A Hierarchical Variational Neural Uncertainty Model for Stochastic Video Prediction

Moitreya Chatterjee[1] Narendra Ahuja[1] Anoop Cherian[2]
[1]University of Illinois at Urbana-Champaign Champaign, IL 61820, USA
[2]Mitsubishi Electric Research Laboratories, Cambridge, MA 02139
metro.smiles@gmail.com n-ahuja@illinois.edu cherian@merl.com

Abstract

Predicting the future frames of a video is a challenging task, in part due to the underlying stochastic real-world phenomena. Prior approaches to solve this task typically estimate a latent prior characterizing this stochasticity, however do not account for the predictive uncertainty of the (deep learning) model. Such approaches often derive the training signal from the mean-squared error (MSE) between the generated frame and the ground truth, which can lead to sub-optimal training, especially when the predictive uncertainty is high. Towards this end, we introduce Neural Uncertainty Quantifier *(NUQ) - a stochastic quantification of the model's predictive uncertainty, and use it to weigh the MSE loss. We propose a hierarchical, variational framework to derive NUQ in a principled manner using a deep, Bayesian graphical model. Our experiments on three benchmark stochastic video prediction datasets show that our proposed framework trains more effectively compared to the state-of-the-art models (especially when the training sets are small), while demonstrating better video generation quality and diversity against several evaluation metrics.*

1. Introduction

Extrapolating the present into the future is a task essential to predictive reasoning and planning. When artificial intelligence systems are deployed to work side-by-side with humans, it is critical that they reason about their visual context and generate plausible futures so that they can anticipate the potential needs of humans or catastrophic risks and be better equipped. Such a visual future generation framework could also benefit applications such as video surveillance [57], human action recognition and forecasting [49, 55] as well as simulation of real-world scenarios to train robot learning algorithms, including autonomous driving [28]. However, such applications have a high element of stochasticity, which makes this prediction task challenging.

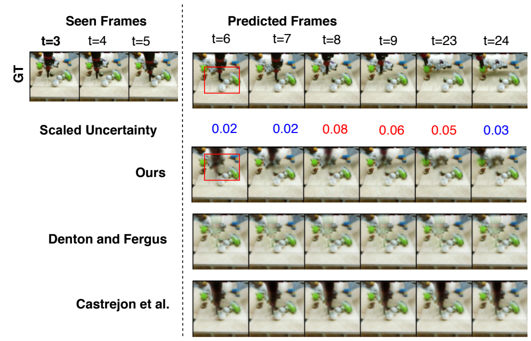


Figure 1. Qualitative results vis-á-vis state-of-the-art video prediction baselines using the proposed NUQ framework on the BAIR Push dataset [15], trained using only 2,000 samples (rather than the full 40K samples). Regions with high motion are shown by a red box. Also shown is an estimate of the per-frame scaled uncertainty estimated by our model. Note that the robotic arm changes direction at $t = 8$, which is reflected in the predicted uncertainty.

The resurgence of deep neural networks, especially the advent of generative adversarial networks [20], has enabled significant progress in the development of frameworks for generating visual data, such as images [30]. While, temporally-evolving extensions of such image generation techniques have shown benefits in artificially producing video sequences for deterministic visual contexts [54, 56, 19, 37, 29], they usually fail to model real-world sequences that are often highly stochastic. Several recent works in video generation, thus design modules to factor in data stochasticity while making predictions [39, 2, 13, 8]. Specifically, such methods assume a latent stochastic prior, from which random samples are drawn, in order to generate future frames. In Babaeizadeh *et al.* [2], this stochastic prior is assumed to follow a fixed normal distribution, which is sampled at every time step, while Denton and Fergus [13], learn this prior from data. The latter's key insight is to use a variational posterior to guide the learning of the prior to produce the sufficient statistics of the normal distribution governing the prior. Such stochastic methods typically employ a deterministic decoder (a neural network) that combines an embedding of the visual context and a random sample from the stochastic prior to generate a future video frame. The variance in this prior accounts for the stochasticity underlying the data. To train such models, the mean-squared error (MSE) is then minimized by comparing the predictions against the true video frames.

Nonetheless existing stochastic methods have largely ignored the predictive uncertainty (aleatoric uncertainty) [31] of the models, which might adversarially impact downstream tasks that leverage these predictions. From a machine learning stand point, ignoring the predictive uncertainty might lead to the model being unnecessarily penalized (via the MSE), even if it makes a very uncertain prediction that ends up being different from the ground-truth. This can destabilize the training of the underlying neural networks, leading to slower convergence or requiring larger training data. This is of importance because such data might be expensive or sometimes even difficult to collect (e.g., predicting the next human actions in instruction videos, or a rare traffic incident), and thus effective training with limited data is essential.

In this work, we rise up to these challenges by quantifying the predictive uncertainty of a stochastic frame prediction model and using it to calibrate its training objective. In particular a stochastic estimate of the predictive uncertainty, derived from the latent space of the model, is used to weigh the MSE. That is, when the uncertainty is high, the MSE is down-weighted proportionately, and vice versa; thereby regularizing the backpropagation gradients to train the frame generation module. Moreover, this uncertainty estimate can be used for downstream tasks, such as for example, regulating the manuevers in autonomous driving [28, 55]. We call our scheme, *Neural Uncertainty Quantifier* (NUQ).

We observe that the weight on the MSE that NUQ introduces, basically amounts to the variance of the normal distribution governing the generated output. Thus, an obvious consideration would be to estimate the variance directly from the output. However, this may be cumbersome due to the very high dimensionality of the output space (order of the number of pixels). We instead, choose to derive it from the variance of the latent space prior, which has far fewer dimensions. Specifically, NUQ leverages a variational, deep, hierarchical, graphical model to bridge the variance of the latent space prior and that of the output. Our framework is trained end-to-end. Sample generations by our framework is shown in Figure 1. In addition, inspired by the recent successes of generative adversarial networks [20, 37, 39], we propose a variant of our framework that uses a novel sequence discriminator, in an adversarial setting. This discriminator module helps to constrain the space of possible output frames, while enforcing motion regularities in the generated videos.

To empirically verify our intuitions, we present experiments on a synthetic (Stochastic Moving MNIST [13]) and two challenging real world datasets: KTH-Action [47], and BAIR push [15] for the task of future frame generation. Our results show that our framework converges faster than prior stochastic video generation methods, and leads to state-of-the-art video generation quality, even when the dataset size is small, while exhibiting generative diversity in the predicted frames. Below, we summarize the main contributions of this paper:

1. We present *Neural Uncertainty Quantifier* (NUQ), a deep, Bayesian network that learns to estimate the predictive uncertainty of stochastic frame generation models, which can be leveraged to control the training updates, for faster and improved convergence of predictive models.

2. We propose a novel, hierarchical, variational training scheme that allows for incorporating problem-specific knowledge into the predictions via hyperpriors on the uncertainty estimate.

3. Experimental results demonstrate our framework's better video generation and faster training capabilities, even with small training sets compared to recent state-of-the-art methods on stochastic video generation tasks, across multiple datasets.

