

Diffusion Probabilistic Modeling for Video Generation

Ruihan Yang ^{*}, Prakhar Srivastava and Stephan Mandt 

Department of Computer Science, University of California, Irvine, CA 92697, USA; prakhs2@uci.edu (P.S.); mandt@uci.edu (S.M.)

* Correspondence: ruihan.yang@uci.edu

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 six baselines on four datasets involving natural and simulation-based videos. We find significant improvements in terms of perceptual quality and probabilistic frame forecasting ability for all datasets.

Keywords: diffusion models; deep generative models; video generation; autoregressive models

1. Introduction

The ability to anticipate future frames of a video is intuitive for humans but challenging for a computer [1]. Applications of such *video prediction* tasks include anticipating events [2], model-based reinforcement learning [3], video interpolation [4], predicting pedestrians in traffic [5], precipitation nowcasting [6], neural video compression [7–12], 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 [13–15] tend to be stronger in goals (i) and (ii), while sequential extensions of generative adversarial networks [16–18] tend to perform better in goal (iii). A probabilistic method that succeeds in all 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 [19–23]. In this paper, we extend diffusion probabilistic models for stochastic video generation. Our ideas are inspired by the principles of predictive coding [24,25] and neural compression algorithms [26] and draw on the intuition that residual errors are easier to model than dense observations [27]. 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 denoising diffusion process, also conditioned on a temporal context (see Figure 1a,b). 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 comparing it against multiple baselines, we prove superior results



Citation: Yang, R.; Srivastava, P.; Mandt, S. Diffusion Probabilistic Modeling for Video Generation. *Entropy* **2023**, *25*, 1469. <https://doi.org/10.3390/e25101469>

Academic Editor: Jakub Tomczak

Received: 30 August 2023

Revised: 16 October 2023

Accepted: 18 October 2023

Published: 20 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

on both probabilistic (CRPS) and perceptual (LPIPS, FID) metrics. In more detail, our achievements are as follows:

1. We show how to use diffusion probabilistic models to generate videos autoregressively. 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 our model based on three metrics, FVD/LPIPS/CRPS, that can cover both forecasting ability and perceptual quality. The result demonstrates that our model performs better than modern GAN and VAE baselines such as IVRNN, SVG-LP, SLAMP, RetroGAN, DVD-GAN and FutureGAN [14–17,28,29].
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 of our experimental findings along with ablation studies. We then discuss connections to the literature and summarize our contributions.

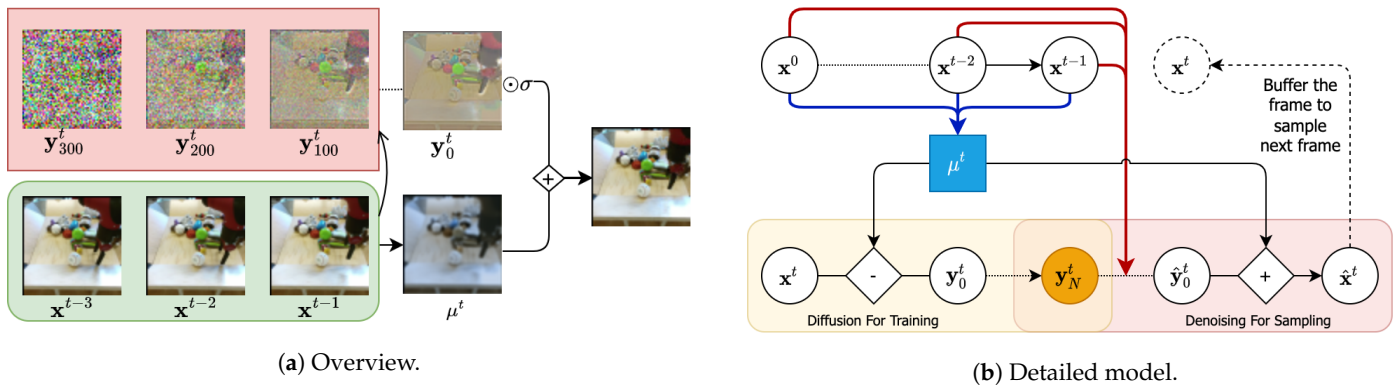


Figure 1. Overview: Our approach predicts the next frame μ_t of a video autoregressively along with an additive correction y_0^t generated by a denoising process. Detailed model: Two convolutional RNNs (blue and red arrows) operate on a frame sequence $x^{0:t-1}$ to predict the most likely next frame μ^t (blue box) and a context vector for a denoising diffusion model. The diffusion model is trained to model the scaled residual $y_0^t = (x^t - \mu^t) / \sigma$ 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 $x^t = \mu^t + \sigma y_0^t$.

2. 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.

2.1. Video Generation Models

Since the advent of modern deep learning, video generation and prediction has been a topic of ongoing interest; see [1] and this paper’s introduction. Videos can be generated based on side information of various types, such as images [30,31], text [32–37] or other videos [38]. Alternatively, videos can also be generated unconditionally, e.g., from white noise [29,39,40]. This survey focuses on conditional video generation, where one conditions the generation of future frames in the context of past frames.

Conditional video prediction is sometimes treated as a supervised problem, where the focus is often on error metrics such as PSNR and SSIM [41–43]. Early works leverage deterministic methods to predict the most likely next frames [42–47]. Generally speaking, the downside of supervised approaches is that real-world videos display multi-modal behavior, i.e., the future is not uniquely predictable from the past (e.g., a traffic light may

switch from yellow to red or green, a new object may or may not enter the scene, etc.). Treating video prediction as a supervised problem can, therefore, lead to mode averaging, perceived as blurriness.

Most recent methods focus on *stochastic generation* using deep generative models. In contrast to learning the average video dynamics, these methods try to match the conditional distribution of future frames given the past, typically by minimizing a divergence measure (as in GANs) or by optimizing a variational bound to a model's log-likelihood (as in VAEs). Consequently, the evaluation has shifted to held-out likelihoods or perceptual metrics, such as FID or LPIPS. A large body of video generation research relies on variational deep sequential latent variable models [13,28,29,48–56], to name a few. These works often draw on earlier works for modeling stochastic dynamics, e.g., Bayer and Osendorfer [57], Chung et al. [58], who included latent variables in recurrent neural networks. Later work [14] extended the sequential VAE by incorporating more expressive priors conditioned on a longer frame context. IVRNN [15] 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 [59]. Normalizing flow-based models for video have been proposed while typically suffering from high demands on memory and compute [49]. Some works [25,27,60,61] explored the use of residuals for improving video generation in sequential VAEs but did not achieve state-of-the-art results.

Overall, the downside of VAE-based models is that they are trained to reconstruct the data. Blau and Michaeli [62] theoretically showed that generative models are typically in conflict between data reconstruction tasks and achieving a high degree of *realism* (defined as matching the target distribution unconditionally, without artifacts). This suggests that VAEs may not be the final answer when it comes to video prediction.

Another line of sequential models relies on GANs [16,17,40,63–65], which—at inference time—can be either deterministic or stochastic. In contrast to VAE-based models, these models tend to show fewer blurry artifacts. A downside of GANs is that their loss function is mode-seeking (as opposed to mass covering), meaning that the data distribution is not covered at sufficient breadth, reflected in their typically worse performance in probabilistic distribution matching and forecasting metrics.

2.2. Diffusion Probabilistic Models

DDPMs have recently shown impressive performance in high-fidelity image generation. Sohl-Dickstein et al. [19] first introduced and motivated this model class by drawing on a non-equilibrium thermodynamic perspective. Song and Ermon [20] proposed a single-network model for score estimation, using annealed Langevin dynamics for sampling. Furthermore, Song et al. [22] 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. [21] 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 [66] or hybridizing these models with VAEs [67]. Apart from the traditional computer vision tasks, diffusion models were proven to be effective in audio synthesis [68,69], while Luo and Hu [70] hybridized normalizing flows and diffusion model to generative 3D point cloud samples.

To the best of our knowledge, TimeGrad [71] 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. Two concurrent preprints also study a video diffusion model [72,73]. Both works are based on alternative architectures and focus primarily on perceptual metrics.

We will extend this survey to the camera-ready version.

2.3. 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 [7–11]. Some of these models show impressive rate-distortion performance with hierarchical structures that separately encode the prediction and error residual. Although compression models have different goals from generative models, both benefit from predictive sequential priors [26]. Note, however, that these models are ill-suited for generation since compression models typically have a small spatio-temporal context and are constructed to *preserve* local information rather than to generalize [12].

