Supplementary Material: Amortized Population Gibbs Samplers with Neural Sufficient Statistics

A. Gradient of the generative model

We show that the gradient of the marginal $\nabla_\theta \log p_\theta(x)$ can be estimated using self-normalized importance sampling. First of all, we express the expected gradient of the log joint as

$$\mathbb{E}_{p_\theta(z|x)}\left[\nabla_\theta \log p_\theta(x,z)\right] = \mathbb{E}_{p_\theta(z|x)}\left[\nabla_\theta \log p_\theta(x) + \nabla_\theta \log p_\theta(z|x)\right] = \mathbb{E}_{p_\theta(z|x)}\left[\nabla_\theta \log p_\theta(x)\right] = \nabla_\theta \log p_\theta(x)$$

Here we make use of a standard identity that is also used in likelihood-ratio estimators

$$\mathbb{E}_{p_\theta(z|x)}\left[\nabla_\theta \log p_\theta(z|x)\right] = \int p_\theta(z|x)\nabla_\theta \log p_\theta(z|x)\, dz = \int \nabla_\theta p_\theta(z|x)\, dz = \nabla_\theta \int p_\theta(z|x)\, dz = \nabla_\theta 1 = 0$$

Therefore, we have the the following equality

$$\nabla_\theta \log p_\theta(x) = \mathbb{E}_{p_\theta(z|x)}\left[\nabla_\theta \log p_\theta(x,z)\right] \simeq \sum_{l=1}^{L} \frac{w^l}{\sum_{l'} w^{l'}} \nabla_\theta \log p_\theta(x, z^l).$$

which is the self-normalized gradient estimator in Equation 7.

B. Importance weights in sequential importance sampling

We will prove the form of importance weight w^k in sequential importance sampling. At step $k = 1$, we use exactly the standard importance sampler, thus it is obvious that the following is a valid importance weight

$$w^1 = \frac{\gamma^1(z^1)}{q^1(z^1)}.$$

When step $k > 2$, we are going to prove that the importance weight relative to the intermediate densities has the form

$$w^k = \frac{\gamma^k(z^{1:k})}{q^1(z^1)\prod_{k'=2}^{k} q^{k'}(z^{k'} \mid z^{1:k'-1})}. \tag{9}$$

At step $k = 2$, the importance weight is defined as

$$w^k = v^2\, w^1 \;= \frac{\gamma^2(z^{1:2})}{\gamma^1(z^1)\, q^2(z^2 \mid z^1)} \frac{\gamma^1(z^1)}{q^1(z^1)} \;= \frac{\gamma^2(z^{1:2})}{q^1(z^1)\, q^2(z^2 \mid z^1)}.$$

which is exactly that form. Now we prove weights in future steps by induction. At step $k \geq 2$, we assume that the weight has the form in Equation 9, i.e.

$$w^k = \frac{\gamma^k(z^{1:k})}{q^1(z^1)\prod_{k'=2}^{k} q^{k'}(z^{k'} \mid z^{1:k'-1})}.$$

then at step $k+1$, the importance weight is the product of incremental weight and incoming weight

$$w^{k+1} = v^{k+1}\, w^k = \frac{\gamma^{k+1}(z^{1:k+1})}{\gamma^k(z^{1:k})\, q^{k+1}(z^{k+1} \mid z^{1:k})} \frac{\gamma^k(z^{1:k})}{q^1(z^1)\prod_{k'=2}^{k} q^{k'}(z^{k'} \mid z^{1:k'-1})} = \frac{\gamma^{k+1}(z^{1:k+1})}{q^1(z^1)\prod_{k'=2}^{k+1} q^{k'}(z^{k'} \mid z^{1:k'-1})}.$$

Thus the importance weight w^k has the form of Equation 9 at each step $k > 2$ in sequential importance sampling.

C. Derivation of Posterior Invariance

We consider a *sweep* of conditional proposals at step K as

$$p_\theta\big(z^k \mid x, z^{k-1}\big) = \prod_{b=1}^{B} p_\theta(z_b^k \mid x, z_{\prec b}^k, z_{\succ b}^{k-1}), \tag{10}$$

where $z_{\prec b} = \{z_i \mid i < b\}$ and $z_{\succ b} = \{z_i \mid i > b\}$. Additionally we define $z_{\preceq b} = \{z_i \mid i \leq b\}$.

We will show that any partial update within a sweep, i.e.

$$p_\theta\big(z_{\preceq b}^k \mid x, z^{k-1}\big) = \prod_{v=1}^{b} p_\theta(z_v^k \mid x, z_{\prec v}^k, z_{\succ v}^{k-1}), \quad \forall b \in \{1, 2, ..., B\} \tag{11}$$

will leave the posterior invariant. In fact, for any choice of b we have

$$\begin{aligned}
\int p_\theta(z^{k-1} \mid x)\, p_\theta(z_{\preceq b}^k \mid x,\, z^{k-1})\, dz_{\preceq b}^{k-1} &= \int p_\theta(z_{\preceq b}^{k-1}, z_{\succ b}^{k-1} \mid x)\, dz_{\preceq b}^{k-1} \prod_{v=1}^{b} p_\theta(z_v^k \mid x, z_{\prec v}^k, z_{\succ v}^{k-1}) \\
&= p_\theta(z_{\succ b}^{k-1} \mid x)\, p_\theta(z_{\preceq b}^k \mid x, z_{\succ b}^{k-1}) \\
&= p_\theta(z_{\preceq b}^k,\, z_{\succ b}^{k-1} \mid x).
\end{aligned}$$

When we require the APG proposal $q_\phi(z_b' \mid x, z_{-b})$ leaves the posterior invariant (by minimizing the inclusive KL divergence relative to the conditional posterior $p_\theta(z_b \mid x, z_{-b})$), then any sweep or part of one sweep will also leave the posterior invariant, as what we prove above. This means that at test time we can apply arbitrary number of APG sweeps, each of which will results in samples that approximate the posterior $p_\theta(z \mid x)$.

D. Resampling Algorithm

Algorithm 3 Multinomial Resampler

1: **Input** Weighted samples $\{z^l, w^l\}_{l=1}^L$
2: **for** $i = 1$ **to** L **do**
3: $\quad a^i \sim \text{Discrete}(\{w^l / \sum_{l'=1}^L w^{l'}\}_{l=1}^L)$ ▷ Index Selection
4: $\quad$ Set Set $\tilde{z}^{\,i} = z^{a^i}$
5: $\quad$ Set $\tilde{w}^{\,i} = \frac{1}{L} \sum_{l=1}^L w^l$ ▷ Re-weigh
6: **end for**
7: **Output** Equally weighted samples $\{\tilde{z}^l, \tilde{w}^l\}_{l=1}^L$

E. Proof of the amortized population Gibbs samplers algorithm

Here, we provide an alternative proof of correctness of the APG algorithm given in Algorithm 2, based on the construction of proper weights (Naesseth et al., 2015) which was introduced after SMC samplers (Del Moral et al., 2006). In section E.1, we will introduce proper weights; In section E.2, we then present several operations that preserve the proper weighting property; In section E.3, we will take use of these properties to prove the correctness of APG samplers algorithm (Algorithm 2).

