Stochastic Image-to-Video Synthesis using cINNs

Michael Dorkenwald[1] Timo Milbich[1] Andreas Blattmann[1] Robin Rombach[1]
Konstantinos G. Derpanis[2,3,4*] Björn Ommer[1*]
[1]IWR/HCI, Heidelberg University, Germany [2]Department of Computer Science, Ryerson University, Canada
[3]Vector Institute for AI, Canada [4]Samsung AI Centre Toronto, Canada

Abstract

Video understanding calls for a model to learn the characteristic interplay between static scene content and its dynamics: Given an image, the model must be able to predict a future progression of the portrayed scene and, conversely, a video should be explained in terms of its static image content and all the remaining characteristics not present in the initial frame. This naturally suggests a bijective mapping between the video domain and the static content as well as residual information. In contrast to common stochastic image-to-video synthesis, such a model does not merely generate arbitrary videos progressing the initial image. Given this image, it rather provides a one-to-one mapping between the residual vectors and the video with stochastic outcomes when sampling. The approach is naturally implemented using a conditional invertible neural network (cINN) that can explain videos by independently modelling static and other video characteristics, thus laying the basis for controlled video synthesis. Experiments on diverse video datasets demonstrate the effectiveness of our approach in terms of both the quality and diversity of the synthesized results. Our project page is available at https://bit.ly/3dg90fV.

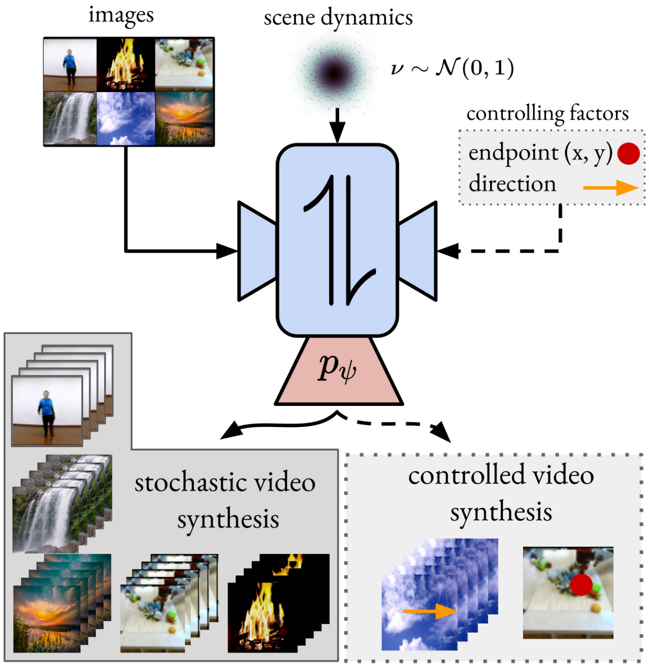
images
scene dynamics
$\nu \sim \mathcal{N}(0,1)$
controlling factors
endpoint (x, y) ●
direction →
p_ψ
stochastic video synthesis
controlled video synthesis

Figure 1. Our approach establishes a bijective mapping between the image and the video domain by introducing a residual representation ν describing the latent scene dynamics. This allows us not only to synthesize diverse videos but also to extend our approach to gain control over the video synthesis task.

1. Introduction

Anticipating and predicting what happens next are key features of human intelligence that allow us to understand and deal with the ever-changing environment that governs our everyday life [8]. Consequently, the ability to foresee and hallucinate the future progression of a scene is a cornerstone of artificial visual understanding with applications including autonomous driving [48, 49, 28], medical treatment [5, 18, 6], and robotic planning [20, 24, 10].

Predicting and synthesizing plausible future progressions from a given image requires a deep understanding of how scenes and objects within video are depicted, interplay with each other, and evolve over time. While an image provides information about the observed scene content, such as object appearance and shape, the challenge is to understand the missing information constituting potential futures, such as the scene dynamics setting the scene in motion. Due to the ambiguity and complexity of capturing this information, many works [42, 25, 11, 74] directly focus on predicting likely video continuations, often resorting to simplifying assumptions (e.g., dynamics modelled by optical flow [21, 59, 76]) and side information (e.g., semantic keypoints [56, 72, 46, 22]). However, truly understanding the synthesis problem not only requires to infer such image continuations but, conversely, also demands when observing a video sequence to describe and represent the instantiated scene dynamics animating its initial frame.

[*] Indicates equal supervision.

Consequently, the image-to-video synthesis task should be modelled as a translation between the image and video domains, ideally by an invertible mapping between them. Since the content information describing an image only accounts for a small fraction of the video information, in particular missing the temporal dimension, learning an invertible mapping requires a dedicated residual representation that captures all missing information. Once learned, given an initial image and an instantiation of the latent residual, we can combine them to synthesize the corresponding future video sequence.

In this paper, we frame image-to-video synthesis as an invertible domain transfer problem and implement it using a conditional invertible neural network (cINN). To account for the domain gap between images and videos, we introduce a dedicated probabilistic residual representation. The bijective nature of our mapping ensures that only information complementary to that in the initial image is captured. Using a probabilistic formulation, the residual representation allows to sample and thus synthesize novel future progressions in video with the same start frame. To reduce the complexity of the learning task, we train a separate conditional variational encoder-decoder architecture to compute a compact, information preserving representation for the video domain. Moreover, our specific framing of learning the residual representation allows to easily incorporating extra conditioning information to exercise control over the image-to-video synthesis process. Our contributions can be summarized as follows:

- We frame image-to-video synthesis as an invertible domain transfer problem and learn a dedicated residual representation to capture the domain gap.
- Our framework naturally extends to incorporate explicit conditioning factors for exercising control over the synthesis process.
- Extensive evaluations on four video datasets, ranging from structured human motion synthesis to subtle dynamic textures, show strong results demonstrating the effectiveness of our approach.

2. Related Work

Video synthesis. Video synthesis involves a wide range of tasks including video-to-video translation [73], image animation [64, 65], frame interpolation [52, 4, 53], and video prediction. The latter can be divided into unconditional [69, 14] and conditional video generation (the focus of our work). Conditional video generation can be described as finding a future progression given a set of context frames in a deterministic [76, 71, 75, 50] or stochastic manner [42, 25, 11, 15, 3], as pursued here. Several works decrease the complexity of the synthesis task by using keypoint annotations [51, 56] as conditioning information. A major drawback of this approach is the requirement of semantic keypoint labels which limit consideration to highly structured objects, like humans, and thus exclude the broader range of imagery we consider, e.g., natural scenes. Recent methods aim at improving video prediction quality by use of high capacity architectures with high computational demands, operating in the latent [58] or pixel-space [74], or using attention [14]. In contrast, we propose a model for understanding the image-to-video synthesis process by learning a bijective transformation between the image and video domains using a dedicated residual representation.