3. 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.

3.1. Background on Diffusion Probabilistic Models

Denoising diffusion probabilistic models (DDPMs) are a recent class of generative models with promising properties [19,21]. 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 [74].

Similar to hierarchical variational autoencoders (VAEs) [75], 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 [19,21]. The *forward*, or *diffusion* process is given by

$$q(\mathbf{x}_{1:N}|\mathbf{x}_0) = \prod_{n=1}^N q(\mathbf{x}_n|\mathbf{x}_{n-1}); \tag{1}$$

$$q(\mathbf{x}_n|\mathbf{x}_{n-1}) = \mathcal{N}(\mathbf{x}_n|\sqrt{1-\beta_n}\mathbf{x}_{n-1}, \beta_n\mathbf{I}).$$

Besides a predefined incremental variance schedule with $\beta_n \in (0, 1)$ for $n \in \{1, \dots, N\}$, this process is parameter-free [20,21]. The reverse process is called *denoising process*,

$$p_\theta(\mathbf{x}_{0:N}) = p(\mathbf{x}_N) \prod_{n=1}^N p_\theta(\mathbf{x}_{n-1}|\mathbf{x}_n); \tag{2}$$

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

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 the closed form [21].

In this paper, we use a simplified loss due to Ho et al. [21], who showed that the variational bound could be simplified to the following denoising *score-matching loss*,

$$L(\theta) = \mathbb{E}_{\mathbf{x}_0, n, \epsilon} \|\epsilon - f_\theta(\mathbf{x}_n, n)\|^2 \tag{3}$$

where $\mathbf{x}_n = \sqrt{\bar{\alpha}_n}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_n}\epsilon$.

We therefore define $\bar{\alpha}_n = \prod_{i=1}^n (1 - \beta_i)$, where $1 - \bar{\alpha}_n$ is the variance schedule whose square root is the standard deviation to reparametrize the injected noise ϵ . We note that this schedule ensures $\bar{\alpha}_n \rightarrow 0$ when $n \rightarrow N$. 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 [21]. 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 increased structure through an annealed Langevin dynamics procedure [20,22].

3.2. 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 [76] 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 [77]. Residual errors also play an important role in modern theories of the brain. For example, predictive coding [24] postulates that neural circuits estimate probabilistic models of other neural activity, iteratively exchanging information about error residuals. This theory has interesting connections to VAEs [25,27] and neural video compression [9,11], 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:

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

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) [76] applied to the frame sequence. This involves an autoregressive prediction network outputting μ_ϕ and a scale parameter σ ,

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

Conditioned on \mathbf{y}_0^t , this transformation is deterministic. The forward MAF transform ($\mathbf{y} \rightarrow \mathbf{x}$) converts the residual into the data sequence; the inverse transform ($\mathbf{x} \rightarrow \mathbf{y}$) decorrelates the sequence. The temporally decorrelated, sparse residuals $\mathbf{y}_0^{1:T}$ involve a simpler modeling task than generating the frames themselves. Although 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 [27] as well as in neural video compression architectures [9–11,26]. These approaches separately compress latent variables that govern the next-frame prediction as well as frame residuals, therefore achieving state-of-the-art rate-distortion performance on high-resolution video content. Although these works focused on compression, this paper focuses on generation.

We now specify the second factor in Equation (4), the generative process of the residual variable, as

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

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 Equation (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 $\mathbf{x}^{0:T}$:

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

Since the residual noise is a deterministic function of the observed and predicted frame, the first factor is deterministic and can be expressed as $q_{\phi}(\mathbf{y}_0^t|\mathbf{x}^{<t}) = \delta(\mathbf{y}_0^t - \frac{\mathbf{x}^t - \mu_{\phi}(\mathbf{x}^{<t})}{\sigma})$. The remaining N factors are identical to Equation (1) with \mathbf{x}_n being replaced by \mathbf{y}_n . Following Nichol and Dhariwal [78], 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 [71] method. This approach showed promising performance in forecasting the 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 4.6 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 the derivation of Equation (3) [21] and using the same definitions of $\bar{\alpha}_n$ and ϵ , this results in

$$L(\theta, \phi) = \mathbb{E}_{\mathbf{x}, n, \epsilon} \sum_{t=1}^T \|\epsilon - f_{\theta}(\mathbf{y}_n^t(\phi), n, \mathbf{x}^{<t})\|^2; \tag{9}$$

$$\mathbf{y}_n^t(\phi) = \sqrt{\bar{\alpha}_n} \mathbf{y}_0^t(\phi) + \sqrt{1 - \bar{\alpha}_n} \epsilon; \mathbf{y}_0^t(\phi) = \frac{\mathbf{x}^t - \mu_{\phi}(\mathbf{x}^{<t})}{\sigma}.$$

We can optimize this function using the reparameterization trick [75], 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 [79]. See Algorithms 1 and 2 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

```

while not converged do
  Sample  $\mathbf{x}^{0:T} \sim q(\mathbf{x}^{0:T})$ ;
   $n \sim \mathcal{U}(0, 1, 2, \dots, N)$ ;
   $L = 0$ ;
  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

```

Algorithm 2 Generation

```

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

```

4. 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.

4.1. 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 [80] and **KTH Actions** [81]. Among the high-resolution datasets (frame sizes of 128×128), we use **CityScape** [82], 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 [83]. These datasets cover various complexities, resolutions, and types of dynamics.

Preprocessing. For KTH and BAIR, we preprocess the videos as commonly proposed [14,27]. For CityScape, we download the portion titled *leftImg8bit_sequence_trainvaltest* from the official website (<https://www.cityscapes-dataset.com/downloads/> (accessed on 1 March 2022)). Each video is a 30-frame sequence from which we randomly select a sub-sequence. All the videos are center-cropped and downsampled to 128×128 . 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 10,000 frames sampled every 128 ticks, using 8000 for training and 2000 for testing.

4.2. 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 batch size 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 four NVIDIA RTX Titan GPUs in parallel for around 4–5 days. The number of diffusion depth is fixed to $N = 1600$, and the 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. The model size is about 123 megabytes (32-bit float numbers) for high-resolution models (128×128). To sample a 128×128 video frame, the model takes 18.5 s per 1000 iterations.

4.3. Baseline Models

SVG-LP [14] is an established sequential VAE baseline. It leverages recurrent architectures in all of the encoder and decoder, and prior to capturing the dynamics in videos. We adopt the official implementation from the authors while replacing all the LSTM with ConvLSTM layers, which helps the model scale to different video resolutions. **IVRNN** [15] 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. **SLAMP** [28] is a recent algorithm yielding stochastic predictions similar to ours. It is also similar in spirit to our idea because it incorporates “motion history” to predict the dynamics for future frames. **FutureGAN** [16] relies on an *encoder-decoder* GAN model that uses spatio-temporal 3D convolutions to process video tensors. 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** [17] 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 (https://github.com/SaulZhang/Video_Prediction_ZOO/tree/master/RetrospectiveCycleGAN (accessed on 1 March 2022)). **DVD-GAN** [29] 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 (<https://github.com/Harrypotterrrr/DVD-GAN> (accessed on 1 March 2022)).

4.4. 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 [84] and LPIPS [85], while the latter is evaluated using CRPS [86] to assess the marginal (pixel-based) predictions of future frames.

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 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 ℓ_2 distance between deep embeddings across all the layers of a pretrained network which are then averaged spatially. The LPIPS score is calculated based on individual frames and then averaged.

Another desirable property of video prediction methods is to forecast future frames reliably. Since ground truth videos exhibit multi-modal conditional distributions (e.g., a traffic light may switch from yellow to green or red), such multi-modality is best captured by proper Scoring Rules such as the Continuous Ranked Probability Score (CRPS). A brief introduction about the metric is available in Appendix A. These metrics are frequently used in probabilistic forecasting problems in, e.g., meteorology or finance [87,88].

In a nutshell, CRPS compares a single-sample estimate of the ground truth CDF (a step function) with the model’s CDF for the next frame. The latter can be efficiently estimated in one dimension by repeatedly sampling from the model. In expectation, CRPS not only rewards high accuracy of the mean prediction but also good uncertainty estimates of the model. Although CRPS is not commonly used in evaluating video prediction methods, we argue that it adds a valuable perspective on a model’s uncertainty calibration.

4.5. 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, a no-reference metric that measures frame quality irrespective of context and without reference to the ground truth. For LPIPS, a reference metric, 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.

