Continuous-Time Video Generation via Learning Motion Dynamics with Neural ODE

Kangyeol Kim*[1,7]
kangyeolk@kaist.ac.kr
Sunghyun Park*†[1,2]
psh01087@kaist.ac.kr
Junsoo Lee[3]
junsoolee93@webtoonscorp.com
Joonseok Lee[4,5]
joonseok2010@gmail.com
Sookyung Kim[6]
sookim@parc.com
Jaegul Choo[1,7]
jchoo@kaist.ac.kr
Edward Choi[1]
edwardchoi@kaist.ac.kr

[1] KAIST AI
Korea
[2] Kakao Enterprise
Korea
[3] Naver Webtoon
Korea
[4] Seoul National University
Korea
[5] Google Research
United States
[6] Xerox PARC
United States
[7] Letsur Inc.
Korea

Abstract

In order to perform unconditional video generation, we must learn the distribution of the real-world videos. In an effort to synthesize high-quality videos, various studies attempted to learn a mapping function between noise and videos, including recent efforts to separate motion distribution and appearance distribution. Previous methods, however, learn motion dynamics in discretized, fixed-interval timesteps, which is contrary to the continuous nature of motion of a physical body. In this paper, we propose a novel video generation approach that learns separate distributions for motion and appearance, the former modeled by neural Ordinary Differential Equation (ODE) to learn natural motion dynamics. Specifically, we employ a two-stage approach where the first stage converts a noise vector to a sequence of keypoints in arbitrary frame rates, and the second stage synthesizes videos based on the given keypoints sequence and the appearance noise vector. Our model not only quantitatively outperforms recent baselines for video generation, but also demonstrates versatile functionality such as dynamic frame rate manipulation and motion transfer between two datasets, thus opening new doors to diverse video generation applications.

1 Introduction

Creating realistic videos from scratch (*i.e.*, unconditional video generation (UVG)) requires the model to learn the distribution of the video data. In other words, we are essentially learn-

* indicates equal contribution.
† This work was done during an internship at Kakao Enterprise.

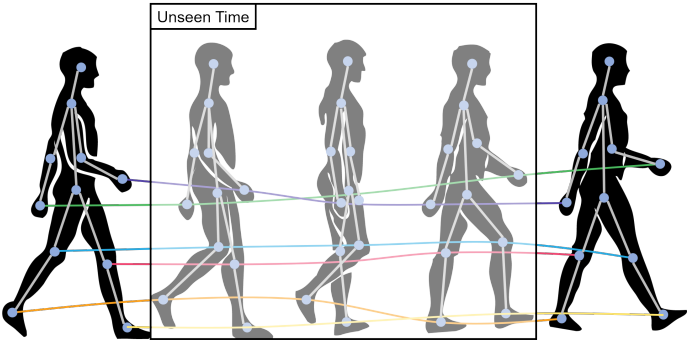


Figure 1: A walking human and corresponding keypoints. Rigid treatment of time as discretized, fixed-interval timesteps (*the 1st and 5th steps*) prevents the model from learning the underlying motion dynamics by missing out frames at unseen timesteps (*motion in the box*).

ing a density function from which we can sample unseen videos. This is typically achieved by employing an adversarial training framework, thanks to the advances in deep generative models where GAN-based approaches have shown impressive performance in static image generation [6, 12, 13, 16]. Video generation, however, differs from image generation since we must consider the temporal aspect in addition to the spatial aspect.

Initial UVG approaches tried to learn a single density function for both temporal and spatial aspects, where based on a single noise vector, a sequence of video frames were generated at each timestep [24, 30]. Such approaches, however, were limited to generating short videos with simple dynamics (*e.g.*, linear motions such as moving trains). This limitation stems most likely from failing to treat the two aspects effectively. Specifically, a video can be decomposed into two orthogonal elements: the motion (*i.e.*, movement of an object(s)) and the appearance (*i.e.*, background, style of the object(s), etc.), where the former is concerned with the temporal aspect, and the latter the spatial aspect.

Based on this observation, a couple of recent works tried to learn separate distributions (one for motion and another for appearance) and have shown improved video quality as well as more natural motion dynamics [29, 33]. However, existing approaches, while separately learning motion and appearance to some extent, fail to learn the true dynamics of motion as they all treat videos as a sequence of frames bound by discretized, fixed-interval timesteps. Such rigid treatment of the temporal aspect limits the model's capacity to learn natural motions as indicated in Fig. 1, which eventually leads to generating videos of suboptimal quality.

