Consistent Jumpy Predictions for Videos and Scenes

Ananya Kumar, S. M. Ali Eslami, Danilo Rezende, Marta Garnelo
Fabio Viola, Edward Lockhart, Murray Shanahan
DeepMind, London, UK
{ananyak, aeslami, danilor, garnelo, fviola, locked, mshanahan}@google.com

Abstract

Stochastic video prediction models take in a sequence of image frames, and generate a sequence of consecutive future image frames. For the most part, these models generate future frames in an autoregressive fashion. This is slow and requires the input and output to be consecutive. We introduce a model that overcomes these drawbacks. Our model learns to generate a latent representation from an arbitrary set of frames within a video. This representation can then be used to simultaneously and efficiently sample temporally consistent frames at arbitrary time-points in the video. For example, our model can "jump" and directly sample frames at the end of the video, without sampling intermediate frames. We apply our model to synthetic video datasets. On top of having greater functionality and speed, our model produces image frames of comparable quality to existing models. In addition, we demonstrate the flexibility of our model by applying it to a 3D scene reconstruction dataset with occlusion. To the best of our knowledge, our model is the first to provide flexible and coherent prediction on stochastic video datasets and stochastic 3D scenes. Please check the project website `https://bit.ly/2DkvUV3` to view scene reconstructions and videos produced by our model.

1 Introduction

The ability to fill in the gaps in high-dimensional data is a fundamental cognitive skill. Suppose you glance out of the window and see a person in uniform approaching your gate carrying a letter. You can easily imagine what will (probably) happen a few seconds later. The person will walk up the path and push the letter through your door. Now suppose you glance out of the window the following day and see a person in the same uniform walking down the path, away from the house. You can easily imagine what (probably) happened a few seconds earlier. The person came through your gate, walked up the path, and delivered a letter. Moreover you can visualize the scene from different viewpoints – you can imagine how things might look from the gate, or from the front door.

Replicating this ability is a significant challenge for artificial intelligence. To make this more precise, let's first consider the traditional video prediction setup. Video prediction is typically framed as a sequential forward prediction task. Given a sequence of frames $f_1, ..., f_t$, a model is tasked to generate future frames that follow, $f_{t+1}, ..., f_T$. This traditional setup is often limiting – in many cases, we are interested in what happens a few seconds after the input sequence. For example, in model based planning tasks, we might want to predict what frame f_T is, but might not care what intermediate frames $f_{t+1}, ..., f_{T-1}$ are. Existing models must still produce the intermediate frames, which is inefficient. Instead, our model, JUMP, is "jumpy" – it can directly sample frames in the future, bypassing intermediate frames. For example, in a 40-frame video, our model can sample the final frame 12 times faster than an autoregressive model like SV2P (Babaeizadeh et al., 2018).

More generally, existing forward video prediction models are not flexible at filling in gaps in data. Instead, our model can sample frames at arbitrary time points of a video, given a set of frames at arbitrary time points as context. So it can, for example, be used to infer backwards or interpolate

Preprint. Work in progress.

between frames, as easily as forwards. In our setup of “jumpy" video prediction, our model JUMP is given frames $f_1, ..., f_n$ from a single video along with the arbitrary time-points $t_1, ..., t_n$ at which those frames occurred. The model is then asked to sample plausible frames at arbitrary time points $t'_1, ..., t'_k$. In many cases there are multiple possible predicted frames given the context, making the problem stochastic. For example, a car moving towards an intersection could turn left or turn right. Although our model is not autoregressive, each set of k sampled frames is consistent with a single coherent possibility, while maintaining diversity across sets of samples. That is, in each sampled set all k sampled frames correspond to the car moving left, or all correspond to the car moving right.

