Accurate and Diverse Sampling of Sequences based on a "Best of Many" Sample Objective (Supplementary Material)

Apratim Bhattacharyya, Bernt Schiele, Mario Fritz
Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany
{abhattac, schiele, mfritz}@mpi-inf.mpg.de

1. Additional Details of our "Best of Many" Sample Objective

Here we provide additional details of our "Best of Many" samples objective and include additional qualitative results. We begin with the formal statement of the First Mean Value Theorem of Integration [1]. The First Mean Value Theorem of Integration states that, if $f_1 : [a, b] \to \mathbb{R}$ is continuous and f_2 is an integrable function that does not change sign on $[a, b]$, then $\exists z' \in (a, b)$ such that,

$$\int_a^b f_1(z)\, f_2(z)\, dz = f_1(z') \int_a^b f_2(z)\, dz \tag{S1}$$

The data log-likelihood Equation (3) in the main paper, estimated using importance sampling using a recognition network q_ϕ is given by,

$$\begin{aligned} &\log(p_\theta(y \mid x)) = \\ &\log\Big(\int p_\theta(y|z,x)\, \frac{p(z|x)}{q_\phi(z|x,y)}\, q_\phi(z|x,y)\, dz \Big). \end{aligned} \tag{S2}$$

We apply the First Mean Value Theorem of Integration to derive Equation (4) in the main paper, which is,

$$\begin{aligned} \log(p_\theta(y|x)) = \log\Big(\int_a^b p_\theta(y|z,x)\, q_\phi(z|x,y)\, dz \Big) \\ + \log\Big(\frac{p(z'|x)}{q_\phi(z'|x,y)} \Big),\ z' \in (a,b). \end{aligned} \tag{S3}$$

To do this, we set $f_1(z) = {p(z|x)}/{q_\phi(z|x,y)}$ and $f_2(z) = p_\theta(y|z,x) \times q_\phi(z|x,y)$ (from the data log-likelihood in (S2)). The integral in (S2) can be very well approximated on a large enough bounded interval $[a, b]$. This leads to,

$$\begin{aligned} &\Big(\int_a^b p_\theta(y|z,x)\, \frac{p(z|x)}{q_\phi(z|x,y)}\, q_\phi(z|x,y)\, dz \Big) \\ &= \frac{p(z'|x)}{q_\phi(z'|x,y)} \ \Big(\int_a^b p_\theta(y|z,x)\, q_\phi(z|x,y)\, dz \Big). \end{aligned} \tag{S4}$$

Taking $\log$ on both sizes of (S4) leads to (S3). We can further lower bound (S3), leading to Equation (5) in the main paper, which is,

$$\begin{aligned} \log(p_\theta(y|x)) \geq \log\Big(\int_a^b p_\theta(y|z,x)\, q_\phi(z|x,y)\, dz \Big) \\ + \min_{z' \in (a,b)} \Big(\log\Big(\frac{p(z'|x)}{q_\phi(z'|x,y)} \Big) \Big) \end{aligned} \tag{S5}$$

However, as mentioned in the main paper, the minimum in (S5) is difficult to estimate. Therefore, we use the following approximation. From (S3), we know that $\exists z' \in (a, b)$ which lower bounds the data log-likelihood. To maximize this data log-likelihood, we would like to maximize $\log(f_1(z'))$. However, as we do not know z', we instead choose to maximize it for a set of N points in (a, b),

$$\begin{aligned} &\log\Big(\int_a^b p_\theta(y|z,x)\, q_\phi(z|x,y)\, dz \Big) \\ &+ \log\Big(\frac{p(z'_1|x)}{q_\phi(z'_1|x,y)} \Big) + .. + \log\Big(\frac{p(z'_N|x)}{q_\phi(z'_N|x,y)} \Big). \end{aligned} \tag{S6}$$

As values of both p and q_ϕ are bounded above by 1, the value of the function $f_2(z'_i) = {p(z'_i|x)}/{q_\phi(z'_i|x,y)}$ is likely to be low when is p low and q_ϕ is high. Therefore, to give more importance to such points z'_i, we weight each point by $q_\phi(z'_i|x,y)$,

$$\begin{aligned} &\log\Big(\int_a^b p_\theta(y|z,x)\, q_\phi(z|x,y)\, dz \Big) \\ &+ q_\phi(z'_1|x,y) \times \log\Big(\frac{p(z'_1|x)}{q_\phi(z'_1|x,y)} \Big) \\ &+ \ldots + q_\phi(z'_N|x,y) \times \log\Big(\frac{p(z'_N|x)}{q_\phi(z'_N|x,y)} \Big). \end{aligned} \tag{S7}$$

