Diffusion Probabilistic Modeling for Video Generation

Anonymous authors
Paper under double-blind review

Abstract

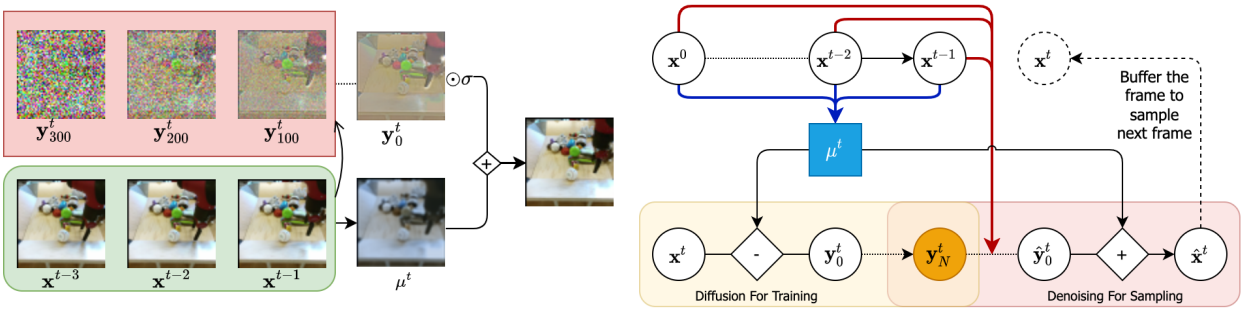
Denoising diffusion probabilistic models are a promising new class of generative models that mark a milestone in high-quality image generation. This paper showcases their ability to sequentially generate video, surpassing prior methods in perceptual and probabilistic forecasting metrics. We propose an autoregressive, end-to-end optimized video diffusion model inspired by recent advances in neural video compression. The model successively generates future frames by correcting a deterministic next-frame prediction using a stochastic residual generated by an inverse diffusion process. We compare this approach against five baselines on four datasets involving natural and simulation-based videos. We find significant improvements in terms of perceptual quality for all datasets. Furthermore, by introducing a scalable version of the Continuous Ranked Probability Score (CRPS) applicable to video, we show that our model also outperforms existing approaches in their probabilistic frame forecasting ability.

1 Introduction

The ability to anticipate future frames of a video is intuitive for humans but challenging for a computer (Oprea et al., 2020). Applications of such *video prediction* tasks include anticipating events (Vondrick et al., 2016a), model-based reinforcement learning (Ha & Schmidhuber, 2018), video interpolation (Liu et al., 2017), predicting pedestrians in traffic (Bhattacharyya et al., 2018), precipitation nowcasting (Ravuri et al., 2021), neural video compression (Han et al., 2019; Lu et al., 2019; Agustsson et al., 2020; Yang et al., 2020; 2021a; 2022), and many more.

The goals and challenges of video prediction include (i) generating multi-modal, stochastic predictions that (ii) accurately reflect the high-dimensional dynamics of the data long-term while (iii) identifying architectures that scale to high-resolution content without blurry artifacts. These goals are complicated by occlusions, lighting conditions, and dynamics on different temporal scales. Broadly speaking, models relying on sequential variational autoencoders (Babaeizadeh et al., 2018; Denton & Fergus, 2018; Castrejon et al., 2019) tend to be stronger in goals (i) and (ii), while sequential extensions of generative adversarial networks (Aigner & Körner, 2018; Kwon & Park, 2019; Lee et al., 2018) tend to perform better in goal (iii). A probabilistic method that succeeds in all the three desiderata on high-resolution video content is yet to be found.

Recently, diffusion probabilistic models have achieved considerable progress in image generation, with perceptual qualities comparable to GANs while avoiding the optimization challenges of adversarial training (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020; Song et al., 2021c;b). In this paper, we extend diffusion probabilistic models for stochastic video generation. Our ideas are inspired by the principles of predictive coding (Rao & Ballard, 1999; Marino, 2021) and neural compression algorithms (Yang et al., 2021b) and draw on the intuition that residual errors are easier to model than dense observations (Marino et al., 2021). Our architecture relies on two prediction steps: first, we employ a deterministic convolutional RNN to deterministically predict the next frame conditioned on a sequence of frames. Second, we correct this prediction by an additive residual generated by a conditional denoising diffusion process (see Figs. 1a and 1b). This approach is scalable to high-resolution video, stochastic, and relies on likelihood-based principles. Our ablation studies strongly suggest that predicting video frame residuals instead of naively predicting the next frames improves generative performance. By investigating our architecture on various datasets and



(a) Overview.

(b) Detailed model.

Figure 1: Overview: Our approach predicts the next frame μ_t of a video autoregressively along with an additive correction $\mathbf{y}_0^t$ generated by a denoising process. Detailed model: Two convolutional RNNs (blue and red arrows) operate on a frame sequence $\mathbf{x}^{0:t-1}$ to predict the most likely next frame μ^t (blue box) and a context vector for a denoising diffusion model. The diffusion model is trained to model the scaled residual $\mathbf{y}_0^t = (\mathbf{x}^t - \mu^t)/\sigma$ conditioned on the temporal context. At generation time, the generated residual is added to the next-frame estimate μ^t to generate the next frame as $\mathbf{x}^t = \mu^t + \sigma\mathbf{y}_0^t$.

comparing it against multiple baselines, we achieve a new state-of-the-art in video generation that produces sharp frames at higher resolutions. In more detail, our achievements are as follows:

1. We show how to use diffusion probabilistic models to generate videos. This enables a new path towards probabilistic video forecasting while achieving perceptual qualities better than or comparable with likelihood-free methods such as GANs.

2. We also study a marginal version of the Continuous Ranked Probability Score and show that it can be used to assess video prediction performance. Our method is also better at probabilistic forecasting than modern GAN and VAE baselines such as IVRNN, SVG-LP, RetroGAN, DVD-GAN and FutureGAN (Castrejon et al., 2019; Denton & Fergus, 2018; Kwon & Park, 2019; Clark et al., 2019; Aigner & Körner, 2018).

3. Our ablation studies demonstrate that modeling residuals from the predicted next frame yields better results than directly modeling the next frames. This observation is consistent with recent findings in neural video compression. Figure 1a summarizes the main idea of our approach (Figure 1b has more details).

The structure of our paper is as follows. We first describe our method, which is followed by a discussion about our experimental findings along with ablation studies. We then discuss connections to the literature, summarize our contributions.

2 A Diffusion Probabilistic Model for Video

We begin by reviewing the relevant background on diffusion probabilistic models. We then discuss our design choices for extending these models to sequential models for video.

2.1 Background on Diffusion Probabilistic Models

Denoising diffusion probabilistic models (DDPMs) are a recent class of generative models with promising properties (Sohl-Dickstein et al., 2015; Ho et al., 2020). Unlike GANs, these models rely on the maximum likelihood training paradigm (and are thus stable to train) while producing samples of comparable perceptual quality as GANs (Brock et al., 2019).

Similar to hierarchical variational autoencoders (VAEs) (Kingma & Welling, 2013), DDPMs are deep latent variable models that model data $\mathbf{x}_0$ in terms of an underlying sequence of latent variables $\mathbf{x}_{1:N}$ such that $p_\theta(\mathbf{x}_0) = \int p_\theta(\mathbf{x}_{0:N}) d\mathbf{x}_{1:N}$. The main idea is to impose a diffusion process on the data that incrementally destroys the structure. The diffusion process's incremental *posterior* yields a stochastic denoising process that can be used to *generate* structure (Sohl-Dickstein et al., 2015; Ho et al., 2020). The *forward*, or *diffusion* process is given by

$$\begin{aligned} q(\mathbf{x}_{1:N}|\mathbf{x}_0) &= \prod_{n=1}^{N} q(\mathbf{x}_n|\mathbf{x}_{n-1}); \\ q(\mathbf{x}_n|\mathbf{x}_{n-1}) &= \mathcal{N}(\mathbf{x}_n|\sqrt{1-\beta_n}\mathbf{x}_{n-1}, \beta_n \mathbf{I}). \end{aligned} \tag{1}$$

Besides a predefined incremental variance schedule with $\beta_n \in (0,1)$ for $n \in \{1, \cdots, N\}$, this process is parameter-free (Song & Ermon, 2019; Ho et al., 2020). The reverse process is called *denoising process*,

$$\begin{aligned} p_\theta(\mathbf{x}_{0:N}) &= p(\mathbf{x}_N) \prod_{n=1}^{N} p_\theta(\mathbf{x}_{n-1}|\mathbf{x}_n); \\ p_\theta(\mathbf{x}_{n-1}|\mathbf{x}_n) &= \mathcal{N}(\mathbf{x}_{n-1}|M_\theta(\mathbf{x}_n, n), \gamma \mathbf{I}). \end{aligned} \tag{2}$$

The reverse process can be thought of as approximating the posterior of the diffusion process. Typically, one fixes the covariance matrix (with hyperparameter γ) and only learns the posterior mean function $M_\theta(\mathbf{x}_n, n)$. The prior $p(\mathbf{x}_N) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ is typically fixed. The parameter θ can be optimized by maximizing a variational lower bound on the log-likelihood, $L_{\text{variational}} = \mathbb{E}_q[-\log \frac{p_\theta(\mathbf{x}_{0:N})}{q(\mathbf{x}_{1:N}|\mathbf{x}_0)}]$. This bound can be efficiently estimated by stochastic gradients by subsampling time steps n at random since the marginal distributions $q(\mathbf{x}_n|\mathbf{x}_0)$ can be computed in closed form (Ho et al., 2020).

