

# Motion Prediction Under Multimodality with Conditional Stochastic Networks

Katerina Fragkiadaki <sup>\*</sup>

katef@cs.cmu.edu

Jonathan Huang <sup>†</sup>

jonathanhuang@google.com

Alex Alemi <sup>†</sup>

alemi@google.com

Sudheendra Vijayanarasimhan <sup>†</sup>

svnaras@google.com

Susanna Ricco <sup>†</sup>

ricco@google.com

Rahul Sukthankar <sup>†</sup>

sakthankar@google.com

## Abstract

*Given a visual history, multiple future outcomes for a video scene are equally probable, in other words, the distribution of future outcomes has multiple modes. Multimodality is notoriously hard to handle by standard regressors or classifiers: the former regress to the mean and the latter discretize a continuous high dimensional output space. In this work, we present stochastic neural network architectures that handle such multimodality through stochasticity: future trajectories of objects, body joints or frames are represented as deep, non-linear transformations of random (as opposed to deterministic) variables. Such random variables are sampled from simple Gaussian distributions whose means and variances are parametrized by the output of convolutional encoders over the visual history. We introduce novel convolutional architectures for predicting future body joint trajectories that outperform fully connected alternatives [29]. We introduce stochastic spatial transformers through optical flow warping for predicting future frames, which outperform their deterministic equivalents [17]. Training stochastic networks involves an intractable marginalization over stochastic variables. We compare various training schemes that handle such marginalization through a) straightforward sampling from the prior, b) conditional variational autoencoders [23, 29], and, c) a proposed K-best-sample loss that penalizes the best prediction under a fixed “prediction budget”. We show experimental results on object trajectory prediction, human body joint trajectory prediction and video prediction under varying future uncertainty, validating quantitatively and qualitatively our architectural choices and training schemes.*

## 1. Introduction

Humans live in the future: we constantly predict what we are about to perceive (see, hear or feel) during our inter-

actions with the world, and learn from the deviations of our expectations from the reality [4]. While watching a scene, we are perfectly happy with one of its *many* possible future evolutions, but unexpected outcomes cause surprise. Predictive computational models should similarly be able to *predict any of the different modes of the distribution of future outcomes*.

We present stochastic neural network architectures that predict samples of future outcomes conditioned on a history of visual glimpses, and apply them to the tasks of object trajectory, body joint trajectory and frame forecasting (Figure 1). Stochastic neural networks are networks where some of the activations are not deterministic but rather sampled from (often Gaussian) distributions. Such normally distributed random variables are decoded after a series of convolutions and non-linear layers to highly multimodal distributions of the output space. In our work, the means and variances of the Gaussian distributions are deep non-linear transformations of the network input (the past visual history), that is, the stochastic layer is “sandwiched” between deterministic encoder and decoder sub-networks, generalizing previous architectures of [29, 34] which sampled from Gaussian unit variance noise. Once trained, sampling the stochastic variables and decoding them provides plausible samples of the future outcomes.

We present novel architectures and training schemes for stochastic neural networks. For forecasting human body joint trajectories we introduce *conv-indexed decoders*, depicted in Figure 1(b). Previous works on body joint trajectory prediction use fully connected layers since joints are present only on a handful of pixels, not at every pixel [29]. This limits generalization performance as each trajectory depends on the whole body configuration. We instead use fully convolutional networks and index into the final convolutional feature map using the body joint pixel coordinates -that are assumed known for the forecasting task. We thus predict body trajectories with a set of linear layers, one per body joint, each conditioned on a different feature embedding, extracted from the corresponding pixel location of the deconvolutional map. We empirically show

<sup>\*</sup>Carnegie Mellon University

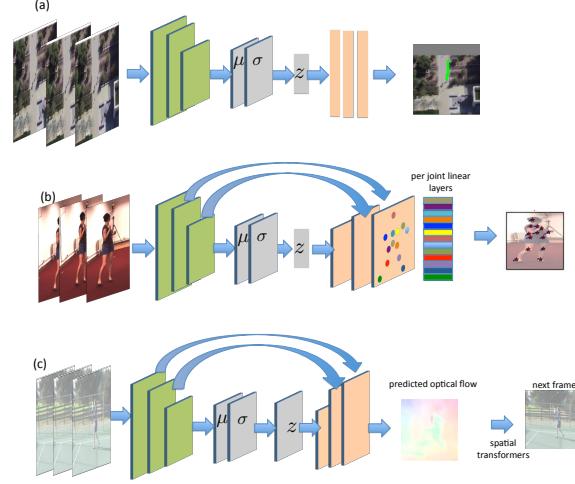
<sup>†</sup>Google Research

that the proposed “as-convolutional-as possible” architecture outperforms by a margin fully connected alternatives [29]. Second, for frame prediction, we present *stochastic dense spatial transformers*: we compose the future video frame through prediction of a dense pixel motion field (optical flow) and differentiable backward warping that essentially re-arranges pixels of the input frame. Different samples from our model result in different future motion fields and corresponding frame warps.