Dynamic texture synthesis. Previous work has given special attention to generating dynamic textures. This work can be divided into two groups: (i) methods that exploit the statistics of dynamics textures [68, 12, 77, 79] and (ii) learning-based approaches [80, 83, 45, 21, 78]. To generalize to other video domains, beyond dynamic textures, we introduce a learning-based approach. MDGAN [80] generates landscape videos from a static scene in a deterministic manner. Several methods (e.g., [21, 83]) consider optical flow in their video generation pipeline. The use of optical flow limits application to specific types of imagery, like clouds, at the exclusion of other dynamic textures which grossly violate standard optical flow assumptions [68]. DeepLandscape [45] extends the structure of StyleGAN to animate landscape images. Their model does not attempt to learn full temporal dynamics of videos and works only by a complex optimization scheme for inference, similar to [27] for style transfer. In contrast, our approach allows for efficient feedforward image-to-video synthesis while also maintaining visual quality and temporal coherence.

Invertible Neural Networks. Invertible neural networks (INNs) are bjective functions which makes them attractive for a variety of tasks, such as analyzing inverse problems [1], interpreting neural networks [23], and representation learning [34]. In particular, INNs can be implemented as normalizing flows [60], a special class of likelihood-based generative models which have recently been applied to various tasks, such as image synthesis [38, 2, 57], domain transfer [62, 61, 23, 82], superresolution [47, 82], and video synthesis [40]. In contrast, we use a conditional normalizing flow model to learn a dedicated residual latent, capturing information not contained in the input image. This allows us to both more efficiently learn the bijective mapping and to consider explicit controlling factors.

3. Method

Our goal is to learn the interplay between images and video by explaining video in terms of a single image and the (stochastic) information not captured by the image about the

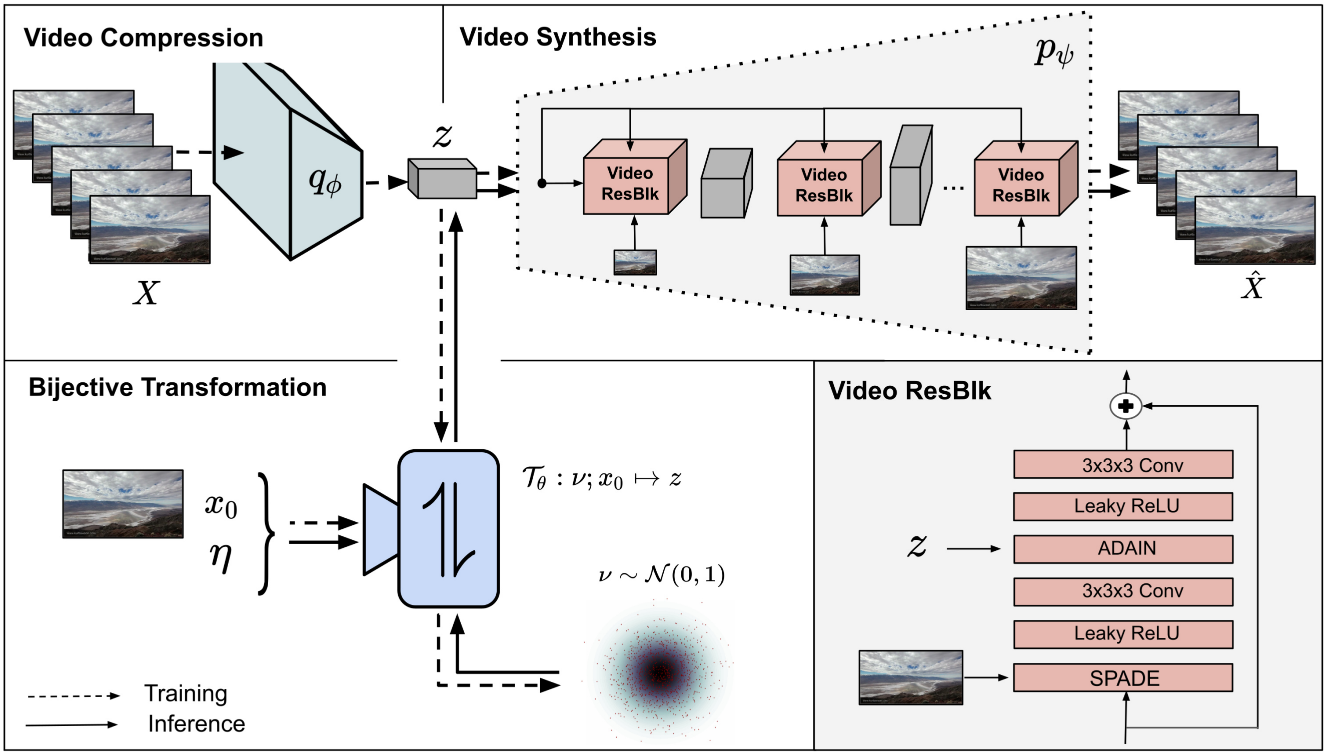

Figure 2. Overview of our proposed framework. We learn an information preserving video representation z using our conditional generative model consisting of an encoder q_ϕ as well as the corresponding decoder p_ψ. The decoder consists of dedicated video residual blocks shown in bottom right. After establishing the video representation, we learn a bijective transformation $\mathcal{T}$ conditioned on the starting frame x_0 and an optionally provided control factor η. During inference, we sample a residual ν, encapsulating the scene dynamics, from the prior distribution and use $\mathcal{T}_\phi$ to obtain the video representation z. Using our decoder we can then synthesize novel video sequences. Training and inference is indicated by the dotted and solid lines, respectively.

video. Together the deterministic and stochastic content allow us to tackle the problem of image-to-video synthesis. In Sec. 3.1, we begin by motivating and introducing our conditional bijective framework for image-to-video mappings and Sec. 3.2 describes the learning process. Sec. 3.3 presents our generative model for video synthesis operating on our learned transformation. Finally, in Sec. 3.4 we extend our model to directly exercise control over factors captured in the residual latent, e.g., direction of motion. Figure 2 provides an overview of our approach.

3.1. Bijection for Image-to-Video Synthesis

Given an initial image, $x_0 \in \mathbb{R}^{d_x}$, image-to-video synthesis generates a video sequence, $X = [x_1, \ldots, x_T]$. This problem is inherently underdetermined with many possible videos conceivable based on x_0. As a result, we cannot synthesize or explain a video merely with a single frame, but require additional information, ν, such as the scene dynamics. Video synthesis can then be framed as mapping x_0 and a residual ν onto a video X or, equivalently, a representation z thereof,

$$z = \mathcal{T}(\nu; x_0) \ . \tag{1}$$

Commonly, stochastic video prediction methods [42, 25, 51] only focus on synthesizing *arbitrary realistic* videos for a single initial or a sequence of frames. In contrast, understanding this synthesis process not only demands to explain the missing information ν to be inferred, but also to recover the residual information from video so that it can be modified subsequently. Explaining a video thus requires to estimate this residual information ν, so that x_0 and ν together are isomorphic to the representation z of the video X. Consequently, $\mathcal{T}$ needs to be a conditional bijective mapping between videos and their description in terms of a starting frame x_0 and the remaining residual information ν.

