODE`s) [3]. Our work borrows the intuition from dynamical systems and treats the trajectory of a sequence as the solution to a first-order non-autonomous ODE. By considering latent codes as moving particles and the latent space as a high-dimensional dynamical system, a video sequence is seen as the discrete-time observations of a continuous trajectory of the initial latent code. The original latent space is therefore mapped into a dynamic space. The latent codes corresponding to different frames are reformulated as state transitions of the first frame, which can then be modeled by using neural ODEs.

There are many advantages to the ODE-based approaches to our problem. First, they specify a continuous latent state. The neural ODEs are encouraged to learn the holistic geometry of the video dynamic space, allowing them to produce video frames at unseen timesteps. It enables our method to perform *continuous frame interpolation*. This is especially helpful when the frames are irregular-sampled and when a temporally smooth video is required. Furthermore, the learned trajectory facilitates *consistent video manipulation* by operating primarily the first frame. Our novel setting for consistent video manipulation only applies core operations to the first frame, promoting the benefits of modeling such trajectories. Some methods have been proposed for world- or time-consistent video manipulation in a frame-by-frame setting, but to our knowl-

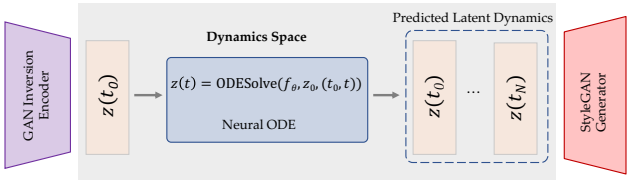


Figure 1. Our goal with DynODE is to model the latent dynamics of StyleGAN using a neural ODE.

edge, this is the first time such a task has been accomplished by focusing primarily on the first frame, without repeating operations on all frames. Our method can also be taken as a temporal counterpart of SinGAN [26]. Trained using a single image, SinGAN produces assorted high-quality images of arbitrary sizes that semantically resemble the training image but contain new object configurations and structures. Analogously, trained in a data-efficient way, our method learns the dynamics from a single video and produces arbitrary frames that contain temporally-consistent contents and motions. Extensive experiments on a wide range of datasets manifest that the proposed method learns the temporal correlation among the independently-inverted latent codes. Experimental results also show that our approach is comparable to state-of-the-art face video editing methods that perform frame-by-frame editing, but requires much fewer operations.

The novelty and contributions of our work are summarized as follows. 1) We model the video dynamics by learning the temporal trajectory of the non-temporal latent codes with neural ODEs. 2) We present a dynamical view of the latent space, in which video frames become discrete-time observations of a continuous trajectory of the initial latent code. With this perspective, existing GAN inversion and latent editing techniques can be applied to video tasks in a different manner. 3) Our method allows for infinite frame interpolation and consistent video manipulation. The latter task is introduced under a novel setting where the core operations need only be applied to the first frame. Compared with state-of-the-art methods that perform frame-by-frame editing, our approach achieves comparable performance with much less computation.

2. Related Work

Video Frame Interpolation. Video frame interpolation aims to synthesize intermediate frames between two consecutive video frames. The key of video interpolation is to model the motion and generate the intermediate frame. In [17], Meyer *et al*. propose a phase-based video interpolation scheme that demonstrates impressive performance on videos with small displacements. Niklaus *et al*. [19] develop a kernel-based framework to predict a sampling kernel for every pixel and interpolate each pixel by convolution on corresponding patches of adjacent frames. Many studies [5, 14] use optical flows to interpolate videos with large motions. Different from existing methods, we present a data-efficient method that models video dynamics from a single video and allows for infinite frame interpolation.

Consistent Video Manipulation. Despite steady progress in image editing, video editing remains difficult since such edits need to be applied consistently to all frames. Kasten *et al*. [9] present a way to edit natural videos by parameterizing them through layered neural 2D atlases. Mallya *et al*. [16] convert semantic inputs into time- and view-consistent videos, based on projections from the point cloud to the camera. More recently, some methods [30, 39, 40] based on GAN inversion [1, 2, 23, 37] and latent editing [27, 40] have recently been proposed. They follow a similar pipeline of altering the inverted latent code of each aligned-and-cropped frame. The same operations are applied repeatedly to the latent code for each frame. We observe that such frame-by-frame processing contains lots of redundancy in operations. By removing the redundant operations from the aforementioned methods, our framework achieves comparable performance at a considerably lower cost. The produced video remains highly consistent in spite of applying the core operations to the first frame only. In contrast with previous studies, we propose a method for editing video in a time-coherent manner without the need to apply redundant operations frame-by-frame.

Video Applications of Neural ODEs. Neural ODE [3] and its variants have recently been explored for video generation.Kanaa *et al*. [6] combine a typical encoder-decoder architecture with Neural ODEs. Park *et al*. [20] propose an ODE convolutional GRU as the encoder for continuous-time video generation. Unlike [6, 20], we use a neural ODE network to model the temporal correlation among latent codes, which are derived separately by using existing GAN inversion algorithms. Our work have significant differences from them in at least three key aspects: 1) *Task*. Their methods are proposed for video generation while our work is for consistent video interpolation and manipulation; 2) *Mechanism*. They train an encoder-decoder with neural ODEs, while the only component of our method that needs training is the neural ODE network; and 3) *Applicability*. Our method can handle videos at resolutions up to 1024×1024. This presents a substantial improvement over previous methods, results of which are typically of much lower resolutions. More importantly, our method is easily applicable to current video editing methods, freeing them from tedious frame-by-frame processing.