In this paper, we propose Motion Ordinary Differential Equation GAN (MODE-GAN), a novel two-stage UVG model that separately learns motion and appearance distributions. In the first stage, the motion generator is responsible for converting the motion noise vector to a sequence of keypoints (*i.e.*, representation of motion). To learn continuous-time dynamics of a physical body, we employ a neural ODE [5], which is a continuous-time model by interpreting the forward pass of the neural networks as solving an ODE. With neural ODE, our motion generator can produce a motion in an arbitrary frame rate, which is especially useful for generating non-linear motion dynamics (*e.g.*, sports) where higher frame rates at certain segments can help the viewer understanding. In the second stage, given a sequence of keypoints and an appearance noise vector, the motion-conditioned video generator synthesizes a video sequence by combining the two. With this two-stage approach, MODE-GAN provides full control of the spatio-temporal aspects of the generated video, such as mixing the motion and appearance from two different datasets. This lends more power to the user to generate diverse videos that even may not exist in the real world.

To sum up, MODE-GAN not only generates high-quality videos with continuous motion dynamics, but also learns completely independent motion density function and appearance density function. We summarize our contributions to the domain of UVG as follows:

- We present a *novel two-stage unconditional video generation framework* MODE-GAN, which demonstrates high-quality videos in terms of pixel distribution as well as motion smoothness.
- By employing the neural ODE to generate continuous-time motions, MODE-GAN is able to *dynamically change the frame rate* even when generating a single video sample.
- MODE-GAN learns completely independent density functions for motion and appearance, enabling *disentanglement of the spatio-temporal aspects* of the generated video.

2 Related Work

Video Generation from Random Noise. The goal of unconditional video generation is to learn a mapping function that generates a realistic video given a random noise vector. Existing approaches tried to decompose a video into several independent components. In an earlier study, VGAN [30] decomposed a video into a foreground object and a background during video synthesis. In addition, TGAN [24] tried to split each frame into a fast and slow part. MoCoGAN [29] was the first approach to divide the video signal into appearance and motion. Lastly, G^{3}AN [33] proposed a three-stream video generator to promote the disentanglement of appearance and motion to improve video quality. Our work is distinguished from previous approaches by learning completely independent density functions for motion and appearance where the former is modeled in continuous-time via neural ODE. This choice or architecture enables not only improved video quality with smooth motion dynamics in arbitrary frame rates, but also transferring motion from one dataset to another.

Conditional Video Generation with Additional Input. Generating videos with the additional inputs such as semantic segmentation [19, 31, 32], pose keypoints [9, 27, 28] relates to learning the marginal distributions instead of modeling the joint distributions [22]. There exist a large body of work, where given a single video frame [14, 18, 36] or a sequence of frames [15, 34, 35], the model predicts the in-between frames or future frames. Some works among them are related to our work in that they predict video frames by extracting the pose keypoints from an input image [14, 36]. Although these works for conditional video generation also handle video data, our contribution lies in learning separate density functions for motion and appearance, rather than making predictions given an initial video frame(s).

Neural ODE. Neural ODEs [6] represent one of the continuous-depth deep learning models which employ a neural network to model the dynamics (*i.e.*, vector field) of the latent state. Equipped with widely used numerical solvers such as Runge-Kutta and Dormand–Prince method, neural ODE has the capacity to express the latent state in continuous-depth, or equivalently continuous-time. The continuous nature of neural ODE paved a way to design the continuous time-series modeling as shown in following studies [7, 8, 20, 23, 37]. Latent ODE [23] introduced ODE-RNN as an encoder and demonstrated the effectiveness of handling the time-series data taken at non-uniform intervals. Furthermore, ODE^{2}VAE [37] and Vid-ODE [20] performed continuous-time video prediction conditioned on input video frames, demonstrating the potential to apply neural ODE to computer vision.

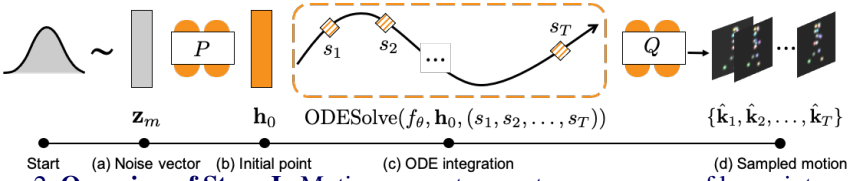


Figure 2: **Overview of Stage I** Motion generator creates a sequence of keypoints representing the motion of an object. (a) Sample a noise vector $\mathbf{z}_m$ from $\mathcal{N}(\mathbf{0}, I)$. (b) $\mathbf{z}_m$ is mapped to an initial point $\mathbf{h}_0$ via function P. (c) By integrating $\mathbf{h}_0$ over the target timesteps $\{s_1, s_2, \ldots, s_T\}$, latent states for each timestep $\mathbf{h}_{1:T}$ are obtained. (d) Latent states are projected via function Q to a set of evolving keypoints $\hat{\mathbf{k}}_{1:T}$.

3 Method

Problem Statement. Our task is unconditionally generating a video $\hat{\mathbf{v}}_{1:T} \in \mathbb{R}^{T\times 3\times H\times W}$ given two noise vectors, the motion noise vector $\mathbf{z}_m \in \mathcal{Z}_M$ and the appearance noise vector $\mathbf{z}_a \in \mathcal{Z}_A$, where T denotes the number of frames, H and W the height and width of the generated image, respectively.