Training stochastic networks involves an intractable marginalization over the stochastic variables. We present and compare three training schemes that handle in different ways such intractable marginalization: a) Straightforward sampling for the prior. We simply collect samples from our Gaussian variables for each training example and maximize the empirical likelihood. b) Variational approximations, a.k.a. *conditional variation autoencoders* that sample from an approximate posterior distribution represented by a recognition model that has access to the output we want to predict. During training we minimize the Kullback–Leibler (KL) divergence between such informed posterior distribution and the prior distribution provided by our generator model. c) *K-best-loss*, a loss we introduce in this work, which penalizes only *the best sample* from a fixed “prediction budget” for each training example and training iteration.

We evaluate the proposed architectures and training schemes on three forecasting tasks: (1) predicting motion trajectories (of pedestrians, vehicles, bicyclists etc.) from overhead cameras in the Stanford drone dataset [19], (2) predicting future body joint motion trajectories in the H3.6M dataset [11], and predicting future video frames “in the wild” using Freiburg-Berkeley Motion Segmentation [3] and H3.6M datasets. We evaluate each training scheme under varying future uncertainty: the longer the frame history we condition upon, the lesser the future uncertainty, the fewer the modes in the output distribution. We show *K-best-loss* can better handle situations of *high uncertainty*, with many *equiprobable* outcomes, such as when predicting the future from a single image. We show extensive quantitative and qualitative results on paper and on our [online website](#) which confirm the validity of architectural innovations and facility of proposed training schemes.

The closest works to ours are the work of Walker et al. [29] that predicts dense future trajectories from a single image and the work of Xue et al. [34] that predicts a future frame given a pair of frames. Both use stochastic units by sampling zero mean and unit variance Gaussian noise concatenated with the input feature maps and train using conditional variational autoencoders. Our work extends those in the following ways: a) proposes indexed convolutional decoders in place of fully connected decoders for trajectory forecasting and shows dramatic improvements, b) pro-



**Figure 1. Conditional stochastic networks** for object trajectory forecasting (a), body joint trajectory forecasting (b) and video forecasting (c). Encoder, stochastic and decoder subnetworks are depicted in green, gray and orange, respectively. (a): Visual glimpses here depict overhead camera views. (b): *Conv-indexed decoders* select feature vectors from a deconvolution network based on pixel coordinates, and predict each body joint trajectory using a separate linear layer on top of the selected feature vector. (c): Samples of future dense motion fields (future optical flows) differentiably warp the current frame to generate samples of the future one.

poses a novel *K-best* for training stochastic networks under extreme uncertainty, c) analyses the training difficulties by comparing multiple training schemes under varying uncertainty, d) proposes a simpler architecture than [34] based on optical flow warping for frame prediction and e) applies stochastic networks in diverse tasks and datasets.

## 2. Related Work

**Multimodality in forecasting** Handling multimodality in prediction is a notoriously difficult problem [26]. Models that minimize standard regression losses perform poorly as they tend to regress towards a mean (blurry) outcome [15]. Models that combine regression and adversarial losses [15] suffer less from regression to the mean, but *latch onto one mode of the distribution and neglect the rest*, as noted in [26]. Classifiers that employ softmax losses need to discretize potentially high-dimensional output spaces, such as pixel motion fields, and suffer from the corresponding discretization errors. Mixture component networks [8, 27] parametrize a Gaussian Mixture Model (GMM) (mixture components weights, means and variances) where each mixture represents a different outcome. They have been used successfully for hand writing generation in [8] to predict the next \$(x, y)\$ pen stroke coordinate, given the gener-

ated writing so far. However, it is hard to train many mixtures of high dimensional output spaces, and, as it has been observed, many components often remain un-trained, with one component dominating the rest [7], unless careful mixture balancing is designed [22].

Many recent data driven approaches predict motion directly from image pixels. In [30], a large, nonparametric image patch vocabulary is built for patch motion regression. In [31], dense optical flow is predicted from a single image and the multimodality of motion is handled by considering a different softmax loss for every pixel. Work of [6] predicts ball trajectories in synthetic “billiard-like” worlds directly from a sequence of visual glimpses using a regression loss. Work of [7] uses recurrent networks to predict a set of body joint heatmaps at a future frame. Such representation though cannot possibly group the heatmap peaks into coherent 2D pose proposals. Work of [12] casts frame prediction as sequential conditional prediction, and samples from a categorical distribution of 255 pixel values at every pixel location, conditioning at the past history and image generated so far. It is unclear how to handle the computational overhead of such models effectively.

**Stochastic neural networks.** Stochastic variables have been used in a variety of settings in the deep learning literature e.g., for generative modeling, regularization, reinforcement learning, etc. A key issue is how to train networks with stochastic variables. Restricted Boltzmann machines (RBMs), for example [9] are typically trained using MCMC sampling. REINFORCE learning rules were introduced by Williams [32] in order to compute sample approximations to gradients in reinforcement learning settings and have also been useful in learning hard attention mechanisms [16, 33, 35]. In recent years papers have also proposed stochastic units that can be trained using ordinary SGD, with Dropout [24] being a ubiquitous example. Another example is the so-called *reparametrization trick* proposed by [13, 18] for backpropagation through certain stochastic units (e.g., Gaussian) that allow one to differentiate quantities of the form  $\mathbb{E}[f(x)]$  where  $x \sim P(x; \theta)$  with respect to distribution parameters  $\theta$ . Schulman *et al.* [21] recently united the REINFORCE and reparameterization tricks to handle backpropagation on general “stochastic computation graphs”.