In this paper, we use a simplified loss due Ho et al. (2020) who showed that the variational bound could be simplified to the following denoising *score matching loss*,

$$\begin{aligned} L(\theta) &= \mathbb{E}_{\mathbf{x}_0, n, \epsilon} ||\epsilon - f_\theta(\mathbf{x}_n, n)||^2 \\ \text{where } \mathbf{x}_n &= \sqrt{\bar{\alpha}_n}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_n}\epsilon. \end{aligned} \tag{3}$$

We thereby define $\bar{\alpha}_n = \prod_{i=1}^{n}(1-\beta_i)$. The intuitive explanation of this loss is that f_θ tries to predict the noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ at the denoising step n (Ho et al., 2020). Once the model is trained, it can be used to generate data by ancestral sampling, starting with a draw from the prior $p(\mathbf{x}_N)$ and successively generating more and more structure through an annealed Langevin dynamics procedure (Song & Ermon, 2019; Song et al., 2021c).

2.2 Residual Video Diffusion Model

Experience shows that it is often simpler to model *differences* from our predictions than the predictions themselves. For example, masked autoregressive flows (Papamakarios et al., 2017) transform random noise into an additive prediction error residual and boosting algorithms train a sequence of models to predict the error residuals of earlier models (Schapire, 1999). Residual errors also play an important role in modern theories of the brain. For example, predictive coding (Rao & Ballard, 1999) postulates that neural circuits estimate probabilistic models of other neural activity, iteratively exchanging information about error residuals. This theory has interesting connections to VAEs (Marino, 2021; Marino et al., 2021) and neural video compression (Agustsson et al., 2020; Yang et al., 2021a), where one also compresses the residuals to the most likely next-frame predictions.

This work uses a diffusion model to generate *residual corrections* to a deterministically predicted next frame, adding stochasticity to the video generation task. Both the deterministic prediction as well as the denoising process are conditioned on a long-range context provided by a convolutional RNN. We call our approach "Residual Video Diffusion" (RVD). Details will be explained next.

Notation. We consider a frame sequence $\mathbf{x}^{0:T}$ and a set of latent variables $\mathbf{y}^{1:T} \equiv \mathbf{y}_{0:N}^{1:T}$ specified by a diffusion process over the lower indices. We refer to $\mathbf{y}_0^{1:T}$ as the (scaled) frame residuals.

Generative Process. We consider a joint distribution over $\mathbf{x}^{0:T}$ and $\mathbf{y}^{1:T}$ of the following form:

$$p(\mathbf{x}^{0:T}, \mathbf{y}^{1:T}) = p(\mathbf{x}^0) \prod_{t=1}^{T} p(\mathbf{x}^t|\mathbf{y}^t, \mathbf{x}^{<t}) p(\mathbf{y}^t|\mathbf{x}^{<t}). \tag{4}$$

We first specify the data likelihood term $p(\mathbf{x}^t|\mathbf{y}^t, \mathbf{x}^{<t})$, which we model autoregressively as a *Masked Autoregressive Flow* (MAF) (Papamakarios et al., 2017) applied to the frame sequence. This involves an autoregressive prediction network outputting μ_ϕ and a scale parameter σ,

$$\mathbf{x}^t = \mu_\phi(\mathbf{x}^{<t}) + \sigma \odot \mathbf{y}_0^t \iff \mathbf{y}_0^t = \frac{\mathbf{x}^t - \mu_\phi(\mathbf{x}^{<t})}{\sigma}. \tag{5}$$

Conditioned on $\mathbf{y}_0^t$, this transformation is deterministic. The forward MAF transform ($\mathbf{y} \to \mathbf{x}$) converts the residual into the data sequence; the inverse transform ($\mathbf{x} \to \mathbf{y}$) decorrelates the sequence. The temporally decorrelated, sparse residuals $\mathbf{y}_0^{1:T}$ involve a simpler modeling task than generating the frames themselves. While the scale parameter σ can also be conditioned on past frames, we did not find a benefit in practice.

The autoregressive transform in Equation 5 has also been adapted in a VAE model (Marino et al., 2021) as well as in neural video compression architectures (Agustsson et al., 2020; Yang et al., 2020; 2021a;b). These approaches separately compress latent variables that govern the next-frame prediction as well as frame residuals, thereby achieving state-of-the-art rate-distortion performance on high-resolution video content. While these works focused on compression, this paper focuses on generation.

We now specify the second factor in Eq. 4, the generative process of the residual variable, as

$$p_\theta(\mathbf{y}_{0:N}^t|\mathbf{x}^{<t}) = p(\mathbf{y}_N^t) \prod_{n=1}^{N} p_\theta(\mathbf{y}_{n-1}^t|\mathbf{y}_n^t, \mathbf{x}^{<t}). \tag{6}$$

We fix the top-level prior distribution to be a multivariate Gaussian with identity covariance. All other denoising factors are conditioned on past frames and involve prediction networks M_θ,

$$p_\theta(\mathbf{y}_{n-1}|\mathbf{y}_n, \mathbf{x}^{<t}) = \mathcal{N}(\mathbf{y}_{n-1}|M_\theta(\mathbf{y}_n, n, \mathbf{x}^{<t}), \gamma\mathbf{I}). \tag{7}$$

As in Eq. 2, γ is a hyperparameter. Our goal is to learn θ.

Inference Process. Having specified the generative process, we next specify the inference process conditioned on the observed sequence $\mathbf{x}^{0:T}$:

$$q_\phi(\mathbf{y}_{0:N}^t|\mathbf{x}^{\leq t}) = q_\phi(\mathbf{y}_0^t|\mathbf{x}^{\leq t}) \prod_{n=1}^{N} q(\mathbf{y}_n^t|\mathbf{y}_{n-1}^t). \tag{8}$$

Since the residual noise is a deterministic function of the observed and predicted frame, the first factor is deterministic and can be expressed as $q_\phi(\mathbf{y}_0^t|\mathbf{x}^{\leq t}) = \delta(\mathbf{y}_0^t - \frac{\mathbf{x}^t - \mu_\phi(\mathbf{x}^{<t})}{\sigma})$. The remaining N factors are identical to Eq. 1 with $\mathbf{x}_n$ being replaced by $\mathbf{y}_n$. Following Nichol & Dhariwal (2021), we use a cosine schedule to define the variance $\beta_n \in (0, 1)$. The architecture is shown in Figure 1b.

Equations 7 and 8 generalize and improve the previously proposed TimeGrad (Rasul et al., 2021) method. This approach showed promising performance in forecasting time series of comparatively smaller dimensions such as electricity prices or taxi trajectories and not video. Besides differences in architecture, this method neither models residuals nor considers the temporal dependency in posterior, which we identify as a crucial aspect to make the model competitive with strong VAE and GAN baselines (see Section 3.6 for an ablation).

Optimization and Sampling. In analogy to time-independent diffusion models, we can derive a variational lower bound that we can optimize using stochastic gradient descent. In analogy to to the derivation of Eq. 3 (Ho et al., 2020) and using the same definitions of $\bar{\alpha}_n$ and ϵ, this results in

$$\begin{aligned} L(\theta, \phi) &= \mathbb{E}_{\mathbf{x}, n, \epsilon} \sum_{t=1}^{T} ||\epsilon - f_\theta(\mathbf{y}_n^t(\phi), n, \mathbf{x}^{<t})||^2; \\ \mathbf{y}_n^t(\phi) &= \sqrt{\bar{\alpha}_n}\mathbf{y}_0^t(\phi) + \sqrt{1 - \bar{\alpha}_n}\epsilon; \mathbf{y}_0^t(\phi) = \frac{\mathbf{x}^t - \mu_\phi(\mathbf{x}^{<t})}{\sigma}. \end{aligned} \tag{9}$$

We can optimize this function using the reparameterization trick Kingma & Welling (2013), i.e., by randomly sampling ϵ and n, and taking stochastic gradients with respect to ϕ and θ. For a practical scheme involving multiple time steps, we also employ teacher forcing Kolen & Kremer (2001). See Algorithm 1 for the detailed training and sampling procedure, where we abbreviated $f_{\theta,\phi}(\mathbf{y}_n^t, n, \mathbf{x}^{<t}) \equiv f_\theta(\mathbf{y}_n^t(\phi), n, \mathbf{x}^{<t})$.

Algorithm 1: Training (left) and Video Generation (right)

while *not converged* **do**
- Sample $\mathbf{x}^{0:T} \sim q(\mathbf{x}^{0:T})$;
- $n \sim \mathcal{U}(0, 1, 2, .., N)$;
- $L = 0$;
- **for** *t=1 to T* **do**
  - $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
  - $\mathbf{y}_0^t = (\mathbf{x}^t - \mu_\phi(\mathbf{x}^{<t}))/\sigma$;
  - $\mathbf{y}_n^t = \sqrt{\bar{\alpha}_n}\mathbf{y}_0^t + \sqrt{1-\bar{\alpha}_n}\epsilon$;
  - $L = L + ||\epsilon - f_{\theta,\phi}(\mathbf{y}_n^t, n, \mathbf{x}^{<t})||^2$
- **end**
- $(\theta, \phi) = (\theta, \phi) - \nabla_{\theta,\phi} L$

end

Get initial context frame $\mathbf{x}^0 \sim q(\mathbf{x}^0)$;

for *t=1 to T* **do**
- $\mathbf{y}_N^t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
- **for** *n=N to 1* **do**
  - $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
  - $\epsilon_\theta = f_{\theta,\phi}(\mathbf{y}_n^t, n, \mathbf{x}^{<t})$;
  - $\tilde{\mathbf{y}}_{n-1}^t = \mathbf{y}_n^t - \frac{\beta_n}{\sqrt{1-\bar{\alpha}_n}}\epsilon_\theta$;
  - $\mathbf{y}_{n-1}^t = \frac{1}{\sqrt{\alpha_n}}\tilde{\mathbf{y}}_{n-1}^t + \frac{1-\bar{\alpha}_{n-1}}{1-\bar{\alpha}_n}\beta_n \mathbf{z}$;
- **end**
- $\mathbf{x}^t = \sigma \odot \mathbf{y}_0^t + \mu_\phi(\mathbf{x}^{<t})$

end

3 Experiments

We compare Residual Video Diffusion (RVD) against five strong baselines, including three GAN-based models and two sequential VAEs. We consider four different video datasets and consider both probabilistic (CRPS) and perceptual (FVD, LPIPS) metrics, discussed below. Our model achieves a new state of the art in terms of perceptual quality while being comparable with or better than the best-performing sequential VAE in its frame forecasting ability.