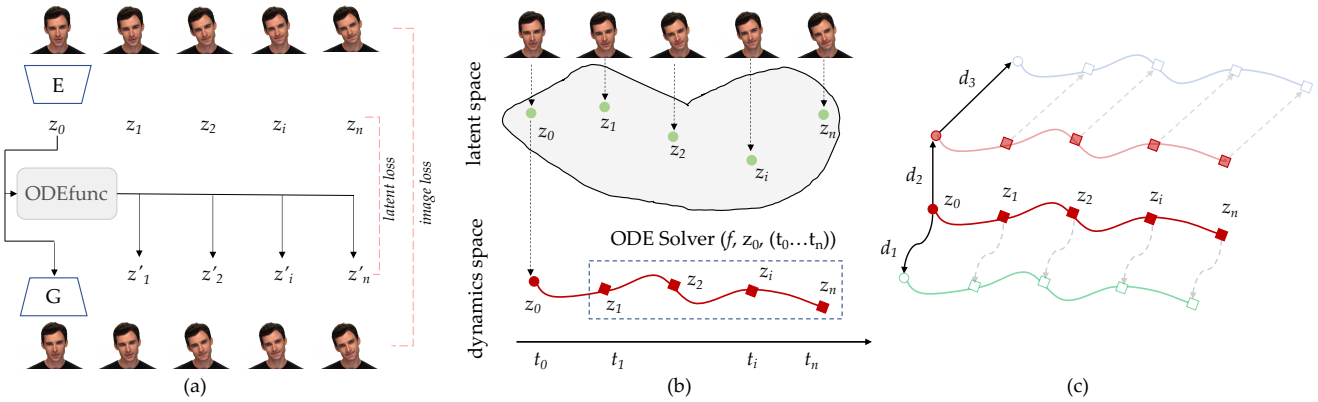


Figure 2. Framework of the proposed DynODE. **(a)** The neural ODE network (ODEfunc, f_θ) is trained to predict the subsequent latent states given the initial state by minimizing the losses in the latent, feature, and image spaces. **(b)** The original latent space is mapped into a dynamic space. These latent codes $\{\mathbf{z}_0, \cdots, \mathbf{z}_n\}$ in the latent space becomes discrete-time observations $\{\mathbf{z}(t_0), \cdots, \mathbf{z}(t_n)\}$ of a continuous trajectory of the initial latent state in the dynamic space. The latent codes corresponding to different frames are reformulated as state transitions of the first frame, which can be modeled by using neural ODEs. **(c)** The desired video attributes could be edited by changing the first frame and extending such modifications to all subsequent frames. These attributes could be altered by applying a direction, either nonlinear (d_1) or linear (d_2 and d_3), to the initial latent code.

3. Method

3.1. Preliminary

GAN Inversion aims to invert a given image back into the latent space of a pretrained GAN model so that the image can be faithfully reconstructed from the inverted code by the generator. The generator of an unconditional GAN learns the mapping $\mathcal{G} : \mathcal{Z} \rightarrow \mathcal{X}$. When $\mathbf{z}_1, \mathbf{z}_2 \in \mathcal{Z}$ are close in the $\mathcal{Z}$ space, the corresponding images $x_1, x_2 \in \mathcal{X}$ are visually similar. GAN inversion maps data x back to latent representation $\mathbf{z}^*$ or, equivalently, finds an image x^* that can be entirely synthesized by the well-trained generator $\mathcal{G}$ and remain close to the real image x. Formally, denoting the signal to be inverted as $x \in \mathbb{R}^n$, the well-trained generative model as $\mathcal{G} : \mathbb{R}^{n_0} \rightarrow \mathbb{R}^n$, and the latent vector as $\mathbf{z} \in \mathbb{R}^{n_0}$, GAN inversion studies the following problem:

$$\mathbf{z}^* = \arg\min_{\mathbf{z}} \ell(\mathcal{G}(\mathbf{z}), x), \tag{1}$$

where $\ell(\cdot)$ is a distance metric in the image or feature space, and $\mathcal{G}$ is assumed to be a feed-forward neural network. With the inverted $\mathbf{z}^*$, we can obtain the original and manipulated images. These pretrained GAN models, the inversion methods, and the latent editing techniques are trained on and developed for images, limiting their applications in video processing. *Instead of developing alternative GAN architectures or inversion methods for videos, we model the trajectory of the isolated latent codes to apply existing inversion techniques to video tasks.*

Neural ODEs are a family of continuous-time models which define a hidden state $h(t)$ to be the solution to an ODE initial-value problem:

$$\dot{h}(t) = \frac{dh(t)}{dt} = f_\theta(h(t), t; \theta) \quad s.t. \quad h(t_0) = h_0. \tag{2}$$

The function f_θ specifies the dynamics of the hidden state, using a neural network with parameters θ. $t \in [0, T]$ is time and $h(t) \in \mathbb{R}^d$. The hidden state $h(t)$ is defined at all times, and can be evaluated at any desired time step by using a numerical ODE solver denoted as ODESolve:

$$h_0, \ldots, h_N = \texttt{ODESolve}(f_\theta, h_0, (t_0, \ldots, t_N)), \tag{3}$$

where $(t_0, \ldots, t_N)$ are time points where $h(t)$ is evaluated. The gaps between consecutive time points t_i are not necessarily equal. Neural ODEs specify $h(t)$ as a continuous function over time, even though it is evaluated at discrete time points $(t_0, \ldots, t_N)$.