**Motion prediction as Inverse Optimal Control (IOC)** Works of [38, 14, 10] model pedestrians as rational agents that move in a way that optimizes a reward function dependent on image features, e.g., obstacles in the scene, and form a distribution over optimal future paths to follow. They train a reward function such that the resulting Markov Decision Process best matches observed human trajectories in expectation. Many times though our motions/behaviours

are not goal oriented and their framework fails to account for those. In these cases, learning the possible future outcomes directly from data is important, as noted also in [37]. However, to deal with much longer temporal horizons we do expect that combination of planning and learning to be beneficial.

### 3. Conditional Stochastic Networks

Figure 1 depicts our three stochastic network models for the three tasks we consider: (1) object trajectory forecasting, (2) body joint trajectory forecasting and (3) video prediction. Samples from our models correspond to plausible future motion trajectories of the object, future body joint trajectories of a person or future frame optical flow fields, respectively, for the three tasks we consider.

**Object-centric prediction** For dense pixel motion prediction, visual glimpse correspond to the whole video frame. For the case of object and body joint trajectory forecasting however, visual glimpses are centered around the object of interest. Such object-centric glimpses have been shown to generalize better to novel environments [6, 2], as they provide translation invariance. Further, the same forecasting model can be shared by all objects in the scene to predict their future trajectories.

Our model has three components: an encoder  $E$ , a stochastic layer  $z$ , and a decoder  $D$ . The encoder is a deep convolutional neural network that takes as input a short sequence of visual glimpses  $x$  and computes a feature hierarchy. The stochastic layer is a  $d$ -dimensional random variable  $z$  drawn from a fully factored Gaussian distribution, whose means and standard deviations are parametrized by the top layer activations of the encoder  $E(x)$  through linear transformations  $(W_\mu, b_\mu)$  and  $(W_\sigma, b_\sigma)$ :

$$\begin{aligned}\mu(x) &= W_\mu \cdot E(x; \theta_E) + b_\mu, \\ \sigma(x) &= \log(\exp(W_\sigma \cdot E(x; \theta_E) + b_\sigma) + 1), \\ z &\sim \mathcal{N}(\mu, \text{diag}(\sigma)),\end{aligned}$$

For the standard deviations we use a softplus nonlinearity to ensure positivity. The decoder  $D$  transforms samples of  $z$  (as well as encoder feature maps when skip-layers are present) to produce output  $y$ . While each of our stochastic variables  $z$  follow a (unimodal) Gaussian distribution, the distribution of  $y$  can in general be highly multimodal as shown qualitatively in Figure 3 since Gaussian samples go through multiple non-linear layers.

The advantage of using a normal distribution for stochastic units is that back-propagation through the stochastic layer can be performed efficiently via the so-called *reparametrization trick* [13, 18], where  $\hat{z}$  denotes stochastic

samples:

$$\hat{z} = \mu(x) + \epsilon' \cdot \sigma(x), \quad \epsilon' \sim \mathcal{N}(0, I),$$

Given a visual glimpse sequence  $x$ , we obtain samples of future outcomes  $\hat{y}$  by sampling our stochastic variables  $z$ :

$$\begin{aligned} \hat{z} &= \mu(x) + \epsilon' \cdot \sigma(x), \quad \epsilon' \sim \mathcal{N}(0, I), \\ \hat{y} &= D(\hat{z}; E(x; \theta_E); \theta_D), \end{aligned}$$

We train our model using maximum likelihood. We employ the standard assumption that ground-truth future outcomes  $y$  are normally distributed with means that depend on the output of our model. Let  $\theta$  denote the weights of our stochastic network model, then we have:

$$y \sim \mathcal{N}(D(z; E(x; \theta_E); \theta_D), \nu I). \quad (1)$$

and maximize the marginal log likelihood  $\mathcal{L}$  of the observed future outcomes conditioned on the input visual glimpses, as previous works:

$$\begin{aligned} \mathcal{L}(\theta) &\equiv \frac{1}{N} \sum_{i=1}^N \log P(y^{(i)}) \equiv \frac{1}{N} \sum_{i=1}^N \log \int_{z^{(i)}} P(y^{(i)}, z^{(i)} | x^{(i)}) dz^{(i)} \\ &\equiv \frac{1}{N} \sum_{i=1}^N \log \int_{z^{(i)}} P(y^{(i)}, | x^{(i)}, z^{(i)}) P(z^{(i)} | x^{(i)}) dz^{(i)} \end{aligned} \quad (2)$$

The marginalization across latent variables in Eq. 2 is intractable. We will use the following three approximations: (1) a sample approximation to the marginal log likelihood by sampling from the prior (MCML), (2) variational approximations (VA) and (3)  $K$ -best-sample loss, as detailed below.