2. Related Work

Early works in video frame prediction mostly resorted to end-to-end deterministic architectures [44, 48, 17]. Ranzato *et al*. [44] proposed to divide frames into patches and extrapolate their evolution in time. Srivastava *et al*. [48] use image encoders with pre-trained weights to encode the frames. ContextVP [7] and PredNet [41] leverage Convolutional LSTMs [60] for video prediction. Fragkiadaki *et al*. [19] proposes pose extrapolation using LSTMs. More recent approaches [37, 62] seek to predict frames bidirectionally (future and past), during training. However, the inherent deterministic nature of such models [29] often becomes a bottleneck to their performance. Instead, we seek to investigate approaches that allow modeling of the underlying stochasticity in the data while generating an assessment of the model's predictive uncertainty.

Stochastic approaches constitute a recently emerging and one of the most promising classes of video prediction methods [39, 2, 13]. These approaches model the data stochasticity using a latent prior distribution and are thus readily generalizable to real-world scenarios. Popular among them are STORNs [5], VRNNs [11], SRNNs [18], and DMMs [35]. SV2P [2] is a more recent method that uses a single set of stochastic latent variables that are assumed to follow a fixed prior distribution. Denton and Fergus [13] improve upon SV2P [2] by allowing the prior distribution to be adapted at every time step by casting the prior as a trainable neural network. Their method is shown to achieve superior empirical performance, thus underlining the importance of learning to model data stochasticity. We also note that generative models have recently been adapted to incorporate stochastic information through a hierarchical latent space [51, 52]. Such networks have also been applied to frame prediction tasks [8]. None of these approaches however, explore the effectiveness of modeling the predictive uncertainty. While, technically it might be possible that the stochastic modules in these prior approaches can learn to quantify this uncertainty implicitly, it may need longer training periods or larger datasets. Instead we show that explicitly incorporating the predictive uncertainty into the learning objective, via a hierarchical, variational framework improves training and inference.

Another line of work in frame prediction seeks to decouple the video into static and moving components [53, 14, 40, 26, 59, 21]. Some of these approaches are deterministic, others stochastic. Denton *et al*. [14] extracts content and pose information for this purpose. Villegas *et al*. [53] adopt a multiscale approach towards frame prediction which works by building a model of object motion, however they require supervisory information, such as annotated pose, during training. Ye *et al*. [61] propose a compositional approach to video prediction by stitching the motion of individual entities. While promising, their approach relies on auxiliary information such as spatial locations of the entities, and as a result, is difficult to generalize. Jin *et al*. [29] investigates decoupling in the frequency space, however they do not model the data stochasticity explicitly. Hsieh *et al*. [27] describes a similar approach by modeling the motion and appearance of each object in the video, but without requiring any auxiliary information. Different from these set of approaches, our proposed framework models frames holistically and is thus agnostic to the video content.

Modeling the predictive uncertainty in deep networks has garnered significant attention lately [6, 12, 36, 42, 50]. Some of these works [1, 36, 42] investigate it in a classification setting, while some others [6, 24] in the context of regression. Uncertainty has also recently been explored in the context of generative models [38, 43, 58]. However, predictive uncertainty modeling in the context of frame prediction has remained largely unexplored. NUQ attempts to fill this gap.

3. Background

Suppose $\boldsymbol{x}_{1:T} := \langle \boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_T \rangle$ denotes a sequence of random variables, each $\boldsymbol{x}_t$ representing a video frame at time step t. Assuming we have access to a few initial frames $x_{1:F}$, to set the visual context (where $1 \leq F < T$), our goal is to generate the rest of the frames $\boldsymbol{x}_{F+1}$ onwards autoregressively, i.e., conditioned on the seen frames and what has been generated hitherto. This task amounts to finding a prediction model $p_\theta(\cdot)$, parameterized by θ, that minimizes the expected negative log-likelihood. When unknown factors of variation are involved in the data generation process, a determinstic predictive model is insufficient. A standard way to incorporate stochasticity is by assuming the generated frames are in turn conditioned on a latent prior model $p(\boldsymbol{z}_t)$; i.e., $\boldsymbol{z}_t \sim p(\boldsymbol{z}_t), \boldsymbol{x}_t \sim p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}, \boldsymbol{z}_t)$. Specifically, the stochasticity in the generative process is characterized by the variance in $p(\boldsymbol{z}_t)$, that produces diversity in $\boldsymbol{z}_t \sim p(\boldsymbol{z}_t)$. Diversity among predicted frames emerges as a result of this variance.

A well-known problem with the use of such latent stochastic priors is the intractability that it brings into the estimation of the *evidence* or the log-partition function: $p(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}) = \int_{\boldsymbol{z}_t} p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}, \boldsymbol{z}_t) p(\boldsymbol{z}_t) d\boldsymbol{z}_t$. This problem is typically avoided by casting this estimation in an encoder-decoder setup, where the encoder embeds $\boldsymbol{x}_{1:t}$ as $\boldsymbol{z}_t \sim p(\boldsymbol{z}_t|\boldsymbol{x}_{1:t})$, while the decoder outputs $\boldsymbol{x}_t \sim p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}, \boldsymbol{z}_t)$. In order to train efficiently, access to a variational posterior $q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t})$ – that approximates the true posterior $p(\boldsymbol{z}_t|\boldsymbol{x}_{1:t})$ of the encoder – is assumed. Using this approximate posterior, learning the model parameters θ and ϕ amounts to maximizing the variational lower bound, $\mathcal{L}_{\theta,\phi}$ [33]:

$$\begin{aligned} \log p(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}) \geq \mathcal{L}_{\theta,\phi}, \text{ where} \\ \mathcal{L}_{\theta,\phi} := \int_{\boldsymbol{z}_t} q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t}) \log p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}, \boldsymbol{z}_t) d\boldsymbol{z}_t \\ - \int_{\boldsymbol{z}_t} q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t}) \log \frac{q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t})}{p(\boldsymbol{z}_t)} d\boldsymbol{z}_t \end{aligned} \quad (1)$$

From the definition, this amounts to:

$$\begin{aligned} \mathcal{L}_{\theta,\phi} = \mathbb{E}_{q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t})} \log p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}, \boldsymbol{z}_t) - \\ \mathrm{KL}\left(q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t}) \| p(\boldsymbol{z}_t)\right), \text{for } t > F. \end{aligned} \quad (2)$$