3.2. Inferring an Explicit Residual Representation

Given a single frame x_0, a multitude of videos are possible with a corresponding z,

$$z \sim p(z|x_0) \ . \tag{2}$$

Since ν contains all the information of z not captured in x_0 and $\mathcal{T}$ is conditionally bijective, we can invert (1) to obtain the residual

$$\nu = \mathcal{T}^{-1}(z; x_0) \ . \tag{3}$$

Then, by the change-of-variables theorem for probability distributions, $\mathcal{T}^{-1}$ transforms $p(z|x_0)$ as

$$p(z|x_0) = \frac{p(\nu|x_0)}{|\det J_{\mathcal{T}}(\nu; x_0)|} \quad (4)$$
$$= p(\mathcal{T}^{-1}(z; x_0)) \cdot |\det J_{\mathcal{T}^{-1}}(z; x_0)| \; , \quad (5)$$

where $J_{\mathcal{T}}$ denotes the Jacobian of the transformation $\mathcal{T}$ and $|\det[\cdot]|$ the absolute value of the determinant of its input. Using the transformed distribution, $p(z|x_0)$, we can now directly learn our transformation $\mathcal{T}$ and the distribution $p(\nu|x_0)$ by maximum likelihood estimation (MLE). To this end, we need to choose an appropriate prior distribution, which can be analytically evaluated and easily sampled. Since we factorize the residual information ν from the starting frame x_0, we can assume $p(\nu|x_0) = q(\nu)$ and, thus, resort to the widely used standard normal distribution $q(\nu) = \mathcal{N}(\nu|0, \mathbf{1})$ [39, 81, 29]. Moreover, we parametrize $\mathcal{T}$ as an invertible neural network [54, 17, 16] $\mathcal{T}_\theta$ with parameters θ which, given the image x_0, translates between the representations z and ν. Thus, we arrive at the negative log-likelihood minimization problem

$$\min_{\theta \in \Theta} \; \mathbb{E}_{z, x_0} \left[\log q(\mathcal{T}_\theta^{-1}(z; x_0)) - \log |\det J_{\mathcal{T}_\theta^{-1}}(z; x_0)| \right] . \quad (6)$$

By simplifying using the standard normal prior and dropping resulting constant terms, we finally arrive at our final objective function

$$\min_{\theta \in \Theta} \; \mathbb{E}_{z, x_0} \left[\|\mathcal{T}_\theta^{-1}(z; x_0)\|_2^2 - \log |\det J_{\mathcal{T}_\theta^{-1}}(z; x_0)| \right] . \quad (7)$$

Due to the information-preserving, isomorphic mapping $\mathcal{T}_\theta$, ν indeed captures the latent information in X not explained by x_0.

To generate a video representation z based on an initial frame x_0, we first sample a residual representation $\nu \sim q(\nu)$ and then apply (1) to obtain $z = \mathcal{T}_\theta(x_0, \nu)$.

3.3. Generative Model for Video Synthesis

We now learn a decoding $p(X|z)$ to synthesize video sequences based on z. Since we require z to be a compact, information-preserving video representation, we also need to learn the corresponding encoding $q(z|X)$. Simultaneously learning both is naturally expressed by an autoencoder [39]. Moreover, to optimally enable learning the transformation $\mathcal{T}_\theta$, we consider the following modelling constraints: *(i)* the representation z of the input should be maximally information-preserving to fully capture the residual dynamics information, *(ii)* we model the residual ν to be a continuous probabilistic model, thus the bijection property of $\mathcal{T}_\theta$ requires $q(z|X)$ to be a strictly positive density, and *(iii)* reducing the complexity of the representation z eases the task of learning the bijective mapping $\mathcal{T}_\theta$. Thus, while still fully capturing scene dynamics in z, we ideally exclude all information in the video which is already present in the initial image x_0.

Learning $p(X|z)$ and $q(z|X)$. Variational latent models [39] are a straightforward choice for stochastic autoencoders. To address *(iii)* above, we use a conditional variational autoencoder [81] with a parametrized encoder $q_\phi(z|X)$ and a parametrized, conditional decoder $p_\psi(X|x_0, z)$ with (ϕ, ψ) being their trainable parameters. Such models encourage the distribution of information among latent variables due to the regularization of the capacity of the latent encoding [13, 84, 9]. Thus, using x_0 as a conditioning to represent most of the scene content, the complexity of z can be reduced by forcing the network capacity to focus on capturing the latent information in X. To balance this with maximally preserving the latent residual information in X, we introduce a weighting parameter β to the standard variational lower bound [9],

$$\mathcal{L}_{p_\psi, q_\phi} = \mathbb{E}_{z \sim q_\phi(z|X)} \left[\log p_\psi(X|x_0, z) \right] - \beta D_{\mathrm{KL}}(q_\phi(z|X) || q(z)) \; , \quad (8)$$

where $q(z)$ denotes a standard normal prior on the encoder q_ϕ. The first term optimizes the synthesis quality of the decoding process, thus maximizing information-preservation. While the second term regularizes $q_\phi(z|X)$ to match the prior $q(z)$ which constrains its capacity and, thus, encourages the distribution of information among x_0 and z to ease subsequent learning of $\mathcal{T}_\theta$. Hence, β allows us to directly balance the informativeness of z and its complexity [13, 84, 9].

Building the video synthesis model. The design of generative architectures significantly influences their synthesis capabilities, especially when dealing with highly complex data. In our conditional model this particularly affects the interplay between information in x_0 and z in $p_\phi(X|z, x_0)$. To this end, we construct the conditional decoder p_ψ using a sequence of n dedicated video residual blocks operating on increasing spatial and temporal feature resolutions. To optimally facilitate the interplay between z and the content information in x_0, we combine them both in each block and, thus, at all scales of p_ψ. Fig. 2 illustrates the general structure of our video residual blocks used for decoding to a video. The conditioning x_0 is incorporated using a SPADE [55] normalization layer to preserve semantic information throughout the generator. The video representation z is added by means of an ADAIN [36] layer to provide video information at all scales of the decoder. Our encoder q_ϕ is implemented as a 3D-ResNet [31] to capture the scene dynamics evolving over time in an input video.