**Sampling from the prior (MCML)** We maximize an approximation to the marginal log likelihood  $\mathcal{L}$  using a set of  $K$  samples for each of our  $N$  training examples:

$$\mathcal{L}^{MCML}(\theta) \equiv \frac{1}{N} \sum_{i=1}^N \log \sum_{j=1}^K P(y^{(i)} | \hat{z}^{(i,j)}, x^{(i)}), \quad (3)$$

$$= \frac{1}{N} \sum_{i=1}^N \log \sum_{j=1}^K \exp \left( -\frac{\|\hat{y}^{(i,j)} - y^{(i)}\|^2}{2\nu} \right) \quad (4)$$

We use  $K = 15$  samples for each training example.

**Variational approximation (VA)** With a stochastic encoder and multimodal predictions, the likelihood of our model is sensitive to the sampling of the stochastic codes, which means, we would require a *large amount of samples* to provide a good approximation to the marginal likelihood, which is the disadvantage of using MCML. Another alternative is to do importance sampling in the stochastic codes,

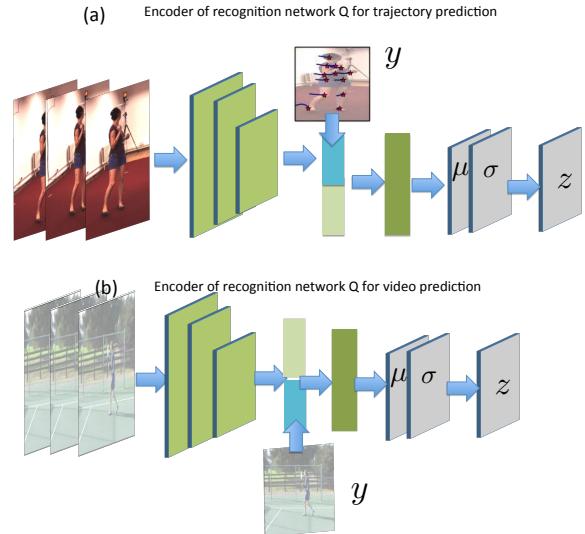


Figure 2. **Recognition networks  $Q$**  for the tasks of body joint prediction and video prediction. The output  $y$  is supplied as input to the model to provide an informative distribution over stochastic  $z$  codes which our generator network  $P$  tries to approximate.

by taking into account the likelihood of each one. Returning to Equation 2 from the previous section, but doing importance sampling over our stochastic codes, our loss takes the form:

$$\begin{aligned} \mathcal{L}(\theta) &= \frac{1}{N} \sum_{i=1}^N \log \int_{z^{(i)}} P(y^{(i)}, | x^{(i)}, z^{(i)}) \\ &\times \frac{P(z^{(i)} | x^{(i)})}{Q(z^{(i)} | x^{(i)} y^{(i)})} Q(z^{(i)} | x^{(i)} y^{(i)}) dz^{(i)}, \end{aligned} \quad (5)$$

$$\begin{aligned} &\geq \mathcal{L}^{VA}(\theta) = \frac{1}{N} \sum_{i=1}^N \int_{z^{(i)}} Q(z^{(i)} | x^{(i)} y^{(i)}) dz^{(i)} \\ &\times \log P(y^{(i)}, | x^{(i)}, z^{(i)}) \frac{P(z^{(i)} | x^{(i)})}{Q(z^{(i)} | x^{(i)} y^{(i)})}, \end{aligned} \quad (6)$$

$$\begin{aligned} &= \frac{1}{N} \sum_{i=1}^N \mathbb{E}_{z \sim Q(z^{(i)} | x^{(i)} y^{(i)})} \left[ \log P(y^{(i)}, | x^{(i)}, z^{(i)}) \right] \\ &- D_{KL} \left( Q(z^{(i)} | x^{(i)} y^{(i)}) || P(z^{(i)} | x^{(i)}) \right), \end{aligned} \quad (7)$$

Equation 6 is a lower bound of the importance weighted maximum likelihood objective (Equation 5), formed by taking the log of an expectation as the expectation of a log using Jensen's inequality. This requires creating a new approximation ( $Q$ ) to the true posterior. Here  $Q$  attempts to give us the best  $z$  that can create a particular path  $y$ . In our final objective in Equation 7 we see that this takes the form of a variational autoencoder style objective, as first noted in [23]; since our  $Q$  network further conditions on

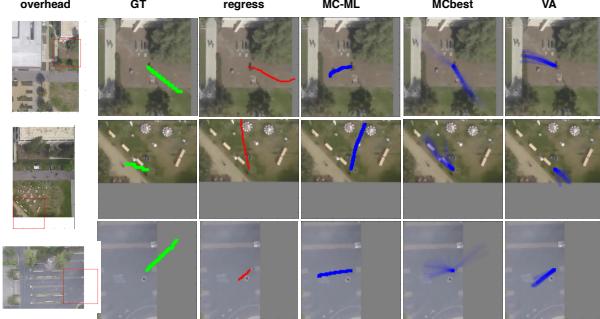


Figure 3. **Object motion trajectory prediction** in the drone Stanford dataset of [20]. A single frame is used as input ( $N^f = 1$ ). This is a case of *extreme uncertainty*: we do not have any idea regarding the direction of object’s motion. The first column shows the overhead scene and the input visual glimpse and second column shows the groundtruth trajectory. Dark blue denotes high sample density and light blue denotes low sample density. MCbest predicts more diverse outcomes (e.g., row 3) in comparison to the rest.