E.1. Proper weights

Definition 1 (Proper weights)**.** Given an unnormalized density $\tilde{p}(z)$, with corresponding normalizing constant $Z_p := \int \tilde{p}(z)\, \mathrm{d}z$ and normalized density $p \equiv \tilde{p}/Z_p$, the random variables $z, w \sim P(z, w)$ are properly weighted with respect to $\tilde{p}(z)$ if and only if for any measurable function f

$$\mathbb{E}_{P(z,w)}\left[w f(z)\right] = Z_p \mathbb{E}_{p(z)}[f(z)]. \tag{12}$$

We will also denote this as

$$z, w \overset{\text{p.w.}}{\sim} \tilde{p}.$$

Using proper weights. Given independent samples $z^l, w^l \sim P$, we can estimate Z_p by setting $f \equiv 1$:

$$Z_p \approx \frac{1}{L} \sum_{l=1}^{L} w^l.$$

This estimator is unbiased because it is a Monte Carlo estimator of the left hand side of (12). We can also estimate $\mathbb{E}_{p(z)}[f(z)]$ as

$$\mathbb{E}_{p(z)}[f(z)] \approx \frac{\frac{1}{L}\sum_{l=1}^{L} w^l f(z^l)}{\frac{1}{L}\sum_{l=1}^{L} w^l}.$$

While the numerator and the denominator are unbiased estimators of $Z_p \mathbb{E}_{p(z)}[f(z)]$ and Z_p respectively, their fraction is biased. We often write this estimator as

$$\mathbb{E}_{p(z)}[f(z)] \approx \sum_{l=1}^{L} \bar{w}^l f(z^l), \tag{13}$$

where $\bar{w}^l := w^l / \sum_{l'=1}^{L} w^{l'}$ is the normalized weight.

E.2. Operations that preserve proper weights

Proposition 1 (Nested importance sampling)**.** *This is similar to Algorithm 1 in (Naesseth et al., 2015). Given unnormalized densities $\tilde{q}(z), \tilde{p}(z)$ with the normalizing constants Z_q, Z_p and normalized densities $q(z), p(z)$, if*

$$z, w \overset{p.w.}{\sim} \tilde{q}, \tag{14}$$

then

$$z, \frac{w\tilde{p}(z)}{\tilde{q}(z)} \overset{p.w.}{\sim} \tilde{p}.$$

Proof. First define the distribution of z, w as Q. For measurable $f(z)$

$$\mathbb{E}_{Q(z,w)}\left[\frac{w\tilde{p}(z)}{\tilde{q}(z)} f(z)\right] = Z_q \mathbb{E}_{q(z)}\left[\frac{\tilde{p}(z)f(z)}{\tilde{q}(z)}\right] = Z_q \int q(z) \frac{\tilde{p}(z)f(z)}{\tilde{q}(z)} \,\mathrm{d}z = \int \tilde{p}(z) f(z) \,\mathrm{d}z = Z_p \mathbb{E}_{p(z)}[f(z)].$$

□

Proposition 2 (Resampling)**.** *This is similar to Section 3.1 in (Naesseth et al., 2015). Given an unnormalized density $\tilde{p}(z)$ (normalizing constant Z_p, normalized density $p(z)$), if we have a set of properly weighted samples*

$$z^l, w^l \overset{p.w.}{\sim} \tilde{p}, \quad l = 1, \ldots, L \tag{15}$$

then the resampling operation preserves the proper weighting, i.e.

$$z'^l, w'^l \overset{p.w.}{\sim} \tilde{p}, \quad l = 1, \ldots, L$$

where $z'^l = z^a$ with probability $P(a = i) = w^i / \sum_{l=1}^{L} w^l$ and $w'^l := \frac{1}{L}\sum_{l=1}^{L} w^l$.

Proof. Define the distribution of z^l, w^l as $\hat{P}$. We show that for any f, $\mathbb{E}[f(z^a)w'^l] = Z_p\mathbb{E}_{p(z)}[f(z)]$.

$$\begin{aligned}
&\mathbb{E}_{(\prod_{l=1}^L \hat{P}(z^l,w^l))p(a|w^{1:L})}\left[f(z^a)w'^l\right] \\
&= \mathbb{E}_{\prod_{l=1}^L \hat{P}(z^l,w^l)}\left[\sum_{i=1}^L f(z^i)w'\, P(a=i)\right] \\
&= \mathbb{E}_{\prod_{l=1}^L \hat{P}(z^l,w^l)}\left[\sum_{i=1}^L f(z^i)w'\frac{w^i}{\sum_{l'=1}^L w^{l'}}\right] \\
&= \mathbb{E}_{\prod_{l=1}^L \hat{P}(z^l,w^l)}\left[\frac{1}{L}\sum_{i=1}^L f(z^i)w^i\right] \\
&= \frac{1}{L}\sum_{i=1}^L \mathbb{E}_{\hat{P}(z^i,w^i)}\left[f(z^i)w^i\right] = \frac{1}{L}\sum_{i=1}^L Z_p\mathbb{E}_{p(z)}[f(z)] = Z_p\,\mathbb{E}_{p(z)}[f(z)].
\end{aligned}$$

□

Therefore, the resampling will return a new set of samples that are still properly weighted relative to the target distribution in the APG sampler (Algorithm 2).

Proposition 3 (Move)**.** *Given an unnormalized density $\tilde{p}(z)$ (normalizing constant Z_p, normalized density $p(z)$) and normalized conditional densities $q(z'|z)$ and $r(z|z')$, the proper weighting is preserved if we apply the transition kernel to a properly weighted sample, i.e. if we have*

$$z^l, w^l \overset{p.w.}{\sim} \tilde{p}, \tag{16}$$

$$z'^l \sim q(z'^l|z^l), \tag{17}$$

$$w'^l = \frac{\tilde{p}(z'^l)r(z^l|z'^l)}{\tilde{p}(z^l)q(z'^l|z^l)}w^l, \qquad l = 1,\dots,L \tag{18}$$

then we have

$$z'^l, w'^l \overset{p.w.}{\sim} \tilde{p}, \qquad l = 1,\dots,L \tag{19}$$

Proof. Firstly we simplify the notation by dropping the superscript l without loss of generality. Define the distribution of z, w as $\hat{P}$. Then, due to (16), for any measurable $f(z)$, we have

$$\mathbb{E}_P[wf(z)] = Z_p E_p[f(z)].$$

To prove (19), we show $\mathbb{E}_{\hat{P}(z,w)q(z'|z)}[w'f(z')] = Z_p\mathbb{E}_{p(z')}[f(z')]$ for any f as follows:

$$\begin{aligned}
\mathbb{E}_{\hat{P}(z,w)q(z'|z)}[w'f(z')] &= \mathbb{E}_{\hat{P}(z,w)q(z'|z)}\left[\frac{\tilde{p}(z')r(z|z')}{\tilde{p}(z)q(z'|z)}wf(z')\right] \\
&= \int \hat{P}(z,w)q(z'|z)\frac{\tilde{p}(z')r(z|z')}{\tilde{p}(z)q(z'|z)}wf(z')\,\mathrm{d}z\,\mathrm{d}w\,\mathrm{d}z' \\
&= \int \hat{P}(z,w)\frac{\tilde{p}(z')r(z|z')}{\tilde{p}(z)}wf(z')\,\mathrm{d}z\,\mathrm{d}w\,\mathrm{d}z' \\
&= \int \tilde{p}(z')f(z')\left(\int \hat{P}(z,w)w\frac{r(z|z')}{\tilde{p}(z)}\,\mathrm{d}z\,\mathrm{d}w\right)\mathrm{d}z' \\
&= \int \tilde{p}(z')f(z')Z_p\mathbb{E}_{p(z)}\left[\frac{r(z|z')}{\tilde{p}(z)}\right]\mathrm{d}z'.
\end{aligned} \tag{20}$$

Using the fact that $\mathbb{E}_{p(z)}\left[\frac{r(z|z')}{\tilde{p}(z)}\right] = \int p(z)\frac{r(z|z')}{\tilde{p}(z)}\,\mathrm{d}z = \int r(z|z')\,\mathrm{d}z/Z_p = 1/Z_p$. Equation 20 simplifies to

$$\int \tilde{p}(z')f(z')\,\mathrm{d}z' = Z_p\mathbb{E}_{p(z')}[f(z')].$$

□

E.3. Correctness of APG Sampler

We prove the correctness of the APG sampler (Algorithm 2) by induction. We firstly prove the correctness of the initial proposing step ($k = 1$, line 4 - line 9); Then we prove that the algorithm is still correct when we perform one Gibbs sweep step ($k = 2$, line 11 - line 22), given that the previous step is already proved to be correct. By induction we can conclude that its correctness still holds if we perform more Gibbs sweeps.