3.1 Datasets

We consider four video datasets of varying complexities and resolutions. Among the simpler datasets of frame dimensions of 64×64, we consider the **BAIR** Robot Pushing arm dataset (Ebert et al., 2017) and **KTH Actions** (Schuldt et al., 2004). Amongst the high-resolution datasets (frame sizes of 128×128), we use **Cityscape** (Cordts et al., 2016), a dataset involving urban street scenes, and a two-dimensional **Simulation** dataset for turbulent flow of our own making that has been computed using the Lattice Boltzmann Method (Chirila, 2018). These datasets cover various complexities, resolutions, and types of dynamics.

Preprocessing. For KTH and BAIR, we preprocess the videos as commonly proposed (Denton & Fergus, 2018; Marino et al., 2021). For Cityscape, we download the portion titled *leftImg8bit_sequence_trainvaltest* from the official website[1]. Each video is a 30-frame sequence from which we randomly select a sub-sequence. All the videos are center-cropped and downsampled to 128x128. For the simulation dataset, we use an LBM solver to simulate the flow of a fluid (with pre-specified bulk and shear viscosity and rate of flow) interacting with a static object. We extract 10000 frames sampled every 128 ticks, using 8000 for training and 2000 for testing.

3.2 Training and Testing Details

The diffusion models are trained with 8 consecutive frames for all the datasets of which the first two frames as used as context frames. We set the batchsize to 4 for all high-resolution videos and to 8 for all low-resolution videos. The pixel values of all the video frames are normalized to $[-1, 1]$. The models are optimized using the Adam optimizer with an initial learning rate of 5×10^{-5}, which decays to 2×10^{-5}. All the models are trained on a NVIDIA RTX Titan GPU. The number of diffusion depth is fixed to $N = 1600$ and the

[1] `https://www.cityscapes-dataset.com/downloads/`

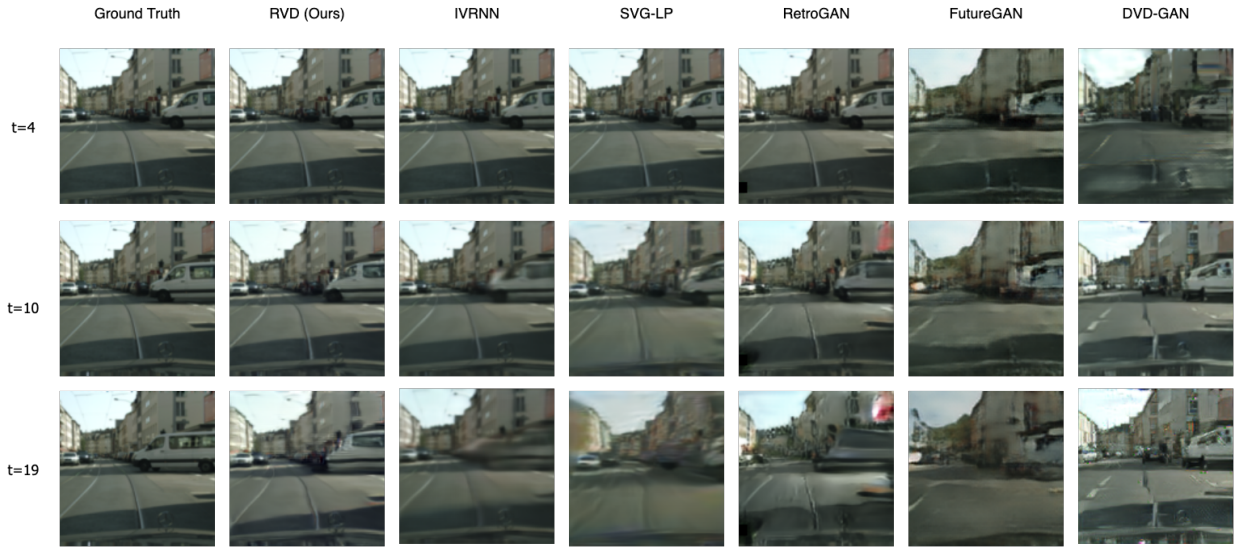


Figure 2: Generated frames on Cityscape (128x128). RVD (ours) has less artifacts and blurriness than its competitors.

scale term is set to $\sigma = 2$. For testing, we use 4 context frames and predict 16 future frames for each video sequence. Wherever applicable, these frames are recursively generated.

3.3 Baseline Models

SVG-LP (Denton & Fergus, 2018) is an established sequential VAE baseline. It leverages recurrent architectures in all of encoder, decoder, and prior to capture the dynamics in videos. We adapt the official implementation from the authors while replacing all the LSTM with ConvLSTM layers, which helps the model scale to different video resolutions. **IVRNN** (Castrejon et al., 2019) is currently the state-of-the-art video-VAE model trained end-to-end from scratch. The model improves SVG by involving a hierarchy of latent variables. We use the official codebase to train the model. **FutureGAN** (Aigner & Körner, 2018) relies on an *encoder-decoder* GAN model that uses spatio-temporal 3D convolutions to process video tensors. In order to make the quality of the output more perceptually appealing, the paper employs the concept of progressively growing GANs. We use the official codebase to train the model. **Retrospective Cycle GAN** (Kwon & Park, 2019) employs a single generator that can predict both future and past frames given a context and enforces retrospective cycle constraints. Besides the usual discriminator that can identify fake frames, the method also introduces sequence discriminators to identify sequences containing the said fake frames. We used an available third-party implementation [2]. **DVD-GAN** (Clark et al., 2019) proposes an alternative dual-discriminator architecture for video generation on complex datasets. We also adapt a third-party implementation of the model to conduct our experiment [3].

3.4 Evaluation Metrics

We address two key aspects for determining the quality of generated sequences: perceptual quality and the models' probabilistic forecasting ability. For the former, we adopt FVD (Unterthiner et al., 2019) and LPIPS (Zhang et al., 2018), while the latter is evaluated using a new extension of CRPS (Matheson & Winkler, 1976) that we propose for video evaluation.

Fréchet Video Distance (FVD) compares sample realism by calculating 2-Wasserstein distance between the ground truth *video* distribution and the distribution defined by the generative model. Typically, an I3D

[2] `https://github.com/SaulZhang/Video_Prediction_ZOO/tree/master/RetrospectiveCycleGAN`
[3] `https://github.com/Harrypotterrrr/DVD-GAN`

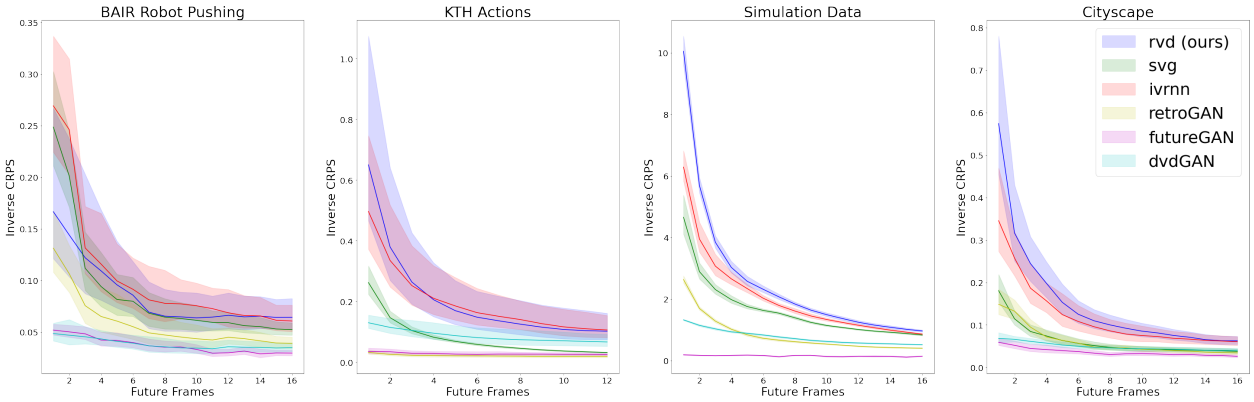


Figure 3: Inverse CRPS scores (higher is better) as a function of the future frame index. The best performances are obtained by RVD (proposed) and IVRNN. Scores also monotonically decrease as the predictions worsen over time.