We also reported CRPS scores in Table 1. Figure 3 shows $1/\text{CRPS}$ (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 Appendix D for the generated videos on other datasets.

Table 1. Test set perceptual (FVD, LPIPS) and forecasting (CRPS) metrics, lower is better (see Section 4 for details). **Bold** numbers denote the best performance.

	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
SLAMP	1451	1749	2998	1853	0.05	0.08	0.30	0.23	6.18	24.8	2.53	23.6
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

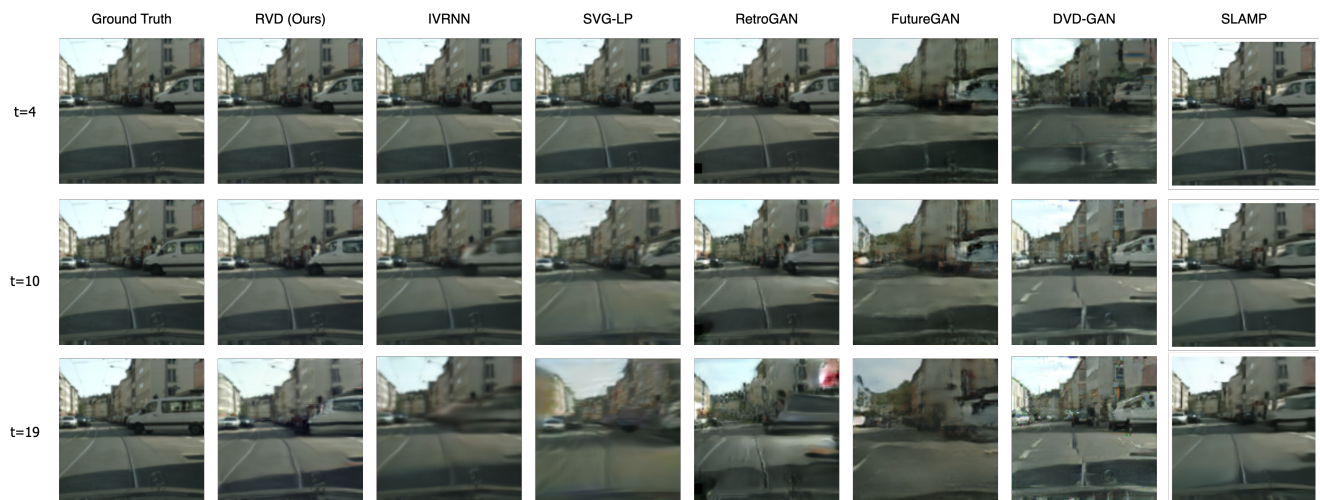


Figure 2. Generated frames on Cityscape (128×128). Compared to RVD (proposed), VAE-based models tend to become blurrier over time, while GAN-based methods generate artifacts and temporal inconsistencies.

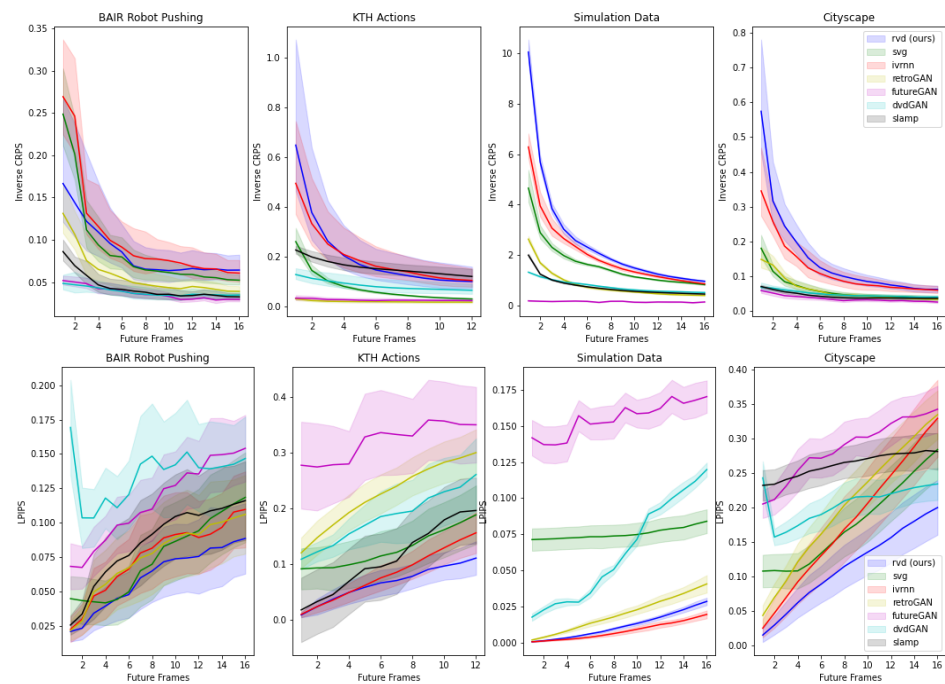


Figure 3. 1st row: Inverse CRPS scores (higher is better) as a function of the future frame index. The best performances are obtained by RVD (proposed) and IVRNN. Scores also monotonically decrease as the predictions worsen over time. 2nd row: LPIPS scores (lower is better) show the per-frame-step perceptual quality, where the shaded region reflects the standard deviation of the sampled frames at the corresponding index.

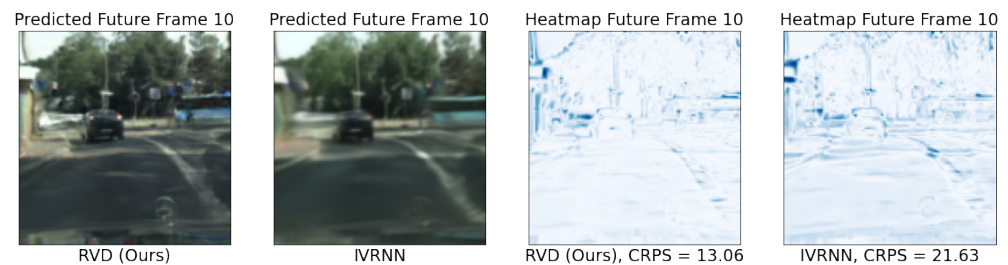


Figure 4. Spatially resolved CRPS scores (right two plots, lower is better). We compare the performance of RVD (proposed) against IVRNN on predicting the 10th future frame of a video from Cityscape. Darker areas point to larger disagreements with respect to the ground truth.

4.6. 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. Dense 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 x_0^t instead of the residual y_0^t . To address this, we make the following adjustment. Since y_0^t and 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). Please note that this ablation can be considered a customized version of TimeGrad [71] 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.

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. **Bold** numbers denote the best performance.

	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
SimpleRVD (2 + 6)	1532	1338	45	1824	0.065	0.063	0.024	0.163	5.80	12.98	0.65	10.78
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

Frame Differences vs. Prediction Residuals One may also wonder if similar results could have been obtained by modeling the residual relative to the last frame as opposed to modeling the residual relative to a predicted frame. We called this ablation “SimpleRVD” (shown in Table 2), where we set $\mu_t = \hat{x}_{t-1}$. Although the model can run more efficiently with this simplified scheme, Table 2 shows that the video quality is typically negatively affected. We conjecture that, since the predicted next frame may be already motion-compensated, the resulting residual is sparser and, hence, easier to capture by the diffusion model. Similar observations have been made in neural video compression [9,26].

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 is more sensitive to sequence length. We note that in most experiments, we outperform IVRNN even though we trained our model on shorter sequences. We also note that IVRNN leverages dense-connected hierarchical latent variables to capture long-sequence dependency. Hence, optimizing the high-level latent variables can be challenging when the training sequence is not sufficiently long. Although the diffusion model is also hierarchical, the model only learns a denoising mapping, which is a much simpler scheme than dense-connected latent variables (as shown in Figure 1).

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. To achieve the unconditional generation of a complete video sequence, we need an auxiliary image gen-

erative model to sample the initial context frames. It is 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 [89] 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 [90] or model distillation [91].

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.