Model Overview. We employ a two-stage approach with two separate components: the *motion generator* and the *motion-conditioned video generator*. Starting from $\mathbf{z}_m$, the motion generator creates a sequence of keypoints, which conveys a plausible movement of an object. Given the sequence of keypoints and $\mathbf{z}_a$, the motion-conditioned video generator synthesizes a realistic video following the geometric information in the keypoints. In the following, we describe each stage in detail.

3.1 Stage I: Motion Generation

Fig. 2 depicts the overall process of motion generation via neural ODE. The motion generator aims to learn a distribution of sequential 2D keypoints coordinates $\mathbf{k}_{1:T} \in \mathbb{R}^{T\times K\times 2}$, where K denotes the number of keypoints, beginning with a noise vector $\mathbf{z}_m$ drawn from $\mathcal{N}(\mathbf{0}, \mathbf{I})$. As an initial step, $\mathbf{z}_m$ is fed into the initial value mapper P to generate the initial value $\mathbf{h}_0$ for the ODE solver. Integrating over the target timesteps $\{s_1, s_2, \ldots, s_T\}$, the ODE solver produces a sequence of latent states $\mathbf{h}_{1:T}$ which are then transformed into the a sequence of keypoints $\hat{\mathbf{k}}_{1:T} \in \mathbb{R}^{T\times K\times 2}$ via a fully connected neural network Q. Overall, the motion generator is described as

$$\begin{aligned} \mathbf{h}_0 &= P(\mathbf{z}_m), \\ \mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_T &= \text{ODESolve}(f_\theta, \mathbf{h}_0, (s_1, s_2, \ldots, s_T)), \\ \text{each} \quad \hat{\mathbf{k}}_t &= Q(\mathbf{h}_t) \quad t = 1, 2, \ldots, T, \end{aligned} \tag{1}$$

where f_θ indicates a fully-connected neural network to approximate $d\mathbf{h}_t/dt$. A straightforward way to train the motion generator is to learn the distribution of real coordinates $\mathbf{k}_{1:T}$ via adversarial training. However, we observed that the 2D coordinates of the keypoints led to unstable training, and therefore used 2D Gaussian heatmaps as an alternative representation. These K Gaussian heatmaps $\mathcal{H}_t \in \mathbb{R}^{K\times H\times W}$ are obtained by utilizing a Gaussian-like function centered at $\mathbf{k}_t$, which can be formulated as

$$\mathcal{H}_t^{(c,\mathbf{u})} = \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{u} - \mathbf{k}_t^c\|^2\right), \tag{2}$$

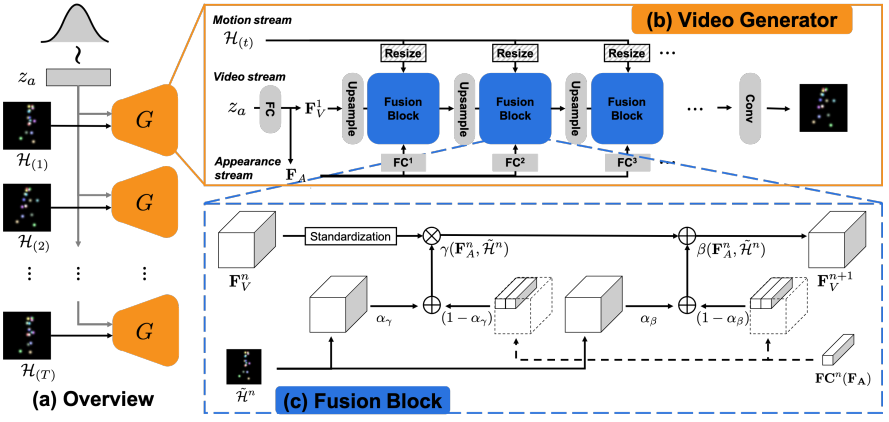


Figure 3: **Overview of Stage II** (a) Given an appearance noise vector $\mathbf{z}_a$ and a sequence of Gaussian keypoint heatmaps $\mathcal{H}_t$ from Stage I, the motion-conditioned video generator G synthesizes a realistic video. (b) The video generator G consists of three streams: motion (top), video (middle), and appearance (bottom). At first, $\mathbf{z}_a$ is fed into a neural network to produce an appearance feature vector, which is reshaped to an initial video feature map $\mathbf{M}_t^1$. Afterwards, a series of N up-sampling layers and composition blocks are used to generate a single video frame at timestep t. (c) In the composition block, the resized Gaussian heatmap $\tilde{\mathcal{H}}_t^n$ and a projected appearance feature $\mathbf{FC}^n(\mathbf{f}_a)$ are used together to determine learnable parameters α_γ and α_β, which are applied to the video feature $\mathbf{M}_t^n$, transforming it according to the motion and appearance information.

where $\mathbf{u} \in \Omega$ is the pixel coordinates and $\mathbf{k}_t^c$ indicates the coordinates of the c-th keypoint at time t.