Overall training objective. Following common practice [39], we train our conditional model, (8), using an L_1 reconstruction loss. To emphasize perceptual quality [41] we use a frame-wise perceptual loss ℓ^ϕ [19, 35]. Similar to previous work [14, 73], we use a discriminator $\mathcal{D}_S$ applied

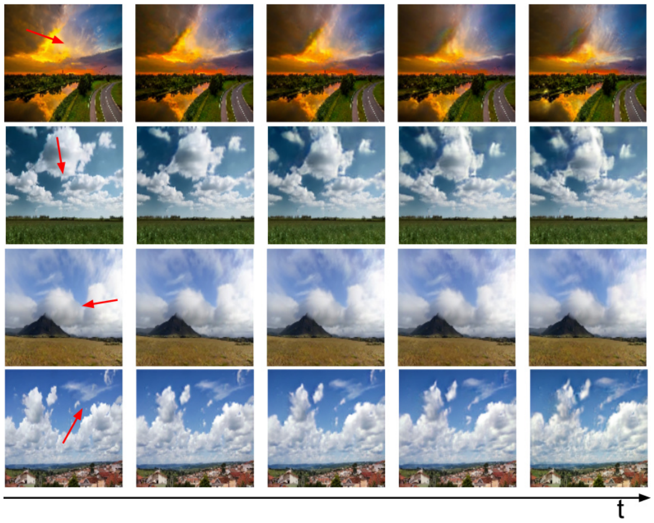

Figure 3. Stochastic video synthesis on Landscape [80] showing subtle motions. Red arrows indicate the direction of motion. Best viewed as video provided in the supplemental.

Method	LPIPS ↓	FID ↓	DTFVD ↓	FVD ↓	DIV VGG ↑
MDGAN^2 [80]	0.49	68.9	2.35	385.1	–
DTVNet^2 [83]	0.35	90.3	2.78	693.4	0.00
AL^2 [21]	0.26	19.5	1.24	307.0	**1.84**
DL^{2,†} [45]	0.41	45.2	1.96	351.5	–
Ours	**0.25**	**16.0**	**0.83**	**160.4**	1.55

Table 1. Quantitative evaluation of video synthesis quality and diversity on Landscape [80]. Numeric superscripts indicate the source of the results, cf. Sec. 4.3. The diversity score based on the I3D [67] trained on DTDB [30] can be found in the supplemental. † provided pretrained model from DL was trained on their unreleased dataset.

to each frame and $\mathcal{D}_T$ on the temporal level. Both discriminators are optimized using the hinge formulation [43, 7]. Thus, the overall training objective can be summarized as

$$\mathcal{L} = \mathcal{L}_{p_\psi, q_\phi} + \mathcal{L}_{\mathcal{D}_T} + \mathcal{L}_{\mathcal{D}_S}. \tag{9}$$

Please see the supplemental for further details of our loss.

3.4. Controllable Video Synthesis

There are many factors comprising the latent residual ν. Understanding the image-to-video process allows us to directly exercise control over such factors and thus over the progression of the depicted scene in the input image x_0. Assuming $\eta \in \mathbb{R}^{d_\eta}$ represents such a factor, e.g., the target location of a moving object, we can explicitly model it while learning our bijective mapping $\mathcal{T}_\theta$ as $\mathcal{T}_\theta(\nu; x_0, \eta)$. Note, now ν constitutes the residual latent information to *both* x_0 and η. Since such individual factors are typically low in information themselves, in general there is no benefit in considering them when learning the conditional decoder p_ψ in contrast to the richer information in x_0. Image-to-video synthesis now extends to manually additionally adjusting η to a fixed value η^* to infer a video representation

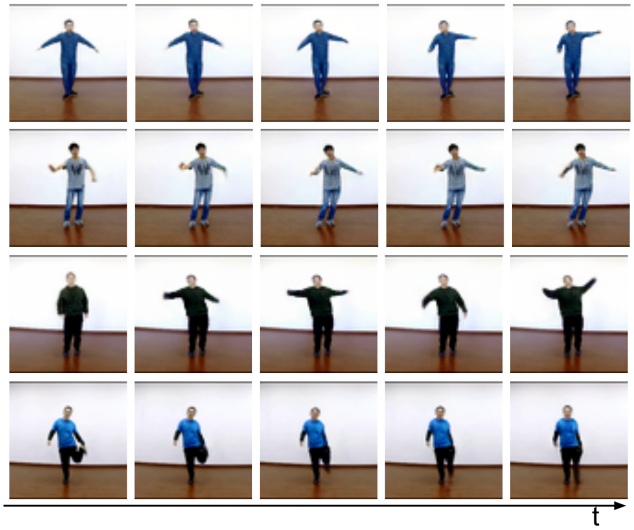

Figure 4. Stochastic video synthesis on iPER [44] showing structured, diverse human motion. Best viewed as video provided in the supplemental.

Method	FVD ↓	DIV VGG ↑	DIV I3D ↑
SAVP^3 [42]	368.6	0.00†	0.01†
SRVP^3 [26]	336.3	0.34	1.01
IVRNN^3 [11]	206.9	0.23	0.57
Ours w/o cINN	255.2	0.31	1.11
Ours w/o x_0	582.6	1.19*	2.87*
Ours w/o ADAIN	213.1	0.51	1.64
Ours	**176.9**	**0.58**	**1.76**

Table 2. Quantitative evaluation of video synthesis quality and diversity on iPER [44]. Numeric superscripts indicate the source of the results, cf. Sec. 4.3. † SAVP experienced mode collapse due to training instabilities originating from the two involved discriminators. * denotes high diversity due to artifacts.

$z = \mathcal{T}_\theta(\nu; x_0, \eta^*)$ which is then used to synthesize a video sequence using p_ψ.

4. Experiments

We evaluate the efficacy of our video synthesis method on a diverse set of video datasets which range from human motion to stochastic dynamics as encompassed by natural landscape scenery. Video prediction results and comparisons are best viewed as videos which are available in the supplemental and on our project page[1]. Implementation details can be found in the supplemental.

4.1. Datasets

Here, we summarize the four diverse datasets used in our evaluation. We train all models on a sequence length of 16. A detailed description of the evaluation protocol for each dataset is in the supplemental.