3.2. DynODE

Given the inverted codes $\mathbf{z}_0, \cdots, \mathbf{z}_n$ corresponding to each frame in the sequence $c_0, \cdots, c_n$, our objective is to model the trajectory or dynamics of a video sequence in the latent space. This specifically targets the latent dynamics of StyleGAN, illustrated in **??**. If we treat the latent space as a dynamic system, change in latent codes along a certain direction is analogous to the motion of particles. Our work borrows intuition from dynamical systems and treats the trajectory of the entire sequence as the solution to a first-order non-autonomous ODE. By considering the latent space as a dynamical system and modeling latent codes as moving particles, the entire sequence can be viewed as discrete-time observations of a continuous trajectory of the initial latent

code $\mathbf{z}_0$. The subsequent latent codes become state transitions of the initial one. Therefore, the dynamics of the entire video is actually determined by the first frame and its trajectory.

We consider the dynamics of a state $\mathbf{z}(t)$ in the phase space Ω ($= \mathbb{R}^{2n}$) of a dynamical system. These non-temporal latent codes $\{\mathbf{z}_0, \cdots, \mathbf{z}_n\}$ become observations $\{\mathbf{z}(t_0), \cdots, \mathbf{z}(t_n)\}$ of a motion trajectory of $\mathbf{z}_0$ at specified times $t_0, \cdots, t_n$. $\mathbf{z}(t_0)$ represents the initial state equal to $\mathbf{z}_0$. This trajectory can be treated as the solution to a non-autonomous dynamical system determined by

$$\frac{d\mathbf{z}}{dt} = f\left(\mathbf{z}(t), t\right) \quad \text{for } t \in \mathbb{R},\ \mathbf{z} \in \Omega, \tag{4}$$

where $f : \Omega \times \mathbb{R} \mapsto T\Omega$ is assumed to be continuous, and $T\Omega$ is the tangent space. By approximating the differential with an estimator $f_\theta \simeq f$, where f_θ is a θ-parameterized neural network, the neural ODEs allow to model the evolution across time of such a dynamical system and learn the dynamics or trajectories from relevant data. For an arbitrary time t_i, the `ODESolve` computes a numerical approximation of the integral of the dynamics from the initial time value t_0 to t_i by Eq. (4) that is equal to

$$\begin{aligned} \mathbf{z}_i = \tilde{\mathbf{z}}(t_i) &= \texttt{ODESolve}\left(f_\theta, \mathbf{z}_0, (t_0, t_i)\right) \\ &\simeq \mathbf{z}_0 + \int_{t_0}^{t_i} f_\theta\left(\mathbf{z}(t), t\right) dt = \mathbf{z}(t_i), \\ \text{where} \quad \mathbf{z}_0 &= \mathcal{E}(c_0), \quad \hat{c}_i = \mathcal{D}(\mathbf{z}_i). \end{aligned} \tag{5}$$

$\tilde{\mathbf{z}}(t_i)$ is a prediction of $\mathbf{z}(t_i)$ using a neural ODE network.

Fig. 2 (a) shows the procedure of our framework[1]. For a source video consisting of N frames $\{x_i\}_{i=1}^N$, we first obtain the preprocessed frames $\{c_i\}_{i=1}^N$ after a cropping-and-alignment step. We then use an off-the-shelf GAN inversion method, which is denoted by $\mathcal{E}$, to produce their latent inversion $\{\mathbf{z}_i\}_{i=1}^N = \{\mathcal{E}\left(c_i\right)\}_{i=1}^N$ for all frames. The reconstructed image can be generated from latent code $\tilde{\mathbf{z}}_i$ with a generator $\mathcal{G}$ by $\mathcal{G}(\tilde{\mathbf{z}}_i)$. These latent codes are then used to train a neural ODE network (`ODEfunc`, f_θ), which aims to predict the subsequent frames. Specifically, we randomly sample a small batch with the same size, $\mathbf{z}_0, \cdots, \mathbf{z}_n$ ($n < N$), for each iteration. Given the initial state $\mathbf{z}_0$, the neural ODE network is trained to predict the known observations, $\mathbf{z}_1, \cdots, \mathbf{z}_n$, at the corresponding times by minimizing the losses in the latent, feature, and image spaces. The obtained latent codes are then fed into the respective generators in order to produce images. Notice that the generators for the latent codes of the entire video might be different, for instance, if the generators are fine-tuned for each latent code as in [24]. The same applies to the GAN inversion module $\mathcal{E}$. The accurate description for $\mathcal{E}$ and $\mathcal{G}$ in Fig. 2 (a) should be $\mathcal{E}_1, \cdots, \mathcal{E}_N$ and $\mathcal{G}_1, \cdots, \mathcal{G}_N$, respectively. One for each is illustrated in the figure for simplicity. This means that our approach is not reliant on any specific methods of GAN inversion, latent editing, or any other related techniques, and thus can be implemented as a plug-and-play module for video dynamic modeling.