Leveraging the re-parametrization trick ([33]) allows efficient optimization of the likelihood loss in Eq. 2, permitting us to learn the parameters θ and ϕ. Note that the expectation term in Eq. 2 boils down to a standard MSE over all predicted frames $\boldsymbol{x}_{F+1:T}$ in the training set when the $p_\theta(\cdot)$ term is assumed to follow a Gaussian distribution with an isotropic constant variance. In this setting, the KL divergence in Eq. 2 acts as a regularizer on $q_\phi(\cdot)$ so that this posterior does not just copy an encoding of $\boldsymbol{x}_t$ available to it as $\boldsymbol{z}_t$, instead captures the density of a latent distribution that is useful to the prediction model in maximizing the first term in Eq. 2. In conditional variational autoencoders [2, 33], the latent prior $p(\boldsymbol{z}_t)$ is typically assumed to be $\mathcal{N}(0, 1)$ - a choice that can be sub-optimal. A better approach is perhaps to learn this prior so that the stochasticity of the future frame can be guided by the data itself. To this end, Denton and Fergus [13] suggests a learned stochastic prior model $p(\boldsymbol{z}_t) = p_\psi(\boldsymbol{z}_t) := p_\psi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t-1})$, parametrized by ψ, which is learned by minimizing its divergence from the variational posterior $q_\phi(\cdot)$ through the KL-term in Eq. 2. As the posterior $q_\phi(\cdot)$ has access to the current input sample $\boldsymbol{x}_t$, it can guide the prior (which does not have access to $\boldsymbol{x}_t$, but only to $\boldsymbol{x}_{1:t-1}$) to produce a distribution on $\boldsymbol{z}_t$ that mimics the posterior (and hence we can discard the posterior at test time). Thus, the training-time sampling pipeline is given by: $\boldsymbol{z}_t \sim q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t}), \boldsymbol{x}_t \sim p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}, \boldsymbol{z}_t)$, and $p_\psi \overset{d}{\longleftrightarrow} q_\phi$ (matching in distribution). While learning the stochastic prior $p_\psi(\cdot)$ allows for characterizing the data stochasticity, the model's predictive uncertainty remains unaccounted for. Our hierarchical framework for estimating this uncertainty, follows a two-step process. The first is the estimation of the learned prior, $p_\psi(\cdot)$. The key idea in the second step is to leverage the variance in the learned prior $p_\psi(\cdot)$ to estimate this uncertainty. Since the prior estimation network, $p_\psi(\cdot)$, is retained both during training and inference (unlike the posterior), this permits its usage for downstream tasks, during inference.

4. Proposed Method

As alluded to above, we seek to control the prediction model using the uncertainty estimated directly from the stochastic prior. Subsequently, we assume the prediction model consists of an LSTM, f_θ, with weights θ such that:

$$\hat{\boldsymbol{x}}_t = f_\theta(\boldsymbol{x}_{1:t-1}, \boldsymbol{z}_t) := f_\theta(\boldsymbol{x}_{t-1}, \boldsymbol{z}_t; \boldsymbol{h}^\theta_{t-1}), \quad (3)$$

where $\hat{\boldsymbol{x}}_t$ denotes the t^{th} generated frame and h^θ_{t-1} captures the internal states of the LSTM. The generated frame $\hat{\boldsymbol{x}}_t$ is then sent through the likelihood model to compute the MSE. With this setup, we are now ready to introduce our neural uncertainty quantifier (NUQ). Figure 2 provides an overview of our framework.

4.1. Neural Uncertainty Quantifier

As is standard practice, let us assume the data likelihood model $p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}, \boldsymbol{z}_t) \sim \mathcal{N}(\boldsymbol{x}_t, \frac{1}{b_t})$, where b_t denotes the precision (inverse variance) of our prediction model, where $b_t > 0$, $b_t \in \mathbb{R}$. Denton and Fergus [13] assumes b_t to be an isotropic constant, such that the negative log-likelihood of the predicted frame $\hat{\boldsymbol{x}}_t$ boils down to computing the ℓ_2-loss. This reduces Eq. 2 to become the evidence lower bound (ELBO) [33]:

$$\mathcal{L}_{\theta,\phi,\psi} := \sum_{t=F+1}^{T} \frac{1}{2} \|\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t\|^2 + \mathrm{KL}\left(q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t}) \,\|\, p_\psi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t-1})\right). \quad (4)$$

Our key insight to the proposed uncertainty measure arises from the observation that the ℓ_2-norm term in Eq. 4 does not include any dependency on the uncertainty associated with the prediction of $\hat{\boldsymbol{x}}_t$. Note that there are two extreme situations when this loss is large that impacts effective training: (i) when there is no uncertainty associated with the generation of $\hat{\boldsymbol{x}}_t$, however the prediction model is not trained well, such that $\hat{\boldsymbol{x}}_t$ does not match $\boldsymbol{x}_t$, and (ii) when there is uncertainty involved in the generative model such that the generated $\hat{\boldsymbol{x}}_t$, while plausible given the context, is different from $\boldsymbol{x}_t$. Thus, the key research question for effective model training becomes: how can we equip the prediction model to differentiate these situations? Our solution is to directly condition the prediction model with the uncertainty derived from the prior $p_\psi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t-1})$, so that when the stochasticity is high for the generated frames, the ℓ_2-loss term is weighed down such that the gradients computed on this term will have a lesser impact in updating the weights of the neural network; thereby stabilizing the training.

Suppose our prior $p_\psi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t-1})$ is a normal distribution $\mathcal{N}(\boldsymbol{\mu}^{\boldsymbol{z}}_t, \boldsymbol{\Sigma}^{\boldsymbol{z}}_t)$, with parameters $\boldsymbol{\mu}^{\boldsymbol{z}}_t$, the mean, and $\boldsymbol{\Sigma}^{\boldsymbol{z}}_t$ the covariance matrix - capturing the predictive uncertainty, in the latent space. For better characterization of this prior model, we assume it to be implemented as an LSTM g_ψ with weights ψ such that $(\boldsymbol{\mu}^{\boldsymbol{z}}_t, \boldsymbol{\Sigma}^{\boldsymbol{z}}_t) = g_\psi(\boldsymbol{x}_{t-1}; \boldsymbol{h}^\psi_{t-1})$, where $\boldsymbol{h}^\psi_{t-1}$ denotes the hidden state. Similarly, we assume the posterior $q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t})$ is normally-distributed: $\mathcal{N}(\boldsymbol{\mu}^q_t, \boldsymbol{\Sigma}^q_t)$, and is implemented using an LSTM $l_\phi(\boldsymbol{x}_t; \boldsymbol{h}^\phi_t)$ with weights ϕ and hidden state $\boldsymbol{h}^\phi_t$. This leads us to the following sampling pipeline:

$$\hat{\boldsymbol{x}}_t = f_\theta(\boldsymbol{x}_{t-1}, \boldsymbol{z}_t; \boldsymbol{h}^\theta_{t-1}) \sim \mathcal{N}(\boldsymbol{x}_t, \frac{1}{b_t}), \quad (5)$$

$$\boldsymbol{z}_t|\boldsymbol{x}_{1:t-1} \sim \mathcal{N}(\mu^{\boldsymbol{z}}_t, \boldsymbol{\Sigma}^{\boldsymbol{z}}_t)\ ;\ (\mu^{\boldsymbol{z}}_t, \boldsymbol{\Sigma}^{\boldsymbol{z}}_t) = g_\psi(\boldsymbol{x}_{t-1}; \boldsymbol{h}^\psi_{t-1}), \quad (6)$$

$$\boldsymbol{z}_t|\boldsymbol{x}_{1:t} \sim \mathcal{N}(\mu^q_t, \boldsymbol{\Sigma}^q_t)\ ;\ (\mu^q_t, \boldsymbol{\Sigma}^q_t) = l_\phi(\boldsymbol{x}_t; \boldsymbol{h}^\psi_t), \quad (7)$$