Our method is not restricted to video prediction. When conditioned on camera position, our model can sample consistent sets of images for an occluded region of a scene, even if there are multiple possibilities for what that region might contain. To summarize, our key contributions are:

1. *We motivate and formulate the problem of jumpy stochastic video prediction*, where a model has to predict consistent target frames at arbitrary time points, given context frames at arbitrary time points. We observe connections with consistent stochastic scene reconstruction.
2. *We present a model for consistent jumpy predictions in videos and scenes.* Unlike existing video prediction models, our model consumes input frames and samples output frames entirely in parallel. It enforces consistency of the sampled frames by training on multiple correlated targets, sampling a global latent, and using a deterministic rendering network.
3. *We show strong experimental results for our model.* Unlike existing sequential video prediction models, our model can also do jumpy video predictions. We show that our model also produces images of similar quality while converging more reliably. Our model is not limited to video prediction – we develop a dataset for stochastic 3D scene reconstruction. Here, we show that our model significantly outperforms GQN.

2 Related Work

There has been a lot of work in stochastic video prediction. Recent works include (Babaeizadeh et al., 2018; Lee et al., 2018; Buesing et al., 2018). Unlike JUMP, these models still generate frames or states one time-step at a time and the input and output frames must be consecutive. Further, unlike prior methods, our models can be used for tasks like stochastic 3D scene reconstruction.

Scene reconstruction is also an active area of research. Traditional approaches include structure-from-motion, structure-from-depth, and multi view geometry techniques, which involve handcrafting the feature space in advance. GQN (Eslami et al., 2018) is a recent deep learning model used for spatial prediction. However, GQN was mainly used for deterministic scene reconstruction. In stochastic scene reconstruction, GQN is unable to capture correlated frames of occluded regions of a scene.

3 Methods

3.1 Problem Description

We consider problems where we have a collection of “scenes”. Scenes could be videos, spatial scenes, or in general any key-indexed collection. A scene $S^{(i)}$ consists of a collection of viewpoint-frame (key-value) pairs $(v_1^{(i)}, f_1^{(i)}), ..., (v_n^{(i)}, f_n^{(i)})$ where $v_j^{(i)}$ refers to the indexing ‘viewpoint’ information and $f_j^{(i)}$ to the frame. For videos the ‘viewpoints’ are timestamps. For spatial scenes the ‘viewpoints’ are camera positions and headings. Each scene S is split into a context and a target. The context C contains m viewpoint-frame pairs $C = \{(v_i, f_i)\}_{i=1}^{m}$. The target T contains the remaining $n - m$ viewpoints $V = \{v_i\}_{i=m+1}^{n}$ and corresponding target frames $F = \{f_i\}_{i=m+1}^{n}$. At evaluation time, the model receives the context C and target viewpoints V and should be able to sample possible values $\hat{F}$ corresponding to the viewpoints V.

3.2 Model (Generation)

We implement JUMP as a latent variable model. Our model can be summarized as follows, and is depicted in Figure 1.

$$r_j = M_\theta(v_j, f_j) \text{ for all } j \in 1, ..., m$$

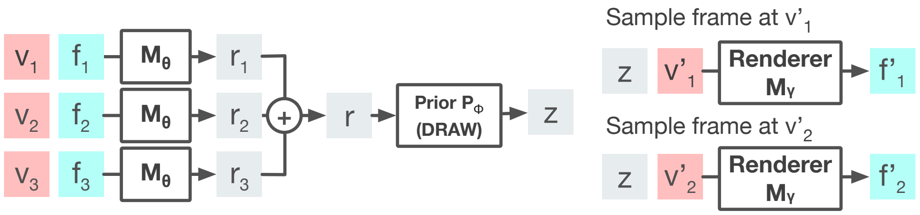


Figure 1: JUMP uses a prior P_ϕ to sample a latent z, conditioned on input frames and viewpoints. JUMP uses a rendering network M_γ to render z at arbitrary viewpoints, in parallel e.g. v'_1, v'_2.