[1] https://bit.ly/3dg90fV

Method	Fire					Vegetation					Waterfall					Clouds				
	LPIPS ↓	FID ↓	FVD ↓	DTFVD ↓	DIV ↑	LPIPS ↓	FID ↓	FVD ↓	DTFVD ↓	DIV ↑	LPIPS ↓	FID ↓	FVD ↓	DTFVD ↓	DIV ↑	LPIPS ↓	FID ↓	FVD ↓	DTFVD ↓	DIV ↑
DG^3 [77]	0.18	29.4	361.3	0.40	–	0.22	71.6	290.3	0.86	–	0.25	143.4	1680.6	2.41	–	0.17	73.5	217.5	0.40	–
AL^3 [21]	0.28	48.4	1475.9	11.42	0.74	0.28	48.9	271.0	1.48	0.93	0.32	124.3	1847.8	5.94	**0.98**	**0.27**	38.7	**142.1**	0.76	**1.52**
Ours	**0.27**	**31.3**	**460.7**	**0.79**	**1.34**	**0.26**	**38.5**	**170.8**	**0.44**	**1.21**	**0.25**	**81.7**	**1072.1**	**2.67**	0.63	**0.27**	**31.3**	179.3	**0.73**	0.96

Table 3. Quantitative evaluation of video synthesis quality and diversity (based on VGG [66]) on DTDB [30]. The diversity score based on the I3D [67] trained on DTDB [30] can be found in the supplemental. Note, DG [77] directly optimizes on test samples. Numeric superscripts indicate the source of the results, cf. Sec. 4.3.

Landscape [80] consists of ∼ 3000 time-lapse videos of dynamic sky scenes, e.g., cloudy skies and night scenes with moving stars. This dataset contains a wide range of sky appearances and motion speeds. Following previous work [80, 83], we evaluate on a sequence length of 32 frames. We compare with recent work on landscape synthesis [80, 21, 83, 45]. To generate sequences of length 32 we apply our model sequentially, meaning we use the last predicted frame as an input for the next prediction.

Dynamic Texture DataBase (DTDB) [30] contains more than 10,000 dynamic texture videos. For evaluation, we focus on the following classes: fire, clouds, vegetation, and waterfall. Each texture class consists of 150 to 300 videos. We train one model for each texture (same as for [21, 77]) on a sequence length of 16 on a resolution of 128 × 128.

BAIR Robot Pushing [20] consists of a randomly moving robotic arm that pushes and grasps objects in a box. It contains around 40k training and 256 test videos. This dataset is used by prior work as a benchmark due its stochastic nature and the real-world application. We follow the standard protocol [74, 70, 14, 58] and evaluate on a sequence length of 16 frames on a resolution of 64 × 64.

Impersonator (iPER) [44] is a recent dataset that contains humans with diverse styles of clothing executing various random actions. The entire dataset contains 206 videos with a total of $241,564$ frames. We follow the train/test split defined in [44] which leads to a training set containing 180k frames and a test set of 49k frames. We evaluate our model on a sequence length of 16 on a 64 × 64 resolution.

4.2. Evaluation Metrics

Synthesis quality. We evaluate the video synthesis quality using the Fréchet video distance (FVD) [70] which is sensitive to both perceptual quality and temporal coherence. This metric represents the spatiotemporal counterpart to FID [32] which is based on an I3D network [67] trained on Kinetics [37], a large-scale human action dataset. To evaluate dynamic textures, we introduce the Dynamic Texture Fréchet Video Distance (DTFVD) by replacing the pretrained network with one we trained on DTDB for classification [30]. The motivation behind introducing DTFVD is that we seek a metric that is sensitive to the types of dynamics encapsulated by dynamic textures, rather than human action-related motions as captured by FVD. To further evaluate dynamic textures, we also evaluate perceptual quality in terms of the Fréchet Inception Distance (FID) [32] and the Learned Perceptual Image Patch Similarity (LPIPS) [19, 35] metrics.

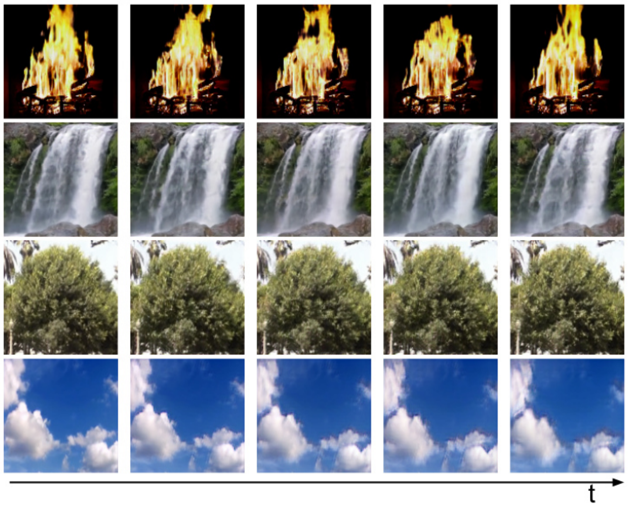

Figure 5. Stochastic video synthesis on DTDB [30] for diverse texture categories. Best viewed as video provided in the supplemental.

Diversity. Photorealism and plausible dynamics are not the only factors we are interested in. In addition, our model is capable of stochastically generating plausible videos from a single image. Following previous work [42] on video synthesis, we measure the diversity between video sequence predictions given an initial frame x_0 as their average mutual distance in the feature space of a VGG-16 network [66] pre-trained on ImageNet [63]. In contrast to [42], we use the Euclidean distance instead of the Cosine distance. Moreover, we also report diversity on pre-trained I3D [67] models (similar to above) which is sensitive to both appearance and motion instead of comparing samples frame-wise. We discuss and compare our chosen diversity measures in the supplemental.

4.3. Quantitative Evaluation

For comparison, we use reported performance from the corresponding paper (marked by ^{1}), where possible, otherwise we report numbers based on pretrained models (marked by ^{2}) or retrained models using the official code (marked by ^{3}) provided by the author.

Landscape. Tab. 1 provides a summary of our evaluation on Landscape in terms of perceptual quality and temporal coherence. As can be seen, we generally outperform all methods across all metrics. Animating Landscape

Method	FVD ↓	DIV VGG ↑	DIV I3D↑
Video Flow[1] [40]	131.0	–	–
SRVP[2] [26]	141.7	0.93	1.65
IVRNN[3] [11]	121.3	0.69	1.13
SAVP[1,2] [42]	116.4	**0.98**	1.70
LVT[1] [58]	125.8	–	–
DVD-GAN[1] [14]	109.8	–	–
Video Transformer[1] [74]	**94.0**	–	–
Ours w/o cINN	134.5	0.59	0.94
Ours w/o x_0	272.6	$2.40^\dagger$	$2.48^\dagger$
Ours w/o ADAIN	131.2	0.78	1.60
Ours	99.6	0.95	**1.75**

Table 4. Quantitative evaluation and ablation study of generation quality and diversity on BAIR [20]. Numeric superscripts indicate the source of the results, cf. Sec. 4.3. † denotes high diversity due to artifacts.