Using this setup, we are now ready to present our hierarchical, generative, variational model for uncertainty estimation, an overview of which is shown in Figure 2. To set the stage, let us assume the precision is sampled from the distribution $p(b_t|\boldsymbol{x}_{1:t-1})$. Then, we can rewrite the log-likelihood in Eq. 1 by including the precision distribution as:

$$\int_{b_t, z_t} \log p(x_t|b_t, z_t, x_{1:t-1}) + \log p(z_t|x_{1:t-1}) + \log p(b_t|x_{1:t-1})\, db_t\, dz_t \quad (8)$$

The above integral is intractable. Hence, we approximate it by sampling b_t and z_t. Note that the first two terms taken together is essentially the left-hand side of Eq. 1, except with the additional conditioning on b_t. Using the variational lower bound [33], like in Eq. 1, we have :

$$\log p(\boldsymbol{x}_t|b_t, \boldsymbol{x}_{1:t-1}) \geq \mathbb{E}_{q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t})} \log p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}, \boldsymbol{z}_t, b_t) - \mathrm{KL}\left(q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t}) \| p_\psi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t-1}\right), \text{for } t > F. \quad (9)$$

Please see the supplementary for the derivation. As stated before, we seek to connect the uncertainty $\boldsymbol{\Sigma}^{\boldsymbol{z}}_t$ in the latent

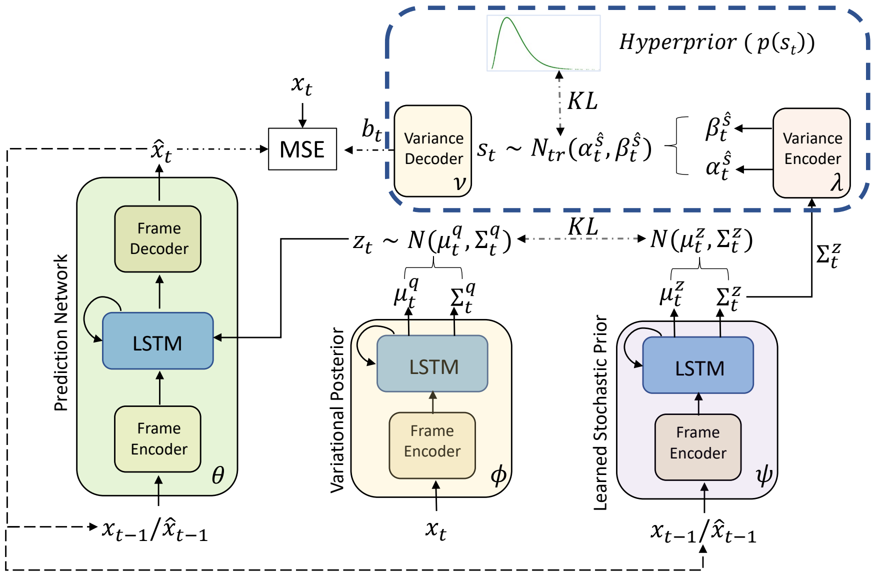


Figure 2. Overview of our approach.

prior $p_\psi(z_t|x_{1:t-1})$ to the precision b_t. We accomplish this via a variational encoder-decoder network. Such a formulation permits the flexibility of introducing customized prior distributions on its latent space. During training, the encoder component of this network, $\zeta_\lambda(\cdot)$, with parameters λ, takes as input $\mathbf{\Sigma}_t^{\boldsymbol{z}}$, and produces the sufficient statistics of the posterior distribution $q_\lambda(\cdot)$ governing the latent space of this network, while the decoder $\tau_\nu(\cdot)$, with parameters ν draws a sample s_t, from this distribution, and decodes it to generate b_t, with a distribution on b_t denoted by $p_\nu(\cdot)$. This sampling scheme is described as follows:

$$b_t \sim p_\nu(b_t|s_t, \boldsymbol{x}_{1:t-1}) = p_\nu(b_t|s_t, \Sigma_t^z), \quad s_t \sim q_\lambda(s_t|\boldsymbol{x}_{1:t-1}) = q_\lambda(s_t|\Sigma_t^z) \tag{10}$$

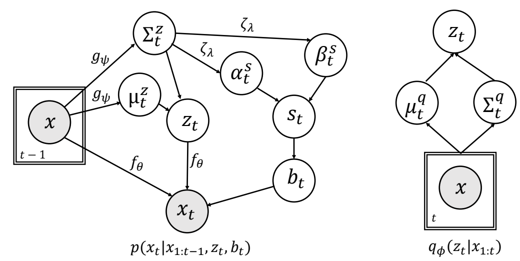
In order to provide appropriate regularization for the latent space distribution, $q_\lambda(\cdot)$, we assume a manually-defined hyper-prior distribution governing the latent space of this module denoted $p(s_t)$. Let the hyper prior $p(s_t) \sim D_\gamma(\alpha_t^s, \beta_t^s)$, with parameters α_t^s, β_t^s chosen by the user and let $q_\lambda(s_t|\Sigma_t^z) \sim D_\lambda(\alpha_t^{\hat{s}}, \beta_t^{\hat{s}})$, where the parameters $\alpha_t^{\hat{s}}, \beta_t^{\hat{s}}$ are estimated by the encoder network $\zeta_\lambda(\cdot)$. With this setup, analogous to Eq. 2, we obtain the following variational lower bound on the likelihood of b_t:

$$\log p(b_t|\boldsymbol{x}_{1:t-1}) \geq \mathbb{E}_{q_\lambda(s_t|\Sigma_t^z)} \log p_\nu(b_t|s_t, \Sigma_t^z) - \mathrm{KL}\left(q_\lambda(s_t|\Sigma_t^z) \parallel p(s_t)\right), \text{for } t > F \tag{11}$$

Please see the supplementary for the derivation. Plugging Eq. 9 and Eq. 11 in Eq. 8, we have the following:

$$\mathbb{E}_{q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t})} \log p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}, \boldsymbol{z}_t, b_t) - \mathrm{KL}\left(q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t}) \| p_\psi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t-1})\right) + \mathbb{E}_{q_\lambda(s_t|\boldsymbol{x}_{1:t-1})} \log p_\nu(b_t|s_t, \Sigma_t^z) - \mathrm{KL}\left(q_\lambda(s_t|\Sigma_t^z) \| p(s_t)\right) \tag{12}$$

Assuming that $p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}, \boldsymbol{z}_t, b_t)$ follows a Gaussian distribution $\mathcal{N}(\boldsymbol{x}_t, \frac{1}{b_t})$, along the lines of Eq. 4, leads us to our final ELBO objective, which we minimize:



(a) Uncertainty and learned prior (b) Posterior

Figure 3. Plate diagrams depicting the graphical model of our NUQ-framework. (a) shows the sampling dependencies between the learned prior and the uncertainty prediction modules, while (b) shows our posterior sampling framework. The plates denote, for example, $t-1$ repetitions of the random variable $\boldsymbol{x}$ in (a).

$$\mathcal{L}^P_{\theta,\phi,\psi,\lambda} = \sum_{t=F+1}^{T} \frac{1}{2}\left[b_t \left\|\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t\right\|^2 - \log b_t\right] - \mathbb{E}_{q_\lambda(s_t|\Sigma_t^z)} \log p_\nu(b_t|s_t, \Sigma_t^z) + \eta_1 \mathrm{KL}(q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t}) \| p_\psi(\boldsymbol{z}_t|\boldsymbol{x}_{1:t-1})) + \eta_2 \mathrm{KL}\left(q_\lambda(s_t|\Sigma_t^z) \| p(s_t)\right) \tag{13}$$

where $\eta_1, \eta_2 \geq 0$ are regularization constants (as suggested in Higgins *et al.* [25]).

Given our setup, a natural choice in a hierarchical Bayesian conjugate sense is to assume $p(s_t) \sim \Gamma(\alpha_t^s, \beta_t^s)$, the gamma distribution, which is a conjugate prior for the precision. Unfortunately, however using the gamma distribution for the posterior does not permit the reparametrization trick [34, 33], which is essential for making sampling-based networks differentiable. While one may resort to approximations to the gamma prior such as using implicit gradients [16] or generalized re-parameterization techniques [46], these approaches can be computationally expensive. Instead, we propose to approximate it by a truncated normal distribution $\mathcal{N}_{tr}(\alpha_t^{\hat{s}}, \beta_t^{\hat{s}})$ (which is amenable to re-parametrization), where now $\alpha_t^{\hat{s}}(\geq 0)$ and $\beta_t^{\hat{s}}$ correspond to the mean and the standard deviation of the truncated normal, respectively and are estimated by the encoder network $\zeta_\lambda(\cdot)$. In practice, the truncation is effected through rejection sampling [45]; i.e., we sample from a normal distribution, and reject samples if they are negative. Empirically, we find this choice of the hyper prior (being a gamma distribution) and our truncated-normal posterior combination to be beneficial. Thus, the KL divergence in Eq. 11 will eventually promote the network $\zeta_\lambda(\cdot)$ to produce the sufficient statistics of a truncated-normal distribution which will closely approximate the true gamma hyper-prior governing $p(s_t)$. Additionally, since s_t is a scalar ($\in \mathbb{R}$), thus b_t ($= \frac{1}{s_t}$) is directly sampled from $q_\lambda(\cdot)$ rather than through the decoder network, $\tau_\nu(\cdot)$. Figure 3 presents a plate diagram of our proposed hierarchical graphical model.

4.2. Sequence Discriminator

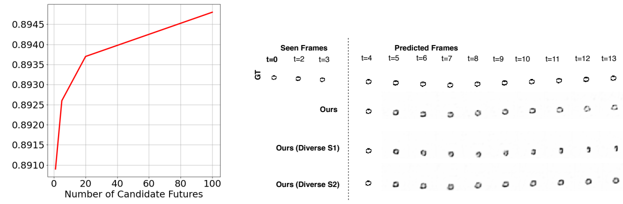
Inspired by the success of generative adversarial networks in generating realistic images and realistic object motions [20, 20, 22, 4, 39, 37], as well as the synergy that GANs bring about in improving the quality of other generative models [10, 23, 3], we propose to integrate NUQ with a sequence discriminator, where the generated frame sequences are input to a discriminator that checks for their realism and motion coherence. Different from prior approaches that employ GANs for future frame prediction [39, 37], our discriminator ($D_{\boldsymbol{w}}$) is a recurrent neural network with weights $\boldsymbol{w}$. It takes as input k contiguous frames with image dimensions $\delta_h \times \delta_w$, and produces a non-negative score, denoting the probability of that sequence being real or fake. Thus, $D_{\boldsymbol{w}} : \mathbb{R}^{\delta_h \times \delta_w \times k} \to \mathbb{R}_+$. Suppose, $\boldsymbol{y}_{1:k} \subset \boldsymbol{x}_{1:T}$ represents k contiguous frames starting at a random time step from video sequences $\boldsymbol{x}_{1:T}$ in the training dataset $\mathcal{X}$. If $\hat{\boldsymbol{x}}_{t-k+1:t}$ represents a sequence of k generated frames, then our discriminator loss is given by:

$$\mathcal{L}_{\boldsymbol{w}}^{D} := -\sum_{t=F+1}^{T} \mathbb{E}_{\boldsymbol{y}_{1:k} \sim \mathcal{X}} \log\left[D_{\boldsymbol{w}}(\boldsymbol{y}_{1:k})\right] + \mathbb{E}_{\hat{\boldsymbol{x}}_{t-k+1:t} \sim p(\hat{\boldsymbol{x}}_{t-k+1:t} | \boldsymbol{x}_{1:t-k}, \boldsymbol{z}_{1:t})} \log\left[1 - D_{\boldsymbol{w}}(\hat{\boldsymbol{x}}_{t-k+1:t})\right], \quad (14)$$

where, the discriminator is trained to distinguish between the generated (with label zero) and real (with label one) input sequences, by minimizing $\mathcal{L}_{\boldsymbol{w}}^{D}$, while the generator tries to maximize it. Combining the ELBO loss in Eq. 13 with the generator loss, we have our modified training loss for this variant, given by $\mathcal{L} = \mathcal{L}_{\theta,\phi,\psi,\lambda}^{P} - \gamma \mathcal{L}_{\boldsymbol{w}}^{D}$, where $\gamma > 0$ is a small regularization constant. For both variants (Eq. 13 or Eq. 14), we optimize the final objective using ADAM [32].

5. Experiments

In this section, we empirically validate the efficacy of NUQ on challenging real-world and synthetic datasets.



(a) SSIM Measure (b) Qualitative

Figure 4. Diversity Results: SSIM score with increasing number of generated futures, per time step, on the SMMNIST test set, as a quantification of output diversity (left) and qualitative generation results (right) using NUQ trained with 2000 training samples.

Datasets: We conduct experiments on three standard stochastic video prediction datasets, namely (i) Stochastic Moving MNIST (SMMNIST) [13, 10], (ii) BAIR Robot Push [15], and (iii) KTH-Action [47]. In SMMNIST, a handwritten digit moves in rectilinear paths within a 48×48 pixel box, bouncing off its walls, where the post bounce movement directions are stochastic. The dataset has a test set size of 1,000 videos [10]. The BAIR Push Dataset [15] consists of 64×64 pixel videos featuring highly stochastic motions of a Sawyer robotic arm pushing objects on a table. This dataset has 257 test samples using the split of Denton and Fergus [13]. Finally, the KTH Action Dataset [47] is a small dataset of 64×64 pixel videos containing a human performing various actions (walking, jogging, etc.), captured in a controlled setting with a static camera. The test set for this dataset consists of 476 videos. We hypothesize that by incorporating the predictive uncertainty, NUQ undergoes more efficient training updates and can thus train with fewer samples, efficiently. We therefore conduct experiments with varying training set sizes for some of these datasets.