Loss Function. We use an adversarial loss to minimize the discrepancy between the distribution of the real keypoints sequences and that of the generated ones. For this purpose, we employ two discriminators $D_{\text{fr}}^{(\text{I})}, D_{\text{sq}}^{(\text{I})}$, where each receives Gaussian heatmaps in individual frame and sequence level, respectively.[1] Since ODE integration is solely determined by the initial value, the initial values $\mathbf{h}_0$ need to be diverse in order to generate diverse motions. To this end, we add an diversity loss of initial value $\mathcal{L}_{\text{div}}^{\text{initial}}$ to enforce two different motion noise vectors $\mathbf{z}_m, \mathbf{z}'_m \in \mathcal{Z}_M$ to be embedded in two different places in the initial value space.

$$\mathcal{L}_{\text{div}}^{\text{initial}} = \frac{\|\mathbf{z}_m - \mathbf{z}'_m\|_1}{\|P(\mathbf{z}_m) - P(\mathbf{z}'_m)\|_1} \tag{3}$$

The overall objective function for stage I is formulated as

$$\min_{P,Q,f_\theta} \max_{D_{\text{fr}}^{(\text{I})}, D_{\text{sq}}^{(\text{I})}} \mathcal{L}_{\text{adv}}^{(\text{I})} + \lambda_{\text{div}}^{\text{initial}} \mathcal{L}_{\text{div}}^{\text{initial}}, \tag{4}$$

where $\mathcal{L}_{\text{adv}}^{(\text{I})}$ denotes the adversarial loss for stage I and $\lambda_{\text{div}}^{\text{initial}}$ is a hyperparamter controlling the relative importance between the two losses.

[1] Detailed descriptions about adversarial losses are provided in the supplementary material.

3.2 Stage II: Motion Conditioned Video Generation

As shown in Fig. 3(a), given a series of keypoints represented as Gaussian heatmaps $\mathcal{H}_{1:T}$, the motion-conditioned video generator G synthesizes a video $\hat{\mathbf{v}}_{1:T}$ where the object follows the geometric information of the keypoints sequence. Inspired by G^{3}AN [33], we use three parallel streams: a motion stream, a video stream, and an appearance stream. The three streams undergo N number of upsample-then-compose blocks to generate the final video, as depicted in Fig. 3(b). Specifically, the appearance noise vector $\mathbf{z}_a$ sampled from $\mathcal{N}(\mathbf{0}, I)$ is fed into a fully-connected neural network to produce an appearance feature vector $\mathbf{f}_a \in \mathbb{R}^L$, used mainly in the appearance stream to inject appearance information to the video stream. In the video stream, the initial video feature map $\mathbf{M}^1_{1:T} \in \mathbb{R}^{\frac{L}{4} \times T \times 2 \times 2}$ is obtained by reshaping $\mathbf{f}_a$, then stacking it across the time axis. In our implementation, we sequentially upsample the feature map from the initial resolution 2×2 to the final resolution. The motion stream injects the motion information from the Gaussian heatmaps into the video stream where at each injection, the $H \times W$ resolution heatmaps are resized to the spatial size of $\mathbf{M}^n_{1:T}$, enabling the model to consider the motion information in multi-scale. The injection of appearance and motion information is conducted inside the composition block.

Composition Block. The composition block takes three inputs $\mathbf{f}_a$ (appearance feature vector), $\mathbf{M}^n_{1:T}$ (video feature map), $\tilde{\mathcal{H}}^n$ (resized Gaussian heatmaps), and combines them via spatially adaptive denormalization [21] by estimating scale γ and shift parameters β, and produces $\mathbf{M}^{n+1}_{1:T}$. It consists of standardization over all frames (*i.e.,* batch size $\times$ T dimension) followed by an adaptive scaling and shift. Let (B, C_n, H_n, W_n) be the number of frames, the channel dimension, height and the width of $\mathbf{M}^n_{1:T}$. Then, the transformed activation value at each site $(b \in B, c \in C_n, i \in H_n, j \in W_n)$ is given by

$$\gamma_{c,i,j}(\mathbf{f}_a, \tilde{\mathcal{H}}^n)\frac{o_{b,c,i,j} - \mu_c}{\sigma_c} + \beta_{c,i,j}(\mathbf{f}_a, \tilde{\mathcal{H}}^n), \tag{5}$$

where $o_{b,c,i,j}$ is an activation value at the site before transformation, the scale and shift parameters $\gamma_{c,i,j}, \beta_{c,i,j}$ are weighted sums of motion and appearance information obtained from $\mathbf{FC}^n(\mathbf{f}_a)$ and $\tilde{\mathcal{H}}^n_t$, and μ_c, σ_c are the mean and standard deviation of $\mathbf{M}^n_{1:T}$ along channel c, respectively (See Fig. 3(c) for details).