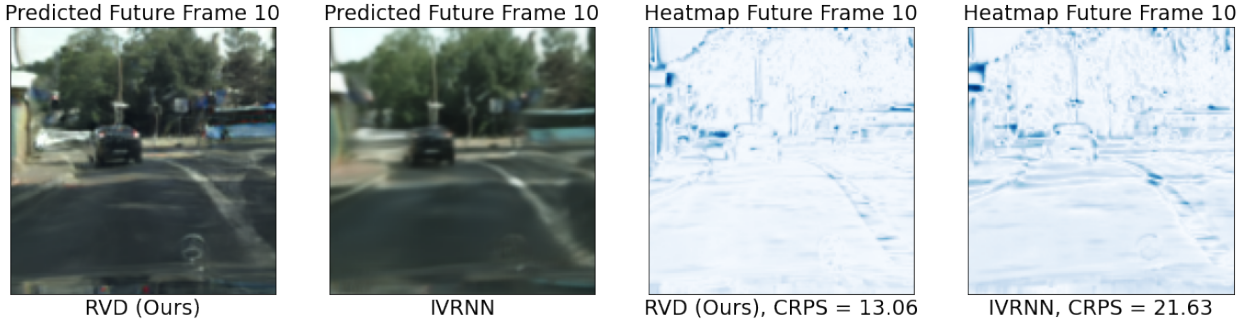


Figure 4: Spatially-resolved CRPS scores (right two plots, lower is better). We compare the performance of RVD (proposed) against IVRNN on predicting the 10th future frame of a video from CityScape. Darker areas point to larger disagreements with respect to the ground truth.

network pretrained on an action-recognition dataset is used to capture low dimensional feature representations, the distributions of which are used in the metric. **Learned Perceptual Image Patch Similarity** (LPIPS), on the other hand, computes the L2 distance between deep embeddings across all the layers of a pretrained network which are then averaged spatially. The LPIPS score is calculated on the basis of individual frames and then averaged.

Apart from realism, it is also important that the generated sequence covers the full distribution of possible outcomes. Given the multi-modality of future video outcomes, such an evaluation is challenging. In this paper, we draw inspiration from the **Continuous Ranked Probability Score** (CRPS), a forecasting metric and proper scoring rule typically associated with time-series data. We calculate CRPS for each pixel in every generated frame and then take an average both spatially and temporally. This results in a CRPS metric on *marginal* (i.e., pixel-level) distributions that scale well to high-dimensional data. To the best of our knowledge, we are the first to suggest this metric for video. We also visualize this quantity spatially.

CRPS measures the agreement of a cumulative distribution function (CDF) F with an observation x, $\text{CRPS}(F, x) = \int_{\mathbb{R}} (F(z) - \mathbb{I}\{x \leq z\})^2 \, dz$, where $\mathbb{I}$ is the indicator function. In the context of our evaluation task, F is the CDF of a single pixel within a single future frame assigned by the generative model. CRPS measures how well this distribution matches the empirical CDF of the data, approximated by a single observed sample. The involved integral can be well approximated by a finite sum since we are dealing with standard 8-bit frames. We approximate F by an empirical CDF $\hat{F}(z) = \frac{1}{S} \sum_{s=1}^{S} \mathbb{I}\{X_s \leq z\}$; we stress that this does not require a likelihood model but only a set of S stochastically generated samples $X_s \sim F$ from the model, enabling comparisons across methods.

	FVD↓				LPIPS↓				CRPS↓			
	KTH	BAIR	Sim	City	KTH	BAIR	Sim	City	KTH	BAIR	Sim	City
RVD (ours)	**1351**	**1272**	**20**	**997**	**0.06**	**0.06**	0.01	**0.11**	6.51	12.86	**0.58**	**9.84**
IVRNN	1375	1337	24	1234	0.08	0.07	**0.008**	0.18	**6.17**	**11.74**	0.65	11.00
SVG-LP	1783	1631	21	1465	0.12	0.08	0.01	0.20	18.24	13.96	0.75	19.34
RetroGAN	2503	2038	28	1769	0.28	0.07	0.02	0.20	27.49	19.42	1.60	20.13
DVD-GAN	2592	3097	147	2012	0.18	0.10	0.06	0.21	12.05	27.2	1.42	21.61
FutureGAN	4111	3297	319	5692	0.33	0.12	0.16	0.29	37.13	27.97	6.64	29.31

Table 1: Test set perceptual (FVD, LPIPS) and forecasting (CRPS) metrics, lower is better (see Section 3 for details).

3.5 Qualitative and Quantitative Analysis

Using the perceptual and probabilistic metrics mentioned above, we compare test set predictions of our video diffusion architecture against a wide range of baselines, which model underlying data density both explicitly and implicitly.

Table 1 lists all the metric scores for our model and the baselines. Our model performs best in all cases in terms of FVD. For LPIPS, our model also performs best in 3 out of 4 datasets. The perceptual performance is also verified visually in Figure 2, where RVD shows higher clarity on the generated frames and shows less blurriness in regions that are less predictable due to the fast motion.

For a more quantitative assessment, we reported CRPS scores in Table 1. These support that the proposed model can predict the future with higher accuracy on high-resolution videos than other models. Besides, we compute CRPS on the basis of individual frames. Figure 3 shows 1/CRPS (higher is better) as a function of the frame index, revealing a monotonically decreasing trend along the time axis. This follows our intuition that long-term predictions become worse over time for all models. Our method performs best in 3 out of 4 cases. We can resolve this score also spatially in the images, as we do in Figure 4. Areas of distributional disagreement within a frame are shown in blue (right). See supplemental materials for the generated videos on other datasets.

3.6 Ablation Studies

We consider two ablations of our model. The first one studies the impact of applying the diffusion generative model for modeling residuals as opposed to directly predicting the next frames. The second ablation studies the impact of the number of frames that the model sees during training.

Modeling Residuals vs. Frames Our proposed method uses a denoising diffusion generative model to generate *residuals* to a deterministic next-state prediction (see Figure 1b). A natural question arises whether this architecture is necessary or whether it could be simplified by directly generating the next frame $\mathbf{x}_0^t$ instead of the residual $\mathbf{y}_0^t$. To address this, we make the following adjustment. Since $\mathbf{y}_0^t$ and $\mathbf{x}_0^t$ have equal dimensions, the ablation can be realized by setting $\mu^t = 0$ and $\sigma = 1$. To distinguish from our proposed "Residual Video Diffusion" (RVD), we call this ablation "Video Diffusion" (VD). Note that this ablation can be considered a customized version of TimeGrad (Rasul et al., 2021) applied to video.

Table 2 shows the results. Across all perceptual metrics, the residual model performs better on all data sets. In terms of CRPS, VD performs slightly better on the simpler KTH and Bair datasets, but worse on the more complex Simulation and Cityscape data. We, therefore, confirm our earlier claims that modeling residuals over frames is crucial for obtaining better performance, especially on more complex high-resolution video.

Influence of Training Sequence Length We train both our diffusion model and IVRNN on video sequences of varying lengths. As Table 2 reveals, we find that the diffusion model maintains a robust performance, showing only a small degradation on significantly shorter sequences. In contrast, IVRNN seems to be more sensitive to the sequence length. We note that in most experiments, we outperform IVRNN even though we trained our model on shorter sequences.

	FVD↓				LPIPS↓				CRPS↓			
	KTH	BAIR	Sim	City	KTH	BAIR	Sim	City	KTH	BAIR	Sim	City
VD(2+6)	1523	1374	37	1321	**0.066**	0.066	0.014	0.127	**6.10**	12.75	0.68	13.42
RVD(2+6)	**1351**	**1272**	**20**	**997**	**0.066**	**0.060**	0.011	**0.113**	6.51	12.86	**0.58**	**9.84**
RVD(2+3)	1663	1381	33	1074	0.072	0.072	0.018	**0.112**	6.67	13.93	0.63	10.59
IVRNN(4+8)	1375	1337	24	1234	0.082	0.075	**0.008**	0.178	**6.17**	**11.74**	0.65	11.00
IVRNN(4+4)	2754	1508	150	3145	0.097	0.074	0.040	0.278	7.25	13.64	1.36	18.24

Table 2: Ablation studies on (1) modeling residuals (RVD, proposed) versus future frames (VD) and (2) training with different sequence lengths, where $(p + q)$ denotes p context frames and q future frames for prediction.

4 Related Work

Our paper combines ideas from video generation, diffusion probabilistic models, and neural video compression. As follows, we discuss related work along these lines.

4.1 Video Generation Models

Video prediction can sometimes be treated as a supervised problem, where the focus is often on error metrics such as PSNR and SSIM (Lotter et al., 2017; Byeon et al., 2018; Finn et al., 2016). In contrast, in *stochastic generation*, the focus is tyically on distribution matching as measured by held-out likelihoods or perceptual no-reference metrics.

A large body of video generation research relies on deep sequential latent variable models (Babaeizadeh et al., 2018; Li & Mandt, 2018; Kumar et al., 2019; Unterthiner et al., 2018; Clark et al., 2019). Among the earliest works, Bayer & Osendorfer (2014) and Chung et al. (2015) extended recurrent neural networks to stochastic models by incorporating latent variables. Later work (Denton & Fergus, 2018) extended the sequential VAE by incorporating more expressive priors conditioned on a longer frame context. IVRNN (Castrejon et al., 2019) further enhanced the generation quality by working with a hierarchy of latent variables, which to our knowledge is currently the best end-to-end trained sequential VAE model that can be further refined by greedy fine-tuning (Wu et al., 2021). Normalizing flow-based models for video have been proposed while typically suffering from high demands on memory and compute (Kumar et al., 2019). Some works (Zhao et al., 2018; Franceschi et al., 2020; Marino, 2021; Marino et al., 2021) explored the use of residuals for improving video generation in sequential VAE settings but did not achieve state of the art results.

Another line of sequential models rely on GANs (Vondrick et al., 2016b; Aigner & Körner, 2018; Kwon & Park, 2019; Wu et al., 2021). While these models do not show blurry artifacts, they tend to suffer from long-term consistency, as our experiments confirm.