Fig. 2 (b) depicts the mapping from the spatial latent space to the temporal dynamic space. By considering the latent space as a high-dimensional dynamical system and latent codes as moving particles, the non-temporal latent codes in the latent space becomes discrete-time observations of a continuous trajectory of the initial latent state in the dynamic space. The latent codes corresponding to different frames are reformulated as state transitions of the first frame. We model the dynamics of these isolated latent codes in the latent space and learn a temporal trajectory in the dynamic space using a neural ODE function. Once trained, dividing the time interval by k, the neural ODE network produces the states at the specified time points in the form of $\mathbf{z}_0, \ldots, \mathbf{z}_k = \texttt{Solve}(f_\theta, \mathbf{z}_0, (t_0, \ldots, t_k))$. Furthermore, with this learned trajectory, as illustrated in Fig. 2 (c), it is possible to perform consistent video manipulation, meaning to change the desired attributes of the target subject in a video by merely altering the first frame and consistently extending those modifications to all subsequent frames. As aforementioned, the editing in the initial frame could be easily done by applying a desired direction to its latent code. The direction (marked as the solid arrow lines in the figure), either nonlinear (d_1) or linear (d_2 and d_3), could be obtained by the off-the-shelf latent editing approaches [27, 40] or some recent language-driven image manipulation methods [21, 35, 36]. This is thought to be equivalent to applying particular directions from latent editing methods (dashed arrow lines) to all latent codes observed at certain time points.

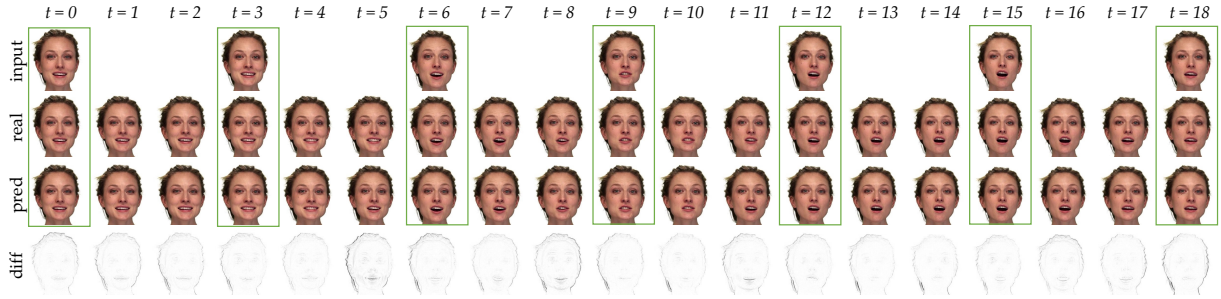
4. Experiments

This section describes datasets, models, and results in terms of three aspects: video dynamic modeling, infinite frame interpolation, and consistent video manipulation.

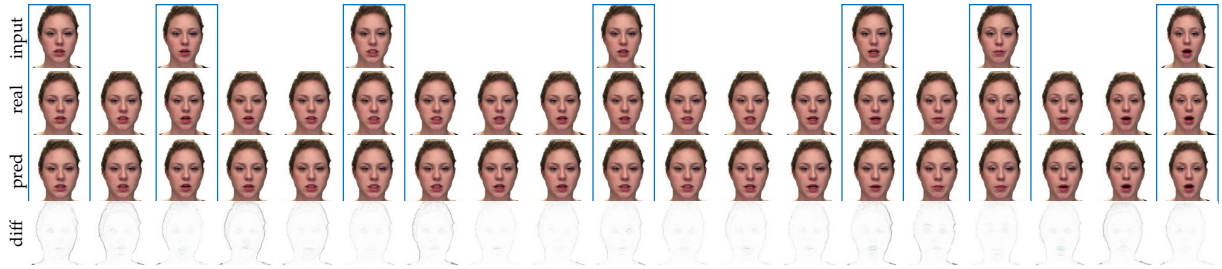
4.1. Experimental Settings

Pretrained models and datasets. Given their widespread popularity, most of the GAN inversion and latent editing methods focus on StyleGANs [7, 8]. Therefore, we use StyleGAN2 [8] as the pretrained model in our experiments. Our method is not dependent on StyleGANs and can be applied to any GAN model. The latent codes are obtained by inverting each frame into $\mathcal{W}^+$ space of StyleGAN2 [8]. Other latent spaces [37], *e.g.* $\mathcal{Z}$ or $\mathcal{W}$ space, are also supported. Experiments are conducted on several categories of

[1] Despite a similar illustration with [6], the methodological and training mechanisms are fundamentally different.



(a) regularly-sampled inputs



(b) irregularly-sampled inputs

Figure 3. Qualitative results of dynamic modeling at both observed and unobserved times. We sample frames at regular and irregular time intervals and compare the predicted frames with the actual ones at both observed and unobserved time points.

Dataset	Face		Scene		Bird		Isaac3D	
Metric	MSE ↓	SSIM ↑	MSE ↓	SSIM ↑	MSE ↓	SSIM ↑	MSE ↓	SSIM ↑
Observed	6.752	98.9	22.956	97.6	25.407	98.1	15.655	99.2
Unobserved	35.397	93.2	48.323	92.4	55.368	90.3	27.651	96.5

Table 1. Quantitative evaluation of dynamic modeling on four datasets. The performance of dynamic modeling here is defined as the reconstruction quality of predicted images at both observed and unobserved time points. Metrics are reported in MSE ↓ (×e-3) and SSIM ↑. ↓ means lower is better, while ↑ means the opposite.

publicly available datasets to demonstrate the effectiveness of our proposed method.

