A Hierarchical Variational Neural Uncertainty Model for Stochastic Video Prediction

Chatterjee, Moitreya; Ahuja, Narendra; Cherian, Anoop

TR2021-096 September 09, 2021

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 four 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.

IEEE International Conference on Computer Vision (ICCV) 2021
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 four 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 [58], human action recognition and forecasting [50, 56] 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.

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 [55, 57, 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 Babaiezadeh 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, regularizing the maneuvers in autonomous driving [28, 56]. 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 three challenging real world datasets: KTH-Action [47], BAIR push [15], and UCF-101 [48] 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, 49, 17]. Ranzato et al. [44] proposed to divide frames into patches and extrapolate their evolution in time. Srivastava et al. [49] use image encoders with pre-trained weights to encode the frames. ContextVP [7] and PredNet [41] leverage Convolutional LSTMs [61] for video prediction. Fragkiadaki et al. [19] proposes pose extrapolation using LSTMs. More recent approaches [37, 63] 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
work. 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 [52, 53]. 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 infersence.

Another line of work in frame prediction seeks to decouple the video into static and moving components [54, 14, 40, 26, 60, 21]. Some of these approaches are deterministic, others stochastic. Denton et al. [14] extracts content and pose information for this purpose. Villegas et al. [54] 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. [62] 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, 51]. 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, 59]. However, predictive uncertainty modeling in the context of frame prediction has remained largely unexplored. NUQ attempts to fill this gap.

### 3. Background

Suppose \( x_{1:T} := (x_1, x_2, \ldots, x_T) \) denotes a sequence of random variables, each \( x_t \) 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 \( x_{F+1:T} \) 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 \( \theta \), that minimizes the expected negative log-likelihood. When unknown factors of variation are involved in the data generation process, a deterministic 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 \( p(z_t) \); i.e., \( z_t \sim p(z_t), x_t \sim p_\theta(x_t|x_{1:t-1}, z_t) \). Specifically, the stochasticity in the generative process is characterized by the variance in \( p(z_t) \), that produces diversity in \( z_t \sim p(z_t) \). 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: \( p(x_t|x_{1:t-1}) = \int_{z_t} p_\theta(x_t|x_{1:t-1}, z_t)p(z_t)dz_t \). This problem is typically avoided by casting this estimation in an encoder-decoder setup, where the encoder embeds \( x_{1:t} \) as \( z_t \sim p(z_t|x_{1:t}) \), while the decoder outputs \( x_t \sim p_\theta(x_t|x_{1:t-1}, z_t) \). In order to train efficiently, access to a variational posterior \( q_\phi(z_t|x_{1:t}) \) — that approximates the true posterior \( p(z_t|x_{1:t}) \) of the encoder — is assumed. Using this approximate posterior, learning the model parameters \( \theta \) and \( \phi \) amounts to maximizing the variational lower bound, \( \mathcal{L}_{\theta,\phi} \) [33]:

\[
\log p(x_t|x_{1:t-1}) \geq \mathcal{L}_{\theta,\phi}, \text{ where}
\]

\[
\mathcal{L}_{\theta,\phi} := \int_{z_t} q_\phi(z_t|x_{1:t}) \log p_\theta(x_t|x_{1:t-1}, z_t)dz_t
\]

\[
- \int_{z_t} q_\phi(z_t|x_{1:t}) \log \frac{q_\phi(z_t|x_{1:t})}{p(z_t)} dz_t
\]

From the definition, this amounts to:

\[
\mathcal{L}_{\theta,\phi} = \mathbb{E}_{q_\phi(z_t|x_{1:t})} \log p_\theta(x_t|x_{1:t-1}, z_t) - \text{KL}(q_\phi(z_t|x_{1:t})||p(z_t)), \text{ for } t > F.
\]