4.2 Diffusion Probabilistic Models

DDPMs have recently shown impressive performance on high-fidelity image generation. Sohl-Dickstein et al. (2015) first introduced and motivated this model class by drawing on a non-equilibrium thermodynamic perspective. Song & Ermon (2019) proposed a single-network model for score estimation, using annealed Langevin dynamics for sampling. Furthermore, Song et al. (2021c) used stochastic differential equations (related to diffusion processes) to train a network to transform random noise into the data distribution.

DDPM by Ho et al. (2020) is the first instance of a diffusion model scalable to high-resolution images. This work also showed the equivalence of DDPM and denoising score-matching methods described above. Subsequent work includes extensions of these models to image super-resolution (Saharia et al., 2021) or hybridizing these models with VAEs (Pandey et al., 2022). Apart from the traditional computer vision tasks, diffusion models were proven to be effective in audio synthesis (Chen et al., 2021; Kong et al., 2021), while Luo & Hu (2021) hybridized normalizing flows and diffusion model to generative 3D point cloud samples.

To the best of our knowledge, TimeGrad (Rasul et al., 2021) is the first sequential diffusion model for time-series forecasting. Their architecture was not designed for video but for traditional lower-dimensional correlated time-series datasets. A concurrent preprint also studies a video diffusion model (Ho et al., 2022). This work is based on an alternative architecture and focuses primarily on perceptual metrics.

4.3 Neural Video Compression Models

Video compression models typically employ frame prediction methods optimized for minimizing code length and distortion. In recent years, sequential generative models were proven to be effective on video compression tasks (Han et al., 2019; Yang et al., 2020; Agustsson et al., 2020; Yang et al., 2021a; Lu et al., 2019; Yang et al., 2022). Some of these models show impressive rate-distortion performance with hierarchical structures that separately encode the prediction and error residual. While compression models have different goals than generative models, both benefit from predictive sequential priors (Yang et al., 2021b). Note, however, that these models are ill-suited for generation.

5 Discussion

We proposed "Residual Video Diffusion": a new model for stochastic video generation based on denoising diffusion probabilistic models. Our approach uses a denoising process, conditioned on the context vector of a convolutional RNN, to generate a *residual* to a deterministic next-frame prediction. We showed that such residual prediction yields better results than directly predicting the next frame.

To benchmark our approach, we studied a variety of datasets of different degrees of complexity and pixel resolution, including Cityscape and a physics simulation dataset of turbulent flow. We compared our approach against two state-of-the-art VAE and three GAN baselines in terms of *both* perceptual and probabilistic forecasting metrics. Our method leads to a new state of the art in perceptual quality while being competitive with or better than state-of-the-art hierarchical VAE and GAN baselines in terms of probabilistic forecasting. Our results provide several promising directions and could improve world model-based RL approaches as well as neural video codecs.

Limitations The autoregressive setup of the proposed model allows conditional generation with at least one context frame pre-selected from the test dataset. In order to achieve unconditional generation of a complete video sequence, we need an auxiliary image generative model to sample the initial context frames. It's also worth mentioning that we only conduct the experiments on *single-domain* datasets with monotonic contents (e.g. Cityscape dataset only contains traffic video recorded by a camera installed in the front of the car), as training a large model for *multi-domain* datasets like Kinetics (Smaira et al., 2020) is demanding for our limited computing resources. Finally, diffusion probabilistic models tend to be slow in training which could be accelerated by incorporating DDIM sampling (Song et al., 2021a) or model distillation (Salimans & Ho, 2022).

Potential Negative Impacts Just as other generative models, video generation models pose the danger of being misused for generating deepfakes, running the risk of being used for spreading misinformation. Note, however, that a probabilistic video prediction model could also be used for anomaly detection (scoring anomalies by likelihood) and hence may help to detect such forgery.

References

Eirikur Agustsson, David Minnen, Nick Johnston, Johannes Balle, Sung Jin Hwang, and George Toderici. Scale-space flow for end-to-end optimized video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8503–8512, 2020.

Sandra Aigner and Marco Körner. Futuregan: Anticipating the future frames of video sequences using spatio-temporal 3d convolutions in progressively growing gans. *arXiv preprint arXiv:1810.01325*, 2018.

Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. In *International Conference on Learning Representations*, 2018.

Nicolas Ballas, Li Yao, Chris Pal, and Aaron C Courville. Delving deeper into convolutional networks for learning video representations. In *ICLR (Poster)*, 2016.

Justin Bayer and Christian Osendorfer. Learning stochastic recurrent networks. *arXiv preprint arXiv:1411.7610*, 2014.

Apratim Bhattacharyya, Mario Fritz, and Bernt Schiele. Long-term on-board prediction of people in traffic scenes under uncertainty. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4194–4202, 2018.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

Wonmin Byeon, Qin Wang, Rupesh Kumar Srivastava, and Petros Koumoutsakos. Contextvp: Fully context-aware video prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 753–769, 2018.

Lluis Castrejon, Nicolas Ballas, and Aaron Courville. Improved conditional vrnns for video prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7608–7617, 2019.

Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. In *International Conference on Learning Representations*, 2021.

Dragos Bogdan Chirila. *Towards lattice Boltzmann models for climate sciences: The GeLB programming language with applications*. PhD thesis, Universität Bremen, 2018.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. *Advances in neural information processing systems*, 28, 2015.

Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial video generation on complex datasets. *arXiv preprint arXiv:1907.06571*, 2019.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *International conference on machine learning*, pp. 1174–1183. PMLR, 2018.

Frederik Ebert, Chelsea Finn, Alex X Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. In *CoRL*, pp. 344–356, 2017.

Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. *Advances in neural information processing systems*, 29, 2016.

Jean-Yves Franceschi, Edouard Delasalles, Mickael Chen, Sylvain Lamprier, and Patrick Gallinari. Stochastic latent residual video prediction. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3233–3246. PMLR, 13–18 Jul 2020.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Jun Han, Salvator Lombardo, Christopher Schroers, and Stephan Mandt. Deep generative video compression. In *International Conference on Neural Information Processing Systems*, pp. 9287–9298, 2019.

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*, pp. 770–778, 2016.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

John F Kolen and Stefan C Kremer. *A field guide to dynamical recurrent networks*. John Wiley & Sons, 2001.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021.

Manoj Kumar, Mohammad Babaeizadeh, Dumitru Erhan, Chelsea Finn, Sergey Levine, Laurent Dinh, and Durk Kingma. Videoflow: A flow-based generative model for video. *arXiv preprint arXiv:1903.01434*, 2 (5), 2019.

Yong-Hoon Kwon and Min-Gyu Park. Predicting future frames using retrospective cycle gan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1811–1820, 2019.

Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

Yingzhen Li and Stephan Mandt. Disentangled sequential autoencoder. In *International Conference on Machine Learning*, pp. 5670–5679. PMLR, 2018.

Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4463–4471, 2017.

William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. *International Conference on Learning Representations*, 2017.

Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. Dvc: An end-to-end deep video compression framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11006–11015, 2019.

Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2837–2845, 2021.

Joseph Marino. Predictive coding, variational autoencoders, and biological connections. *Neural Computation*, 34(1):1–44, 2021.

Joseph Marino, Lei Chen, Jiawei He, and Stephan Mandt. Improving sequential latent variable models with autoregressive flows. *Machine Learning*, pp. 1–24, 2021.

James E Matheson and Robert L Winkler. Scoring rules for continuous probability distributions. *Management science*, 22(10):1087–1096, 1976.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

Sergiu Oprea, Pablo Martinez-Gonzalez, Alberto Garcia-Garcia, John Alejandro Castro-Vargas, Sergio Orts-Escolano, Jose Garcia-Rodriguez, and Antonis Argyros. A review on deep learning techniques for video prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Kushagra Pandey, Avideep Mukherjee, Piyush Rai, and Abhishek Kumar. Diffusevae: Efficient, controllable and high-fidelity generation from low-dimensional latents. *arXiv preprint arXiv:2201.00308*, 2022.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

Rajesh PN Rao and Dana H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87, 1999.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*, pp. 8857–8868. PMLR, 2021.

Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, et al. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677, 2021.

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

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *arXiv preprint arXiv:2104.07636*, 2021.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Robert E Schapire. A brief introduction to boosting. In *Ijcai*, volume 99, pp. 1401–1406. Citeseer, 1999.

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, pp. 32–36. IEEE, 2004.

Lucas Smaira, João Carreira, Eric Noland, Ellen Clancy, Amy Wu, and Andrew Zisserman. A short note on the kinetics-700-2020 human action dataset. *arXiv preprint arXiv:2010.10864*, 2020.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a.

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

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems*, 34, 2021b.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021c.

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

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Fvd: A new metric for video generation. *ICLR 2019 Workshop for Deep Generative Models for Highly Structured Data*, 2019.

Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 98–106, 2016a.

Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. *Advances in neural information processing systems*, 29, 2016b.

Bohan Wu, Suraj Nair, Roberto Martin-Martin, Li Fei-Fei, and Chelsea Finn. Greedy hierarchical variational autoencoders for large-scale video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2318–2328, 2021.

Ren Yang, Fabian Mentzer, Luc Van Gool, and Radu Timofte. Learning for video compression with recurrent auto-encoder and recurrent probability model. *IEEE Journal of Selected Topics in Signal Processing*, 15 (2):388–401, 2020.

Ruihan Yang, Yibo Yang, Joseph Marino, and Stephan Mandt. Hierarchical autoregressive modeling for neural video compression. In *International Conference on Learning Representations*, 2021a.

Ruihan Yang, Yibo Yang, Joseph Marino, and Stephan Mandt. Insights from generative modeling for neural video compression. *arXiv preprint arXiv:2107.13136*, 2021b.

Yibo Yang, Stephan Mandt, and Lucas Theis. An introduction to neural data compression. *arXiv preprint arXiv:2202.06533*, 2022.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris Metaxas. Learning to forecast and refine residual motion for image-to-video generation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 387–403, 2018.