Baselines and Evaluation Metrics: To carefully evaluate the performance improvement brought about by incorporating our uncertainty estimation method into a stochastic video generation framework, we choose three competitive and closely-related state-of-the-art methods within the stochastic video prediction realm as baselines, namely: (i) Denton and Fergus [13], (ii) Castrejon *et al*. [8], and (ii) Hsieh *et al*. [27]. At test time, we follow the standard protocol of generating 100 sequences for all models and report performances on sequences that matches best with the ground truth [13]. To quantify the generation quality, we use standard evaluation metrics: (i) per-frame Structural Similarity (SSIM) ([9]), (ii) Peak Signal to Noise Ratio (PSNR), and (iii) Learned Perceptual Image Patch Similarity (LPIPS) [63] - with a VGG backbone. We report the average scores on these metrics across all predicted frames.

Experimental Setup: For all three datasets SMMNIST, BAIR Push, and KTH Action, we train all methods with 5 seen and 15 unseen frames, while at test time 20 frames are predicted after the first 5 seen ones. For the baseline methods [13, 8, 27], we use the publicly available implementations from the authors. To ensure our proposed NUQ-

Table 1. Best SSIM, PSNR, and LPIPS scores on the test set for different datasets after @1, @5, and @Convergence (C) (upto 150 epochs) epochs of training with varying training set sizes. [Key: Best results in **bold** and second-best in blue.]

Dataset: SMMNIST	SSIM ↑			PSNR ↑			LPIPS ↓		
	@1	@5	@C	@1	@5	@C	@1	@5	@C
Number of training samples - 2,000									
Ours	**0.8686**	**0.8638**	**0.8948**	17.76	**18.13**	18.14	**0.3087**	**0.2836**	**0.1803**
Ours (w/o discriminator)	0.8599	0.8825	0.8929	**17.82**	18.07	**18.48**	0.3283	0.3158	0.1967
Denton and Fergus [13]	0.8145	0.8650	0.8696	17.07	18.05	18.13	0.3429	0.3345	0.2321
Castrejon *et al.* [8]	0.8564	0.8748	0.8868	17.36	17.98	18.12	0.3392	0.3432	0.2262
Hsieh *et al.* [27]	0.4538	0.8419	0.8569	11.27	16.40	16.70	0.4370	0.3696	0.2842
Number of training samples - 8,000 (full training set)									
	@1	@5	@ C	@1	@5	@ C	@1	@5	@C
Ours	**0.8524**	**0.8610**	**0.9088**	**17.93**	18.14	**19.07**	**0.3787**	**0.3013**	0.1149
Denton and Fergus [13]	0.8154	0.8607	0.8819	17.49	**18.22**	18.30	0.4061	0.3626	0.2813
Castrejon *et al.* [8]	0.8640	0.8708	0.8868	17.23	18.06	18.27	0.3939	0.3316	**0.1040**
Hsieh *et al.* [27]	0.5328	0.8374	0.8801	11.46	16.65	16.70	0.4217	0.4039	0.2747
Dataset: BAIR Push	**SSIM ↑**			**PSNR ↑**			**LPIPS ↓**		
	@1	@5	@C	@1	@5	@C	@1	@5	@C
Number of training samples - 2,000									
Ours	**0.7709**	**0.8230**	**0.8314**	**18.40**	**19.15**	**19.26**	**0.3394**	**0.2014**	**0.1574**
Denton and Fergus [13]	0.7351	0.7853	0.8196	17.32	17.44	18.49	0.3531	0.3197	0.1725
Castrejon *et al.* [8]	0.7094	0.7961	0.8221	17.19	17.92	18.79	0.3433	0.2560	0.1742
Hsieh *et al.* [27]	0.4979	0.7901	0.7989	11.32	15.28	16.00	0.4159	0.3899	0.1891
Number of training samples - 43,264 (full training set)									
	@1	@5	@ C	@1	@5	@ C	@1	@5	@C
Ours	**0.8135**	**0.8336**	**0.8460**	**19.03**	**19.14**	19.31	**0.1656**	**0.1470**	0.1296
Denton and Fergus [13]	0.7782	0.8198	0.8328	18.30	18.38	18.81	0.2119	0.1843	0.1499
Castrejon *et al.* [8]	0.7816	0.8309	0.8437	18.29	18.56	**19.59**	0.1878	0.1720	**0.1181**
Hsieh *et al.* [27]	0.7507	0.8123	0.8323	16.52	16.61	16.61	0.2140	0.1829	0.1713
Dataset: KTH Action	**SSIM ↑**			**PSNR ↑**			**LPIPS ↓**		
	@1	@5	@C	@1	@5	@C	@1	@5	@C
Number of training samples - 1,911 (full training set)									
Ours	**0.7990**	**0.8192**	0.8448	**22.62**	22.89	**24.02**	**0.4309**	**0.3390**	**0.2238**
Denton and Fergus [13]	0.7028	0.8056	0.8374	21.29	**22.93**	24.73	0.4621	0.3580	0.2497
Castrejon *et al.* [8]	0.6345	0.8054	**0.8510**	21.31	21.12	24.82	0.4513	0.3471	0.2395
Hsieh *et al.* [27]	0.4647	0.5335	0.7057	11.25	12.32	16.44	0.5189	0.3939	0.2771

Table 2. Best SSIM, PSNR, and LPIPS scores on the SMMNIST test set after @1, @5, and @Convergence (C) (upto 150 epochs) epochs of training with alternative formulations of our model using 2,000 training samples. [Key: Best results in **bold**].

Dataset: SMMNIST	SSIM ↑			PSNR ↑			LPIPS ↓		
	@1	@5	@C	@1	@5	@C	@1	@5	@C
$p(s_t) \sim$ Uniform$[0, 1]$	0.8173	0.8374	0.8523	17.6	17.95	18.06	0.3442	0.3038	0.198
Estimate b_t from the decoder $p_\theta(\cdot)$	0.7627	0.7628	0.7828	17.54	17.55	17.55	0.3463	0.3259	0.2225
Estimate b_t w/o variance encoder-decoder	0.7450	0.7454	0.7648	16.22	16.53	16.78	0.3469	0.3263	0.2328
NUQ (Ours)	**0.8686**	**0.8638**	**0.8948**	**17.76**	**18.13**	**18.14**	**0.3087**	**0.2836**	**0.1803**

Table 3. Human preference scores for samples generated using NUQ versus competing baselines across different datasets,