Author Contributions: Conceptualization, R.Y. and S.M.; Methodology, R.Y.; Validation, R.Y. and P.S.; Formal analysis, R.Y.; Investigation, R.Y. and P.S.; Data curation, P.S.; Writing—original draft, R.Y., P.S. and S.M.; Writing—review & editing, R.Y., P.S. and S.M.; Visualization, R.Y. and P.S.; Supervision, S.M.; Project administration, S.M.; Funding acquisition, S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the IARPA WRIVA program, the National Science Foundation (NSF) under the NSF CAREER Award 2047418; NSF Grants 2003237 and 2007719, the Department of Energy, Office of Science under grant DE-SC0022331, as well as gifts from Intel, Disney, and Qualcomm.

Acknowledgments: The authors acknowledge support by the IARPA WRIVA program, the National Science Foundation (NSF) under the NSF CAREER Award 2047418; NSF Grants 2003237 and 2007719, the Department of Energy, Office of Science under grant DE-SC0022331, as well as gifts from Intel, Disney, and Qualcomm.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. About CRPS

CRPS measures the agreement of a cumulative distribution function (CDF) F with an observation x , $CRPS(F, x) = \int_{\mathbb{R}} (F(z) - \mathbb{I}\{x \leq z\})^2 dz$, 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.

Appendix B. Architecture

Our architecture extends the previously proposed DDPM [21] architecture to a temporally conditioned version. Figures A1 and A2 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 A1 for more details). Codes are available at: <https://github.com/buggyyang/RVD>

- *Channel Dim* refers to the channel dimension of all the components in the first downsampling layer of the U-Net [92] 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 [93] leverages a standard implementation of the ResNet block with 3×3 kernel, LeakyReLU activation and Group Normalization.
- All ConvGRU [94] 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 A1).
- The Upsample/Downsample components in Figures A1 and A2 involve Deconvolutional/Convolutional networks that spatially scale the feature map with scaling factors 2 and 1/2, respectively.

Figure A1a shows the overall U-Net style architecture that has been adopted for the 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 the downsampling layer and the first ResBlock of the corresponding upsampling layer as detailed in Figure A1b. Context conditioning is provided by a ConvGRU block within the downsampling layers that generates a context that is concatenated along the residual processing stream in the second ResBlock module. Additionally, each ResBlock module in either layer receives a positional encoding indicating the denoising step.

Figure A2a 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 the downsampling layer and first ResBlock of the corresponding upsampling layer as detailed in Figure A2b.

Table A1. Configuration Table, see Appendix B for definitions.

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 ($L^{\text{transform}} = 4$)

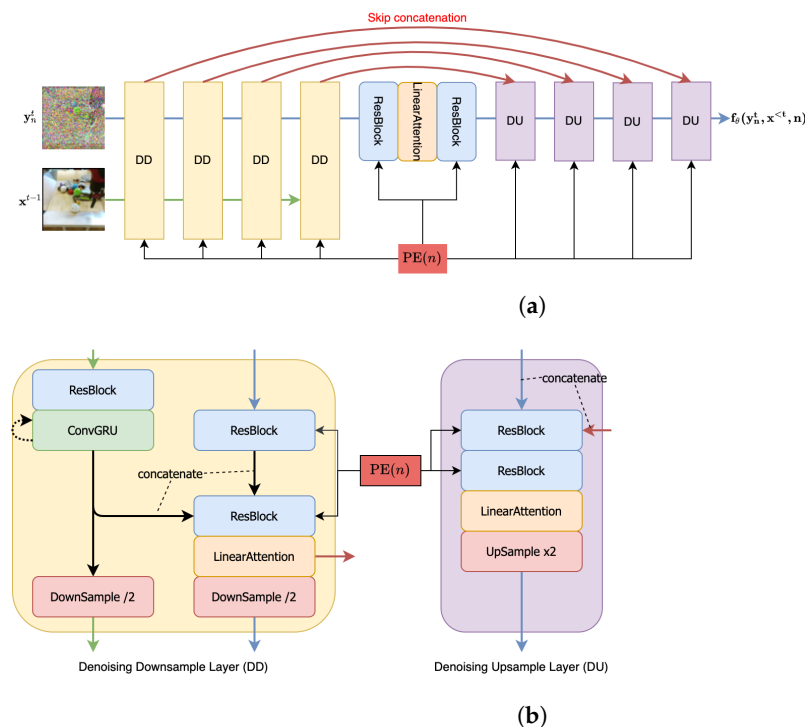


Figure A1. Autoregressive denoising module. (a) Overview figure of the autoregressive denoising module using a U-Net-inspired architecture with skip-connections. We focus on the example of

$L^{\text{denoise}} = 4$ downsampling and upsampling layers. Each of these layers, DD and DU, are explained in (b). 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 (a). As in (a), each residual block is conditioned on a positional encoding (PE) of the denoising step n .

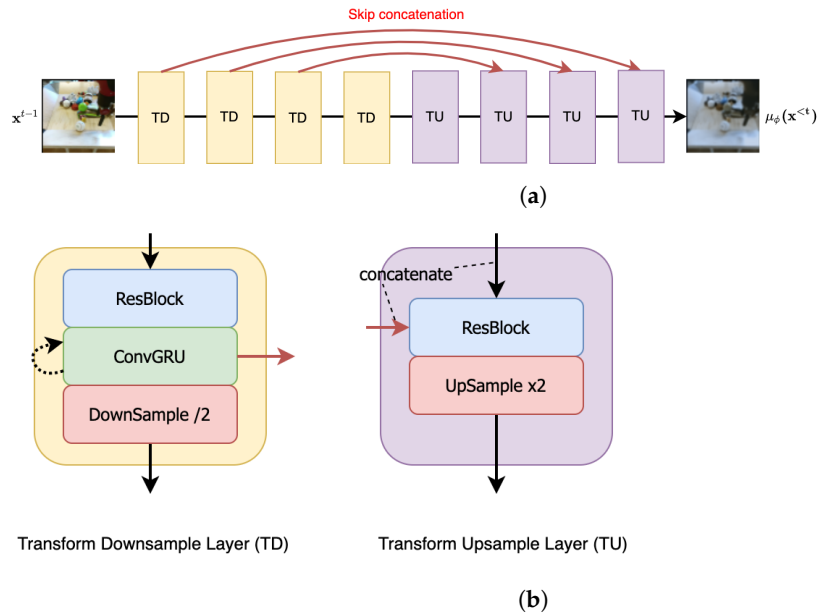


Figure A2. Autoregressive transform module for predicting the next frame μ_t . (a) Overview figure of the autoregressive transform module, predicting the mean next frame μ_t . We use a U-Net-inspired architecture with an example size of $L^{\text{transform}} = 4$ upsampling and downsampling steps. Each arrow corresponds to the arrows with the same color in (b). TD and TU layers are elaborated in (b). (b) Downsample/upsample layer design for autoregressive transform module. Each arrow corresponds to the arrows with the same color in (a).

Appendix C. Deriving the Optimization Objective

The following derivation closely follows Ho et al. [21] 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 observed scaled residuals, given by $\mathbf{y}_0^t = \frac{\mathbf{x}^t - \mu_\phi(\mathbf{x}^{<t})}{\sigma}$ for $t = 1, \dots, T$. The variational bound is as follows:

$$\begin{aligned}
 L &= \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \sum_{t=1}^T \mathbb{E}_{\mathbf{y}_{1..N}^t \sim q(\cdot | \mathbf{y}_0^t)} \left[-\log \frac{p_\theta(\mathbf{y}_{0..N}^t | \mathbf{x}^{<t})}{q(\mathbf{y}_{1..N}^t | \mathbf{y}_0^t, \mathbf{x}^{<t})} \right] \\
 &= \mathbb{E}_{\mathcal{D}} \sum_{t=1}^T \mathbb{E}_q \left[-\log \frac{p(\mathbf{y}_N^t)}{q(\mathbf{y}_N^t | \mathbf{x}^{<t})} - \sum_{n>1}^N \log \frac{p_\theta(\mathbf{y}_{n-1}^t | \mathbf{y}_n^t, \mathbf{x}^{<t})}{q(\mathbf{y}_{n-1}^t | \mathbf{y}_n^t, \mathbf{x}^{<t})} \right. \\
 &\quad \left. - \log p_\theta(\mathbf{y}_0^t | \mathbf{y}_1^t, \mathbf{x}^{<t}) \right].
 \end{aligned} \tag{A1}$$

The first term in Equation (A1), $-\log \frac{p(\mathbf{y}_N^t)}{q(\mathbf{y}_N^t | \mathbf{x}^{<t})}$, tries to match the $q(\mathbf{y}_N^t | \mathbf{x}^{<t})$ 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 $q(\mathbf{y}_N^t | \mathbf{x}^{<t}) = \mathcal{N}(\sqrt{\bar{\alpha}_n} \mathbf{y}_0^t, (1 - \bar{\alpha}_n) \mathbf{I})$. Because the variance of q is also fixed, the only effect of the 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 Equation (A1), we simplify

$$\begin{aligned} & -\log p_\theta(\mathbf{y}_0^t | \mathbf{y}_1^t, \mathbf{x}^{<t}) \\ & = \frac{1}{2\sigma\gamma_1} \|\mathbf{x}^t - \mu(\mathbf{x}^{<t}) - \sigma M_\theta(\mathbf{y}_1^t, \mathbf{x}^{<t}, 1)\|^2 + \text{const.}, \end{aligned} \quad (\text{A2})$$

Equation (A2) 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.