Loss Function. The motion-conditioned video generator aims at generating a realistic video retaining the motion information from $\mathcal{H}_{1:T}$.[2] In addition, we employ the pixel-level diversity loss to make output videos distinctive given two different appearance noise vectors $\mathbf{z}_a, \mathbf{z}'_a \in \mathcal{Z}_A$.

$$\mathcal{L}^{\text{pixel}}_{\text{div}} = \frac{\|\mathbf{z}_a - \mathbf{z}'_a\|_1}{\|G(\mathbf{z}_a; \mathcal{H}_{1:T}) - G(\mathbf{z}'_a; \mathcal{H}_{1:T})\|_1} \tag{6}$$

The complete objective function for stage II is described as

$$\min_G \max_{D_{\text{img}}, D_{\text{vid}}} \mathcal{L}^{(\text{II})}_{\text{adv}} + \lambda^{\text{pixel}}_{\text{div}} \mathcal{L}^{\text{pixel}}_{\text{div}}, \tag{7}$$

where $\mathcal{L}^{(\text{II})}_{\text{adv}}$ denotes the adversarial loss for stage II and $\lambda^{\text{pixel}}_{\text{div}}$ adjusts the relative importance between losses.

[2] Detailed descriptions about adversarial losses are provided in the supplementary material.

	Weizmann	MUG	UvA	KTH
VGAN [30]	99.03	104.71	103.70	103.31
TGAN-v2 [25]	83.90	72.60	86.91	65.03
MoCoGAN [29]	93.93	35.12	49.58	36.90
G^3AN [33]	64.07	21.76	39.24	44.84
MODE-GAN	**55.06**	**17.90**	**15.38**	**28.28**

Table 1: Video FID scores on 4 datasets. Lower values are better.

4 Experiments

Datasets. We evaluate our method on four datasets: 1) *Weizmann Action* [2] consists of 90 videos of 9 subjects performing 10 actions (*e.g.,* walk, jumping-jack). 2) From *KTH Action* [26], we select videos of 25 subjects performing three types of actions (boxing, hand waving, and hand clapping) always having single person in the video. 3) *MUG* [1] contains 1,254 video sequences of six facial expressions, such as anger, disgust, and happiness. 4) *UvA-NEMO* [8] is comprised of 1,240 videos of 400 smiling subjects. We resize the videos to the 64×64 resolution for all datasets.

For training the motion generator, the ground truth keypoints are obtained by using a pre-trained keypoint detector [3, 10, 17]. In particular, we use 13 out of 68 face keypoints [3] including eyes and nose to represent the facial expression for *MUG* and *UvA-NEMO*. And we use 17 pose keypoints for *Weizmann Action* and *KTH Action*.

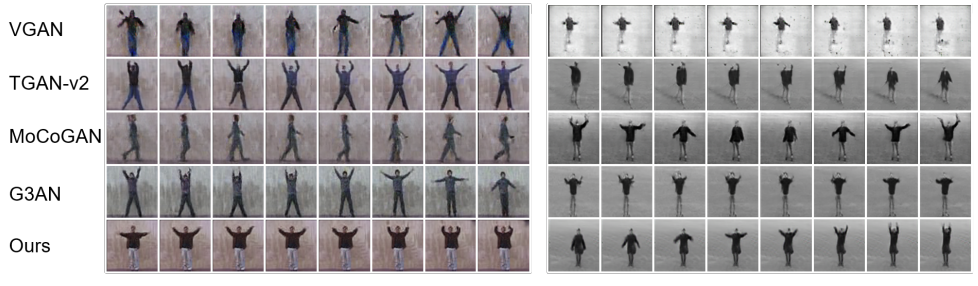
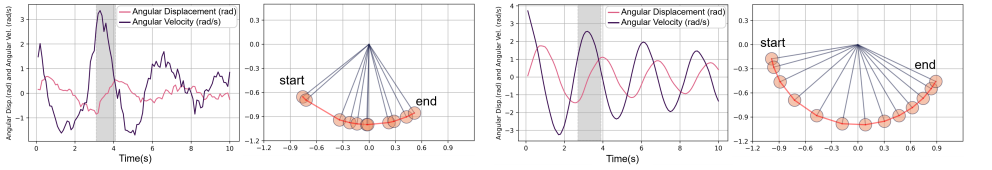


Figure 4: Qualitative comparison with baselines on human action datasets.

Evaluation on Fixed Frame Rate. We compare our model with unconditional video generation baselines [25, 29, 30, 33] on four datasets. We followed the hyperparameters as presented in the papers. Table 1 shows qualitative comparison of MODE-GAN with VGAN [30], TGAN-v2 [25], MoCoGAN [29] and G^3AN [33] on four different datasets by measuring video Fréchet Inception Distance (FID) [33], which is a widely used metric for evaluating the quality of videos. As seen in Table 1, MODE-GAN consistently outperforms all baseline models. This indicates our synthetic videos are not just visually realistic but keeps the same realistic quality consistently across time compared to the baselines. Also, Fig. 4 shows visual comparisons using human action dataset, where MODE-GAN produces the competitive results compared to the baseline models.[4] This state-of-the-art performance shows that explicitly decomposing the video generation process into spatial (*i.e.* appearance) and tem-

[3] 2, 9, 16, 20, 25, 38, 42, 45, 47, 49, 52, 55, 58th facial landmark locations detected using open face alignment library (https://github.com/1adrianb/face-alignment). We provide a detailed illustration in supplementary material.