Step $k = 1$. We initialize the proposal of all the blocks $z := z_{1:B}$ from an initial encoder $z \sim q_\phi(z|x)$ (line 5), which is trained using the wake-ϕ phase objective in the standard reweighted wake-sleep(Le et al., 2019), where the objective is

$$\mathbb{E}_{\hat{p}(x)}\left[\mathrm{KL}\left(p_\theta(z|x)||q_\phi(z|x)\right)\right].$$

We take gradient w.r.t. variational parameter ϕ and compute a self-normalized gradient estimate (line 8) as

$$\begin{aligned} g_\phi &:= -\nabla_\phi\, \mathbb{E}_{\hat{p}(x)}\left[\mathrm{KL}\left(p_\theta(z \mid x)\,||\,q_\phi(z \mid x)\right)\right] && (21)\\ &= \mathbb{E}_{\hat{p}(x)}\left[\mathbb{E}_{p_\theta(z|x)}\left[\nabla_\phi \log q_\phi(z \mid x)\right]\right] && (22)\\ &= \sum_{l=1}^{L} \frac{w^l}{\sum_{l'=1}^{L} w^{l'}} \nabla_\phi \log q_\phi(z^l \mid x), \quad z^l \sim q_\phi(z \mid x), \quad w^l = \frac{p_\theta(x, z^l)}{q_\phi(z^l \mid x)}. && (23)\end{aligned}$$

Equation 13 will guarantee the validity of this gradient estimate g_ϕ, as long as we show that samples are properly weighted

$$z^l, w^l \overset{\text{p.w.}}{\sim} p_\theta(z, x), \qquad l = 1, \ldots, L. \tag{24}$$

In fact, $\{(w^l, z^l)\}_{l=1}^L$ are properly weighted because z^l are proposed using importance sampling Naesseth et al. (2015), where $q_\phi(z|x)$ is the proposal density and $p_\theta(z^l, x)$ is the unnormalized target density. Note that the resampling step (line 13) will preserve the proper weighting because of Proposition 2.

Step $k = 2$. Now we iteratively update each block of the variable z_b for $b = 1, 2, ..., B$, using the corresponding conditional proposal $q_\phi(z_b \mid x, z_{-b})$, which is trained by the objective

$$\mathcal{K}_b(\phi) := \mathbb{E}_{\hat{p}(x)p_\theta(z_{-b}|x)}\Big[\mathrm{KL}\left(p_\theta(z_b \mid x, z_{-b})\,||\,q_\phi(z_b \mid x, z_{-b})\right)\Big], \quad b = 1, 2, ..., B.$$

We take gradient w.r.t ϕ as

$$\begin{aligned} g_\phi^b &:= -\nabla_\phi \mathbb{E}_{p(x)}\left[\mathbb{E}_{p_\theta(z_{-b}|x)}\left[\mathrm{KL}\left(p_\theta(z_b|z_{-b}, x)||q_\phi(z_b|z_{-b}, x)\right)\right]\right] && (25)\\ &= \mathbb{E}_{p(x)}\left[\mathbb{E}_{p_\theta(z_{1:B}|x)}\left[\nabla_\phi \log q_\phi(z_b|z_{-b}, x)\right]\right], \quad b = 1, 2, ..., B. && (26)\end{aligned}$$

We compute a self-normalized gradient estimate (line 19) in a propose-weigh-reassign manner (line 15, line 16, line 17).

We will validate this gradient estimate (line 19) using the proper weighting again, i.e. we want to prove that

$$z_{1:B}^l, w^l \overset{\text{p.w.}}{\sim} p_\theta(z_{1:B}, x), \qquad l = 1, \ldots, L. \tag{27}$$

so that Equation 13 will guarantee the validity of this gradient estimate.

To prove that one Gibbs sweep (line 11 - line 22) also preserves proper weighting, we will show that each block update satisfies all the 3 conditions (Equation 16, 18 and 27) in Proposition 3, by which we can conclude the samples are still properly weighted after each block update.

Without loss of generality, we drop all l superscripts in the rest of the proof. Before any block update (before line 15), we already know that samples are properly weighted, i.e.

$$z, w \overset{\text{p.w.}}{\sim} p_\theta(z, x). \tag{28}$$

which corresponds to Equation 16. Next we define a conditional distribution $q(z' \mid z) := q_\phi(z'_b|x, z_{-b})\delta_{z_{-b}}(z'_{-b})$, from which we propose a new sample

$$z' \sim q_\phi(z'_b|x, z_{-b})\delta_{z_{-b}}(z'_{-b}), \tag{29}$$

where the density of z'_{-b} is a delta mass on z_{-b} defined as $\delta_{z_{-b}}(z'_{-b}) = 1$ if $z_{-b} = z'_{-b}$ and 0 otherwise. In fact, this form of sampling step is equivalent to firstly sample $z'_b \sim q_\phi(z_b \mid x, z_{-b})$ (line 15) and let $z'_{-b} = z_{-b}$ (line 17), which is exactly what the APG sampler assumes procedurally in Algorithm 2. This condition corresponds to Equation 17.

Finally, we define the weight w'

$$w' = \frac{p_\theta(x, z'_b, z'_{-b})r(z_b|x, z_{-b})\delta_{z_{-b}}(z_{-b})}{p_\theta(x, z_b, z_{-b})q_\phi(z'_b|x, z_{-b})\delta_{z_{-b}}(z'_{-b})}w, \tag{30}$$

where the terms in blue are treated as densities (normalized or unnormalized) of $z'_{1:B}$ and the terms in red are treated as densities of $z_{1:B}$. Since both delta mass densities evaluate to one, this weight is equal to the weight computed after each block update (line 16). This condition corresponds to Equation 18.

Now we can apply the conclusion (19) in Proposition 3 and claim

$$z'_{1:B}, w' \overset{\text{p.w.}}{\sim} p_\theta(z'_{1:B}, x).$$

since $z_{-b} = z'_{-b}$ and $z_b = z'_b$ due to the re-assignment (line 17). Note that the resampling step (line 13) will preserve the proper weighting because of Proposition 2.

Based on the fact that proper weighting is preserved at both the initial proposing step $k = 1$ and one Gibbs sweep $k = 2$, we have proved that both gradient estimates (line 8 and line 19) are correct.

F. Architectures of Amortized Population Gibbs samplers using Neural Sufficient Statistics

Based on our proposed parameterization in terms of neural sufficient statistics (see section 4), we will explain how we design the approximate Gibbs (neural) proposals in the experiments in section 6.

In general, we consider a structured model $p_\theta(x, z)$ where we can partition the latent variables $z = \{z^{\text{G}}, z^{\text{L}}\}$ into global variables z^{G} and local variables z^{L}. The dimensionality of global variables is typically constant, whereas local variables $z^{\text{L}} = \{z^{\text{L}}_1, \ldots, z^{\text{L}}_N\}$ have a dimensionality that increases with the instance size N. For models with this structure, the local variables are typically conditionally independent

$$z^{\text{L}}_n \perp z^{\text{L}}_{-n} \mid x, z^{\text{G}}. \tag{31}$$

We assume that the priors $p(z^{\text{G}}; \lambda^{\text{G}})$ and $p(z^{\text{L}}; \lambda^{\text{L}})$ are in the exponential family form, where λ^{G} and λ^{L} are natural parameters of the corresponding distributions. By the conditional independencies, we parameterize the conditional neural proposals (i.e. variational distributions) using neural sufficient statistics $T_\phi(\cdot)$ as

$$\tilde{\lambda}^{\text{G}} = \lambda^{\text{G}} + \sum_{n=1}^{N} T^{\text{G}}_\phi(x_n, z^{\text{L}}_n), \qquad \tilde{\lambda}^{\text{L}}_n = \lambda^{\text{L}}_n + T^{\text{L}}_\phi(x_n, z^{\text{G}}). \tag{32}$$

where $\tilde{\lambda}^{\text{G}}$ and $\tilde{\lambda}^{\text{L}}$ are natural parameters of proposals of the global variables z^{G} and local variables z^{L} respectively.