Deriving a parameterization for the second term of Equation (A1), $L^{\text{mid}} := -\sum_{n>1}^N \log \frac{p_\theta(\mathbf{y}_{n-1}^t | \mathbf{y}_n^t, \mathbf{x}^{<t})}{q(\mathbf{y}_{n-1}^t | \mathbf{y}_n^t, \mathbf{x}^{<t})}$, we recognize that it is a KL divergence between two Gaussians with fixed variances:

$$q(\mathbf{y}_{n-1}^t | \mathbf{y}_n^t, \mathbf{x}^{<t}) = q(\mathbf{y}_{n-1}^t | \mathbf{y}_n^t, \mathbf{y}_0^t) = \mathcal{N}(\mathbf{y}_{n-1}^t; M(\mathbf{y}_n^t, \mathbf{y}_0^t), \gamma_n \mathbf{I}), \quad (\text{A3})$$

$$p_\theta(\mathbf{y}_{n-1}^t | \mathbf{y}_n^t, \mathbf{x}^{<t}) = \mathcal{N}(\mathbf{y}_{n-1}^t; M_\theta(\mathbf{y}_n^t, \mathbf{x}^{<t}, n), \gamma_n \mathbf{I}). \quad (\text{A4})$$

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

$$L_n^{\text{mid}} = \frac{1}{2\gamma_n} \|M(\mathbf{y}_n^t, \mathbf{y}_0^t) - M_\theta(\mathbf{y}_n^t, \mathbf{x}^{<t}, n)\|^2 \quad (\text{A5})$$

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 Equation (A5) to the following form:

$$L_n^{\text{mid}} = \frac{1}{2\gamma_n} \left\| \frac{1}{\sqrt{1-\beta_n}} \left(\mathbf{y}_n^t - \frac{\beta_n}{\sqrt{1-\bar{\alpha}_n}} \epsilon \right) - M_\theta(\mathbf{y}_n^t, \mathbf{x}^{<t}, n) \right\|^2. \quad (\text{A6})$$

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

$$\begin{aligned} & L_n^{\text{mid}} \\ & = \frac{1}{2\gamma_n} \left\| \frac{1}{\sqrt{1-\beta_n}} \left(\mathbf{y}_n^t - \frac{\beta_n}{\sqrt{1-\bar{\alpha}_n}} \epsilon \right) \right. \\ & \quad \left. - \frac{1}{\sqrt{1-\beta_n}} \left(\mathbf{y}_n^t - \frac{\beta_n}{\sqrt{1-\bar{\alpha}_n}} f_\theta(\mathbf{y}_n^t, \mathbf{x}^{<t}, n) \right) \right\|^2 \end{aligned} \quad (\text{A7})$$

The resulting stochastic objective can be simplified by dropping all untrainable parameters, as suggested in [21]. As a result, one obtains the simplified objective

$$L_n^{\text{mid}} \equiv \|\epsilon - f_\theta(\mathbf{y}_n^t(\phi), n, \mathbf{x}^{<t})\|^2, \quad (\text{A8})$$

which is the denoising score-matching objective revealed in Equation (9) of our paper.

Appendix D. 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 method and IVRNN are the strongest contenders. Perceptually, the FVD metric indicates that we outperform 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.

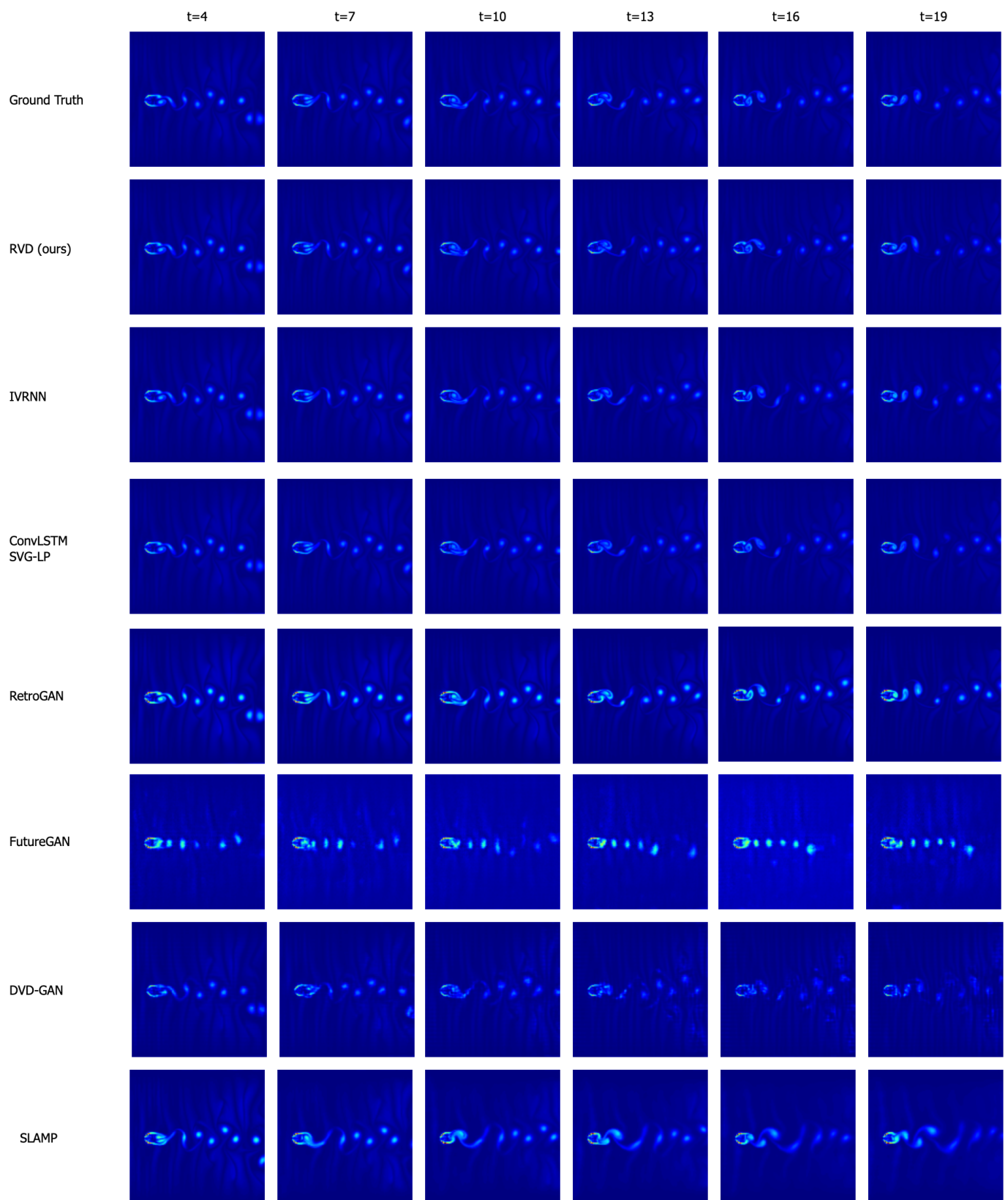
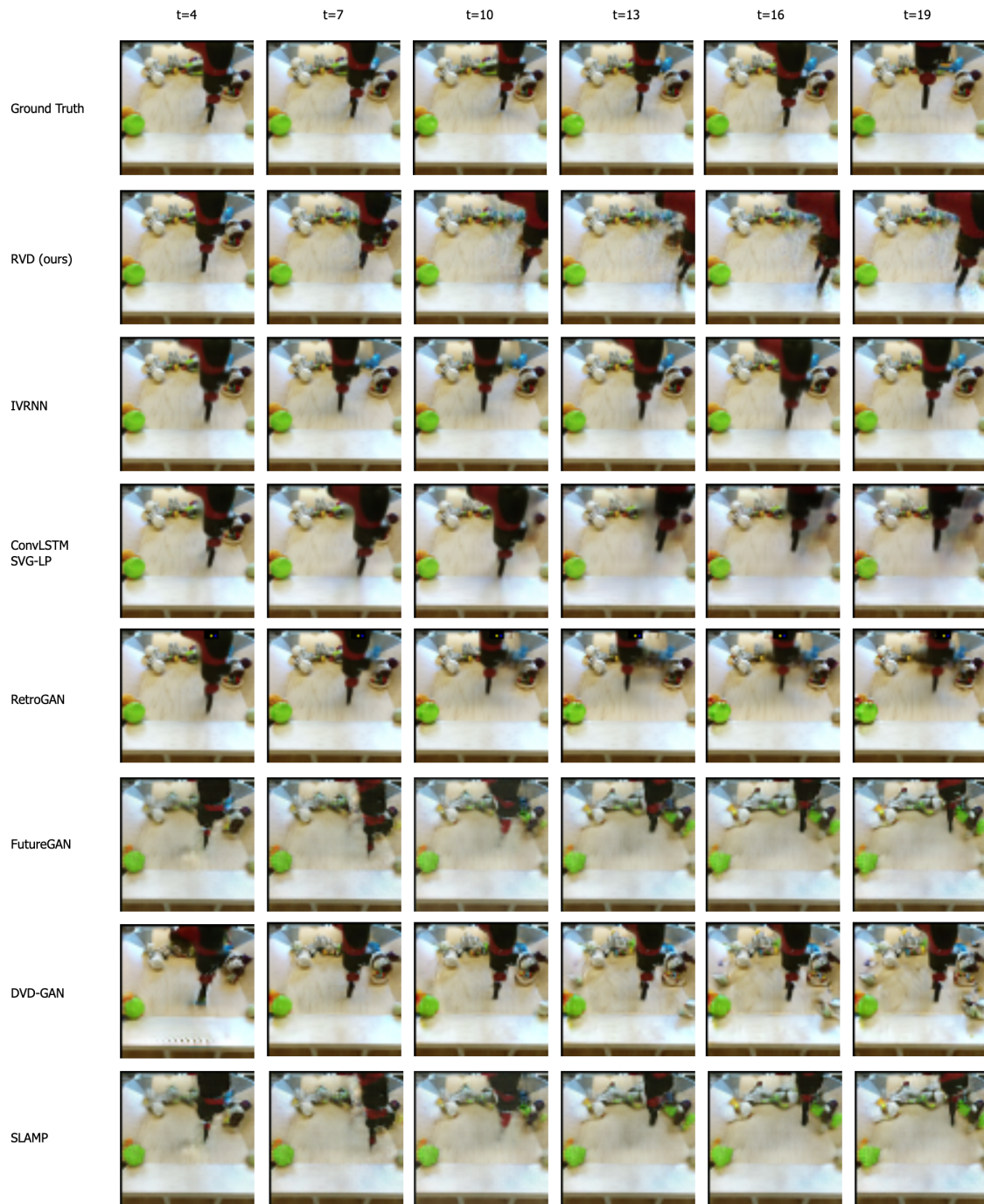