- **Face.** The StyleGAN2 model is trained on LLHQ [7] at a resolution of 1024×1024. The real face videos are collected from publicly available talking-head datasets [15, 33] or downloaded from YOUTUBE.
- **Scene.** The StyleGAN2 is trained using Place365 [41] at the resolution of 256×256. We use [10] to generate temporally-consistent frames by editing the time-varying attributes, *e.g.*, NIGHT, DAWNDUSK, and SUNRISESUNSET. We also download real videos of outdoor natural scenes from YOUTUBE and obtain latent codes of sampled frames by direct optimization.
- **Bird.** The StyleGAN2 model is trained using CUB-200-2011 dataset [32] at the resolution of 256×256. We collect bird videos from YOUTUBE.
- **Isaac3D.** The StyleGAN2 model is trained using Isaac3D [18] with a resolution of 128×128. The dataset contains nine factors of variation, such as background color, object shape, robot movement, and camera height. As it is a synthetic dataset, we generate consecutive frames by moving the robot or camera.

Implementation details. We implement the proposed method in PyTorch on an Nvidia GeForce RTX 2080. The neural ODE function is parameterized by a Multi-Layer Perceptron (MLP). The parameters in the neural ODE function are optimized using the Adam optimizer [12]. We train the neural ODE network with 5000 gradient descent steps with a learning rate of $0.01, \beta_1 = 0.9, \beta_2 = 0.999$, and $\epsilon = 1e^{-8}$. We use a adaptive-step DOPRI5 (Runge-Kutta [4] of order 5 of Dormand-Prince-Shampine) as the default ODE Solvers.

Performance metrics. The performance metrics for quantitative comparisons are divided into two groups. To evaluate dynamic modeling, we use pixel-wise Mean Square Error (MSE) and Structural Similarity Index (SSIM) [34]. For comparisons of consistent video manipulation, we focus on evaluating the temporal coherence of edited videos, which is measured by the identity similarity between the frame

Figure 4. Results of continuous frame interpolation for talking faces and outdoor natural scenes. Based on given frames in (a), our method can generate in-between video frames in diverse time intervals.

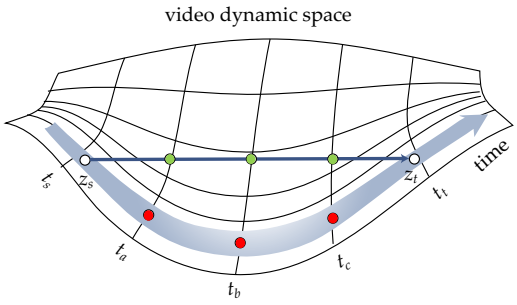


Figure 5. In the video dynamic space, the direct morphing creates the intermediate states (green) by accumulating the differences between the two states at t_s and t_t. In contrast, neural ODEs estimate in-between states (red) at unseen times by accounting for the holistic geometry of the video dynamics. These states produce time-oriented and motion-coherent frames.

pairs as in [30]. To evaluate temporal consistency of facial identity, we use two metrics, temporally-local (TL-ID) and temporally-global (TG-ID) identity preservation introduced in [30], and a variant of the Average Content Distance (ACD) [29]. We use Fréchet Video Distance (FVD) [31] to evaluate the quality of motion.

4.2. Dynamic Modeling

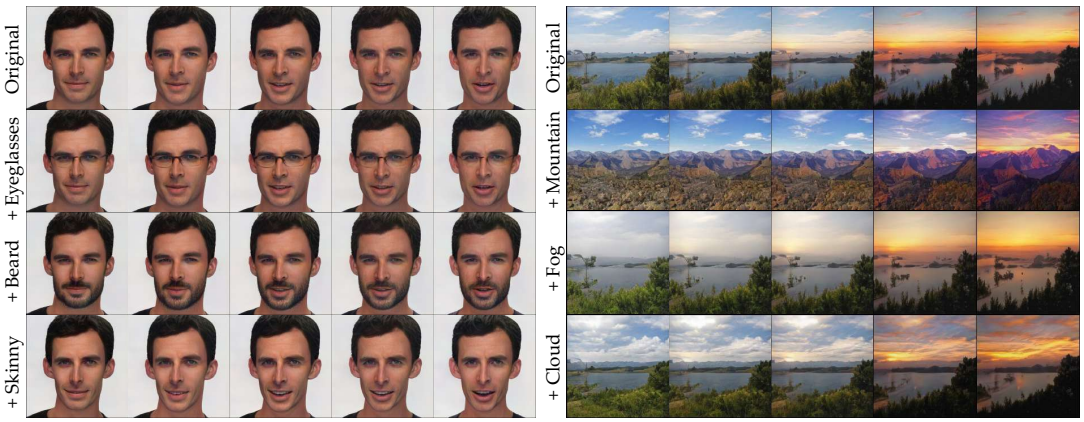
We conduct experiments to evaluate the performance of video dynamic modeling. Neural ODEs are expected to learn the temporal properties of a trajectory, allowing it to approximate the actual states at the observed time points and even those between observations. Thus, the performance of dynamic modeling can be understood as the reconstruction quality of predicted images. Specifically, we sample frames at regular and irregular time intervals and compare the predicted frames with the actual ones at both observed and unobserved time points. Fig. 3 shows the sampled frames of the original videos and the predicted results. Tab. 1 shows the performance of dynamic modeling characterized by the reconstruction quality. Considering that images would be perfectly fitted if given enough time, we set a fixed number of iterations to maintain a reasonable running time. The quantitative evaluation on four categories demonstrates that our method not only models the video dynamics for the known observations but also generalizes to the unobserved time. This capability indicates that neural ODEs are able to learn the entire geometry of video dynamic rather than simply remembering given observations, enabling them to accept irregularly sampled frames or create video frames at unknown times.