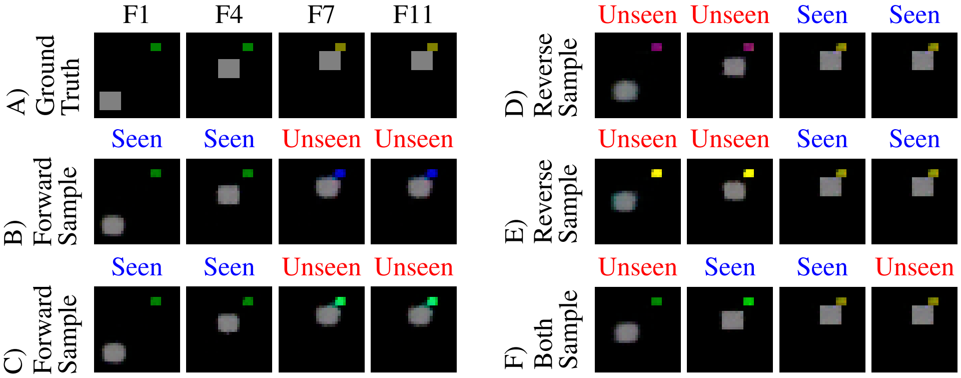


Figure 2: For each video in this dataset, shape 1 moves to shape 2 over frames 1 - 6, shape 2 changes color and stays that color from frames 7 - 12. Initial positions, shapes, sizes, colors, are randomized for each video. An example ground truth video is shown in sub-figure A. Our model gets 2 'seen' frames and samples 2 'unseen' samples. Our model is flexible with respect to input-output structure, and can roll a video forwards (Figures 2B, 2C), backwards (Figures 2D, 2E), or both ways (Figure 2F).

$$r = r_1 + ... + r_m$$
$$z \sim P_\phi(z|r)$$
$$\hat{f}_i = M_\gamma(z, v_i) \text{ for all } i \in m+1, ..., n$$

JUMP is trained using an approximate posterior that has access to a representation of multiple targets. We train the model by maximizing the ELBO lower bound, as in (Gregor et al., 2016).

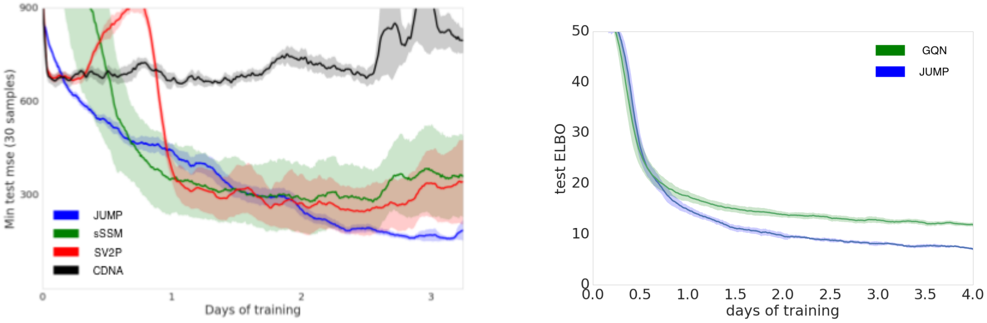
4 Experiments

We evaluate JUMP against a number of strong existing baselines on two tasks: a synthetic, combinatorial video prediction task and a 3D scene reconstruction task with occlusion.

Flexibility of JUMP: JUMP is more flexible than existing video prediction models. JUMP can take arbitrary sets of frames as input, and directly predict arbitrary sets of output frames (see Figure 2).

Quantitative Comparisons: We quantitatively compare the image frame quality in JUMP with sSSM (Buesing et al., 2018), SV2P (Babaeizadeh et al., 2018), and CDNA (Finn et al., 2016). The other models cannot do jumpy prediction – we compare these models when used for forward prediction on a synthetic dataset where one shape sequentially "visits" 4 other shapes. All the shapes, colors, sizes, positions are randomized. The plot shows that JUMP converges much more reliably. Averaged across 15 runs, our model performs better than sSSM, SV2P, and CDNA.