Leveraging the re-parametrization trick ([33]) allows efficient optimization of the likelihood loss in Eq. 2, permitting us to learn the parameters \( \theta \) and \( \phi \). Note that the expectation term in Eq. 2 boils down to a standard MSE over all predicted frames \( x_{F+1:T} \) 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 \( x_t \) available to it as \( z_t \), 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 \( p(z_t) \) 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 \( p_\psi(z_t) = q_\psi(z_t) := q_\psi(z_t|x_{1:t-1}) \), parametrized by \( \psi \), 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 \( x_t \),
it can guide the prior (which does not have access to \(x_t\), but only to \(x_{1:t-1}\)) to produce a distribution on \(z_t\) that mimics the posterior (and hence we can discard the posterior at test time). Thus, the training-time sampling pipeline is given by:

\[
z_t \sim q_\psi(z_t|x_{1:t}), \quad x_t \sim p_\theta(x_t|x_{1:t-1}, z_t), \quad \text{and} \quad p_\psi \xrightarrow{d} q_\phi \quad \text{(matching in distribution)}. 
\]

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 during both 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, \(f_\theta\), with weights \(\theta\) such that:

\[
\hat{x}_t = f_\theta(x_{1:t-1}, z_t) := f_\theta(x_{t-1}, z_t; h_t^{\theta - 1}), 
\]

where \(\hat{x}_t\) denotes the \(t^{th}\) generated frame and \(h_t^{\theta - 1}\) captures the internal states of the LSTM. The generated frame \(\hat{x}_t\) 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 \(p_\theta(x_t|x_{1:t-1}, z_t) \sim \mathcal{N}(x_t, \frac{1}{b_t})\), 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 \(\hat{x}_t\) boils down to computing the \(\ell_2\)-loss. This reduces Eq. 2 to become the evidence lower bound (ELBO) [33]:

\[
\mathcal{L}_{\theta, \phi, \psi} := \sum_{t=F+1}^T \frac{1}{2} \| \hat{x}_t - x_t \|^2 + \text{KL}(q_\phi(z_t|x_{1:t}) \parallel p_\psi(z_t|x_{1:t-1})). 
\]

Our key insight to the proposed uncertainty measure arises from the observation that the \(\ell_2\)-norm term in Eq. 4 does not include any dependency on the uncertainty associated with the prediction of \(\hat{x}_t\). 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 \(\hat{x}_t\), however the prediction model is not trained well, such that \(\hat{x}_t\) does not match \(x_t\), and (ii) when there is uncertainty involved in the generative model such that the generated \(\hat{x}_t\), while plausible given the context, is different from \(x_t\). 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 \(p_\psi(z_t|x_{1:t-1})\), so that when the stochasticity is high for the generated frames, the \(\ell_2\)-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 \(p_\psi(z_t|x_{1:t-1})\) is a normal distribution \(\mathcal{N}(\mu_t^\psi, \Sigma_t^\psi)\), with parameters \(\mu_t^\psi\), the mean, and \(\Sigma_t^\psi\) 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 \(g_\phi\) with weights \(\psi\) such that \((\mu_t^\psi, \Sigma_t^\psi) = g_\phi(x_{t-1}; h_t^{\psi - 1})\), where \(h_t^{\psi - 1}\) denotes the hidden state. Similarly, we assume the posterior \(q_\phi(z_t|x_{1:t})\) is normally-distributed: \(\mathcal{N}(\mu_t^\phi, \Sigma_t^\phi)\), and is implemented using an LSTM \(l_\phi(x_t; h_t^\phi)\) with weights \(\phi\) and hidden state \(h_t^\phi\). This leads us to the following sampling pipeline:

\[
\hat{x}_t = f_\theta(x_{t-1}, z_t; h_t^{\theta - 1}) \sim \mathcal{N}(x_t, \frac{1}{b_t}), 
\]

\[
z_t|x_{1:t-1} \sim \mathcal{N}(\mu_t^\phi, \Sigma_t^\phi); \quad (\mu_t^\phi, \Sigma_t^\phi) = g_\phi(x_{t-1}; h_t^{\psi - 1}), 
\]

\[
z_t|x_{1:t} \sim \mathcal{N}(\mu_t^\phi, \Sigma_t^\phi); \quad (\mu_t^\phi, \Sigma_t^\phi) = l_\phi(x_t; h_t^\phi), 
\]

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 \(p(b_t|x_{1:t-1})\). Then, we can rewrite the log-likelihood in Eq. 1 by including the precision distribution as:

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

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:

\[
\log p(x_t|b_t, x_{1:t-1}) \geq \mathbb{E}_{q_\phi(z_t|x_{1:t})} \log p_\theta(x_t|x_{1:t-1}, z_t, b_t) - \text{KL}(q_\phi(z_t|x_{1:t}) \parallel p_\psi(z_t|x_{1:t-1})), \quad \text{for } t > F.
\]