In the Gaussian mixture model we know the analytic forms of conditional (conjugate) posteriors. This means that we have analytic expressions for true sufficient statistics in equation 33. Here, the APG sampler learns neural sufficient statistics that approximate the true statistics. In the deep generative mixture model and the time series model in bouncing MNIST, we no longer know the analytic forms of the conditionals. As a result, the APG samplers for these two models will employ neural

networks f_ϕ^{G} and f_ϕ^{L} that take the (learned) neural sufficient statistics and the parameters of the priors as input, and predict the parameters of the proposals as output, i.e.

$$\tilde{\lambda}^{\text{G}} = \lambda^{\text{G}} + \sum_{n=1}^{N} T_\phi^{\text{G}}(x_n, z_n^{\text{L}}) \approx f_\phi^{\text{G}}(\lambda^{\text{G}}, \sum_{n=1}^{N} T_\phi^{\text{G}}(x_n, z_n^{\text{L}})), \qquad \tilde{\lambda}_n^{\text{L}} = \lambda_n^{\text{L}} + T_\phi^{\text{L}}(x_n, z^{\text{G}}) \approx f_\phi^{\text{L}}(\lambda_n^{\text{L}}, T_\phi^{\text{L}}(x_n, z^{\text{G}})).$$

Since there is always a deterministic transformation between a natural parameter and the corresponding distribution parameters (i.e. the parameters in canonical form), we can always convert any exponential family to a canonical form. For convenience, our networks output the canonical parameters directly, rather than returning natural parameters that then need to be converted to canonical form.

F.1. Gaussian Mixture Model

In the APG sampler for the Gaussian mixture model (GMM), we employ neural proposals of the form

$$q_\phi(\mu_{1:M}, \tau_{1:M} \mid x_{1:N}) = \prod_{m=1}^{M} \text{NormalGamma}\Big(\mu_m, \tau_m \;\Big|\; \tilde{\alpha}_m, \tilde{\beta}_m, \tilde{\mu}_m, \tilde{\nu}_m\Big), \tag{33}$$

$$q_\phi(\mu_{1:M}, \tau_{1:M} \mid x_{1:N}, c_{1:N}) = \prod_{m=1}^{M} \text{NormalGamma}\Big(\mu_m, \tau_m \;\Big|\; \tilde{\alpha}_m, \tilde{\beta}_m, \tilde{\mu}_m, \tilde{\nu}_m\Big), \tag{34}$$

$$q_\phi(c_{1:N} \mid x_{1:N}, \mu_{1:M}, \tau_{1:M}) = \prod_{n=1}^{N} \text{Categorical}\Big(c_n \;\Big|\; \tilde{\pi}_n\Big). \tag{35}$$

where $M = 3$ is the number of clusters in a GMM. We use the tilde symbol $\tilde{}$ to denote the parameters of the conditional neural proposals (i.e. approximate Gibbs proposals). The NormalGamma on the vector-valued mean $\mu_m \in \mathbb{R}^2$ and diagonal precision $\tau_m \in \mathbb{R}_+^2$ follows the standard definition

$$\tau_m \sim \text{Gamma}(\alpha_0, \beta_0), \qquad \mu_m \sim \text{Normal}(\mu_0, 1/(\nu_0 \tau)). \tag{36}$$

where $\mu_0 = 0, \nu_0 = 0.1, \alpha_0 = 0.2, \beta_0 = 0.2$. The natural parameters $\lambda^{\text{G}} := (\lambda_1^{\text{G}}, \lambda_2^{\text{G}}, \lambda_3^{\text{G}}, \lambda_4^{\text{G}})$ of this distribution are defined in terms of the canonical parameters as

$$\lambda_1^{\text{G}} = \alpha_0 - \frac{1}{2}, \qquad \lambda_2^{\text{G}} = -\beta_0 - \frac{\nu_0 \mu_0^2}{2}, \qquad \lambda_3^{\text{G}} = \nu_0 \mu_0, \qquad \lambda_4^{\text{G}} = -\frac{\nu_0}{2}. \tag{37}$$

We employ neural sufficient statistics that approximate the true pointwise sufficient statistics

$$\Big\{ \text{I}[c_n = m],\ \text{I}[c_n = m]\, x_n,\ \text{I}[c_n = m]\, x_n^2 \;\Big|\; m = 1, 2, \ldots, M \Big\}$$

We use fully-connected networks for the statistics $T_\phi^{\text{G}}(x_n)$ of the initial proposal $q_\phi(\mu_{1:M}, \tau_{1:M} \mid x_{1:N})$ and the statistics $T_\phi^{\text{G}}(x_n, c_n)$ for the conditional $q_\phi(\mu_{1:M}, \tau_{1:M} \mid x_{1:N}, c_{1:N})$,

$T_\phi^{\text{G}}(x_n),\ x_n \in \mathbb{R}^2$	
FC. 2. (s_n)	FC. 3. Softmax. (t_n)

$T_\phi^{\text{G}}(x_n, c_n),\ x_n \in \mathbb{R}^2, c_n \in \{0,1\}^3$	
Concatenate$[x_n,\ c_n]$	
FC. 2. (s_n)	FC. 3. Softmax. (t_n)

The output of each network consists of two elements s_n and t_n. s_n approximates the variable x_n; $t_{n,m}$ approximates the identity function $\text{I}[c_n = m]$. Then we sum over all the points and compute the parameters of the conjugate posterior in analytic forms

$$N_m = \sum_{n=1}^{N} t_{n,m}, \quad \tilde{x}_m = \sum_{n=1}^{N} t_{n,m} \cdot s_n, \quad \tilde{x}_m^2 = \sum_{n=1}^{N} t_{n,m} \cdot s_n^2, \tag{38}$$

$$\tilde{\alpha}_m = \alpha_0 + \frac{N_m}{2}, \tag{39}$$

$$\tilde{\beta}_m = \beta_0 + \frac{1}{2}\left(\tilde{x}_m^2 - \frac{(\tilde{x}_m)^2}{N_m}\right) + \frac{N_m \nu_0}{N_m + \nu_0} \frac{(\frac{\tilde{x}_m}{N_m} - \mu_0)^2}{2}, \tag{40}$$

$$\tilde{\mu}_m = \frac{\mu_0 \nu_0 + \tilde{x}_m}{\nu_0 + N_m}, \tag{41}$$

$$\tilde{\nu}_m = \nu_0 + N_m. \tag{42}$$

We assume a Categorical prior on the assignment of each point c_n of the form

$$c_n \sim \text{Categorical}(\pi) \tag{43}$$

where $\pi = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. The natural parameter is

$$\lambda^{\text{L}} = \log \pi \tag{44}$$

We employ a vector of neural statistics

$$T_\phi^{\text{L}}(x_n, \mu_{1:M}, \tau_{1:M}) := \big(T_\phi^{\text{L}}(x_n, \mu_1, \tau_1),\ T_\phi^{\text{L}}(x_n, \mu_2, \tau_2),\ \ldots,\ T_\phi^{\text{L}}(x_n, \mu_M, \tau_M)\big), \tag{45}$$

where each element is parameterized by the network

$T_\phi^{\text{L}}(x_n, \mu_m, \tau_m),\ x_n \in \mathbb{R}^2,\ \mu_m \in \mathbb{R}^2,\ \tau_m \in \mathbb{R}_+^2$
Concatenate$[x_n\ \mu_m,\ \tau_m]$
FC. 32. Tanh. FC. 1.