A Architecture

Our architecture extends the previously proposed DDPM Ho et al. (2020) architecture to a temporally conditioned version. Figures 5 and 6 show the design of the proposed *denoising* and *transform* modules. Before describing these figures, we list important definitions and parameter choices below (see also Table 3 for more details). Codes are available at: `https://anonymous.4open.science/r/RVDC1FC`

- *Channel Dim* refers to the channel dimension of all the components in the first downsampling layer of the U-Net Ronneberger et al. (2015) style structure used in our approach.
- *Denoising/Transform Multipliers* are the channel dimension multipliers for subsequent downsampling layers (including the first layer) in the denoising/transform modules. The upsampling layer multipliers follow the reverse sequence.
- Each `ResBlock` He et al. (2016) leverages a standard implementation of the ResNet block with 3×3 kernel, LeakyReLU activation and Group Normalization.
- All `ConvGRU` Ballas et al. (2016) use a 3×3 kernel to deal with the temporal information.
- Each `LinearAttention` module involves 4 attention heads, each involving 16 dimensions.
- To condition our architecture on the denoising step n, we use positional encodings to encode n and add this encoding to the `ResBlocks` (as in Figure 5).
- The `Upsample / Downsample` components in Figures 5 & 6 involve Deconvolutional / Convolutional networks that spatially scale the feature map with scaling factors 2 and 1/2, respectively.

Figure 5a shows the overall U-Net style architecture that has been adopted for denoising module. It predicts the noise information from the noisy residual (flowing through the blue arrows) at an arbitrary n^{th} step (note that we perform a total of $N = 1600$ steps in our setup), conditioned on all the past context frames (flowing through the green arrows). The figure shows a low-resolution setting, where the number of downsampling and upsampling layers has been set to $L^{\text{denoise}} = 4$. Skip concatenations (shown as red arrows) are performed between the `Linear Attention` of downsampling layer and first `ResBlock` of the corresponding upsampling layer as detailed in Figure 5b. Context conditioning is provided by a `ConvGRU` block within the downsampling layers that generates a context which is concatenated along the residual processing stream in the second `ResBlock` module. Additionally, each `ResBlock` module in either layers receives a positional encoding indicating the denoising step.

Figure 6a shows the U-Net style architecture that has been adopted for the transform module with the number of downsampling and upsampling layers $L^{\text{transform}} = 4$ in the low-resolution setting. Skip concatenations (shown as red arrows) are performed between the `ConvGRU` of downsampling layer and first `ResBlock` of the corresponding upsampling layer as detailed in Figure 6b.

Video Resolutions	Channel Dim	Denoising Multipliers	Transform Multipliers
64×64	48	1,2,4,8 ($L^{\text{denoise}} = 4$)	1,2,2,4 ($L^{\text{transform}} = 4$)
128×128	64	1,1,2,2,4,4 ($L^{\text{denoise}} = 6$)	1,2,3,4 ($L^{\text{transform}} = 4$)

Table 3: Configuration Table, see Section A for definitions.

B Deriving the Optimization Objective

The following derivation closely follows Ho et al. Ho et al. (2020) to derive the variational lower bound objective for our sequential generative model.

As discussed in the main paper, let $\mathbf{x}^{0:T}$ denote observed frames, and $\mathbf{y}_{0:N}^{1:T}$ the variables associated with the diffusion process. Among them, only $\mathbf{y}_{1:N}^{1:T}$ are the latent variables, while $\mathbf{y}_0^{1:T}$ are the the observed scaled residuals, given by $\mathbf{y}_0^t = \frac{\mathbf{x}^t - \mu_\phi(\mathbf{x}^{<t})}{\sigma}$ for $t = 1, ..., T$. The variational bound is as follows:

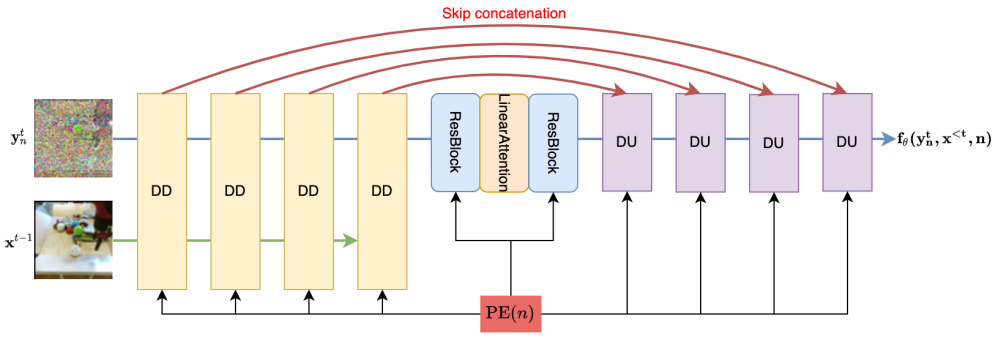
$$
\begin{aligned}
L &= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}} \sum_{t=1}^{T} \mathbb{E}_{\mathbf{y}_{1..N}^t \sim q(\cdot|\mathbf{y}_0^t)}[-\log \frac{p_\theta(\mathbf{y}_{0..N}^t|\mathbf{x}^{<t})}{q(\mathbf{y}_{1..N}^t|\mathbf{y}_0^t, \mathbf{x}^{<t})}] \\
&= \mathbb{E}_{\mathcal{D}} \sum_{t=1}^{T} \mathbb{E}_q[-\log \frac{p(\mathbf{y}_N^t)}{q(\mathbf{y}_N^t|\mathbf{x}^{\leq t})} - \sum_{n>1}^{N} \log \frac{p_\theta(\mathbf{y}_{n-1}^t|\mathbf{y}_n^t, \mathbf{x}^{<t})}{q(\mathbf{y}_{n-1}^t|\mathbf{y}_n^t, \mathbf{x}^{\leq t})} \\
&- \log p_\theta(\mathbf{y}_0^t|\mathbf{y}_1^t, \mathbf{x}^{<t})].
\end{aligned} \tag{10}
$$

The first term in Eq. 10, $-\log \frac{p(\mathbf{y}_N^t)}{q(\mathbf{y}_N^t|\mathbf{x}^{\leq t})}$, tries to match the $q(\mathbf{y}_N^t|\mathbf{x}^{\leq t})$ to the prior $p(\mathbf{y}_N^t) = \mathcal{N}(0, \mathbf{I})$. The prior has a fixed variance and is centered around zero, while $q(\mathbf{y}_N^t|\mathbf{x}^{\leq t}) = \mathcal{N}(\sqrt{\bar{\alpha}_n}\mathbf{y}_0^t, (1 - \bar{\alpha}_n)\mathbf{I})$. Because the variance of q is also fixed, the only effect of the the first term is to pull $\sqrt{\bar{\alpha}_n}\mathbf{y}_0^t$ towards zero. However, in practice, $\bar{\alpha}_n = \prod_{i=0}^{N}(1 - \beta_i) \approx 0$, and hence the effect of this term is very small. For simplicity, therefore, we drop it.

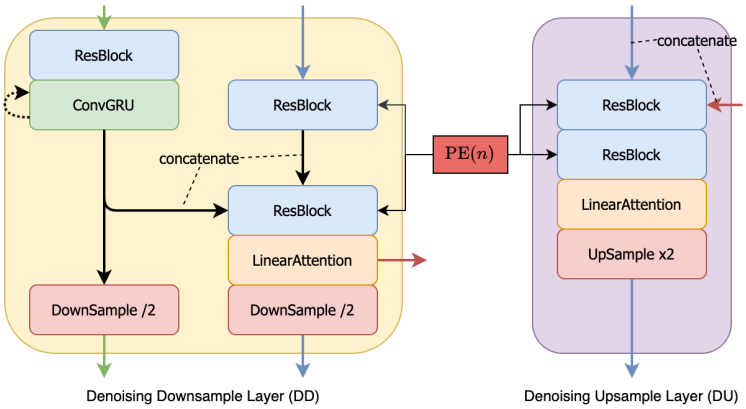
To understand the third term in Eq. 10, we simplify

$$
\begin{aligned}
&- \log p_\theta(\mathbf{y}_0^t|\mathbf{y}_1^t, \mathbf{x}^{<t}) \\
&= \frac{1}{2\sigma\gamma_1} ||\mathbf{x}^t - \mu(\mathbf{x}^{<t}) - \sigma M_\theta(\mathbf{y}_1^t, \mathbf{x}^{<t}, 1)||^2 \; + \; const.,
\end{aligned} \tag{11}
$$

Eq. 11 suggests that the third term matches the diffusion model's output to the frame residual, which is also a special case of L^{mid} we elaborate below.



(a) Overview figure of the autoregressive denoising module using a U-net inspired architecture with skip-connections. We focus on the example of $L^{\text{denoise}} = 4$ downsampling and upsampling layers. Each of these layers, DD and DU, are explained in Figure b below. All DD and DU layers are furthermore conditioned on a positional encoding (PE) of the denoising step n.



(b) Downsampling/Upsampling layer design for the autoregressive denoising module. Each arrow corresponds to the arrows with the same color in Figure (a). As in Figure (a), each residual block is conditioned on a positional encoding (PE) of the denoising step n.

Figure 5: Autoregressive denoising module.