Please see the supplementary for the derivation. As stated before, we seek to connect the uncertainty \(\Sigma_t^\phi\) in the latent
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, \( \psi(z_t|\cdot) \), with parameters \( \lambda \), takes as input \( \Sigma_t \), and produces the sufficient statistics of the posterior distribution \( q_\lambda(\cdot) \) governing the latent space of this network, while the decoder \( r_\nu(\cdot) \), with parameters \( \nu \) 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:

\[
\begin{align*}
  b_t &\sim p_\nu(b_t|s_t, x_{1:t-1}) = p_\nu(b_t|s_t, \Sigma_t), \\
  s_t &\sim q_\lambda(s_t|x_{1:t-1}) = q_\lambda(s_t|\Sigma_t) \\
\end{align*}
\]

(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 \( p(s_t) \sim D_N(\alpha_t^s, \beta_t^s) \), with parameters \( \alpha_t^s, \beta_t^s \) chosen by the user and let \( q_\lambda(s_t|\Sigma_t) \sim D_N(\alpha_t^s, \beta_t^s) \), where the parameters \( \alpha_t^s, \beta_t^s \) are estimated by the decoder network \( \lambda(\cdot) \). With this setup, analogous to Eq. 2, we obtain the following variational lower bound on the likelihood of \( b_t \):

\[
\log p(b_t|x_{1:t-1}) \geq \mathbb{E}_{q_\lambda(s_t|\Sigma_t)} \log p_\nu(b_t|s_t, \Sigma_t) - KL(q_\lambda(s_t|\Sigma_t) \parallel p(s_t)) , \text{ for } t > F
\]

(11)

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

\[
\begin{align*}
  \mathbb{E}_{q_\phi(z_t|x_{1:t-1})} \log p_\theta(z_t|x_{1:t-1}, z_{t-1}, b_t) - &\quad KL(q_\phi(z_t|x_{1:t-1}) || p_\psi(z_t|x_{1:t-1})) + \mathbb{E}_{q_\lambda(s_t|x_{1:t-1})} \log p_\nu(b_t|s_t, \Sigma_t) - KL(q_\lambda(s_t|\Sigma_t) || p_\lambda(\cdot))
\end{align*}
\]

(12)

Assuming that \( p_\theta(z_t|x_{1:t-1}, z_{t-1}, b_t) \) follows a Gaussian distribution \( \mathcal{N}(x_t, \Sigma_t) \), along the lines of Eq. 4, leads us to our final ELBO objective, which we minimize:

\[
\begin{align*}
  \mathcal{L}_{\theta, \phi, \psi, \lambda}^F = \sum_{t=F+1}^T \left[ b_t \| \hat{x}_t - x_t \|^2 - \log b_t \right] - &\quad \mathbb{E}_{q_\phi(z_t|x_{1:t-1})} \log p_\psi(z_t|x_{1:t-1}) + \eta_1 KL(q_\phi(z_t|x_{1:t-1}) || p_\psi(z_t|x_{1:t-1})) + \eta_2 KL(q_\lambda(s_t|\Sigma_t) || p_\lambda(\cdot))
\end{align*}
\]

(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 re-parameterization 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 distribu-

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 \( x \) in (a).
tion $\mathcal{N}_{t_r}(\alpha_s^t, \beta_s^t)$ (which is amenable to re-parametrization), where now $\alpha_s^t \geq 0$ and $\beta_s^t$ correspond to the mean and the standard deviation of the truncated normal, respectively and are estimated by the encoder network $\zeta_s(\cdot)$. 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 $\zeta_s(\cdot)$ 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, $\tau_r(\cdot)$. 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 ($D_w$) is a recurrent neural network with weights $w$. 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, $D_w : \mathbb{R}^{\delta_h \times \delta_w \times k} \rightarrow \mathbb{R}_+$. Suppose, $y_{1:k} \subset x_{1:T}$ represents $k$ contiguous frames starting at a random time step from video sequences $x_{1:T}$ in the training dataset $\mathcal{X}$. If $\hat{x}_{t-k+1:t}$ represents a sequence of $k$ generated frames, then our discriminator loss is given by:

$$
\mathcal{L}_w^D := - \sum_{t=F+1}^{T} \mathbb{E}_{y_{1:k} \sim \mathcal{X}} \log [D_w(y_{1:k})] + \quad (14)
$$

$$
\mathbb{E}_{x_{t-k+1:t} \sim p(\hat{x}_{t-k+1:t} | x_{1:t-k}, z_{1:t})} \log [1 - D_w(\hat{x}_{t-k+1:t})],
$$

where, the discriminator is trained to distinguish between the generated (with label zero) and real (with label one) input sequences, by minimizing $\mathcal{L}_w^D$ 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 $\mathcal{L} = \mathcal{L}_D^P + \lambda \mathcal{L}_w^D - \gamma \mathcal{L}_w^D$, 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.

---

**Datasets:** We conduct experiments on four standard stochastic video prediction datasets, namely (i) Stochastic Moving MNIST (SMMNIST) [13, 10], (ii) BAIR Robot Push [15], (iii) KTH-Action [47], and (iv) UCF-101 [48]. In SMMNIST, a hand-written digit moves in rectilinear paths within a $48 \times 48$ 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 $64 \times 64$ 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]. The KTH Action Dataset [47] is a small dataset of $64 \times 64$ pixel videos featuring 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. Finally, the UCF-101 Dataset [48] is a dataset of videos, resized to $64 \times 64$ containing 101 common human action categories (such as pushups, cricket shot, etc.), spanning both indoor and outdoor activities. The test set for this dataset consists of 1,895 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) [64] with a VGG backbone. We report the average scores on these metrics across all predicted frames.

**Experimental Setup:** For SMMNIST, BAIR Push, and
Table 1. 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_.]

<table>
<thead>
<tr>
<th>Dataset: SMMNIST</th>
<th>SSIM ↑</th>
<th>PSNR ↑</th>
<th>LPIPS ↓</th>
<th>Number of training samples - 2,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>0.8686</td>
<td>0.8638</td>
<td>0.8948</td>
<td>17.76 18.13 18.14 0.3087 0.2836 0.1803</td>
</tr>
<tr>
<td>(w/o discriminator)</td>
<td>0.8599</td>
<td>0.8582</td>
<td>0.8929</td>
<td>17.82 18.07 18.48 0.3283 0.3158 0.1967</td>
</tr>
<tr>
<td>Denton and Fergus [13]</td>
<td>0.8145</td>
<td>0.8650</td>
<td>0.8696</td>
<td>17.07 18.05 18.13 0.3429 0.3345 0.2321</td>
</tr>
<tr>
<td>Castrejon et al. [8]</td>
<td>0.8564</td>
<td>0.8748</td>
<td>0.8668</td>
<td>17.36 17.98 18.12 0.3392 0.3432 0.2262</td>
</tr>
<tr>
<td>Hsieh et al. [27]</td>
<td>0.4538</td>
<td>0.8419</td>
<td>0.8569</td>
<td>11.27 16.40 16.70 0.4370 0.3696 0.2842</td>
</tr>
</tbody>
</table>

| Ours (w/o discriminator) | 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 |

Table 2. 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**.

<table>
<thead>
<tr>
<th>Dataset: BAIR Push</th>
<th>SSIM ↑</th>
<th>PSNR ↑</th>
<th>LPIPS ↓</th>
<th>Number of training samples - 2,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>0.7709</td>
<td>0.8230</td>
<td>0.8314</td>
<td>18.40 19.15 19.26 0.3394 0.2014 0.1574</td>
</tr>
<tr>
<td>Denton and Fergus [13]</td>
<td>0.7351</td>
<td>0.7853</td>
<td>0.8196</td>
<td>17.32 17.44 18.49 0.3531 0.3197 0.1725</td>
</tr>
<tr>
<td>Castrejon et al. [8]</td>
<td>0.7094</td>
<td>0.7961</td>
<td>0.8221</td>
<td>17.19 17.92 18.79 0.3433 0.2560 0.1742</td>
</tr>
<tr>
<td>Hsieh et al. [27]</td>
<td>0.4979</td>
<td>0.7901</td>
<td>0.7989</td>
<td>11.32 15.28 16.00 0.4159 0.3899 0.1891</td>
</tr>
</tbody>
</table>