We add these statistics to the natural parameters. The canonical parameters of the approximate posterior $\tilde{\pi}$ are then simply the Softmax normalization of the resulting sum

$$\tilde{\pi}_n = \text{Softmax}\Big(\log \pi + T_\phi^{\text{L}}(x_n,\ \mu_{1:M},\ \tau_{1:M})\Big). \tag{46}$$

F.2. Deep Generative Mixture Model

The APG sampler in the deep generative mixture model (DMM) employs neural proposals of the form

$$q_\phi(\mu_{1:M} \mid x_{1:N}) = \prod_{m=1}^{M} \text{Normal}\Big(\mu_m \,\Big|\, \tilde{\mu}_m, \tilde{\sigma}_m^2 I\Big), \tag{47}$$

$$q_\phi(\mu_{1:M} \mid x_{1:N}, c_{1:N}, h_{1:N}) = \prod_{m=1}^{M} \text{Normal}\Big(\mu_m \,\Big|\, \tilde{\mu}_m, \tilde{\sigma}_m^2 I\Big), \tag{48}$$

$$q_\phi(c_{1:N}, h_{1:N} \mid x_{1:N}, \mu_{1:M}) = q_\phi(c_{1:N} \mid x_{1:N}, \mu_{1:M})\ q_\phi(h_{1:N} \mid c_{1:N}, , x_{1:N}, \mu_{1:M}) \tag{49}$$

$$= \prod_{n=1}^{N} \text{Categorical}\Big(c_n \,\Big|\, \tilde{\pi}_n\Big)\ \text{Beta}\Big(h_n \,\Big|\, \tilde{\alpha}_n, \tilde{\beta}_n\Big). \tag{50}$$

where $M = 4$ is the number of clusters in a GMM. We use the tilde symbol ˜ to denote the parameters of the conditional neural proposals (i.e. approximate Gibbs proposals).

We assume a Gaussian prior on mean of each cluster μ_m of the form

$$\mu_m \sim \text{Normal}(\mu, \sigma_0^2 I) \tag{51}$$

where the hyper-parameters are $\mu = 0$ and $\sigma_0 = 10$. To compute the parameters of neural proposals, we firstly employ some MLPs that predict neural sufficient statistics $T_\phi^{\text{G}}(x_n)$ and $T_\phi^{\text{G}}(x_n, c_n, h_n)$ as

$T_\phi^{\text{G}}(x_n),\ x_n \in \mathbb{R}^2$	
FC. 32. Tanh. FC. 8. (s_n)	FC. 32. Tanh. FC. 4. Softmax. (t_n)

$T_\phi^{\text{G}}(x_n, c_n, h_n),\ x_n \in \mathbb{R}^2,\ c_n \in \{0,1\}^4,\ h_n \in [0,1]^1$	
Concatenate$[x_n,\ c_n,\ h_n]$	
FC. 32. Tanh. FC. 8. (s_n)	FC. 32. Tanh. FC. 4. Softmax. (t_n)

The architectures here are similar to those in the GMM in the sense that output of each network also consists of two features: s_n and t_n. The intuition is that the we can extract useful features in the same way, but without conjugacy relationship. Then we compute the outer product of these two features, resulting in a weighted average like the ones in equation 38 as

$$s_n \otimes t_n := \begin{bmatrix} s_{n,1}t_{n,1} & s_{n,1}t_{n,2} & s_{n,1}t_{n,3} & s_{n,1}t_{n,4} \\ s_{n,2}t_{n,1} & s_{n,2}t_{n,2} & s_{n,2}t_{n,3} & s_{n,2}t_{n,4} \\ \cdots & & & \\ s_{n,8}t_{n,1} & s_{n,8}t_{n,2} & s_{n,8}t_{n,3} & s_{n,8}t_{n,4} \end{bmatrix} \tag{52}$$

We aggregate this outer products over all the points by taking an elementwise sum $\sum_{n=1}^N s_n \otimes t_n$. Then we normalize the aggregation by taking an elementwise division with the sum $\sum_{n=1}^N t_n$, i.e.

$$\tilde{T}_\phi^{\text{G}} := \frac{\sum_{n=1}^N s_n \otimes t_n}{\sum_{n=1}^N t_n} \tag{53}$$

where we call the normalized aggregation $\tilde{T}_\phi^{\text{G}} \in \mathbb{R}^{8\times 4}$, where its second dimension corresponds to the number of cluster $M = 4$. Next we employ MLPs $f_\phi^{\text{G}}(\cdot)$ to predict variational distribution of each cluster given the aggregated neural sufficient statistics and the parameters of the priors as

$$\tilde{\mu}_m, \tilde{\sigma}_m^2 \leftarrow f_\phi^{\text{G}}\Big(\tilde{T}_\phi^{\text{G}}(x_n)[:, m],\ \mu,\ \sigma_0\Big), \qquad \tilde{\mu}_m, \tilde{\sigma}_m^2 \leftarrow f_\phi^{\text{G}}\Big(\tilde{T}_\phi^{\text{G}}(x_n, c_n, h_n)[:, m],\ \mu,\ \sigma_0\Big). \tag{54}$$

where $m = 1, 2, ..., M$. The notation $[:, m]$ mean we take the m-th slice along the second dimension which corresponds the m-th cluster.

We parameterize the $f_\phi^{\text{G}}(\cdot)$ using two separate MLPs, each of which is concatenated with the corresponding pointwise neural sufficient statistics networks, i.e. $T_\phi^{\text{G}}(x_n)$ and $T_\phi^{\text{G}}(x_n, c_n, h_n)$ respectively

$f_\phi^{\text{G}}\Big(\tilde{T}_\phi^{\text{G}}(x_n)[:, m],\ \mu,\ \sigma_0\Big),\ \tilde{T}_\phi^{\text{G}}(x_n)[:, m] \in \mathbb{R}^8,\ \mu \in \mathbb{R}^2,\ \sigma_0 \in \mathbb{R}^2$
Concatenate$[\tilde{T}_\phi^{\text{G}}(x_n)[:, m],\ \mu,\ \sigma_0]$
FC. 2×32. Tanh. FC. 2×2

$f_\phi^{\text{G}}\Big(\tilde{T}_\phi^{\text{G}}(x_n, c_n, h_n)[:, m],\ \mu,\ \sigma_0\Big),\ \tilde{T}_\phi^{\text{G}}(x_n, c_n, h_n)[:, m] \in \mathbb{R}^8,\ \mu \in \mathbb{R}^2,\ \sigma_0 \in \mathbb{R}^2$
Concatenate$[\tilde{T}_\phi^{\text{G}}(x_n, c_n, h_n)[:, m],\ \mu,\ \sigma_0]$
FC. 2×32. Tanh. FC. 2×2

We assume a Categorical on mixture assignments c_n and a Beta prior on the embedding of each point h_n,

$$c_n \sim \text{Categorical}(\pi), \qquad h_n \sim \text{Beta}(\alpha, \beta). \tag{55}$$

where $\pi = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}), \alpha = 1$ and $\beta = 1$.

The neural sufficient statistics is defined as

$$T^{\text{L}}_\phi(x_n, \mu_{1:M}) := \big(T^{\text{L}}_\phi(x_n, \mu_1),\ T^{\text{L}}_\phi(x_n, \mu_2),\ ...,\ T^{\text{L}}_\phi(x_n, \mu_M)\big) \tag{56}$$

where each element is parameterized by the network

$T^{\text{L}}_\phi(x_n, \mu_m),\ x_n \in \mathbb{R}^2,\ \mu_m \in \mathbb{R}^2$
Concatenate$[x_n,\ \mu_m]$
FC. 32. Tanh. FC. 1.