There are some ways to analyze the spatio-temporal dynamics of videos in addition to ODE-based models. Recent studies [11, 20] have demonstrated that they are not as effective as ODE-based models. The RNN-based models, for example, assuming fixed time intervals, are limited to learn the representations only at the observed times. These methods can barely handle datasets collected from wild environments with irregular time intervals and missing states.

4.3. Continuous Frame Interpolation

The learned neural ODE specifies $z(t)$ as a continuous function over time, which facilitates infinite frame interpolation. It enables our framework to interpolate non-existent frames within the time interval from $t = 0$ to $t = N$ at arbitrary time steps. Once trained, dividing the time interval $[0, N]$ by k, the neural ODE network produces the in-between states at the corresponding time points in the form of $\mathbf{z}_0, \ldots, \mathbf{z}_k = \texttt{Solve}(f_\theta, \mathbf{z}_0, (t_0, \ldots, t_k))$. The intrinsic properties of neural ODEs allow to achieve such infinite frame interpolation at a constant memory cost even when the frames are irregularly sampled or partially observed, which is the case that their RNN-based counterparts are often struggling to deal with [11,25]. This is particularly helpful when a temporally smooth video is required. Fig. 4 shows the results of frame interpolation for talking heads and outdoor natural scenes at arbitrary time steps. While the generators are trained on image datasets and not specifically tuned for the video, the predicted frames still exhibit high consistency across time.

It should be noted that intermediate video frames can also be generated by simply blending two adjacent latent codes. This procedure, often called image morphing in literature, is to fuse two images by interpolating their latent codes, which also presents a continuous process. Given z_s



(a) Face (b) Scene

Figure 6. Results of consistent video manipulation for talking heads and outdoor natural scenes. Our method changes the desired attributes of the entire video by altering the initial frame and extending such modifications to the entire sequence, without the need to apply redundant operations to every frame. The manipulated frames of the entire video show identical video dynamics and maintain temporal coherence, even when the facial identity in the first frame appears to have drifted after editing.

and z_t, a series of semantically meaningful images can be generated following $z^* = z_s + \alpha(z_t - z_s)$, where α is a scale between 0 and 1. The term $(z_t - z_s)$ can also be considered as a direction, similar to those discovered by latent editing methods [10, 27, 40]. As opposed to the trajectory learned by neural ODEs, which could be viewed as the *temporal* directions, these are *spatial* or *manipulatable* directions for altering the semantics. Fig. 5 (adopted from [20]) illustrates a video dynamic space from t_s to t_t. The 2D instead of 1D structure of such space indicates the stochasticity of the trajectory between the two observed time points. The difference between direct morphing and ours is obvious in the video dynamic space. The direct morphing creates the intermediate states (green) at t_a, t_b, and t_c by accumulating the differences between $\mathbf{z}_s$ and $\mathbf{z}_t$, without considering the geometry structure of the video dynamic space. The obtained states may fall outside of the video dynamic space and thus produce frames that contain spatial changes instead of temporal motions. In contrast, our method estimates in-between states (red) at unknown times by accounting for the holistic geometry of the video dynamics. These states produce time-oriented and motion-coherent frames. The trajectory obtained from our method, illustrated as the blue arrow curve, fits closely with the video dynamic space and indicates the intrinsic motion of the video.

Due to the inherent limitations of current GANs [8], our method is suited more to videos with a specific category and may not perform well on existing video frame interpolation benchmarks that include different objects or scenes. The existing frame interpolation methods [5, 14, 19, 38] could not be trained for specific categories due to the lack of such high-quality video datasets. As a result, we do not conduct comparisons of video interpolation. The comparison will be applicable if either constraint is lifted in the future.

4.4. Consistent Video Manipulation

Baselines. We compare our method with several state-of-the-art GAN inversion based video editing baselines [30, 40]. These recently-proposed methods follow a pipeline that consists of three key steps: pre-processing (inverting each cropped and aligned frame into the latent space); attribute manipulation (editing images by employing off-the-shelf latent based semantic editing techniques), and seamless cloning (blending the modified faces with the original input frames using [22]). To maintain the inherent temporal consistency of the original video, Tzaban *et al*. [30] further include an Pivotal Tuning Inversion (PTI) module [24] to produce faithful inversions by fine-tuning the generator and introduce a stitching-tuning operation that further tunes the generator to provide spatially-consistent transitions. Xu *et al*. [39] propose to improve inversion quality by utilizing a sequence of frames instead of a single image. We compare with [39] to see how this video inversion scheme contributes to video editing.