| 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 |

<table>
<thead>
<tr>
<th>Dataset: KTH Action</th>
<th>SSIM ↑</th>
<th>PSNR ↑</th>
<th>LPIPS ↓</th>
<th>Number of training samples - 1,911 (full training set)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>0.7990</td>
<td>0.8192</td>
<td>0.8448</td>
<td>22.62 22.89 24.02 0.4309 0.3390 0.2238</td>
</tr>
<tr>
<td>Denton and Fergus [13]</td>
<td>0.7028</td>
<td>0.8056</td>
<td>0.8374</td>
<td>21.29 22.93 24.73 0.4621 0.3580 0.2497</td>
</tr>
<tr>
<td>Castrejon et al. [8]</td>
<td>0.6345</td>
<td>0.8054</td>
<td>0.8510</td>
<td>21.31 21.12 24.82 0.4513 0.3471 0.2395</td>
</tr>
<tr>
<td>Hsieh et al. [27]</td>
<td>0.4647</td>
<td>0.5335</td>
<td>0.7057</td>
<td>11.25 12.32 16.44 0.5189 0.3939 0.2771</td>
</tr>
</tbody>
</table>

| Ours                | 0.7359 | 0.7636 | 0.7729 | 21.25 21.98 22.73 0.3914 0.2865 0.0836 |
| Denton and Fergus [13] | 0.6253 | 0.7540 | 0.7603 | 19.35 20.60 21.64 0.3507 0.3006 0.1259 |
| Castrejon et al. [8] | 0.6712 | 0.7555 | 0.7756 | 20.58 20.58 22.53 0.3414 0.2965 0.1036 |
| Hsieh et al. [27] | 0.6199 | 0.6800 | 0.7103 | 16.65 17.18 18.41 0.3989 0.3239 0.1771 |

<table>
<thead>
<tr>
<th>Dataset: UCF-101</th>
<th>SSIM ↑</th>
<th>PSNR ↑</th>
<th>LPIPS ↓</th>
<th>Number of training samples - 11,425 (full training set)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>0.8135</td>
<td>0.8336</td>
<td>0.8460</td>
<td>19.03 19.14 19.31 0.1656 0.1470 0.1296</td>
</tr>
<tr>
<td>Denton and Fergus [13]</td>
<td>0.7782</td>
<td>0.8198</td>
<td>0.8328</td>
<td>18.30 18.38 18.81 0.2119 0.1843 0.1499</td>
</tr>
<tr>
<td>Castrejon et al. [8]</td>
<td>0.7816</td>
<td>0.8309</td>
<td>0.8437</td>
<td>18.29 18.56 19.59 0.1878 0.1720 0.1181</td>
</tr>
<tr>
<td>Hsieh et al. [27]</td>
<td>0.7507</td>
<td>0.8123</td>
<td>0.8323</td>
<td>16.52 16.61 16.61 0.2140 0.1829 0.1713</td>
</tr>
</tbody>
</table>

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. When training with UCF-101, 15 seen and 10 unseen frames are used, while at test time the number of unseen frames is set to 15. For the baseline methods [13, 8, 27], we use the publicly available im-
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.

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

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SMMNIST (2000 samples)</td>
<td>89% / 11% / 0%</td>
</tr>
<tr>
<td>BAIR Push (2000 samples)</td>
<td>78% / 22% / 0%</td>
</tr>
<tr>
<td>KTH-Action (1911 samples)</td>
<td>78% / 11% / 11%</td>
</tr>
</tbody>
</table>

implementations from the authors. To ensure our proposed NUQ-framework is similar in learning capacity, we use the “DCGAN encoder-decoder” architecture for the frames, as in Denton and Fergus [13], for all datasets. Our variance encoder network $\phi_{\theta} (\cdot)$ 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 $\gamma$ 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. The 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 four 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 $\frac{1}{2}$ 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 Uniform$(0, 1)$ distribution, (ii) estimating $b_t$ from the frame decoder $p_d(\cdot)$ by producing a diagonal covariance matrix, and (iii) assuming a deterministic mapping from $\Sigma_t^f$ 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 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 $p_d(\cdot)$, or computing it deterministically from $\Sigma_t^f$ 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.
References


[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, 8