input  $x$  (rather than just output  $y$ ), it is called *conditional* variational autoencoder. In the term on the left, we are just forming maximum likelihood predictions, but using codes from our recognition network  $Q$  (we found it sufficient to use only one sample per training example and iteration) instead of our generator, while the term on the right forces the distribution of the stochastic codes of our recognition network  $Q$  and our generator to be close (in KL divergence terms). The architecture of our recognition networks for the tasks we consider are illustrated in Figure 2. At test time, of course only the generator model is used.

**K-best-sample-loss (MCbest)** We introduce  $K$ -best-sample-loss  $\mathcal{L}^{best\text{-}sample}(\theta)$  that takes into account only the best sample within a budget of  $K$  predictions, for each training example:

$$\mathcal{L}^{best\text{-}sample}(\theta) \equiv \frac{1}{N} \sum_{i=1}^N \max_{\hat{z}^{(i,j)} \sim P(z|x^{(i)})} P(y^{(i)}|\hat{z}^{(i,j)}, x^{(i)}), \quad (8)$$

$$\equiv \frac{1}{N} \sum_{i=1}^N \min_{j=1\cdots K} \|y^{(i)} - \hat{y}^{(i,j)}\|^2 \quad (9)$$

Intuitively, for each training example during training, our model makes  $K$  guesses (i.e. draws  $K$  samples) about the future outcome, not all of which have to be “correct” — we only penalize the best guess made by the model for its deviation from ground-truth. This approach consequently encourages our model to “cover its bases” by making diversified predictions.

## 4. Experiments

We test the performance of our stochastic network models in three tasks: (a) forecasting future motion of human body joints, (b) forecasting video frames, and (c) forecasting future object motion from overhead cameras, and compare against losses and architectures considered in previous works. We compare the following training schemes presented in Section 3: (a) a sample approximation to the marginal likelihood (*MCML*), (b)  $K$ -best-sample loss (*MCbest*), (c) conditional variational autoencoder (*VA*) for the various architectures we consider in each task. We also consider a regression model (*regress*) that has similar architecture to the stochastic alternatives in each experiment but that does not have any stochastic units and minimizes a regression loss.

**Evaluation metric** We quantify how well our models perform on predicting the distribution of the future outcomes in our datasets, let it be object trajectories, body joint trajectories or future frames, using  $topk$  error,  $k \in 1 \cdots 15$ :  $topk$  error measures the lowest of the errors (against ground-truth) among the top  $k$  predictions, where predictions are ordered according to model’s confidence. For our stochastic networks, we obtain multiple predictions by sampling  $z$  (without ensuring diversity) and decoding it to  $\hat{y}^{out}$  and ordering the predictions according to the  $z$  probability (in decreasing order) while keeping the top  $k$  most confident.  $Topk$  error is a meaningful measure in case of uncertainty, when there is no one single best solution, and has been employed widely, e.g., in the Imagenet Image classification task [5], which has potentially much less uncertainty than our forecasting setup. For regression models, we only have one prediction available and thus all  $topk$  errors are identical.

**Human body joint trajectory forecasting** We use the Human3.6M (H3.6M) dataset of Ionescu et al. [11]. H3.6M contains videos of actors performing activities and provides annotations of body joint locations at every frame, recorded from a Vicon system. We split the videos according to the human subjects, having subject with id=5 as our test subject. We consider videos of all activities. We center our input glimpses around the human actor by using the bounding box provided by the dataset on the latest history frame and crop all history images with the same box (so that only the human moves and the background remains static, as is the case also in the original frame). For training, we pretrained the encoder for body joint detection in the same dataset, and then finetuned the whole model in the forecasting. We consider two types of decoder subnetworks: a four layer fully connected decoder considered in [29] and a conv-indexed decoder, described in Section 3.

Our evaluation criterion is L2 distance between predicted

**Stanford Drone Dataset**-object trajectory prediction

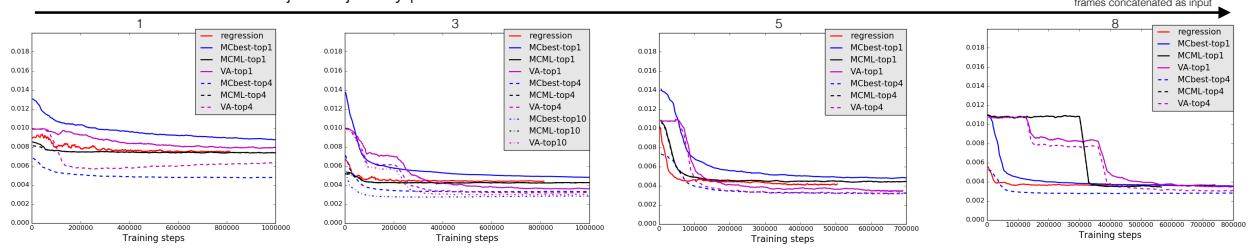


Figure 4. **Object trajectory prediction** in the Drone Stanford dataset [20] under varying number of input frames  $N^f$ . We show top $k$  error, for  $k = 1$  and  $k = 4$ . Col 1: Forecasting future motion trajectory from a single frame. This is a case of extreme uncertainty and MCbest outperforms the rest on top-4 error. When a long visual history is used as input ( $N^f = 8$ ), the benefit of stochastic models over a deterministic regression baseline is smaller, as uncertainty in prediction decreases (column 4).