Figure A3. Prediction Quality for Simulation data. The top row indicates the ground truth, wherein we feed 4 frames as context (from $t = 0$ to $t = 3$) and predict the next 16 frames (from $t = 4$ to $t = 19$).

This is a 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 the flow of a fluid around blunt bodies. Colors indicate the vorticity of the solution to the simulation.



8

Figure A4. 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.

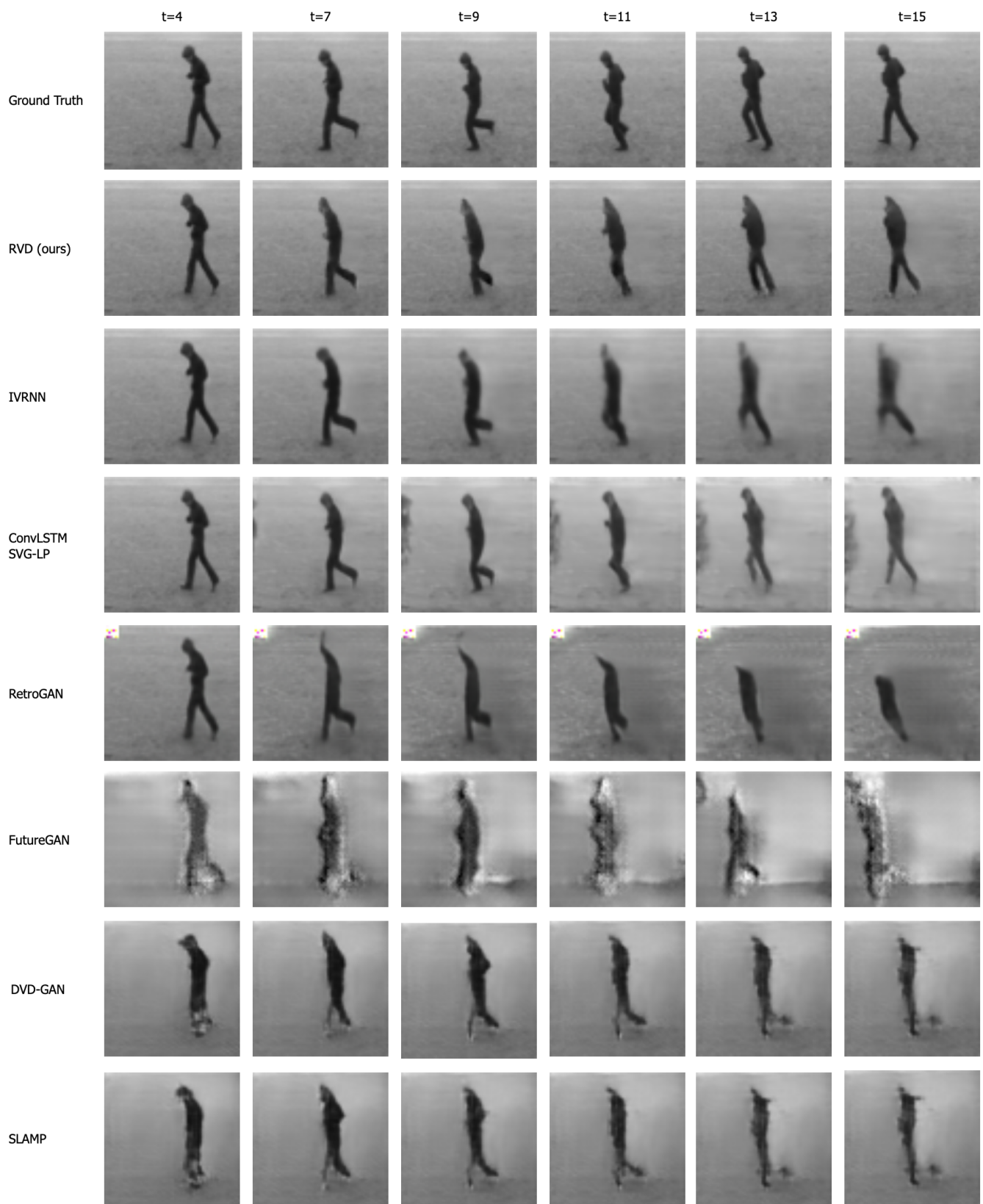


Figure A5. Prediction Quality for KTH Actions. As seen before, the top row indicates the ground truth, wherein we feed 4 frames as context (from $t = 0$ to $t = 3$), but unlike other datasets, we only predict the next 12 frames (from $t = 4$ to $t = 15$).