[4] Other visual comparisons using facial expression dataset are shown in the supplementary material.



(a) Recurrent neural network (b) Neural ODE

Figure 5: Comparison of generated pendulum dynamics (*Left*: sampled θ and $\dot{\theta}$, *Right*: visualization of the bob movement in the gray area of the left. (a) RNN fails to preserve the smoothness of the pendulum movement (θ and $\dot{\theta}$ severely fluctuate across time), yielding a rigid and irregular movement of the bob (b) Neural ODE successfully mimics the pendulum movement maintaining continuous and smooth bob motion.

poral (*i.e.* motion) components, and modeling the latter in the natural continuous-time leads to more realistic synthetic outcomes.

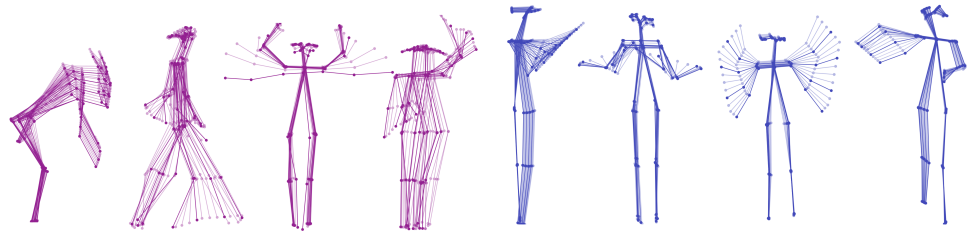
Effectiveness of ODE in Learning Dynamics. We compare the effectiveness of neural ODE with recurrent neural networks (RNNs) in learning the distribution of dynamics starting from a random noise. Motivated by previous work [] that interprets human motion as a relative *pendulum* dynamics of keypoints, we take a *pendulum* system which is mathematically formulated as

$$\ddot{\theta} + \left(\frac{B}{M}\right) \cdot \dot{\theta} + \left(\frac{g}{L}\right) \cdot \sin(\theta) = 0, \tag{8}$$

where θ, B, g, L, and M are the angular displacement, damping factor, gravity force, length of pendulum, and mass of bob, respectively. Our goal is to train the generator which is capable of simulating the dynamics of the pendulum system. For this purpose, the model aims at producing θ and $\dot{\theta}$, a plausible physical parameters for the pendulum system. In experiment, g is a constant and fixed to 9.81 m/s^2, whereas B, L, and M are stochastically determined by sampling each factor from Gaussian distribution with means of 0.2, 1.0, 1.0 and unit variance, respectively. As a baseline, we employ an RNN-based motion generator, where we substitute the neural ODE in stage I with an RNN. The RNN takes Gaussian noises at each time step and aims to generate a plausible θ and $\dot{\theta}$.

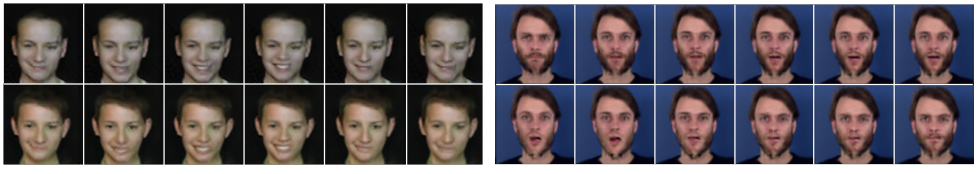
As seen in Fig. 5(a), the RNN-based motion generator fails to generate smooth dynamics, running off the expected trail of the pendulum dynamics. On the other hand, as in Fig 5(b), our ODE-based motion generator successfully simulates the pendulum dynamics. This demonstrates that neural ODE has a benefit in learning the continuous motion dynamics compared to RNN. As seen in Fig. 5, RNN fails to smoothly interpolate the pendulum trajectory (a) whereas neural ODE successfully simulates the smooth dynamics of the pendulum (b), which indicates the potential of neural ODE for learning the dynamics of real-world videos smoothly. In other words, the capacity of learning the smooth dynamics can be extended to simulate the unseen dynamics between two frames in more natural manner without rather unnatural rigid and irregular movements. Such capability can be seen in Fig. 6, where the motion generator successfully fills the plausible keypoints at unseen timesteps. Detailed descriptions about data generation process and model architectures are provided in supplementary material.