Flipping the sign before the terms in the second part of (S7),

$$\begin{aligned}&\log\Big(\int_a^b p_\theta(y|z,x)\,q_\phi(z|x,y)\,dz\Big)\\&-q_\phi(z_1'|x,y)\times\log\Big(\frac{q_\phi(z_1'|x,y)}{p(z_1'|x)}\Big)\\&-\ldots-q_\phi(z_N'|x,y)\times\log\Big(\frac{q_\phi(z_N'|x,y)}{p(z_N'|x)}\Big).\end{aligned}\tag{S8}$$

If we choose a sufficiently large set of points $z_i' \in (a,b)$, we can collect the terms in the second part of (S8) and replace them with a single integral,

$$\begin{aligned}&\log\Big(\int_a^b p_\theta(y|z,x)\,q_\phi(z|x,y)\,dz\Big)\\&-\int_a^b q_\phi(z|x,y)\times\log\Big(\frac{q_\phi(z|x,y)}{p(z_|x)}\Big)\,dz.\end{aligned}\tag{S9}$$

The second integral in (S9) is the KL divergence between the two distributions $q_\phi(z|x,y)$ and $p(z_|x)$,

$$\begin{aligned}&\log\Big(\int_a^b p_\theta(y|z,x)\,q_\phi(z|x,y)\,dz\Big)\\&-D_{\text{KL}}(q_\phi(z|x,y)\parallel p(z|x)).\end{aligned}\tag{S10}$$

We can estimate the data log-likelihood term in (S10) using Monte-Carlo integration. This leads to the "Many Sample" objective from the main paper,

$$\begin{aligned}\hat{\mathcal{L}}_{\text{MS}}=&\log\Big(\frac{1}{T}\sum_{i=1}^{T}p_\theta(y|\hat{z}_i,x)\Big)\\&-D_{\text{KL}}(q_\phi(z|x,y)\parallel p(z|x)),\ \ \hat{z}_i\sim q_\phi(z|x,y).\end{aligned}\tag{S11}$$

As mentioned in the main paper, we use the re-parameterization trick [2] to sample from our recognition network q_ϕ. Therefore, the recognition network predicts the mean and variance $\mathcal{N}(\mu,\sigma)$ of the Gaussian distribution q_ϕ from which the latent variable z is sampled. Thus, we can directly use the predicted μ,σ to estimate the KL divergence as in [2].

Approximating the data log-likelihood term in the first part of (S11) as shown in the main paper, leads to our "Best of Many" sample objective.

2. Additional Details of our Models

Here, we include details of each layer of our models.

2.1. Model for Structured Trajectory Prediction

We provide the details of our structured trajectory prediction model in Table 1. Followed by the details of the recognition network (q_ϕ) in Table 2. We refer to fully connected layers as Dense and Size refers to the number of neurons in the layer.

Layer	Type	Size	Activation	Input	Output
In_1	Input			x	EMB_1
EMB_1	Dense	32	*ReLU*	In_1	LSTM_{enc}
LSTM_{enc}	LSTM	48	*tanh*	EMB_1	EMB_2
EMB_2	Dense	64	*ReLU*	$\{\text{LSTM}_{enc}, q_\phi\}$	LSTM_{dec}
LSTM_{dec}	LSTM	48	*tanh*	EMB_2	Out_1
Out_1	Dense	2		LSTM_{dec}	$\hat{y}$

Table 1: Details our model for Structured Trajectory Prediction. The details of the recognition network q_ϕ used during training follows in Table 2.

Layer	Type	Size	Activation	Input	Output
In_2	Input			y	EMB_3
EMB_3	Dense	64	*ReLU*	In_2	LSTM_{rec}
LSTM_{rec}	LSTM	128	*tanh*	EMB_3	$\{\text{D}_1, \text{D}_2\}$
D_1	Dense	64		LSTM_{rec}	μ
D_2	Dense	64		LSTM_{rec}	σ

Table 2: Details of the recognition network used during training of our model for Structured Trajectory Prediction.

2.2. Extension with Visual Input

This model is similar to the model for Structured Trajectory Prediction, expect that the LSTM_{dec} is additionally conditioned on the output of an CNN encoder. The details are in Table 3 and Table 4. We use the same recognition network as described previously in subsection 2.1.