References

1. Oprea, S.; Martinez-Gonzalez, P.; Garcia-Garcia, A.; Castro-Vargas, J.A.; Orts-Escolano, S.; Garcia-Rodriguez, J.; Argyros, A. A review on deep learning techniques for video prediction. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 2806–2826. [[CrossRef](#)]
2. Vondrick, C.; Pirsivash, H.; Torralba, A. Anticipating visual representations from unlabeled video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 98–106.
3. Ha, D.; Schmidhuber, J. World models. *arXiv* **2018**, arXiv:1803.10122.
4. Liu, Z.; Yeh, R.A.; Tang, X.; Liu, Y.; Agarwala, A. Video frame synthesis using deep voxel flow. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4463–4471.
5. Bhattacharyya, A.; Fritz, M.; Schiele, B. Long-term on-board prediction of people in traffic scenes under uncertainty. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4194–4202.
6. Ravuri, S.; Lenc, K.; Willson, M.; Kangin, D.; Lam, R.; Mirowski, P.; Fitzsimons, M.; Athanassiadou, M.; Kashem, S.; Madge, S.; et al. Skilful precipitation nowcasting using deep generative models of radar. *Nature* **2021**, *597*, 672–677. [[CrossRef](#)] [[PubMed](#)]
7. Han, J.; Lombardo, S.; Schroers, C.; Mandt, S. Deep generative video compression. In Proceedings of the International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 9287–9298.
8. Lu, G.; Ouyang, W.; Xu, D.; Zhang, X.; Cai, C.; Gao, Z. Dvc: An end-to-end deep video compression framework. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 11006–11015.
9. Agustsson, E.; Minnen, D.; Johnston, N.; Balle, J.; Hwang, S.J.; Toderici, G. Scale-space flow for end-to-end optimized video compression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtually, 14–19 June 2020; pp. 8503–8512.
10. Yang, R.; Mentzer, F.; Van Gool, L.; Timofte, R. Learning for video compression with recurrent auto-encoder and recurrent probability model. *IEEE J. Sel. Top. Signal Process.* **2020**, *15*, 388–401. [[CrossRef](#)]
11. Yang, R.; Yang, Y.; Marino, J.; Mandt, S. Hierarchical Autoregressive Modeling for Neural Video Compression. In Proceedings of the International Conference on Learning Representations, Virtually, 3–7 May 2021.
12. Yang, Y.; Mandt, S.; Theis, L. An Introduction to Neural Data Compression. *arXiv* **2022**, arXiv:2202.06533.
13. Babaeizadeh, M.; Finn, C.; Erhan, D.; Campbell, R.H.; Levine, S. Stochastic Variational Video Prediction. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
14. Denton, E.; Fergus, R. Stochastic video generation with a learned prior. In Proceedings of the International Conference on Machine Learning, Alvsjö, Sweden, 10–15 July 2018; pp. 1174–1183.
15. Castrejon, L.; Ballas, N.; Courville, A. Improved conditional vrns for video prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7608–7617.
16. Aigner, S.; Körner, M. Futuregan: Anticipating the future frames of video sequences using spatio-temporal 3d convolutions in progressively growing gans. *arXiv* **2018**, arXiv:1810.01325.
17. Kwon, Y.H.; Park, M.G. Predicting future frames using retrospective cycle gan. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 1811–1820.
18. Lee, A.X.; Zhang, R.; Ebert, F.; Abbeel, P.; Finn, C.; Levine, S. Stochastic adversarial video prediction. *arXiv* **2018**, arXiv:1804.01523.
19. Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 2256–2265.
20. Song, Y.; Ermon, S. Generative modeling by estimating gradients of the data distribution. In Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019.
21. Ho, J.; Jain, A.; Abbeel, P. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6840–6851.
22. Song, Y.; Sohl-Dickstein, J.; Kingma, D.P.; Kumar, A.; Ermon, S.; Poole, B. Score-Based Generative Modeling through Stochastic Differential Equations. In Proceedings of the International Conference on Learning Representations, Virtually, 3–7 May 2021.
23. Song, Y.; Durkan, C.; Murray, I.; Ermon, S. Maximum likelihood training of score-based diffusion models. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 1415–1428.
24. Rao, R.P.; Ballard, D.H. Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects. *Nat. Neurosci.* **1999**, *2*, 79–87. [[CrossRef](#)]
25. Marino, J. Predictive coding, variational autoencoders, and biological connections. *Neural Comput.* **2021**, *34*, 1–44. [[CrossRef](#)] [[PubMed](#)]
26. Yang, R.; Yang, Y.; Marino, J.; Mandt, S. Insights from Generative Modeling for Neural Video Compression. *arXiv* **2021**, arXiv:2107.13136.
27. Marino, J.; Chen, L.; He, J.; Mandt, S. Improving sequential latent variable models with autoregressive flows. In Proceedings of the 2nd Symposium on Advances in Approximate Bayesian Inference, Virtually, 22 June 2021.
28. Akan, A.K.; Erdem, E.; Erdem, A.; Guney, F. Slamp: Stochastic Latent Appearance and Motion Prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Virtually, 11–17 October 2021; pp. 14728–14737.
29. Clark, A.; Donahue, J.; Simonyan, K. Adversarial video generation on complex datasets. *arXiv* **2019**, arXiv:1907.06571.
30. Dorkenwald, M.; Milbich, T.; Blattmann, A.; Rombach, R.; Derpanis, K.G.; Ommer, B. Stochastic image-to-video synthesis using cinns. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtually, 19–25 June 2021; pp. 3742–3753.

31. Nam, S.; Ma, C.; Chai, M.; Brendel, W.; Xu, N.; Kim, S.J. End-to-end time-lapse video synthesis from a single outdoor image. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 1409–1418.
32. Wu, C.; Huang, L.; Zhang, Q.; Li, B.; Ji, L.; Yang, F.; Sapiro, G.; Duan, N. Godiva: Generating open-domain videos from natural descriptions. *arXiv* **2021**, arXiv:2104.14806.
33. Singer, U.; Polyak, A.; Hayes, T.; Yin, X.; An, J.; Zhang, S.; Hu, Q.; Yang, H.; Ashual, O.; Gafni, O.; et al. Make-a-video: Text-to-video generation without text-video data. *arXiv* **2022**, arXiv:2209.14792.
34. Gafni, O.; Polyak, A.; Ashual, O.; Sheynin, S.; Parikh, D.; Taigman, Y. Make-a-scene: Scene-based text-to-image generation with human priors. In Proceedings of the Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, 23–27 October 2022; pp. 89–106.
35. Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv* **2022**, arXiv:2204.06125.
36. Zhang, H.; Koh, J.Y.; Baldridge, J.; Lee, H.; Yang, Y. Cross-modal contrastive learning for text-to-image generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtually, 19–25 June 2021; pp. 833–842.
37. Zhou, Y.; Zhang, R.; Chen, C.; Li, C.; Tensmeyer, C.; Yu, T.; Gu, J.; Xu, J.; Sun, T. LAFITE: Towards Language-Free Training for Text-to-Image Generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2022.
38. Wang, T.C.; Liu, M.Y.; Zhu, J.Y.; Liu, G.; Tao, A.; Kautz, J.; Catanzaro, B. Video-to-Video Synthesis. *arXiv* **2018**, arXiv:1808.06601.
39. Saito, M.; Saito, S.; Koyama, M.; Kobayashi, S. Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal gan. *Int. J. Comput. Vis.* **2020**, *128*, 2586–2606. [[CrossRef](#)]
40. Yu, S.; Tack, J.; Mo, S.; Kim, H.; Kim, J.; Ha, J.W.; Shin, J. Generating Videos with Dynamics-aware Implicit Generative Adversarial Networks. In Proceedings of the International Conference on Learning Representations, Virtually, 25–29 April 2022.
41. Byeon, W.; Wang, Q.; Srivastava, R.K.; Koumoutsakos, P. Contextvp: Fully context-aware video prediction. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 753–769.
42. Finn, C.; Goodfellow, I.; Levine, S. Unsupervised learning for physical interaction through video prediction. In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016.
43. Lotter, W.; Kreiman, G.; Cox, D. Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning. In Proceedings of the International Conference on Learning Representations, Palais des Congres Neptune, France, 24–26 April 2017.
44. Srivastava, N.; Mansimov, E.; Salakhutdinov, R. Unsupervised Learning of Video Representations using LSTMs. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015.
45. Walker, J.; Gupta, A.; Hebert, M. Dense optical flow prediction from a static image. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 11–18 December 2015; pp. 2443–2451.
46. Villegas, R.; Yang, J.; Hong, S.; Lin, X.; Lee, H. Decomposing Motion and Content for Natural Video Sequence Prediction. In Proceedings of the International Conference on Learning Representations, Palais des Congres Neptune, France, 24–26 April 2017.
47. Liang, X.; Lee, L.; Dai, W.; Xing, E.P. Dual motion GAN for future-flow embedded video prediction. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1744–1752.
48. Li, Y.; Mandt, S. Disentangled sequential autoencoder. In Proceedings of the International Conference on Machine Learning, Alvsjo, Sweden, 10–15 July 2018; pp. 5670–5679.
49. Kumar, M.; Babaeizadeh, M.; Erhan, D.; Finn, C.; Levine, S.; Dinh, L.; Kingma, D. Videoflow: A flow-based generative model for video. *arXiv* **2019**, arXiv:1903.01434.
50. Unterthiner, T.; van Steenkiste, S.; Kurach, K.; Marinier, R.; Michalski, M.; Gelly, S. Towards accurate generative models of video: A new metric & challenges. *arXiv* **2018**, arXiv:1812.01717.
51. Villegas, R.; Pathak, A.; Kannan, H.; Erhan, D.; Le, Q.V.; Lee, H. High fidelity video prediction with large stochastic recurrent neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019.
52. Babaeizadeh, M.; Saffar, M.T.; Nair, S.; Levine, S.; Finn, C.; Erhan, D. Fitvid: Overfitting in pixel-level video prediction. *arXiv* **2021**, arXiv:2106.13195.
53. Villegas, R.; Erhan, D.; Lee, H. Hierarchical long-term video prediction without supervision. In Proceedings of the International Conference on Machine Learning, Alvsjo, Sweden, 10–15 July 2018; pp. 6038–6046.
54. Yan, W.; Zhang, Y.; Abbeel, P.; Srinivas, A. Videogpt: Video generation using vq-vae and transformers. *arXiv* **2021**, arXiv:2104.10157.
55. Rakhimov, R.; Volkhonskiy, D.; Artemov, A.; Zorin, D.; Burnaev, E. Latent video transformer. *arXiv* **2020**, arXiv:2006.10704.
56. Lee, W.; Jung, W.; Zhang, H.; Chen, T.; Koh, J.Y.; Huang, T.; Yoon, H.; Lee, H.; Hong, S. Revisiting Hierarchical Approach for Persistent Long-Term Video Prediction. In Proceedings of the International Conference on Learning Representations, Virtually, 3–7 May 2021.
57. Bayer, J.; Osendorfer, C. Learning stochastic recurrent networks. *arXiv* **2014**, arXiv:1411.7610.
58. Chung, J.; Kastner, K.; Dinh, L.; Goel, K.; Courville, A.C.; Bengio, Y. A recurrent latent variable model for sequential data. *arXiv* **2015**, arXiv:1506.02216.