Dataset (# Training Samples)	Prefer: Ours/ [8]/ [13]
SMMNIST (2000 samples)	**89%** / 11% / 0 %
BAIR Push (2000 samples)	**78%** / 22% / 0 %
KTH-Action (1911 samples)	**78%** / 11% / 11 %

framework is similar in learning capacity, we use the "DC-GAN encoder-decoder" architecture for the frames, as in Denton and Fergus [13], for all datasets. Our variance encoder network $\zeta_\lambda(\cdot)$ is a multi-layer perceptron with 2 layers, and our sequence discriminator is an LSTM with a single 256-d hidden layer. More architectural details are in the Supplementary Materials. We set $k = 3$ in Eq. 14, and γ to be 0.00001. We use $\eta_1 = 0.0001$ and $\eta_2 = 0.001$ in Eq. 13. Learning rate is set to 0.002 and no scheduling is used. We observed a Gaussian hyper-prior with mean $(\alpha_t^s) = 0.02$, and variance $(\beta_t^s) = 0.12$ to be just as effective. Additionally, thresholding b_t to be less or equal 0.17 helped. The

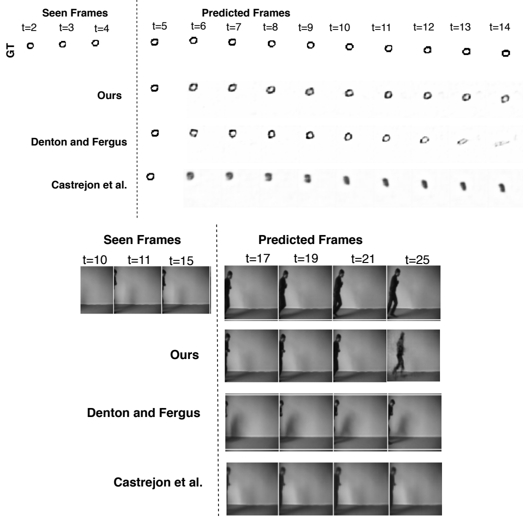


Figure 5. Qualitative results of NUQ against competing baselines on SMMNIST (top) and on KTH-Action (bottom).

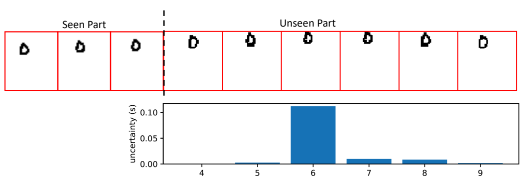


Figure 6. Evolution of scaled uncertainty on the SMMNIST dataset (trained with 2,000 training samples) against time-steps.

hyper-parameters for the baseline methods are chosen using a small validation set ($\sim 5\%$ of training data).

State-of-the-Art Comparisons: In Table 1, we report quantitative evaluations of our model versus competing baselines across the three datasets. We observe that both variants of the NUQ outperform recent competitive baselines by wide margins on all measures (upto 10% in SSIM), with the one with the discriminator being slightly better - underscoring the benefits of adversarial training. While noticeable gains are obtained across the board, NUQ really shines under limited training set sizes. We surmise that this gain is attributable to the failure of the baseline methods in incorporating predictive uncertainty explicitly into the learning objective. From the table, we also see that NUQ converges faster than other methods both for small and large training set sizes. Figures 1 and 5 show sample generation results from SMMNIST, BAIR Push, and KTH-Action datasets versus competing baselines, trained with 2,000 and 1,911 samples, respectively. From these figures, we see that compared to baseline methods, frames generated by our method are superior at capturing both the appearance and the motion of the object (i.e. digit/robot arm/human) even under limited training data. Human annotators, when presented with a few random sample generations by NUQ versus competing methods, overwhelmingly choose NUQ samples for their realism, as shown in Table 3. Figure 6 shows the evolution of a scaled uncertainty estimate derived from $\frac{1}{b_t}$ over different frames. The plot shows the increase in uncertainty co-occurs with the bounce of the digit against the boundary, suggesting that the uncertainty is well grounded in the data.

Alternative Formulations: Next, we discuss the results of some alternative formulations of our model. We consider: (i) replacing the gamma hyperprior on $p(s_t)$ with $\mathrm{Uniform}(0,1)$ distribution, (ii) estimating b_t from the frame decoder $p_\theta(\cdot)$ by producing a diagonal covariance matrix, and (iii) assuming a deterministic mapping from $\mathbf{\Sigma}_t^{\mathbf{z}}$ to b_t through a multi-layer perceptron. Table 2 presents the performance of these alternatives on the SMMNIST dataset. From the first row of the table, we see that choosing the $\mathrm{Uniform}(0,1)$ as priors results in suboptimal variants of NUQ. Further, the results also show that either estimating b_t directly from the decoder $p_\theta(\cdot)$, or computing it deterministically from $\mathbf{\Sigma}_t^{\mathbf{z}}$ performs poorly, suggesting that such estimation techniques are not ideal.

Diversity: In Figure 4 (a), we plot the average SSIM of NUQ for a set of futures, with increasing cardinality of this set. Our plot shows that the SSIM increases with larger number of futures, suggesting that the possibility of matching with a ground truth future increases with more futures, implying better diversity of our model. Figure 4(b) presents diverse generation results on the SMMNIST dataset by NUQ.

6. Conclusions

Recent approaches have demonstrated the need for modeling data uncertainty in video prediction models. However, in this paper we show that the state of the art in such stochastic methods do not leverage the model's predictive uncertainty to the fullest extent. Indeed, we show that explicitly incorporating this uncertainty into the learning objective via our proposed Neural Uncertainty Quantifier (NUQ) framework, can lead to faster and more effective model training even with fewer training samples, as validated by our experiments.

Acknowledgements. The support of the Office of Naval Research under grant N00014- 20-1-2444, and USDA National Institute of Food and Agriculture under grant 2020-67021-32799/1024178 are gratefully acknowledged.

References

[1] Lynton Ardizzone, Radek Mackowiak, Carsten Rother, and Ullrich Köthe. Training normalizing flows with the information bottleneck for competitive generative classification. *Advances in Neural Information Processing Systems*, 33, 2020. 3

[2] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. In *International Conference on Learning Representations*, 2018. 1, 2, 3

[3] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Cvae-gan: fine-grained image generation through asymmetric training. In *Proceedings of the IEEE international conference on computer vision*, pages 2745–2754, 2017. 6

[4] Emad Barsoum, John Kender, and Zicheng Liu. Hp-gan: Probabilistic 3d human motion prediction via gan. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1418–1427, 2018. 6

[5] Justin Bayer and Christian Osendorfer. Learning stochastic recurrent networks. *arXiv preprint arXiv:1411.7610*, 2014. 2

[6] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pages 1613–1622, 2015. 3

[7] 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)*, pages 753–769, 2018. 2

[8] Lluis Castrejon, Nicolas Ballas, and Aaron Courville. Improved conditional vrnns for video prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 1, 3, 6, 7

[9] Sumohana S Channappayya, Alan Conrad Bovik, and Robert W Heath Jr. Rate bounds on ssim index of quantized images. *IEEE Transactions on Image Processing*, 17(9):1624–1639, 2008. 6

[10] Anoop Cherian, Moitreya Chatterjee, and Narendra Ahuja. Sound2sight: Generating visual dynamics from sound and context. In *European Conference on Computer Vision (ECCV)*, 2020. 6