Layer	Type	Filters	Size	Activation	Input	Output
In_2	Input					C_1
C_1	Conv	32	3×3	*tanh*	In_2	P_1
P_1	MaxPool		2×2		C_1	C_2
C_2	Conv	64	3×3	*tanh*	P_1	P_2
P_2	MaxPool		2×2		C_2	C_3
C_3	Conv	128	3×3	*tanh*	P_2	P_3
P_3	MaxPool		2×2		C_3	C_4
C_4	Conv	256	3×3	*tanh*	P_3	P_4
P_4	MaxPool		2×2		C_4	FC_1
FC_1	Dense	1024		*tanh*	P_4	FC_2
FC_2	Dense	32		*tanh*	FC_1	EMB_2

Table 3: Details of the CNN encoder used with the extended Structured Trajectory Prediction model with Visual Input. Conv stands for 2D convolution, MaxPool stands for 2D max pooling and UpSample stands for 2D upsampling operations.

Layer	Type	Size	Activation	Input	Output
In_1	Input			x	EMB_1
EMB_1	Dense	32	*ReLU*	In_1	LSTM_{enc}
LSTM_{enc}	LSTM	48	*tanh*	EMB_1	EMB_2
EMB_2	Dense	64	*ReLU*	$\{\mathrm{LSTM}_{enc}, \mathrm{FC}_2\}$	EMB_3
EMB_3	Dense	64	*ReLU*	$\{\mathrm{EMB}_2, q_\phi\}$	LSTM_{dec}
LSTM_{dec}	LSTM	64	*tanh*	EMB_3	Out_1
Out_1	Dense	2		LSTM_{dec}	$\hat{y}$

Table 4: Details our model for extended Structured Trajectory Prediction model with Visual Input. The details of the recognition network q_ϕ used during training follows in Table 5.

Layer	Type	Size	Activation	Input	Output
In_3	Input			y	EMB_4
EMB_4	Dense	64	*ReLU*	In_3	LSTM_{rec}
LSTM_{rec}	LSTM	128	*tanh*	EMB_3	$\{\mathrm{D}_1, \mathrm{D}_2\}$
D_1	Dense	64		LSTM_{rec}	μ
D_2	Dense	64		LSTM_{rec}	σ

Table 5: Details of the recognition network used during training of our extended Structured Trajectory Prediction model with Visual Input.

2.3. Model for Structured Image Sequence Prediction

We provide the details of our structured image sequence prediction model in Table 6. Followed by the details of the recognition network (q_ϕ) in Table 7. In contrast to the model for structured trajectory prediction, we use Convolutional LSTMs and Convolutional Embedding layers.

Layer	Type	Filters	Size	Input	Output
In_1	Input			x	CEMB_1
CEMB_1	Conv	32	3×3	In_1	P_1
P_1	MaxPool		2×2	CEMB_1	CLSTM_{enc1}
CLSTM_{enc1}	CLSTM	32	3×3	P_1	P_2
P_2	MaxPool		2×2	CLSTM_{enc1}	CLSTM_{enc2}
CLSTM_{enc2}	CLSTM	64	3×3	P_2	CEMB_2
CEMB_2	Conv	32	3×3	$\{\mathrm{CLSTM}_{enc2}, q_\phi\}$	CLSTM_{dec1}
CLSTM_{dec1}	CLSTM	64	3×3	CEMB_2	U_1
U_1	UpSample		2×2	CLSTM_{dec1}	CLSTM_{dec2}
CLSTM_{dec2}	CLSTM	64	3×3	U_1	U_2
U_2	UpSample		2×2	CLSTM_{dec2}	Out_1
Out_1	Conv	32	3×3	U_2	Out_2
Out_2	Conv	1	3×3	Out_1	$\hat{y}$

Table 6: Details our model for Structured Image Sequence Prediction. CLSTM stands for 2D Convolutional LSTM, Conv stands for 2D convolution, MaxPool stands for 2D max pooling and UpSample stands for 2D upsampling operations. The details of the recognition network q_ϕ used during training follows in Table 7.

Layer	Type	Filters	Size	Input	Output
In_2	Input			y	CEMB_3
CEMB_3	Conv	32	3×3	In_2	P_3
P_3	MaxPool		2×2	CEMB_3	CLSTM_{rec1}
CLSTM_{rec1}	CLSTM	32	3×3	P_3	P_4
P_4	MaxPool		2×2	CLSTM_{rec1}	CLSTM_{rec2}
CLSTM_{rec2}	CLSTM	64	3×3	P_4	$\{\mathrm{C}_1, \mathrm{C}_2\}$
C_1	Conv	64	3×3	CLSTM_{rec2}	μ
C_2	Conv	64	3×3	CLSTM_{rec2}	σ

Table 7: Details of the recognition network used during training of our model for Structured Image Sequence Prediction. CLSTM stands for 2D Convolutional LSTM, Conv stands for 2D convolution, MaxPool stands for 2D max pooling and UpSample stands for 2D upsampling operations.