Qualitative evaluation. We demonstrate the effectiveness of our method on a range of in-the-wild videos gathered from popular publicly available content. These include diverse and challenging scenarios characterized by complex backgrounds and considerable movement. Fig. 6 shows results of consistent video editing on talking heads and outdoor natural scenes. The target attribute is edited by employing off-the-shelf latent-based semantic editing techniques (*i.e*., [40] for faces and [10] for others) on the first frame. Our method changes the desired attributes of the entire video by altering the initial frame and extending such modifications to the entire sequence, without the need to ap-

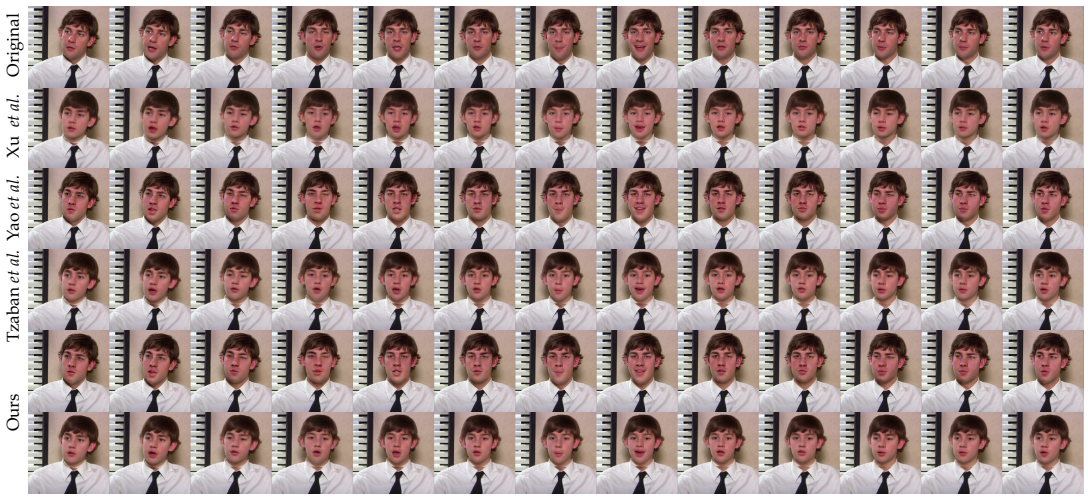


Figure 7. Comparison with recent face video editing methods, including Yao *et al*. [40], Tzaban *et al*. [30], and Xu *et al*. [39]. The last two rows are obtained through plug-and-play integration of our method into existing video editing pipelines [30, 40] respectively.

	Yao *et al*. [40]	Xu *et al*. [39]	Tzaban *et al*. [30]	Ours ([40]/ [30])
TL-ID ↑	0.957	0.969	<u>0.989</u>	0.965/**0.991**
TG-ID ↑	0.839	0.851	**0.912**	0.843/<u>0.907</u>
ACD ↓	1.352	1.485	**0.846**	1.331/<u>0.929</u>
FVD ↓	582.4	632.7	**352.9**	522.3/<u>378.2</u>

Table 2. Quantitative comparison with recent face video editing methods. We reported results of identity preservation metrics and FVD. ↑ means higher is better, while ↓ means the opposite.

ply redundant operations to every frame. The edited video displays identical dynamics and maintains temporal coherence, even when the first frame after editing shows drifted facial identities in some cases (*e.g*. SKINNY). The drifted identities result from the employed latent-based editing process [40] and can be resolved by using alternative methods.

The comparison results of facial attribute editing on videos are shown in Fig. 7. We compare several state-of-the-art GAN inversion based video editing baselines [30, 39, 40]. These results demonstrate highly temporal coherence across all the frames. The reason for this is that the temporal consistency of the source video is maintained during inversion and latent editing. The reconstructed video would be predisposed towards temporal consistency if the image-based GAN inversion methods can faithfully reconstruct each frame, even without considering the temporal relationship between adjacent frames. Furthermore, despite minor inversion inconsistencies introduced by the inversion module, especially those of the backgrounds, the final results would still be smooth in these regions. The blending process only stitches the masked facial regions back to the original images and the inconsistent backgrounds are not involved. The video-based inversion method [39], on the other hand, does not provide extra feasibility for the task of video editing compared with its image-based counterparts. The last two rows are results from implementing our method as a plug-and-play module for dynamic modeling in two existing face video editing pipelines [40] and [30], respectively. Note that our results are obtained by applying core operations only to the first frame, freeing the aforementioned video editing methods from repeating tedious and redundant operations on all frames. The desired video attributes could be edited by altering the first frame and extending such changes to the entire sequence, as we expected, without deteriorating the temporal consistency across all the frames.

Quantitative evaluation. Since the results of all compared methods are generated by the same generator, the evaluation on image quality using the popular metric, *e.g*., FID or LPIPS, are not very meaningful. Instead, we evaluate the temporal coherence and the motion quality of edited videos. The temporal coherence is measured by the identity similarity between the frame pairs. TL-ID computes the identity similarity between adjacent video frames. TG-ID refers to the identity similarity between all possible pairs of video frames. We also evaluate the temporal consistency of facial identity using a variant of ACD [29]. We use FVD [31] to evaluate the quality of motion.

The quantitative evaluation results are shown in Tab. 2. Our method achieves state-of-the-art performance when embedded as a plug-and-play dynamic modeling module in two face video editing frameworks [40] and [30], respectively. In contrast to other video editing studies, our method avoids repeating operations on every frame, without sacrificing editing quality or temporal consistency.