We compute the parameters of the conditional proposal $\tilde{\pi}_n$ by computing the logits normalizing it using Softmax activation

$$\tilde{\pi}_n = \text{Softmax}\Big(\log \pi + T^{\text{L}}_\phi(x_n,\ \mu_{1:M})\Big). \tag{57}$$

Then we can sample assignments c_n from the variational distribution $c_n \sim \text{Categorical}(\tilde{\pi}_n)$. The neural proposal for embedding variable h_n is conditioned on the assignments c_n in a way that it takes as input the mean of the cluster μ_m, to which each point belongs to, i.e.

$$q_\phi(h_n \mid x_n, \mu_{1:M}, c_n) = q_\phi(h_n \mid x_n, \mu_{c_n}). \tag{58}$$

As a result, the network for this neural proposal is

$q_\phi(h_n \mid x_n, \mu_{c_n}),\ x_n \in \mathbb{R}^2,\ \mu_{c_n} \in \mathbb{R}^2$
$x_n - \mu_{c_n}$
FC. 2×32. Tanh. FC. 2×1.

Since the Beta distribution requires its parameters to be positive, this network will output the logarithms of proposal parameters for h_n of the form

$$\log \tilde{\alpha}_n,\ \log \tilde{\beta}_n \leftarrow q_\phi(h_n \mid x_n, \mu_{c_n}). \tag{59}$$

In this experiment we also learn a deep generative model of the form

$$p_\theta(x_{1:N} | \mu_{1:M}, c_{1:N}, h_{1:N}) := \prod_{n=1}^{N} \text{Normal}\Big(x_n \ \Big|\ g_\theta(h_n) + \mu_{c_n}, \sigma^2_\epsilon I\Big). \tag{60}$$

where $\sigma_\epsilon = 0.1$ is a hyper-parameter. The architecture of the MLP decoder $g_\theta(h_n)$ is

$g_\theta(h_n),\ h_n \in \mathbb{R}^1$
FC. 32. Tanh. FC. 2.

F.3. Time Series Model – Bouncing MNIST

We learn a deep generative model of the form

$$p_\theta(x_{1:T} \mid z^{\text{what}}_{1:D}, z^{\text{where}}_{1:T}) = \prod_{t=1}^{T} \text{Bernoulli}\Big(x_t \;\Big|\; \sigma\Big(\sum_d \text{ST}\big(g_\theta(z^{\text{what}}_d),\; z^{\text{where}}_{d,t}\big)\Big)\Big) \tag{61}$$

Given each digit feature z^{what}_d, the APG sampler reconstruct a 28×28 MNIST image using a MLP decoder, the architecture of which is

$g_\theta(z^{\text{what}}_d),\; z^{\text{what}}_d \in \mathbb{R}^{10}$
FC. 200. ReLU.
FC. 400. ReLU.
FC. 784. Sigmoid.

Then we put each reconstructed image $g_\theta(z^{\text{what}}_d)$ onto a 96×96 canvas using a spatial transformer ST which takes position variable $z^{\text{where}}_{d,t}$ as input. To ensure a pixel-wise Bernoulli likelihood, we clip on the composition as

$$\text{For each pixel } p_i \in \Big(\sum_d \text{ST}\big(g_\theta(z^{\text{what}}_d),\; z^{\text{where}}_{d,t}\big)\Big),\; \sigma(p_i) = \begin{cases} p_i = 0 & \text{if } p_i < 0 \\ p_i = p_i & \text{if } 0 \le p_i \le 1 \\ p_i = 1 & \text{if } p_i > 1 \end{cases} \tag{62}$$

The APG sampler in the bouncing MNIST employs neural proposals of the form

$$q_\phi(z^{\text{where}}_{1:D,t} \mid x_t) = \prod_{d=1}^{D} \text{Normal}\Big(z^{\text{where}}_{d,t} \;\Big|\; \tilde{\mu}^{\text{where}}_{d,t}, \tilde{\sigma}^{\text{where}\,2}_{d,t} I\Big), \qquad \text{for } t = 1, 2, \ldots, T, \tag{63}$$

$$q_\phi(z^{\text{where}}_{1:D,t} \mid x_t, z^{\text{what}}_{1:D}) = \prod_{d=1}^{D} \text{Normal}\Big(z^{\text{where}}_{d,t} \;\Big|\; \tilde{\mu}^{\text{where}}_{d,t}, \tilde{\sigma}^{\text{where}\,2}_{d,t} I\Big), \qquad \text{for } t = 1, 2, \ldots, T, \tag{64}$$

$$q_\phi(z^{\text{what}}_{1:D} \mid x_{1:T}, z^{\text{where}}_{1:T}) = \prod_{d=1}^{D} \text{Normal}\Big(z^{\text{what}}_{d} \;\Big|\; \tilde{\mu}^{\text{what}}_{d}, \tilde{\sigma}^{\text{what}\,2}_{d} I\Big). \tag{65}$$

We train the proposals with instances containing $D = 3$ digits and $T = 10$ time steps and test them with instances containing up to $D = 5$ digits and $T = 100$ time steps. We use the tilde symbol $\tilde{}$ to denote the parameters of the conditional neural proposals (i.e. approximate Gibbs proposals).

The APG sampler uses these proposals to iterate over the $T + 1$ blocks

$$\{z^{\text{what}}_{1:D}\},\; \{z^{\text{where}}_{1:D,1}\},\; \{z^{\text{where}}_{1:D,2}\},\; \ldots,\; \{z^{\text{where}}_{1:D,T}\}.$$

For the position features, the proposal $q_\phi(z^{\text{where}}_{1:D,t} \mid x_t)$ and proposal $q_\phi(z^{\text{where}}_{1:D,t} \mid x_t, z^{\text{what}}_{1:D})$ share the same network, but contain different pre-steps where we compute the input of that network. The initial proposal $q_\phi(z^{\text{where}}_{1:D,t} \mid x_t)$ will convolve the frame x_t with the mean image of the MNIST dataset; The conditional proposal $q_\phi(z^{\text{where}}_{1:D,t} \mid x_t, z^{\text{what}}_{1:D})$ will convolve the frame x_t with each reconstructed MNIST image $g_\theta(z^{\text{what}}_d)$. We perform convolution sequentially by looping over all digits $d = 1, 2, ..., D$. Here is pseudocode of both pre-steps:

Algorithm 4 Convolution Processing for $q_\phi(z^{\text{where}}_{1:D,t} \mid x_t)$

Input frame $x_t \in \mathbb{R}^{9216}$, mean image of MNIST dataset $mm \in \mathbb{R}^{784}$
for $d = 1$ **to** D **do**
    $x^{\text{conv}}_{d,t} \leftarrow$ Conv2d(x_t) with kernel mm, stride = 1, no padding.
end for
Output Convolved features $\{x^{\text{conv}}_{d,t} \in \mathbb{R}^{4761}\}_{d=1}^{D}$

Algorithm 5 Convolution Processing for $q_\phi(z^{\text{where}}_{1:D,t} \mid x_t, z^{\text{what}}_{1:D})$

Input frame $x_t \in \mathbb{R}^{9216}$, reconstructed MNIST digits $\{g_\theta(z^{\text{what}}_d) \in \mathbb{R}^{784}\}_{d=1}^{D}$
for $d = 1$ **to** D **do**
    $x^{\text{conv}}_{d,t} \leftarrow$ Conv2d(x_t) with kernel $g_\theta(z^{\text{what}}_d)$, stride = 1, no padding.
end for
Output Convolved features $\{x^{\text{conv}}_{d,t} \in \mathbb{R}^{4761}\}_{d=1}^{D}$

We employ a MLP encoder $f^{\text{L}}_\phi(\cdot)$ that takes the convolved features as input and predict the variational parameters for positions $\{z^{\text{where}}_{d,t}\}_{d=1}^{D}$ at step t, i.e. vector-valued mean $\tilde{\mu}^{\text{where}}_{d,t}$ and logarithm of the diagonal covariance $\log \tilde{\sigma}^{\text{where}\,2}_{d,t}$ as

$$\tilde{\mu}^{\text{where}}_{d,t}, \log \tilde{\sigma}^{\text{where}\,2}_{d,t} \leftarrow f^{\text{L}}_\phi(x^{\text{conv}}_{d,t}), \qquad d = 1, 2, \ldots, D. \tag{66}$$

The architecture of the MLP encoder $f^{\text{L}}_\phi(\cdot)$ is

$f^{\text{L}}_\phi(x^{\text{conv}}_{d,t}),\ x^{\text{conv}}_{d,t} \in \mathbb{R}^{4761}$
FC. 200. ReLU.
FC. 2×100. ReLU.
FC. 2×2.