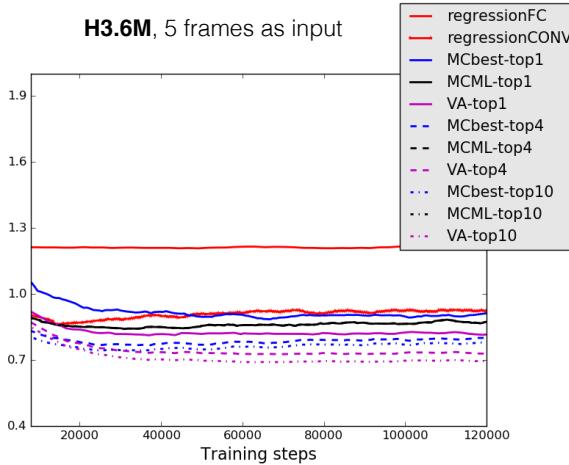


Figure 5. **Body joint trajectory forecasting** in the H3.6M test set. All actions categories are considered and five frames are used as input ( $N^f = 5$ ). All models apart from *regressionFC* share the same architecture that use conv-indexed decoders. The regression baseline *regressionFC* has a fully-connected decoder instead for comparison. **Fully connected decoders under-fit our dataset and failed to minimize the error.** For clarity we show only the fully connected regression model since the stochastic fully connected models exhibit similar behavior. Conditional variational autoencoders outperform all other training schemes in this experiment.

and ground-truth instantaneous velocities of all body joints, averaged over the time horizon of the prediction. We use  $h = 15$ . We show quantitative results in Figure 5 (a) for  $N^f = 5$  frames in the visual history input to the model. All models with fully connected decoders failed to minimize the error in our experiments and under-fit the training set. We show in the figure only the regression baseline for clarity (*regressionFC*), as the other curves are identical. Models with conv-indexed decoders fit the data effectively, and the VA training scheme performed the best. For extensive qual-

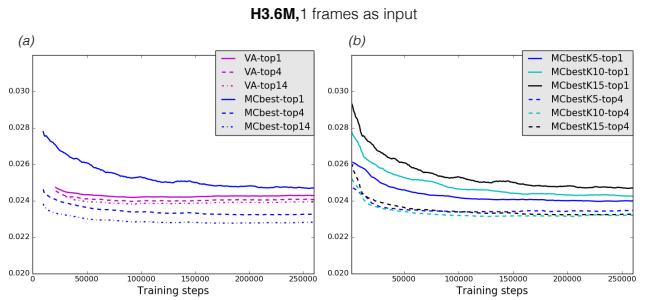


Figure 6. **Video prediction** in the H3.6M test set from a single image ( $N = 1$ ). (a) We compare 15-best-loss (*MCbest*) against conditional variational approximations (VA). 15-best-loss has higher top1 error than VA but lower top4 and top14 error. (b)  $K$ -best loss under varying  $K$ : as we increase number of samples  $K$ , top1 error gets worse and top4 decreases, as expected.  $K$  is a hyperparameters and depends on the amount of anticipated uncertainty.

itative results please see our [online website](#).

Conv-indexed decoders assume we know the pixel coordinates  $(x, y)$  of the joints whose trajectories we want to predict in the latest frame of the visual history. Such knowledge is indeed available, without which prediction of velocities (accumulated in time to form predicted  $(x, y)$  locations) would not be meaningful. We note that fully connected decoders effectively worked for the much easier, close to deterministic, problem of predicting body joint trajectories *only for the Walking activity* which suggests that architectural considerations are important as the forecasting problem becomes harder and more diverse.

**Video prediction** We use the Freiburg-Berkeley Motion Segmentation (MoSeg) dataset of [3] and the Human3.6M (H3.6M) dataset of [11]. MoSeg contains high resolution videos of people, animals, vehicles and other moving objects; the camera may be moving or static and the frame

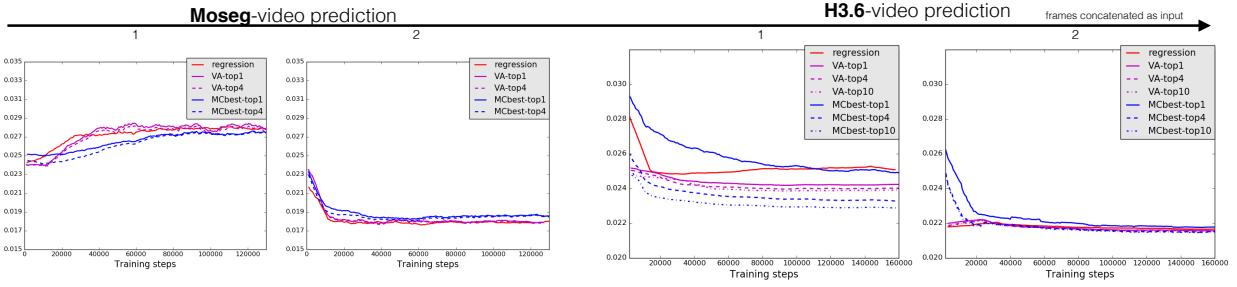


Figure 7. **Video prediction** results in MoSeg and H3.6M datasets.