Deriving a parameterization for the second term of Eq. 10, $L^{\text{mid}} := -\sum_{n>1}^{N} \log \frac{p_\theta(\mathbf{y}_{n-1}^t|\mathbf{y}_n^t, \mathbf{x}^{<t})}{q(\mathbf{y}_{n-1}^t|\mathbf{y}_n^t, \mathbf{x}^{\leq t})}$, we recognize that it is a KL divergence between two Gaussians with fixed variances:

$$q(\mathbf{y}_{n-1}^t|\mathbf{y}_n^t, \mathbf{x}^{\leq t}) = q(\mathbf{y}_{n-1}^t|\mathbf{y}_n^t, \mathbf{y}_0^t) = \mathcal{N}(\mathbf{y}_{n-1}^t; M(\mathbf{y}_n^t, \mathbf{y}_0^t), \gamma_n \mathbf{I}), \tag{12}$$

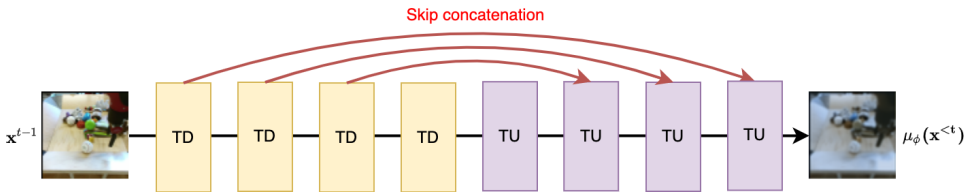
$$p_\theta(\mathbf{y}_{n-1}^t|\mathbf{y}_n^t, \mathbf{x}^{<t}) = \mathcal{N}(\mathbf{y}_{n-1}^t; M_\theta(\mathbf{y}_n^t, \mathbf{x}^{<t}, n), \gamma_n \mathbf{I}). \tag{13}$$

We define $M(\mathbf{y}_n^t, \mathbf{y}_n^0) = \frac{\sqrt{\bar{\alpha}_{n-1}}\beta_n}{1-\bar{\alpha}_n}\mathbf{y}_0^t + \frac{\sqrt{1-\beta_n}(1-\bar{\alpha}_{n-1})}{1-\bar{\alpha}_n}\mathbf{y}_n^t$. The KL divergence can therefore be simplified as the L2 distance between the means of these two Gaussians:

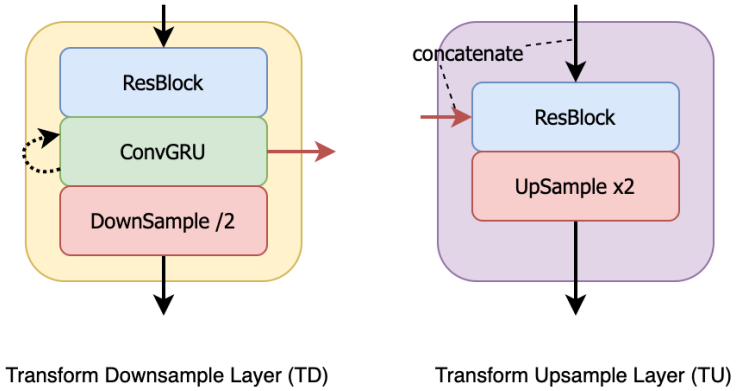
$$L_n^{\text{mid}} = \frac{1}{2\gamma_n}||M(\mathbf{y}_n^t, \mathbf{y}_n^0) - M_\theta(\mathbf{y}_n^t, \mathbf{x}^{<t}, n)||^2 \tag{14}$$

As $\mathbf{y}_n^t$ always has a closed form when $\mathbf{y}_0^t$ is given: $\mathbf{y}_n^t = \sqrt{\bar{\alpha}_n}\mathbf{y}_0^t + \sqrt{1-\bar{\alpha}_n}\epsilon$ (See Section 3.1), we parameterize Eq 14 to the following form:

$$L_n^{\text{mid}} = \frac{1}{2\gamma_n}||\frac{1}{\sqrt{1-\beta_n}}(\mathbf{y}_n^t - \frac{\beta_n}{\sqrt{1-\bar{\alpha}_n}}\epsilon) - M_\theta(\mathbf{y}_n^t, \mathbf{x}^{<t}, n)||^2. \tag{15}$$



(a) Overview figure of the autoregressive transform module, predicting the mean next frame μ_t. We use a U-net inspired architecture with an example size of $L^{\text{transform}} = 4$ upsampling and downsampling steps. Each arrow corresponds to the arrows with the same color in Figure (b). TD and TU layers are elaborated in Figure (b).



(b) Downsample/upsample layer design for autoregressive transform module. Each arrow corresponds to the arrows with the same color in Figure (a).

Figure 6: Autoregressive transform module for predicting the next frame μ_t.

It becomes apparent that M_θ is trying to predict $\frac{1}{\sqrt{1-\beta_n}}(\mathbf{y}_n^t - \frac{\beta_n}{\sqrt{1-\bar{\alpha}_n}}\epsilon)$. Equivalently, we can therefore predict ϵ. To this end, we replace the parameterization M_θ by the following parameterization involving f_θ:

$$\begin{aligned} & L_n^{\text{mid}} \\ & = \frac{1}{2\gamma_n}||\frac{1}{\sqrt{1-\beta_n}}(\mathbf{y}_n^t - \frac{\beta_n}{\sqrt{1-\bar{\alpha}_n}}\epsilon) \\ & - \frac{1}{\sqrt{1-\beta_n}}(\mathbf{y}_n^t - \frac{\beta_n}{\sqrt{1-\bar{\alpha}_n}} f_\theta(\mathbf{y}_n^t, \mathbf{x}^{<t}, n))||^2 \end{aligned} \tag{16}$$

The resulting stochastic objective can be simplified by dropping all untrainable parameters, as suggested in Ho et al. (2020). As a result, one obtains the simplified objective

$$L_n^{\text{mid}} \equiv ||\epsilon - f_\theta(\mathbf{y}_n^t(\phi), n, \mathbf{x}^{<t})||^2, \tag{17}$$

which is the denoising score matching objective revealed in Eq. 9 of our paper.

C Additional Generated Samples

In this section, we present some qualitative results from our study for some additional datasets apart from CityScape. More specifically, we present some examples from our Simulation data, BAIR Robot Pushing data and KTH Actions data. In the following figures, the top row consists of ground truth frames, both contextual and predictive, while the subsequent rows are the generated frames from our method and some other VAE and GAN baselines. It is quite evident that our our method and IVRNN are the strongest contenders. Perceptually, the FVD metric indicates that we outperform on every dataset, whereas statistically, the CRPS metric shows our top performance on high-resolution datasets and a competitive performance against IVRNN on the other two datasets.

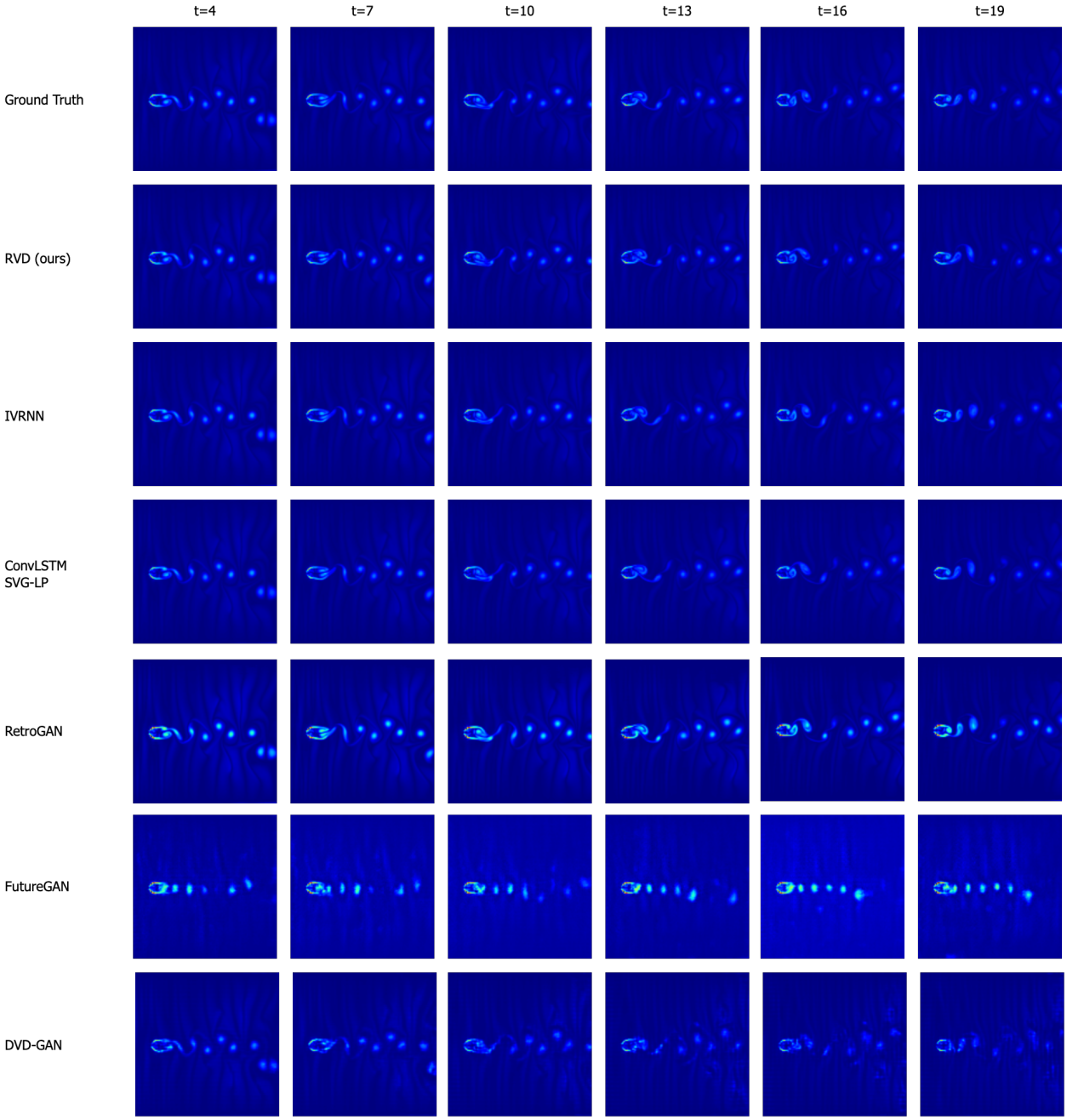


Figure 7: Prediction Quality for Simulation data. Top row indicates the ground truth, wherein we feed 4 frames as context (from $t = 0$ to $t = 3$) and predict the next 16 frames (from $t = 4$ to $t = 19$). This is a high resolution dataset (128×128) which we generated using a Lattice Boltzmann Solver. It simulates the von Kármán vortex street using Navier-Stokes equations. A fluid (with pre-specified viscosity) flows through a 2D plane interacting with a circular obstacle placed at the center left. This leads to the formation of a repeating pattern of swirling vortices, caused by a process known as vortex shedding, which is responsible for the unsteady separation of flow of a fluid around blunt bodies. Colors indicate the vorticity of the solution to the simulation.