3. Additional Results

We show additional qualitative results on the HKO dataset in Figure 1 at $t+5, t+10$ and $t+15$. We generate $T = 50$ samples and show the sample closest to the groundtruth (Best), the mean of all the samples and the per-pixel variance in the samples. As in the main paper, the qualitative examples demonstrate that our model produces samples which are close to the groundtruth (comparing the Best sample and the groundtruth) and diverse samples (comparing the difference between the mean of the samples and the Best sample).

References

[1] M. Comenetz. *Calculus: the elements*. World Scientific Publishing Co Inc, 2002.

[2] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *ICLR*, 2013.

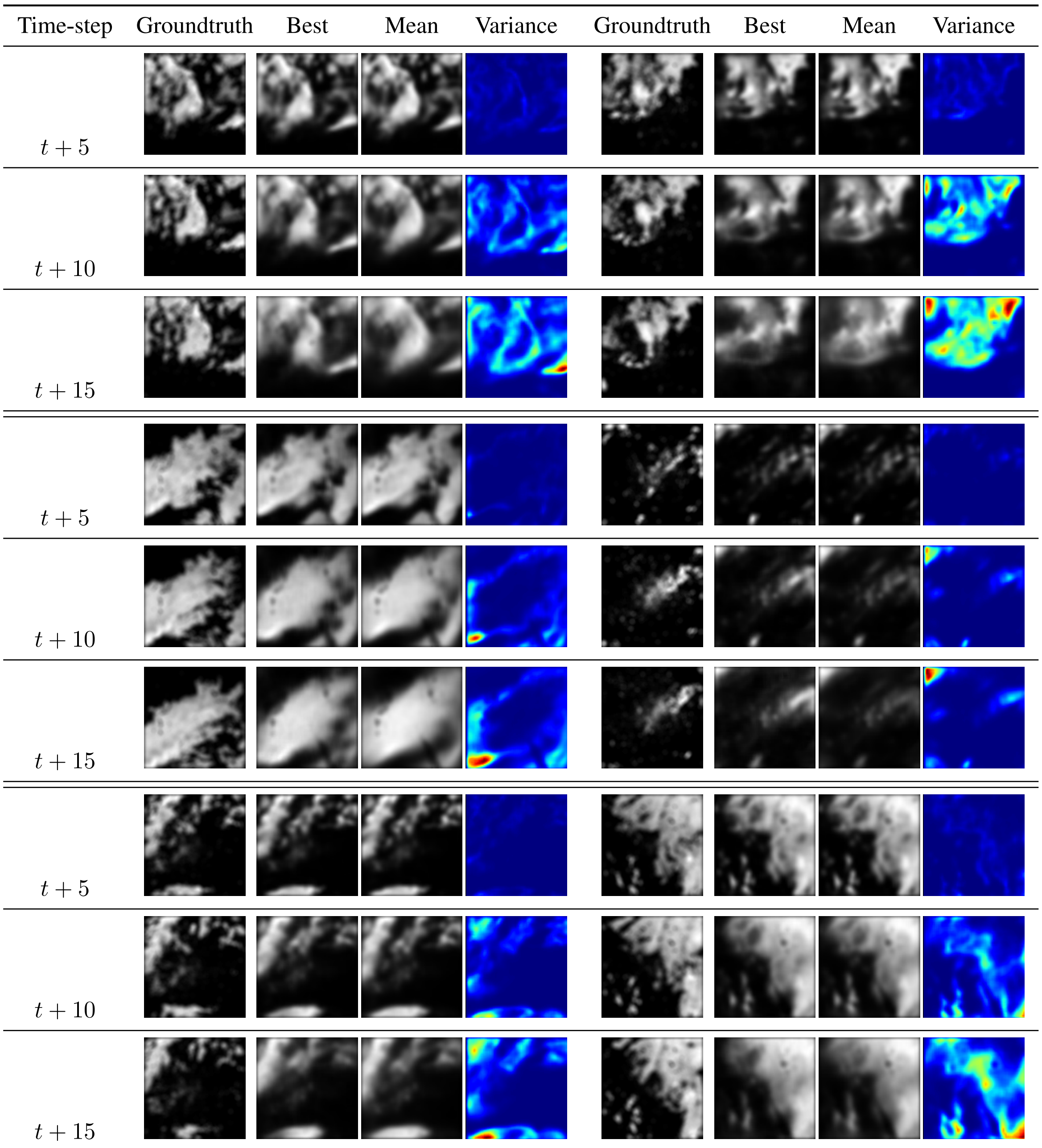

Figure 1: Statistics of samples generated by our LSTM-BMS model on the HKO dataset.