Diverse Motions in Continuous-time Space. The motion generator is to learn the distribution of motions, thereby generating a plausible motion. Specifically, neural ODE allows the motion generator to model dynamics of keypoints in continuous-time domain. Fig. 6 shows



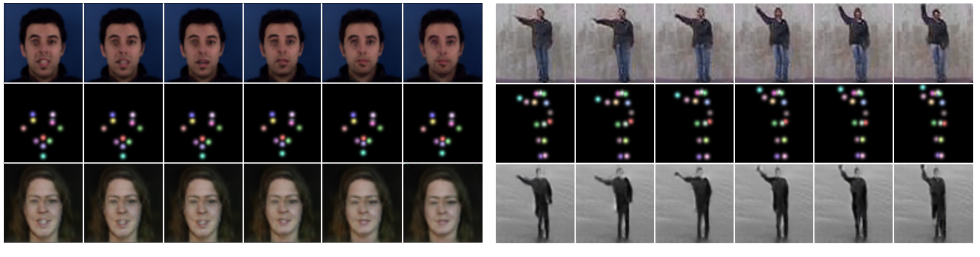
(a) Weizmann Action (b) KTH Action

Figure 6: Examples of sequence of continuously evolving keypoints: We sample a sequence of 64 keypoints from MODE-GAN trained at 16 FPS. To visualize the keypoints dynamics, every 4th frame (*i.e.* 16 / 64) are marked in bold, while other frames are faintly illustrated.



(a) Fixed Motion (b) Fixed Appearance

Figure 7: An example of manipulating motion and appearance: (a) Fixed motion (facial expression) with different appearances, (b) Fixed appearance with different motions.



(a) MUG: *Surprise* ⇒ UvA-NEMO (b) Weizmann: *One hand waving* ⇒ KTH

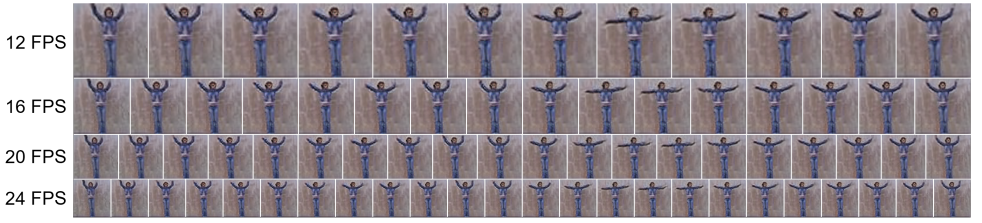
Figure 8: Motion transfer results across different video domains (*First row*: generated output at source domain, *Second row*: corresponding keypoints sequence, *Third row*: generated output at target domain).

keypoints representations of various motions as we integrate densely across time, trained on *Weizmann Action* and *KTH Action* datasets. We see the sampled motions are natural and fluid, demonstrating that the motion generator has successfully learned the underlying distribution of video motions.

Motion and Appearance Manipulation. Recall that our model synthesizes a video by combining separate representations for *motion* and *appearance*. In this experiment, we demonstrate our model's capability to generate videos while maintaining one component (*i.e.,* motion *or* appearance), by fixing either $\mathbf{z}_m$ or $\mathbf{z}_a$ and varying the other.

As shown in Fig. 7, we observe different combinations can successfully produce videos preserving the fixed component, while altering the other. These results indicate that we can successfully manipulate the video generation process by controlling appearance and motion independently.

Motion Transfer between Different Video Domains. One of the distinguishing applications of MODE-GAN is to import a non-existing motion to a different video domain. Such



(a) Weizmann: *Two hand waving*

(b) Weizmann: *Bending*

Figure 9: Videos generated in diverse frame rates by MODE-GAN, trained at 16 FPS.

application stems from the fact that motion-conditioned video generators share the common motion space expressed as keypoints representations. Therefore, by connecting the motion generator trained on one video domain with the motion-conditioned video generator trained on another one, we can combine motion and appearance from each video domain.

Fig. 8 shows motion transfer examples in facial expression and human action datasets. (a) Although UvA-NEMO contains only smile motion, *surprise* expression can be imported by adopting the motion model trained on MUG dataset. (b) In a similar manner, we synthesize the KTH Action domain video using *one hand waving* motion adopted from the Weizmann Action domain.

Arbitrary Frame Rate Video Generation. Another application of our model is to generate videos in arbitrary frame rates based on the continuously generated keypoints from the motion generator. As illustrated in Fig. 9, our model is capable of synthesizing a video at various frame rates, densely dividing the continuous-time domain. Synthesized videos successfully fill in the frame at arbitrary timesteps, demonstrating that our model understand the underlying dynamics of motion (*i.e., two hand waving*). Furthermore, we can dynamically control the frame rate even for a single video (*e.g.* from slow to fast motion) by controlling integration time span of the motion generator. Such results are provided in supplementary material as a video.