For the digit features, the APG sampler performs conditional updates in the sense that we crop each frame x_t into a 28×28 subframe according to $z^{\text{where}}_{d,t}$ using the spatial transformer ST as

$$x^{\text{crop}}_{d,t} \leftarrow \text{ST}\big(x_t, z^{\text{where}}_{d,t}\big), \qquad d = 1, 2, \ldots, D, \quad t = 1, 2, \ldots, T. \tag{67}$$

we employ a MLP encoder $T^{\text{G}}_\phi(\cdot)$ that takes the cropped subframes as input, and predicts frame-wise neural sufficient statistics, which we will sum up over all the time steps. The architecture of this network is

$T^{\text{G}}_\phi(x^{\text{crop}}_{d,t}),\ x^{\text{crop}}_{d,t} \in \mathbb{R}^{784}$
FC. 400. ReLU.
FC. 200. ReLU.

Then we employ another network $f^{\text{G}}_\phi(\cdot)$ that takes the sums as input, and predict the variational parameters for digit features $\{z^{\text{what}}_d\}_{d=1}^{D}$, i.e. the vector-valued means $\{\tilde{\mu}^{\text{what}}_d\}_{d=1}^{D}$ and the logarithms of the diagonal covariances $\{\log \tilde{\sigma}^{\text{what}\,2}_d\}_{d=1}^{D}$. The architecture of this network is

$f^{\text{G}}_\phi(\sum_{t=1}^{T} T^{\text{G}}_\phi(x^{\text{crop}}_{d,t})),\ \sum_{t=1}^{T} T^{\text{G}}_\phi(x^{\text{crop}}_{d,t}) \in \mathbb{R}^{200}$
FC. 2×10

G. Analytical inclusive KL divergence during Training in GMM

In GMM experiment, we train the model with different number of sweeps K under fixed computational budget $K \cdot L = 100$. Figure 5 shows that more number of sweeps results in slightly faster convergence.

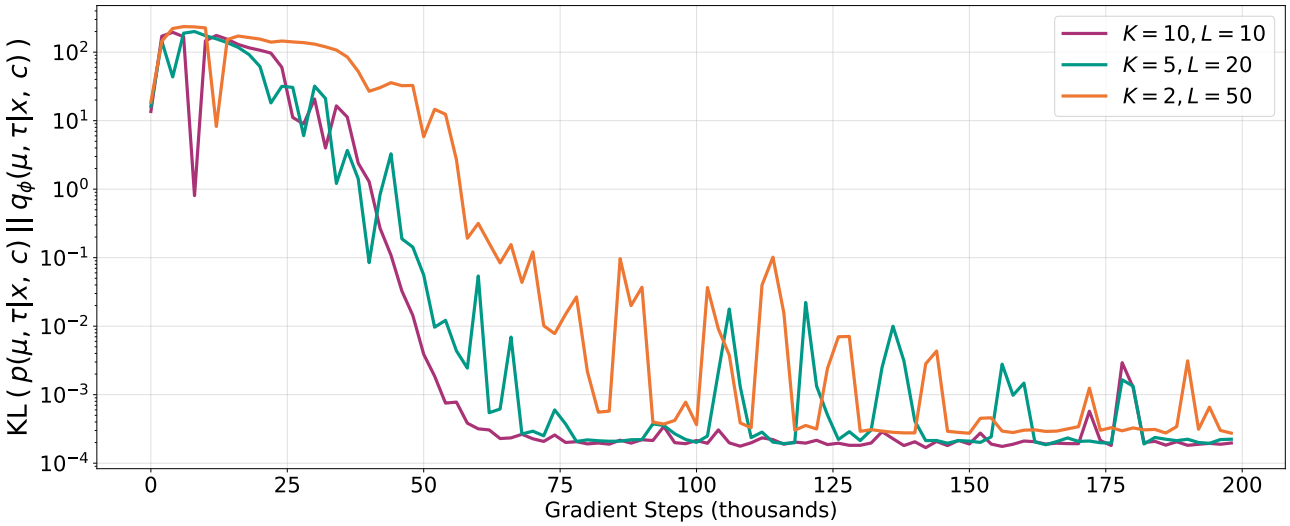

Figure 5. Inclusive KL divergence as a function of gradient steps. Each model is trained with 20000 gradient steps, $K \cdot L = 100$

H. Comparison between APG sampler and RWS method in Bouncing MNIST

We visualize the inference results and reconstruction from the APG sampler and reweighted wake-sleep method. We can see that APG sampler significantly improves both tracking inference results and the reconstruction on instances with $T = 20$ time steps and $D = 5$ digits. We can see that the APG sampler achieves substantial results while the RWS method does not make reasonable prediction at all.

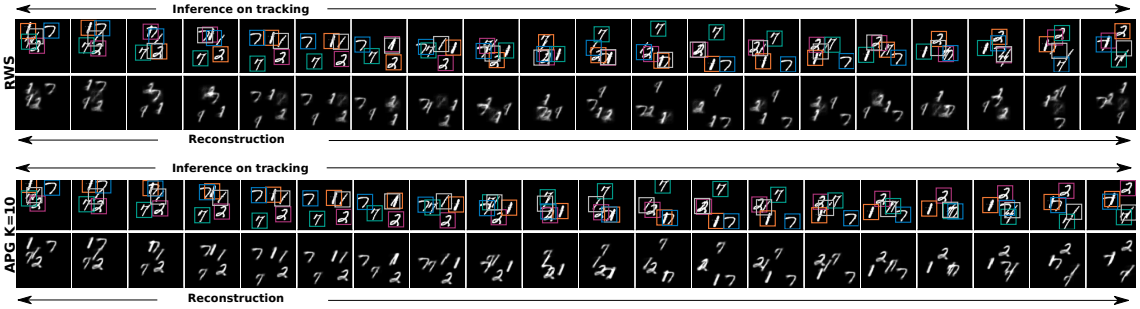

Figure 6. Example 1.

I. Inference Results and Reconstruction on large time steps Bouncing MNIST

We show the inference results on tracking and the reconstruction on test instances with $T = 100$ time steps and $D = 3, 4, 5$ MNIST digits, using models that is trained with instances containing only $T = 10$ time steps and $D = 3$ digits. In each figure below, the 1st, 3rd, 5th, 7th, 9th rows show the inference results, while the other rows show the reconstruction of the series above. We can see the APG sampler is scalable with much large number of latent variables, achieving accurate inference and making impressive reconstruction.