59. Wu, B.; Nair, S.; Martin-Martin, R.; Fei-Fei, L.; Finn, C. Greedy hierarchical variational autoencoders for large-scale video prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtually, 19–25 June 2021; pp. 2318–2328.
60. Zhao, L.; Peng, X.; Tian, Y.; Kapadia, M.; Metaxas, D. Learning to forecast and refine residual motion for image-to-video generation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 387–403.
61. Franceschi, J.Y.; Delasalles, E.; Chen, M.; Lamprier, S.; Gallinari, P. Stochastic Latent Residual Video Prediction. In Proceedings of the 37th International Conference on Machine Learning, Virtually, 13–18 July 2020; pp. 3233–3246.
62. Blau, Y.; Michaeli, T. The perception-distortion tradeoff. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6228–6237.
63. Vondrick, C.; Pirsaviash, H.; Torralba, A. Generating videos with scene dynamics. In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016.
64. Tulyakov, S.; Liu, M.Y.; Yang, X.; Kautz, J. Mocogan: Decomposing motion and content for video generation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1526–1535.
65. Gui, J.; Sun, Z.; Wen, Y.; Tao, D.; Ye, J. A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 3313–3332. [[CrossRef](#)]
66. Saharia, C.; Ho, J.; Chan, W.; Salimans, T.; Fleet, D.J.; Norouzi, M. Image super-resolution via iterative refinement. *arXiv* **2021**, arXiv:2104.07636.
67. Pandey, K.; Mukherjee, A.; Rai, P.; Kumar, A. DiffuseVAE: Efficient, Controllable and High-Fidelity Generation from Low-Dimensional Latents. *arXiv* **2022**, arXiv:2201.00308.
68. Chen, N.; Zhang, Y.; Zen, H.; Weiss, R.J.; Norouzi, M.; Chan, W. WaveGrad: Estimating Gradients for Waveform Generation. In Proceedings of the International Conference on Learning Representations, Virtually, 3–7 May 2021.
69. Kong, Z.; Ping, W.; Huang, J.; Zhao, K.; Catanzaro, B. DiffWave: A Versatile Diffusion Model for Audio Synthesis. In Proceedings of the International Conference on Learning Representations, Virtually, 3–7 May 2021.
70. Luo, S.; Hu, W. Diffusion probabilistic models for 3d point cloud generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtually, 19–25 June 2021; pp. 2837–2845.
71. Rasul, K.; Seward, C.; Schuster, I.; Vollgraf, R. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In Proceedings of the International Conference on Machine Learning, Virtually, 18–24 July 2021; pp. 8857–8868.
72. Ho, J.; Salimans, T.; Gritsenko, A.; Chan, W.; Norouzi, M.; Fleet, D.J. Video Diffusion Models. *arXiv* **2022**, arXiv:2204.03458.
73. Voleti, V.; Jolicoeur-Martineau, A.; Pal, C. MCVD-Masked Conditional Video Diffusion for Prediction, Generation, and Interpolation. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022.
74. Brock, A.; Donahue, J.; Simonyan, K. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
75. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
76. Papamakarios, G.; Pavlakou, T.; Murray, I. Masked autoregressive flow for density estimation. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
77. Schapire, R.E. A brief introduction to boosting. In Proceedings of the Ijcai, Stockholm, Sweden, 31 July–6 August 1999; Volume 99, pp. 1401–1406.
78. Nichol, A.Q.; Dhariwal, P. Improved denoising diffusion probabilistic models. In Proceedings of the International Conference on Machine Learning, Virtually, 18–24 July 2021; pp. 8162–8171.
79. Kolen, J.F.; Kremer, S.C. *A Field Guide to Dynamical Recurrent Networks*; John Wiley & Sons: Hoboken, NJ, USA, 2001.
80. Ebert, F.; Finn, C.; Lee, A.X.; Levine, S. Self-Supervised Visual Planning with Temporal Skip Connections. In Proceedings of the CoRL, Mountain View, CA, USA, 13–15 November 2017; pp. 344–356.
81. Schuld, C.; Laptev, I.; Caputo, B. Recognizing human actions: A local SVM approach. In Proceedings of the 17th International Conference on Pattern Recognition, Cambridge, UK, 23–26 August 2004; Volume 3, pp. 32–36.
82. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016.
83. Chirila, D.B. Towards Lattice Boltzmann Models for Climate Sciences: The GeLB Programming Language with Applications. Ph.D. Thesis, Universität Bremen, Bremen, Germany, 2018.
84. Unterthiner, T.; van Steenkiste, S.; Kurach, K.; Marinier, R.; Michalski, M.; Gelly, S. FVD: A new metric for video generation. In Proceedings of the ICLR 2019 Workshop for Deep Generative Models for Highly Structured Data, New Orleans, LA, USA, 6–9 May 2019.
85. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 586–595.
86. Matheson, J.E.; Winkler, R.L. Scoring rules for continuous probability distributions. *Manag. Sci.* **1976**, *22*, 1087–1096. [[CrossRef](#)]

87. Hersbach, H. Decomposition of the continuous ranked probability score for ensemble prediction systems. *Weather Forecast.* **2000**, *15*, 559–570. [[CrossRef](#)]
88. Gneiting, T.; Ranjan, R. Comparing density forecasts using threshold-and quantile-weighted scoring rules. *J. Bus. Econ. Stat.* **2011**, *29*, 411–422. [[CrossRef](#)]
89. Smaira, L.; Carreira, J.; Noland, E.; Clancy, E.; Wu, A.; Zisserman, A. A short note on the kinetics-700-2020 human action dataset. *arXiv* **2020**, arXiv:2010.10864.
90. Song, J.; Meng, C.; Ermon, S. Denoising Diffusion Implicit Models. In Proceedings of the International Conference on Learning Representations, Virtually, 3–7 May 2021.
91. Salimans, T.; Ho, J. Progressive distillation for fast sampling of diffusion models. *arXiv* **2022**, arXiv:2202.00512.
92. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
93. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
94. Ballas, N.; Yao, L.; Pal, C.; Courville, A.C. Delving Deeper into Convolutional Networks for Learning Video Representations. In Proceedings of the ICLR (Poster), San Juan, Puerto Rico, 2–4 May 2016.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.