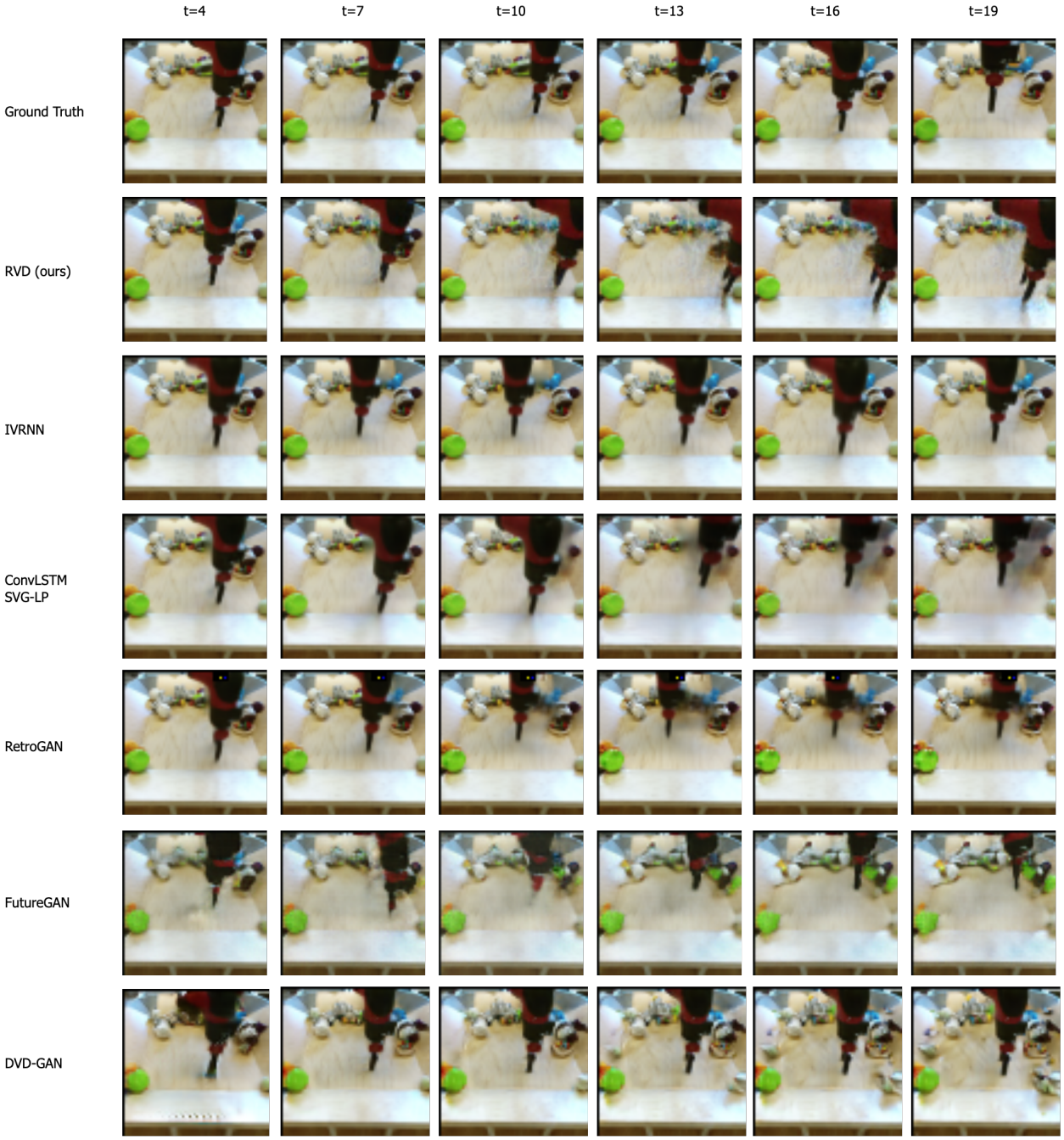


Figure 8: Prediction Quality for BAIR Robot Pushing. As seen before, the top row indicates the ground truth, wherein we feed 4 frames as context (from $t = 0$ to $t = 3$) and predict the next 16 frames (from $t = 4$ to $t = 19$). This is a low resolution dataset (64×64) that captures the motion of a robotic hand as it manipulates multiple objects. Temporal consistency and occlusion handling are some of the big challenges for this dataset.

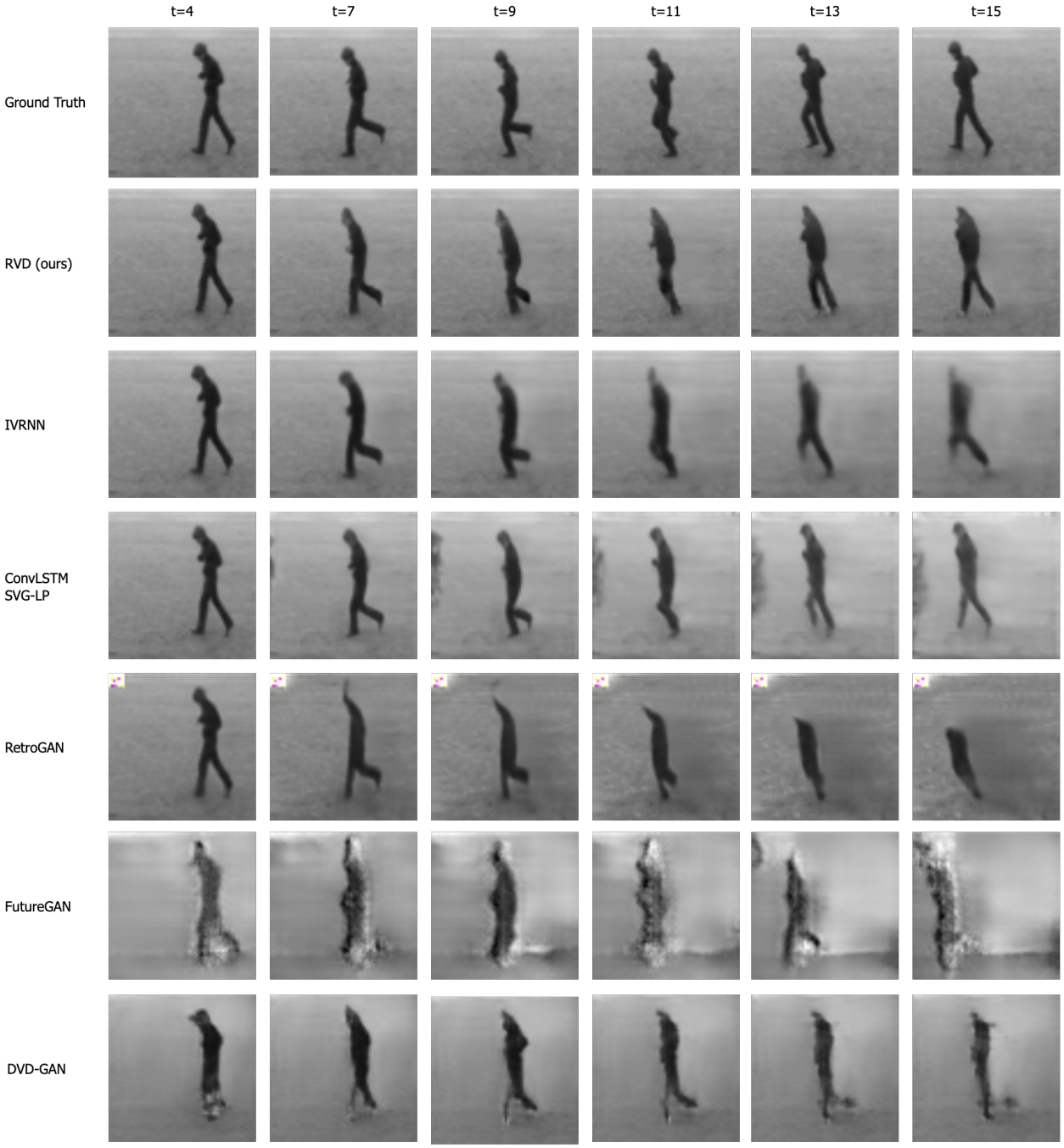


Figure 9: Prediction Quality for KTH Actions. As seen before, the top row indicates the ground truth, wherein we feed 4 frames as context (from $t = 0$ to $t = 3$), but unlike other datasets, we only predict the next 12 frames (from $t = 4$ to $t = 15$).