Scene Reconstruction: We develop a 3D dataset where each scene consists of a cube in a room. Each face of the cube has a random MNIST digit on it. The context frames show at most 3 sides of the dice, but the model may be asked to sample camera snapshots involving the unseen fourth side. Floor colors, wall colors, lighting direction, cube positions are random. Unlike JUMP, GQN samples each frame independently, and does not sample a coherent scene (Figure 4). JUMP reaches much better test ELBO values than GQN over 6 runs (see plot in Figure 3b, with 2 standard errors).



(a) JUMP converges more reliably to lower test errors than video prediction models, over 15 runs.

(b) JUMP consistently achieves lower (better) test ELBO scores than GQN, over 6 runs.

Figure 3: Quantitative comparisons between JUMP and video prediction and scene prediction models.

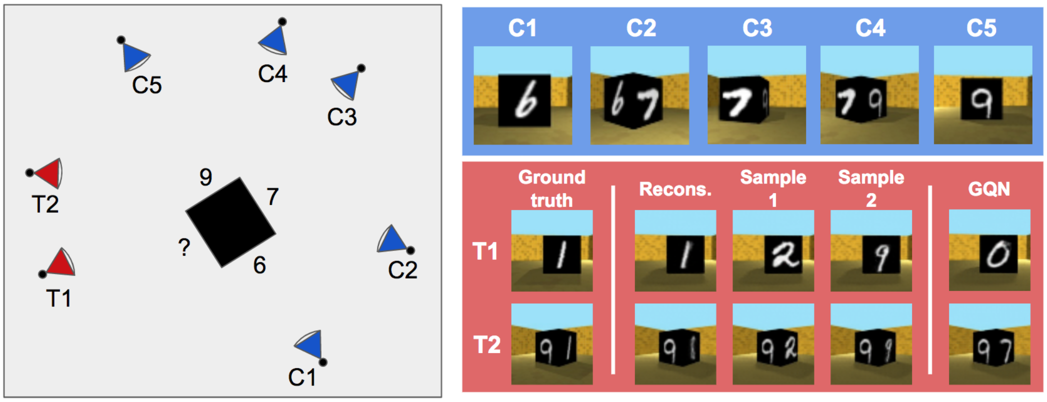


Figure 4: A cube in a room, with MNIST digits engraved on each face (test-set scene). The blue cones are where the context frames were captured from. The red cones are where the model is queried. The context frames see three sides of the cube, but the models are tasked to sample from the fourth, unseen, side. GQN (right column) independently samples a 0 and 7 for the unseen side, resulting in an inconsistent scene. JUMP samples a consistent digit (2 or 9) for the unseen cube face.

References

Babaeizadeh, M., Finn, C., Erhan, D., Campbell, R. H., and Levine, S. (2018). Stochastic variational video prediction. In *International Conference on Learning Representations*.

Buesing, L., Weber, T., Racanière, S., Eslami, S. M. A., Rezende, D. J., Reichert, D. P., Viola, F., Besse, F., Gregor, K., Hassabis, D., and Wierstra, D. (2018). Learning and querying fast generative models for reinforcement learning. *CoRR*, abs/1802.03006.

Eslami, S. M. A., Jimenez Rezende, D., Besse, F., Viola, F., Morcos, A. S., Garnelo, M., Ruderman, A., Rusu, A. A., Danihelka, I., Gregor, K., Reichert, D. P., Buesing, L., Weber, T., Vinyals, O., Rosenbaum, D., Rabinowitz, N., King, H., Hillier, C., Botvinick, M., Wierstra, D., Kavukcuoglu, K., and Hassabis, D. (2018). Neural scene representation and rendering. *Science*, 360(6394).

Finn, C., Goodfellow, I. J., and Levine, S. (2016). Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems*.

Gregor, K., Besse, F., Jimenez Rezende, D., Danihelka, I., and Wierstra, D. (2016). Towards conceptual compression. In *Advances in Neural Information Processing Systems 29*.

Lee, A. X., Zhang, R., Ebert, F., Abbeel, P., Finn, C., and Levine, S. (2018). Stochastic adversarial video prediction. *CoRR*, abs/1804.01523.