5 Summary and Future Work

In this paper, we propose a novel framework MODE-GAN for unconditional video generation. Based on the observation that real world videos can be decomposed into spatial (*i.e.* static appearance) and temporal (*i.e.* continuous motion dynamics) aspects, we employ neural ODE to handle the continuous nature of video motion. In addition, a two-stage approach enables MODE-GAN to focus on modeling the motion itself, allowing our model to separately learn motion and appearance distributions. Experimental results not only demonstrate its ability to generate high-quality videos but also its versatile functionality including continuous-time and cross-domain video generation.

Current motion generator of MODE-GAN focuses on modeling the single person's dynamics, yet it bears a potential to be used for modelling the dynamics of multiple people and complex videos. As a future direction, we plan to explore how to generate highly complicated real world semantics in continuous-time domain.

Acknowledgement

This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2019-0-00075, Artificial Intelligence Graduate School Program (KAIST) and No. 2021-0-01778, Development of human image synthesis and discrimination technology below the perceptual threshold) and by Kakao Enterprise.

References

[1] Niki Aifanti, Christos Papachristou, and Anastasios Delopoulos. The MUG facial expression database. In *International Workshop on Image Analysis for Multimedia Interactive Services*, 2010.

[2] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2005.

[3] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3D facial landmarks). In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[4] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

[5] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[6] Wonwoong Cho, Sungha Choi, David Keetae Park, Inkyu Shin, and Jaegul Choo. Image-to-image translation via group-wise deep whitening-and-coloring transformation. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[7] Edward De Brouwer, Jaak Simm, Adam Arany, and Yves Moreau. GRU-ODE-Bayes: Continuous modeling of sporadically-observed time series. In *Advances in Neural Information Processing Systems (NIPS)*, 2019.

[8] Hamdi Dibeklioğlu, Albert Ali Salah, and Theo Gevers. Are you really smiling at me? spontaneous versus posed enjoyment smiles. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2012.

[9] Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural ODEs. In *Advances in Neural Information Processing Systems (NIPS)*, 2019.

[10] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. RMPE: Regional multi-person pose estimation. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[11] Gunnar Johansson. Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 14(2):201–211, 1973.

[12] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[13] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of styleGAN. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[14] Yunji Kim, Seonghyeon Nam, In Cho, and Seon Joo Kim. Unsupervised keypoint learning for guiding class-conditional video prediction. In *Advances in Neural Information Processing Systems (NIPS)*, 2019.

[15] Yong-Hoon Kwon and Min-Gyu Park. Predicting future frames using retrospective cycle GAN. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[16] Hsin-Ying Lee, Hung-Yu Tseng, Qi Mao, Jia-Bin Huang, Yu-Ding Lu, Maneesh Singh, and Ming-Hsuan Yang. Drit++: Diverse image-to-image translation via disentangled representations. *International Journal of Computer Vision (IJCV)*, pages 1–16, 2020.

[17] Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Hao-Shu Fang, and Cewu Lu. CrowdPose: Efficient crowded scenes pose estimation and a new benchmark. *arXiv:1812.00324*, 2018.

[18] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Flow-grounded spatial-temporal video prediction from still images. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2018.

[19] Junting Pan, Chengyu Wang, Xu Jia, Jing Shao, Lu Sheng, Junjie Yan, and Xiaogang Wang. Video generation from single semantic label map. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[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. *arXiv:2010.08188*, 2020.

[21] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2337–2346, 2019.

[22] Yunchen Pu, Shuyang Dai, Zhe Gan, Weiyao Wang, Guoyin Wang, Yizhe Zhang, Ricardo Henao, and Lawrence Carin Duke. Jointgan: Multi-domain joint distribution learning with generative adversarial nets. In *Proc. of the International Conference on Machine Learning (ICML)*, 2018.

[23] Yulia Rubanova, Tian Qi Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *Advances in Neural Information Processing Systems (NIPS)*, 2019.

[24] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[25] Masaki Saito, Shunta Saito, Masanori Koyama, and Sosuke Kobayashi. Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal GAN. *International Journal of Computer Vision (IJCV)*, 2020.

[26] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 32–36. IEEE, 2004.

[27] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[28] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *Advances in Neural Information Processing Systems (NIPS)*, 2019.

[29] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN: Decomposing motion and content for video generation. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[30] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[31] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[32] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Bryan Catanzaro, and Jan Kautz. Few-shot video-to-video synthesis. In *Advances in Neural Information Processing Systems (NIPS)*, 2019.

[33] Yaohui Wang, Piotr Bilinski, Francois Bremond, and Antitza Dantcheva. G^3AN: Disentangling appearance and motion for video generation. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[34] Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3D LSTM: A model for video prediction and beyond. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2019.

[35] Yunbo Wang, Jianjin Zhang, Hongyu Zhu, Mingsheng Long, Jianmin Wang, and Philip S Yu. Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[36] Ceyuan Yang, Zhe Wang, Xinge Zhu, Chen Huang, Jianping Shi, and Dahua Lin. Pose guided human video generation. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2018.

[37] Cagatay Yildiz, Markus Heinonen, and Harri Lahdesmaki. ODE^{2}VAE: Deep generative second order odes with bayesian neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2019.