(AL) [21] stores the motion embeddings of all training instances in their codebook and uses them to generate videos during inference. In this way, AL is able to reproduce the diversity of the training videos. DTVNet [83] does not enforce a distribution over their representation and consequently are limited to deterministic video generations. DeepLandscape [45] (DL) does not learn dynamics from videos, but rather uses a manually constructed set of homographies. The pretrained model provided by DL was trained on their unreleased dataset. In contrast, we explicitly model and learn the dynamics distribution and by that, are able to synthesize *novel* dynamics to set scenes in motion.

DTDB. We observe similar results on DTDB (Tab. 3) on nearly all dynamic textures (fire, waterfall, and vegetation) across all perceptual quality and coherence metrics. For the clouds texture, AL achieves better results due to the fact that this motion can be faithfully described by optical flow. Here, we also consider results from methods dedicated to dynamic texture synthesis [77, 79] as strong baselines. These methods are not exactly comparable as they directly optimize on *test samples*. We only present results for DG [77], as Xie et al. [79] did not converge when trained on all test samples.

BAIR. We achieve strong results in terms of video quality (Tab. 4), even when compared with the computationally expensive transformer based approach [74]. In terms of diversity, we are on par with the state-of-the-art stochastic video prediction approaches.

iPER. The evaluation of articulated human motion is presented in Tab. 2 on iPER [44]. We achieve superior results to recent approaches for video prediction [25, 11, 42] in terms of FVD and diversity. Note, that we only condition on one frame in comparison to the baselines which use two [42, 11] and eight context frames [26].

4.4. Qualitative Evaluation

Image-to-video synthesis. We provide samples for all datasets. On Landscape [80] we see that our model is able to synthesize realistic samples (see Fig. 3) from diverse, complex scenes captured in the input image. In Fig. 4, we show samples on iPER [44] which illustrates the complexity of motion in the dataset. In Fig. 5, we visualize one sample per DTDB class which shows the variety of dynamic textures used for evaluation. Lastly, Fig. 6 (top two rows) show the diversity in our video samples by way of the differences across the last generated frame per sample on BAIR [20].

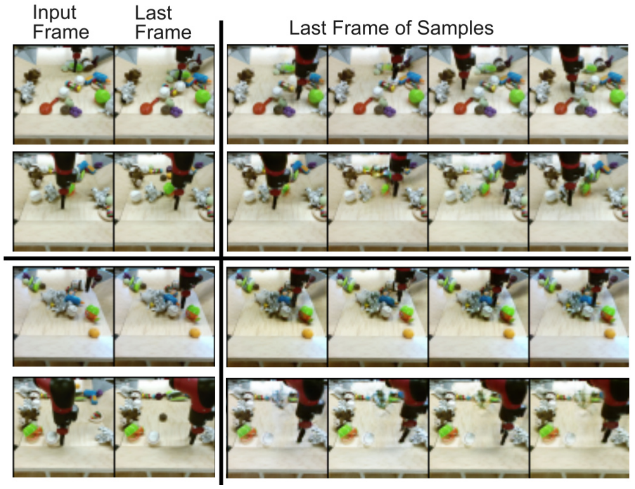

Figure 6. Qualitative evaluation of diversity on BAIR [20]. **Stochastic video synthesis:** (top two rows, left-to-right) input frame and last frame from a BAIR sequence and four frames representing the last frames from sampled videos generated using the input frame alone. The generated frames show a high degree of stochasticity in terms of the end effector position, as desired. **Controlled video synthesis:** (bottom two rows, left-to-right) input frame and last frame from a BAIR sequence, and four frames representing the last frames from sampled videos generated using both the input frame and the 3D end effector position in the last frame. The end effector position in the last frame is in close agreement with the position control input, as desired.

Controllable video synthesis. A strength of our model is the ability to exert explicit control over the synthesis process. As described in Sec. 3.4, we control this process by introducing a factor η. Here, we consider two different factors for controllable video synthesis on BAIR [20] and DTDB [30]. On BAIR we condition the synthesis process on the 3D location of the robot arm's end effector in the last frame; we use the location provided in the groundtruth. Fig. 6 (bottom two rows) shows several samples of the last frame of each sequence of our controllable synthesis. It can clearly be seen that the last frames of our samples match closely to the groundtruth end frame. As a second example, this time on DTDB [30], we condition the video synthesis of clouds based on the 2D direction of motion, again through manipulating η. This is visualized in Fig. 7 where four different directions are considered. To aid in the visualization, we also include the optical flow fields, estimated with [33], to show the consistency between the motion direction used

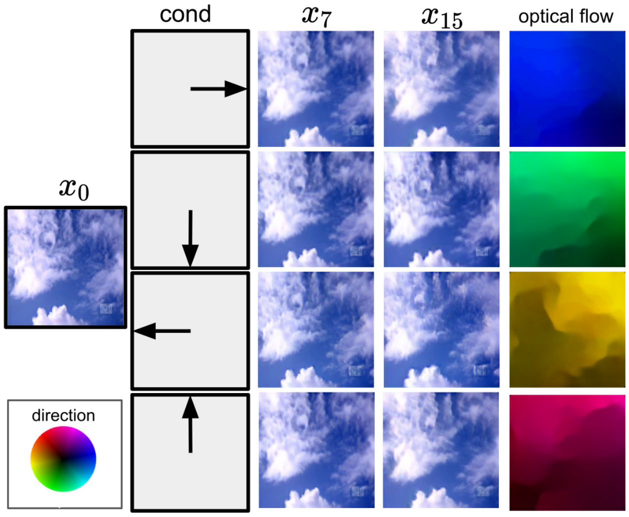

Figure 7. Examples of controlling video synthesis of clouds in DTDB [30] starting at frame x_0 using motion direction inputs (indicated by arrows). We show intermediate frames x_7 and x_{15}. The color wheel indicates flow direction.

for conditioning and the direction realized in the generated videos. As can be seen, the conditioning and generated motion directions are in close agreement. For results on controlled video-to-video synthesis (cf. Sec. 3.4), please refer to the supplemental.

Motion transfer. Finally, we illustrate the capability of our model to transfer a motion contained in one sequence to a set of initial frames for video synthesis. Using Landscape [80] Fig. 8 illustrates this process, where the top row contains the motion to be transferred and the bottom three rows show the generated video sequences realized by combining the transferred motion and the initial frames. As can be clearly seen, the original motion is successfully transferred to each of the scenes.

4.5. Ablation study

To evaluate the design choices of our approach, we now perform ablation studies on BAIR [20] and iPER [44]: (Ours w/o x_0) represents implementing our video generator, p_ψ without conditioning on the input image, x_0, thus z also captures the full scene content information, (Ours w/o ADAIN) similarly denotes removing the ADAIN input of z in our proposed Video ResBlk, i.e., p_ψ only has access to z via the bottleneck and (Ours w/o cVAE) stands for removing the cINN resulting in a cVAE framework.