[11] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pages 2980–2988, 2015. 2

[12] William R Clements, Benoît-Marie Robaglia, Bastien Van Delft, Reda Bahi Slaoui, and Sébastien Toth. Estimating risk and uncertainty in deep reinforcement learning. *arXiv preprint arXiv:1905.09638*, 2019. 3

[13] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *International Conference on Machine Learning*, pages 1174–1183, 2018. 1, 2, 3, 4, 6, 7

[14] Emily L Denton et al. Unsupervised learning of disentangled representations from video. In *Advances in neural information processing systems*, pages 4414–4423, 2017. 3

[15] Frederik Ebert, Chelsea Finn, Alex X Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. *arXiv preprint arXiv:1710.05268*, 2017. 1, 2, 6

[16] Mikhail Figurnov, Shakir Mohamed, and Andriy Mnih. Implicit reparameterization gradients. In *Advances in Neural Information Processing Systems*, pages 441–452, 2018. 5

[17] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in neural information processing systems*, pages 64–72, 2016. 2

[18] Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. Sequential neural models with stochastic layers. In *Advances in neural information processing systems*, pages 2199–2207, 2016. 2

[19] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4346–4354, 2015. 1, 2

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1, 2, 6

[21] Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11474–11484, 2020. 3

[22] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017. 6

[23] Shir Gur, Sagie Benaim, and Lior Wolf. Hierarchical patch vae-gan: Generating diverse videos from a single sample. *Advances in Neural Information Processing Systems*, 33, 2020. 6

[24] Ali Harakeh, Michael Smart, and Steven L Waslander. Bayesod: A bayesian approach for uncertainty estimation in deep object detectors. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 87–93. IEEE, 2020. 3

[25] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2(5):6, 2017. 5

[26] Yung-Han Ho, Chuan-Yuan Cho, Wen-Hsiao Peng, and Guo-Lun Jin. Sme-net: Sparse motion estimation for parametric video prediction through reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10462–10470, 2019. 3

[27] Jun-Ting Hsieh, Bingbin Liu, De-An Huang, Li F Fei-Fei, and Juan Carlos Niebles. Learning to decompose and disentangle representations for video prediction. In *Advances in Neural Information Processing Systems*, pages 517–526, 2018. 3, 6, 7

[28] Ashesh Jain, Hema S Koppula, Bharad Raghavan, Shane Soh, and Ashutosh Saxena. Car that knows before you do: Anticipating maneuvers via learning temporal driving models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3182–3190, 2015. 1, 2

[29] Beibei Jin, Yu Hu, Qiankun Tang, Jingyu Niu, Zhiping Shi, Yinhe Han, and Xiaowei Li. Exploring spatial-temporal multi-frequency analysis for high-fidelity and temporal-consistency video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4554–4563, 2020. 1, 2, 3

[30] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 1

[31] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584, 2017. 2

[32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2015. 6

[33] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. 2014. 3, 4, 5

[34] David A Knowles. Stochastic gradient variational bayes for gamma approximating distributions. *arXiv preprint arXiv:1509.01631*, 2015. 5

[35] Rahul G Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In *Thirty-first aaai conference on artificial intelligence*, 2017. 2

[36] Abhinav Kumar, Tim K Marks, Wenxuan Mou, Ye Wang, Michael Jones, Anoop Cherian, Toshiaki Koike-Akino, Xiaoming Liu, and Chen Feng. Luvli face alignment: Estimating landmarks' location, uncertainty, and visibility likelihood. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8236–8246, 2020. 3

[37] Yong-Hoon Kwon and Min-Gyu Park. Predicting future frames using retrospective cycle gan. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1811–1820, 2019. 1, 2, 6

[38] Minhyeok Lee and Junhee Seok. Estimation with uncertainty via conditional generative adversarial networks. *arXiv preprint arXiv:2007.00334*, 2020. 3

[39] Xiaodan Liang, Lisa Lee, Wei Dai, and Eric P Xing. Dual motion gan for future-flow embedded video prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1744–1752, 2017. 1, 2, 6

[40] Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future frame prediction for anomaly detection–a new baseline. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6536–6545, 2018. 3

[41] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. *International Conference on Learning Representations*, 2017. 2

[42] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems*, pages 7047–7058, 2018. 3

[43] Rowan McAllister, Gregory Kahn, Jeff Clune, and Sergey Levine. Robustness to out-of-distribution inputs via task-aware generative uncertainty. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2083–2089. IEEE, 2019. 3

[44] MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014. 2

[45] Christian P Robert. Simulation of truncated normal variables. *Statistics and computing*, 5(2):121–125, 1995. 6

[46] Francisco R Ruiz, Michalis Titsias RC AUEB, and David Blei. The generalized reparameterization gradient. In *Advances in neural information processing systems*, pages 460–468, 2016. 5

[47] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 32–36. IEEE, 2004. 2, 6

[48] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852, 2015. 2

[49] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Rahul Sukthankar, Kevin Murphy, and Cordelia Schmid. Relational action forecasting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 273–283, 2019. 1

[50] Natasa Tagasovska and David Lopez-Paz. Single-model uncertainties for deep learning. In *Advances in Neural Information Processing Systems*, pages 6417–6428, 2019. 3

[51] Jakub Tomczak and Max Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pages 1214–1223, 2018. 3

[52] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33, 2020. 3

[53] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. *Proceedings of the International Conference on Learning Representations*, 2017. 3

[54] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances in neural information processing systems*, pages 613–621, 2016. 1

[55] Chandrasekar Vuppalapati, Anitha Ilapakurti, Sharat Kedari, Rajasekar Vuppalapati, Jayashankar Vuppalapati, and Santosh Kedari. Human ai symbiosis: The role of artificial intelligence in stratifying high-risk outpatient senior citizen fall events in a non-connected environments. In *International Conference on Intelligent Human Systems Integration*, pages 325–332. Springer, 2020. 1, 2

[56] Jacob Walker, Kenneth Marino, Abhinav Gupta, and Martial Hebert. The pose knows: Video forecasting by generating pose futures. In *Proceedings of the IEEE international conference on computer vision*, pages 3332–3341, 2017. 1

[57] Gang Wang, Bo Li, Yongfei Zhang, and Jinhui Yang. Background modeling and referencing for moving cameras-captured surveillance video coding in hevc. *IEEE Transactions on Multimedia*, 20(11):2921–2934, 2018. 1

[58] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688, 2011. 3

[59] Yue Wu, Rongrong Gao, Jaesik Park, and Qifeng Chen. Future video synthesis with object motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5539–5548, 2020. 3

[60] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015. 2

[61] Yufei Ye, Maneesh Singh, Abhinav Gupta, and Shubham Tulsiani. Compositional video prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10353–10362, 2019. 3

[62] Wei Yu, Yichao Lu, Steve Easterbrook, and Sanja Fidler. Efficient and information-preserving future frame prediction and beyond. In *International Conference on Learning Representations*, 2019. 2

[63] 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*, pages 586–595, 2018. 6