5. Discussion

Limitations and future directions. Our framework has several limitations. For example, we use the simplest implement of the neural differential equations to show their potential applications. From the vast variants, we can provide the dynamic adjustment to the trajectory based on future observations using the neural controlled differential equations (CDEs) [11], or introduce the stochasticity using the neural stochastic differential equations (SDEs) [13]. Furthermore, since the dynamics is modeled from one single video, the learned trajectory is deterministic, which contradicts the stochastic nature of the time-varying videos. The problem can be addressed by training an encoder on a large-scale video dataset. The learned encoder would be capable of stochastic video prediction based on the first frame.

Conclusion. In this paper, we present `DynODE`, a method to model video dynamics by learning the trajectory of independently-inverted latent codes using neural ODEs. Our method estimates time-oriented and motion-coherent frames at unseen timesteps by accounting for the holistic geometry of the video dynamic space. Such a design enables continuous frame interpolation and consistent video manipulation, freeing video editing from tedious and redundant frame-by-frame processing. Extensive experiments on a wide range of datasets show that our method improves upon prior state-of-the-art methods.

References

[1] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In *ICCV*, pages 6711–6720, 2021.

[2] Qingyan Bai, Yinghao Xu, Jiapeng Zhu, Weihao Xia, Yujiu Yang, and Yujun Shen. High-fidelity GAN inversion with padding space. In *ECCV*, 2022.

[3] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *NeurIPS*, 2018.

[4] John R Dormand and Peter J Prince. A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980.

[5] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super SloMo: High quality estimation of multiple intermediate frames for video interpolation. In *CVPR*, pages 9000–9008, 2018.

[6] David Kanaa, Vikram Voleti, Samira Ebrahimi Kahou, and Christopher Pal. Simple video generation using neural ODEs. In *NeurIPS Workshop*, 2019.

[7] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019.

[8] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *CVPR*, 2020.

[9] Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing. *TOG*, 2021.

[10] Valentin Khrulkov, Leyla Mirvakhabova, Ivan Oseledets, and Artem Babenko. Latent transformations via NeuralODEs for GAN-based image editing. In *ICCV*, pages 14428–14437, 2021.

[11] Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural Controlled Differential Equations for Irregular Time Series. In *NeurIPS*, 2020.

[12] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[13] Xuechen Li, Ting-Kam Leonard Wong, Ricky TQ Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. In *AISTATS*, pages 3870–3882, 2020.

[14] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *CVPR*, pages 4463–4471, 2017.

[15] Steven R Livingstone and Frank A Russo. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, 13(5):e0196391, 2018.

[16] Arun Mallya, Ting-Chun Wang, Karan Sapra, and Ming-Yu Liu. World-consistent video-to-video synthesis. In *ECCV*, 2020.

[17] Simone Meyer, Oliver Wang, Henning Zimmer, Max Grosse, and Alexander Sorkine-Hornung. Phase-based frame interpolation for video. In *CVPR*, pages 1410–1418, 2015.

[18] Weili Nie, Tero Karras, Animesh Garg, Shoubhik Debnath, Anjul Patney, Ankit Patel, and Animashree Anandkumar. Semi-supervised StyleGAN for disentanglement learning. In *ICLR*, 2020.

[19] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *ICCV*, 2017.

[20] Sunghyun Park, Kangyeol Kim, Junsoo Lee, Jaegul Choo, Joonseok Lee, Sookyung Kim, and Edward Choi. Vid-ode: Continuous-time video generation with neural ordinary differential equation. In *AAAI*, 2021.

[21] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. StyleCLIP: Text-driven manipulation of StyleGAN imagery. In *ICCV*, pages 2085–2094, 2021.

[22] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *SIGGRAPH*, pages 313–318, 2003.

[23] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a StyleGAN encoder for image-to-image translation. In *CVPR*, 2021.

[24] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *TOG*, 2022.

[25] Yulia Rubanova, Ricky TQ Chen, and David Duvenaud. Latent ODEs for irregularly-sampled time series. In *NeurIPS*, 2019.

[26] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. SinGAN: Learning a generative model from a single natural image. In *ICCV*, pages 4570–4580, 2019.

[27] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of GANs for semantic face editing. In *CVPR*, 2020.

[28] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *CVPR*, 2022.

[29] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN: Decomposing motion and content for video generation. In *CVPR*, 2018.

[30] Rotem Tzaban, Ron Mokady, Rinon Gal, Amit H Bermano, and Daniel Cohen-Or. Stitch it in time: Gan-based facial editing of real videos. *arXiv preprint arXiv:2201.08361*, 2022.

[31] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

[32] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[33] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *CVPR*, pages 10039–10049, 2021.

[34] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *TIP*, 13, 2004.

[35] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. TediGAN: Text-guided diverse face image generation and manipulation. In *CVPR*, pages 2256–2265, 2021.

[36] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. Towards open-world text-guided face image generation and manipulation. *arXiv preprint arXiv:2104.08910*, 2021.

[37] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. GAN inversion: A survey. *TPAMI*, 2022.

[38] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. In *NeurIPS*, pages 1647–1656, 2019.

[39] Yangyang Xu, Yong Du, Wenpeng Xiao, Xuemiao Xu, and Shengfeng He. From continuity to editability: Inverting gans with consecutive images. In *ICCV*, pages 13910–13918, 2021.

[40] Xu Yao, Alasdair Newson, Yann Gousseau, and Pierre Hellier. A latent transformer for disentangled face editing in images and videos. In *ICCV*, pages 13789–13798, 2021.

[41] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *CVPR*, 2017.