In Tab. 2 and Tab. 4, we observe significant performance drops for all ablations compared to our full model (Ours). In particular removing the conditioning image, x_0, from the generator, p_ψ, greatly affects the synthesis quality. This is due to the generator not having direct access to the static information depicted in the initial frame x_0. When removing the ADAIN input of z from our Video ResBlk, the information of z is now only available at the lowest scale of p_ψ, in contrast to the multi-scale information flow in our full model. Moreover, the cVAE only model (w/o cINN) results in worse performance both in quality and diversity, which can be explained by the trade-off between synthesis quality and capacity regularization, as discussed in Sec. 3.3.

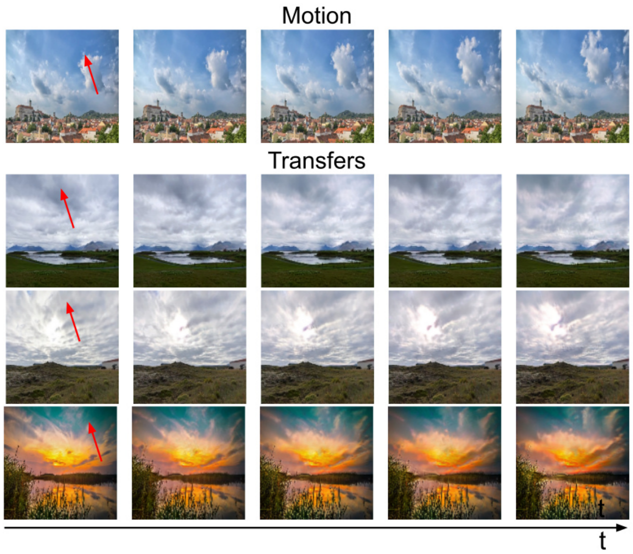

Figure 8. Transferring motion across videos on Landscape [80]. (top row, left-to-right) source video for target motion. (bottom three rows, left-to-right) animating different starting frames by transferring motion from source video. Red arrows indicate the direction of motion. Best viewed as video provided in the supplemental.

5. Conclusion

In summary, we introduced a novel model for understanding image-to-video synthesis based on a bijective transformation, instantiated as a cINN, between the video domain and the image domain plus residual information. The probabilistic residual representation allows to sample and synthesize novel, plausible progressions in video with the same initial frame. Moreover, our framework allows for incorporating additional controlling factors to guide the image-to-video synthesis process. Our empirical evaluation and comparison to strong baselines on four diverse video datasets demonstrated the efficacy of our stochastic image-to-video synthesis approach.

Acknowledgement

This work was started as part of M.D.'s internship at Ryerson University and was supported in part by the DAAD scholarship, the NSERC Discovery Grant program (K.G.D.), the German Research Foundation (DFG) within project 421703927 (B.O.) and the BW Stiftung (B.O.). K.G.D. contributed to this work in his capacity as an Associate Professor at Ryerson University.

References

[1] Lynton Ardizzone, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Analyzing inverse problems with invertible neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[2] Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Guided image generation with conditional invertible neural networks. *CoRR*, 2019.

[3] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H. Campbell, and Sergey Levine. Stochastic variational video prediction. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[4] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3703–3712, 2019.

[5] Biagio Brattoli, Uta Büchler, Michael Dorkenwald, Philipp Reiser, Linard Filli, Fritjof Helmchen, Anna-Sophia Wahl, and Björn Ommer. uBAM: Unsupervised behavior analysis and magnification using deep learning. *CoRR*, 2020.

[6] Biagio Brattoli, Uta Büchler, Anna-Sophia Wahl, Martin E. Schwab, and Björn Ommer. LSTM self-supervision for detailed behavior analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3747–3756, 2017.

[7] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[8] Andreja Bubic, D Yves von Cramon, and Ricarda I Schubotz. Prediction, cognition and the brain. *Frontiers in human neuroscience*, 4:25, 2010.

[9] Christopher P. Burgess, Irina Higgins, Arka Pal, Loïc Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-VAE. *CoRR*, 2018.

[10] Arunkumar Byravan, Felix Leeb, Franziska Meier, and Dieter Fox. Se3-pose-nets: Structured deep dynamics models for visuomotor planning and control. *CoRR*, 2017.

[11] Lluís Castrejón, Nicolas Ballas, and Aaron C. Courville. Improved conditional vrnns for video prediction. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 7607–7616, 2019.

[12] Shiming Chen, Peng Zhang, Xinge You, Qinmu Peng, Xin Liu, and Zehong Cao. Similarity-dt: Kernel similarity embedding for dynamic texture synthesis. *CoRR*, 2019.

[13] Xi Chen, Diederik P. Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[14] Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial video generation on complex datasets, 2019.

[15] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1182–1191, 2018.

[16] Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: non-linear independent components estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[17] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[18] Michael Dorkenwald, Uta Büchler, and Björn Ommer. Unsupervised magnification of posture deviations across subjects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8253–8263, 2020.

[19] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *Neural Information Processing Systems (NeurIPS)*, pages 658–666, 2016.

[20] Frederik Ebert, Chelsea Finn, Alex X. Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. In *Conference on Robot Learning (CoRL)*, pages 344–356, 2017.

[21] Yuki Endo, Yoshihiro Kanamori, and Shigeru Kuriyama. Animating landscape: self-supervised learning of decoupled motion and appearance for single-image video synthesis. *ACM Transactions on Graphics*, pages 175:1–175:19, 2019.

[22] Patrick Esser, Johannes Haux, Timo Milbich, and Björn Ommer. Towards learning a realistic rendering of human behavior. In *ECCV Workshops*, pages 409–425, 2018.

[23] Patrick Esser, Robin Rombach, and Björn Ommer. A disentangling invertible interpretation network for explaining latent representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9220–9229, 2020.

[24] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793, 2017.

[25] Jean-Yves Franceschi, Edouard Delasalles, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari. Stochastic latent residual video prediction. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 3233–3246, 2020.

[26] Jean-Yves Franceschi, Edouard Delasalles, Mickaël Chen, Sylvain Lamprier, and Patrick Gallinari. Stochastic latent residual video prediction. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 3233–3246, 2020.

[27] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *CoRR*, 2015.

[28] Patrick Gebert, Alina Roitberg, Monica Haurilet, and Rainer Stiefelhagen. End-to-end prediction of driver intention using 3D convolutional neural networks. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 969–974, 2019.

[29] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville,

and Yoshua Bengio. Generative adversarial networks. *CoRR*, 2014.

[30] Isma Hadji and Richard P. Wildes. A new large scale dynamic texture dataset with application to convnet understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 334–351, 2018.

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[32] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Neural Information Processing Systems (NeurIPS)*, pages 6626–6637, 2017.