Figure 8. **Video prediction** in MoSeg (first five columns) and H3.6M (last five columns) using  $Nf = 2$  frames as input. *Top row*: last frame of visual history. *Middle Row*: predicted frame by warping. The forecasted flow fields correctly delineate the moving objects. The visual realism of the predicted frames is a benefit of using motion driven differentiable warping [36] as opposed to direct pixel generation through standard deconvolution networks.

rate varies from video to video. H3.6M contains videos of a static camera and constant frame rate. We applied frame-centric video prediction in MoSeg and object-centric prediction in H3.6M (same as above, but this time we predict the whole frame rather than body joint trajectories). In this way, we can test video prediction both in controlled and “in-the-wild” setups. In MoSeg, we follow the official train and test split of the dataset and train our video prediction models from scratch. In H3.6M, we use the same split as for body joint prediction. The input to the network is 128 X 320 for MoSeg and 64X64 for H3.6M. MCML has shown the worst performance so far and since for frame prediction each sample is a whole decoded frame rather than a small set of trajectories we did not consider MCML for this experiment. Due to GPU memory limitations, we could only test 5-best-loss for MoSeg while for H3.6M we test 15-best-loss (since the input to the network is smaller). We also compare against a regression baseline that predicts a single flow field and warps to generate the future frame, similar to the model of [17].

Quantitative results showing the L2 loss between next ground-truth frame and predicted frame for the test set of the two datasets are shown in Figures 7, 6 and qualitative results are in Figure 8. All models fail to minimize the prediction error in MoSeg dataset when only one frame is considered due to camera motion: camera motion dictates the appearance of the next frame and is impossible to be predicted from a single frame. This is one reason previous works use background subtraction to stabilize the video [29, 28]. On the contrary, in H3.6M the camera is not moving and a single frame suffices to make plausible predictions of future flow fields. Video results and diverse samples for future frames are available at our [online website](#) as gifs, since they are often too subtle to see in a static image.

**Object future motion trajectory prediction** We use the Stanford Drone dataset introduced in Robicquet et al. [20], the largest overhead dataset that has been collected so far, with a drone flying above various locations in the Stanford campus. Trajectories of vehicles, pedestrians, bikers, bi-

cyclists, skateboarders etc, as well as their category labels have been annotated. We split trajectories into train and test set randomly. We use a short sequence of  $N^f$  square visual glimpses around the target of interest (again following the principle of object-centric prediction) as input to our model and predict future motion for the subsequent  $h = 100$  frames. We do not use the annotated category labels. We randomly split object trajectories into train and test sets. Our evaluation criterion is L2 distance between predicted and ground-truth instantaneous velocities, averaged over the time horizon of the prediction.

We show quantitative results of our models and a regression baseline in Figure 4. The deterministic regression baseline is similar to the model used in [6] for predicting trajectories of Billiard balls in simulated (deterministic) worlds. Under uncertainty, regression has higher top1 error than VA, and much higher error than top4 errors of VA and 15-best-loss, for any number of input history frames  $N^f$  (for  $N^f = 1$ , VA takes much longer to train than regression).

**As the length of the history increases, uncertainty decreases** and regression becomes more competitive. Indeed, most of the times a linear object motion model suffices. It is at the *intersections* that a multimodal predictive model is needed, but these are precisely the cases when good predictions are crucial, e.g., to adapt a self-driving car’s behavior based on anticipated pedestrian motion. Please see our [online website](#) for video results.

#### 4.1. Implementation details.

Our encoder E is a convolutional network variant of the Inception architecture [25]. We pre-train our encoders on the ImageNet classification task by replicating the weights of the first convolutional layer as many times as the number of frames in the input glimpse volume. For body joint trajectory forecasting, we pretrain our encoder on body joint classification in H3.6M dataset. We use skip layers in the tasks of body joint trajectory forecasting and video forecasting. The stochastic variables are replicated across the width and height of the top encoder layer in case of fully convolutional decoders. We represent output motion trajectories with a sequence of instantaneous velocities  $\mathbf{tr} = \{(\Delta x_t, \Delta y_t), t = 0, \dots, h\}$ , where  $h$  is a prediction horizon. We conduct our training and evaluation using TensorFlow [1], a publicly available deep learning package.

### 5. Conclusion

We presented stochastic neural networks for multimodal motion forecasting for object, body joint and frame predictions. We presented architectural innovations as well as novel training schemes and extensive evaluations of those in various benchmarks. Our work aims to help understanding stochastic networks and their applicability for forecasting, and the difficulties (or not) of their training, building upon

and extending recent previous works. We discussed both object and body joint trajectory prediction that uses human annotations for future outcomes as well as self-supervised video prediction that uses the next frame as ground-truth. The longer the temporal horizon for the prediction, and the shorter the temporal history to condition upon, the larger the uncertainty and the larger the benefit from incorporating stochasticity. Our model offers a simple solution to multimodal prediction of large continuous spaces and we expect it will be useful in domains with uncertainty beyond motion prediction.

### References

- [1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *arXiv:1603.04467*, 2016. [8](#)
- [2] P. W. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, and K. Kavukcuoglu. Interaction networks for learning about objects, relations and physics. *CoRR*, abs/1612.00222, 2016. [3](#)
- [3] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*. 2010. [2, 6](#)
- [4] L. de Wit, B. Machilsen, and T. Putzeys. Predictive coding and the neural response to predictable stimuli. *Journal of Neuroscience*, 30(26):8702–8703, 2010. [1](#)
- [5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. [5](#)
- [6] K. Fragniadaki, P. Agrawal, S. Levine, and J. Malik. Learning visual predictive models of physics for playing billiards. *ICLR*, 2016. [3, 8](#)
- [7] K. Fragniadaki, S. Levine, P. Felsen, and J. Malik. Recurrent network models for human dynamics. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. [3](#)
- [8] A. Graves. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850, 2013. [2](#)
- [9] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006. [3](#)
- [10] D.-A. Huang, A. M. Farahmand, K. M. Kitani, and J. A. Bag nell. Approximate maxent inverse optimal control and its application for mental simulation of human interactions. In *AAAI*, 2015. [3](#)
- [11] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *PAMI*, 36(7):1325–1339, 2014. [2, 5, 6](#)

- [12] N. Kalchbrenner, A. v. d. Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu. Video pixel networks. *arXiv preprint arXiv:1610.00527*, 2016. 3
- [13] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv:1312.6114*, 2013. 3
- [14] K. Kitani, B. Ziebart, J. Bagnell, and M. Hebert. Activity forecasting. In *ECCV*, 2012. 3
- [15] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *CoRR*, abs/1511.05440, 2015. 2
- [16] V. Mnih, N. Heess, A. Graves, et al. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems*, pages 2204–2212, 2014. 3
- [17] V. Patraucean, A. Handa, and R. Cipolla. Spatio-temporal video autoencoder with differentiable memory. *CoRR*, abs/1511.06309, 2015. 1, 7
- [18] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv:1401.4082*, 2014. 3
- [19] A. Robicquet, A. Alahi, A. Sadeghian, B. Anenberg, J. Doherty, E. Wu, and S. Savarese. Forecasting social navigation in crowded complex scenes. *arXiv:1601.00998*, 2016. 2
- [20] A. Robicquet, A. Alahi, A. Sadeghian, B. Anenberg, J. Doherty, E. Wu, and S. Savarese. Forecasting social navigation in crowded complex scenes. *CoRR*, abs/1601.00998, 2016. 5, 6, 7
- [21] J. Schulman, N. Heess, T. Weber, and P. Abbeel. Gradient estimation using stochastic computation graphs. In *NIPS*, pages 3510–3522, 2015. 3
- [22] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. V. Le, G. E. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *CoRR*, abs/1701.06538, 2017. 3
- [23] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *NIPS*, pages 3483–3491. Curran Associates, Inc., 2015. 1, 4
- [24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 3
- [25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 8
- [26] L. Theis, A. van den Oord, and M. Bethge. A note on the evaluation of generative models. In *ICLR*, Apr 2016. 2
- [27] Z. Tüske, M. A. Tahir, R. Schlüter, and H. Ney. Integrating gaussian mixtures into deep neural networks: Softmax layer with hidden variables. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 4285–4289, Brisbane, Australia, Apr. 2015. 2
- [28] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pages 613–621, 2016. 7
- [29] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: Forecasting from static images using variational autoencoders. *CoRR*, abs/1606.07873, 2016. 1, 2, 5, 7
- [30] J. Walker, A. Gupta, and M. Hebert. Patch to the future: Unsupervised visual prediction. In *CVPR*, pages 3302–3309. IEEE, 2014. 3
- [31] J. Walker, A. Gupta, and M. Hebert. Dense optical flow prediction from a static image. *CoRR*, abs/1505.00295, 2015. 3
- [32] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992. 3
- [33] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv:1502.03044*, 2015. 3
- [34] T. Xue, J. Wu, K. Bouman, and B. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 91–99. Curran Associates, Inc., 2016. 1, 2
- [35] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. *arXiv:1511.06984*, 2015. 3
- [36] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros. View synthesis by appearance flow. *CoRR*, abs/1605.03557, 2016. 7
- [37] B. D. Ziebart. Modeling purposeful adaptive behavior with the principle of maximum causal entropy. Dec 2010. 3
- [38] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Petersson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa. Planning-based prediction for pedestrians. In *IROS*, 2009. 3