[33] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1647–1655, 2017.

[34] Jörn-Henrik Jacobsen, Arnold W. M. Smeulders, and Edouard Oyallon. i-revnet: Deep invertible networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[35] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 694–711, 2016.

[36] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4401–4410, 2019.

[37] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The Kinetics human action video dataset. *CoRR*, 2017.

[38] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Neural Information Processing Systems (NeurIPS)*, pages 10236–10245, 2018.

[39] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[40] Manoj Kumar, Mohammad Babaeizadeh, Dumitru Erhan, Chelsea Finn, Sergey Levine, Laurent Dinh, and Durk Kingma. Videoflow: A conditional flow-based model for stochastic video generation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[41] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1558–1566, 2016.

[42] Alex X. Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *CoRR*, 2018.

[43] Jae Hyun Lim and Jong Chul Ye. Geometric gan, 2017.

[44] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. Liquid warping GAN: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 5903–5912, 2019.

[45] Elizaveta Logacheva, Roman Suvorov, Oleg Khomenko, Anton Mashikhin, and Victor Lempitsky. Deeplandscape: Adversarial modeling of landscape videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 256–272, 2020.

[46] Dominik Lorenz, Leonard Bereska, Timo Milbich, and Björn Ommer. Unsupervised part-based disentangling of object shape and appearance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10955–10964, 2019.

[47] Andreas Lugmayr, Martin Danelljan, Luc Van Gool, and Radu Timofte. SRFlow: Learning the super-resolution space with normalizing flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 715–732, 2020.

[48] Manuel Martin, Alina Roitberg, Monica Haurilet, Matthias Horne, Simon Reiß, Michael Voit, and Rainer Stiefelhagen. Drive&act: A multi-modal dataset for fine-grained driver behavior recognition in autonomous vehicles. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2801–2810, 2019.

[49] Sarfaraz Masood, Abhinav Rai, Aakash Aggarwal, Mohammad Najam Doja, and Musheer Ahmad. Detecting distraction of drivers using convolutional neural network. *Pattern Recognition Letters*, pages 79–85, 2020.

[50] Timo Milbich, Miguel Ángel Bautista, Ekaterina Sutter, and Björn Ommer. Unsupervised video understanding by reconciliation of posture similarities. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 4404–4414, 2017.

[51] Matthias Minderer, Chen Sun, Ruben Villegas, Forrester Cole, Kevin P. Murphy, and Honglak Lee. Unsupervised learning of object structure and dynamics from videos. In *Neural Information Processing Systems (NeurIPS)*, pages 92–102, 2019.

[52] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1701–1710, 2018.

[53] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5436–5445, 2020.

[54] George Papamakarios, Eric T. Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *CoRR*, 2019.

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

[56] Karl Pertsch, Oleh Rybkin, Jingyun Yang, Shenghao Zhou, Konstantinos G. Derpanis, Kostas Daniilidis, Joseph J. Lim, and Andrew Jaegle. Keyframing the future: Keyframe discovery for visual prediction and planning. In *L4DC*, Proceedings of Machine Learning Research, pages 969–979, 2020.

[57] Albert Pumarola, Stefan Popov, Francesc Moreno-Noguer, and Vittorio Ferrari. C-flow: Conditional generative flow models for images and 3D point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7946–7955, 2020.

[58] Ruslan Rakhimov, Denis Volkhonskiy, Alexey Artemov, Denis Zorin, and Evgeny Burnaev. Latent video transformer. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP*, pages 101–112, 2021.

[59] Fitsum A. Reda, Guilin Liu, Kevin J. Shih, Robert Kirby, Jon Barker, David Tarjan, Andrew Tao, and Bryan Catanzaro. Sdcnet: Video prediction using spatially-displaced convolution. *CoRR*, 2018.

[60] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1530–1538, 2015.

[61] Robin Rombach, Patrick Esser, and Björn Ommer. Making sense of cnns: Interpreting deep representations and their invariances with inns. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 647–664, 2020.

[62] Robin Rombach, Patrick Esser, and Björn Ommer. Network-to-network translation with conditional invertible neural networks. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[63] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3), 2015.

[64] Aliaksandr Siarohin, Stéphane Lathuiliére, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2377–2386, 2019.

[65] Aliaksandr Siarohin, Stéphane Lathuiliére, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *Neural Information Processing Systems (NeurIPS)*, pages 7135–7145, 2019.

[66] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[67] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[68] Matthew Tesfaldet, Marcus A. Brubaker, and Konstantinos G. Derpanis. Two-stream convolutional networks for dynamic texture synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6703–6712, 2018.

[69] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1526–1535, 2018.

[70] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *CoRR*, 2018.

[71] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[72] Jacob Walker, Kenneth Marino, Abhinav Gupta, and Martial Hebert. The pose knows: Video forecasting by generating pose futures. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 3352–3361, 2017.

[73] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Nikolai Yakovenko, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *Neural Information Processing Systems (NeurIPS)*, pages 1152–1164, 2018.

[74] Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[75] Nevan Wichers, Ruben Villegas, Dumitru Erhan, and Honglak Lee. Hierarchical long-term video prediction without supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 6033–6041, 2018.

[76] Yue Wu, Rongrong Gao, Jaesik Park, and Qifeng Chen. Future video synthesis with object motion prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5538–5547, 2020.

[77] Jianwen Xie, Ruiqi Gao, Zilong Zheng, Song-Chun Zhu, and Ying Nian Wu. Learning dynamic generator model by alternating back-propagation through time. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 5498–5507, 2019.

[78] Jianwen Xie, Yang Lu, Ruiqi Gao, Song-Chun Zhu, and Ying Nian Wu. Cooperative training of descriptor and generator networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, pages 27–45, 2020.

[79] Jianwen Xie, Song-Chun Zhu, and Ying Nian Wu. Synthesizing dynamic patterns by spatial-temporal generative convnet. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1061–1069, 2017.

[80] Wei Xiong, Wenhan Luo, Lin Ma, Wei Liu, and Jiebo Luo. Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2364–2373, 2018.

[81] Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. Attribute2image: Conditional image generation from visual

attributes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 776–791, 2016.

[82] Jason J. Yu, Konstantinos G. Derpanis, and Marcus A. Brubaker. Wavelet flow: Fast training of high resolution normalizing flows. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[83] Jiangning Zhang, Chao Xu, Liang Liu, Mengmeng Wang, Xia Wu, Yong Liu, and Yunliang Jiang. Dtvnet: Dynamic time-lapse video generation via single still image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 300–315, 2020.

[84] Shengjia Zhao, Jiaming Song, and Stefano Ermon. InfoVAE: Information maximizing variational autoencoders. *CoRR*, 2017.