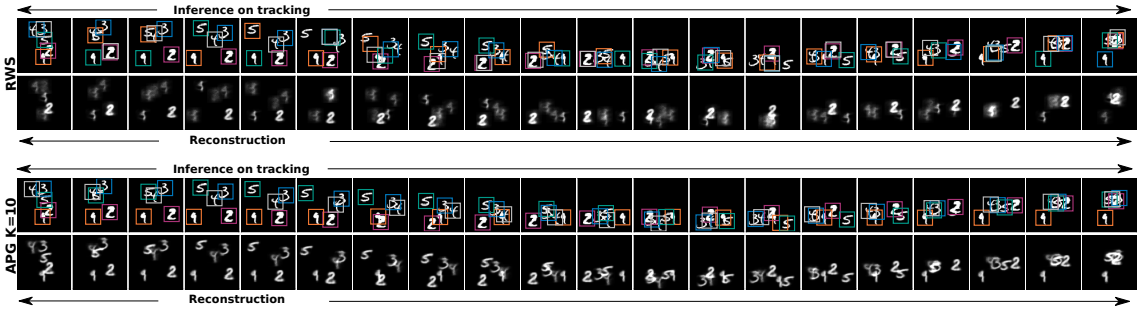

Figure 7. Example 2.

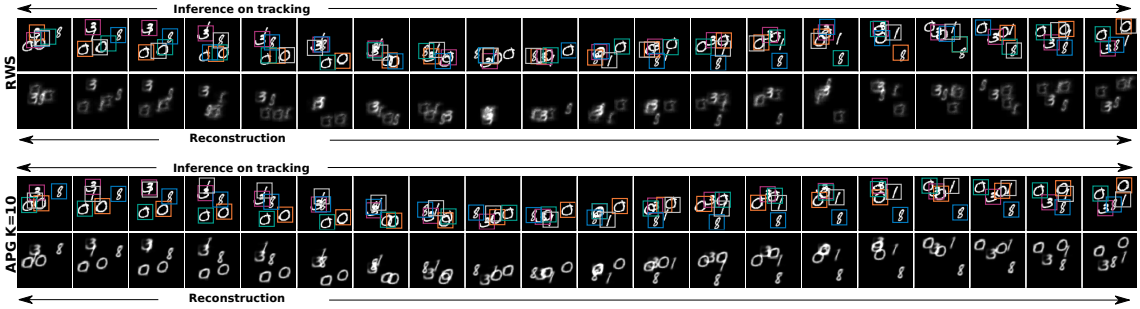

Figure 8. Example 3.

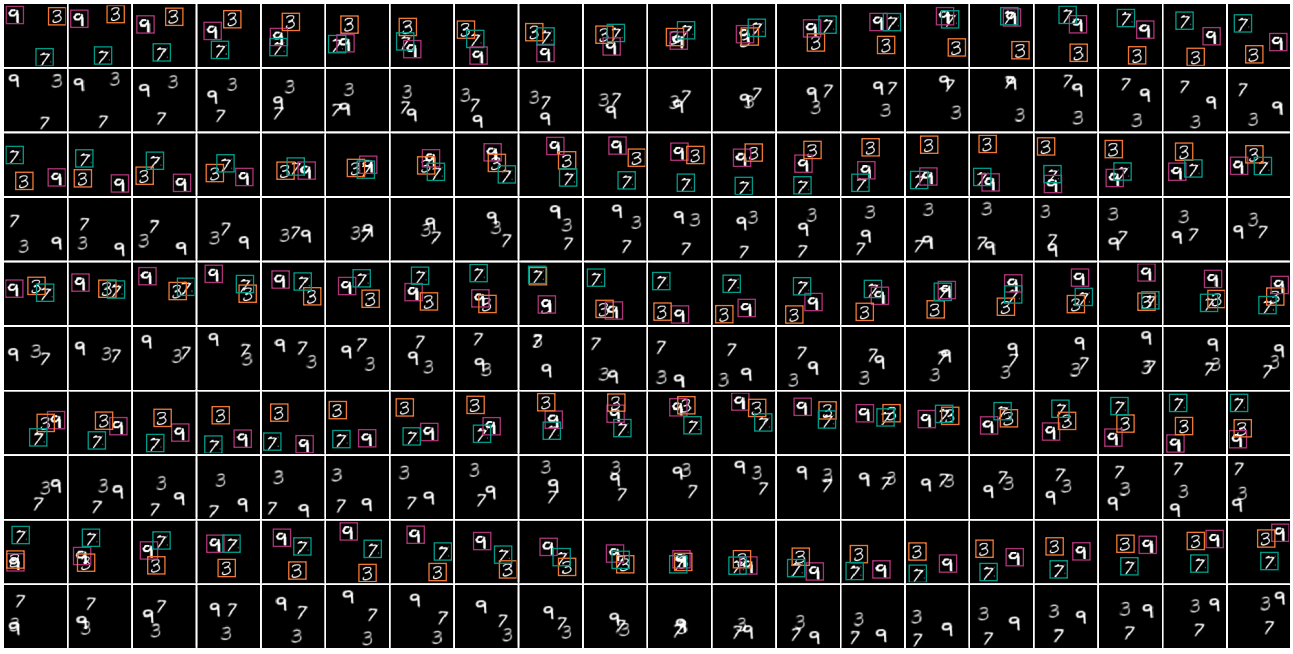

Figure 9. Full reconstruction for a video where $T = 100, D = 3$.

Figure 10. Full reconstruction for a video where $T = 100, D = 4$.

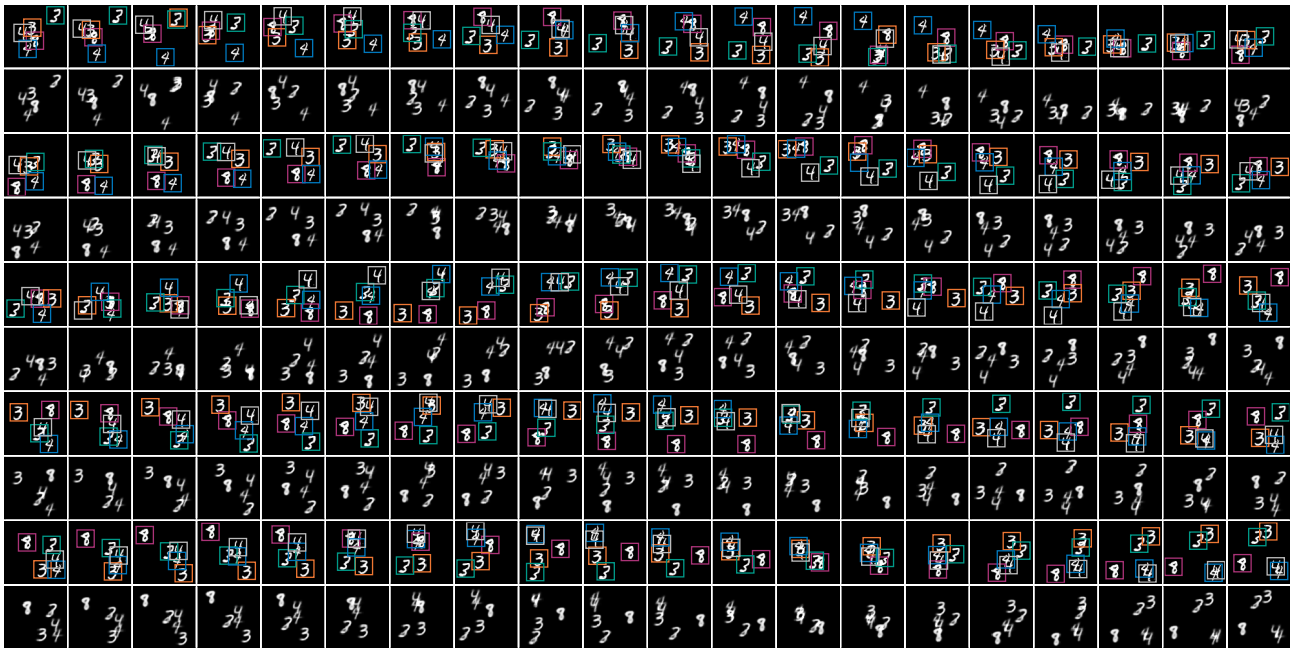

Figure 11. Full reconstruction for a video where $T = 100, D = 5$.

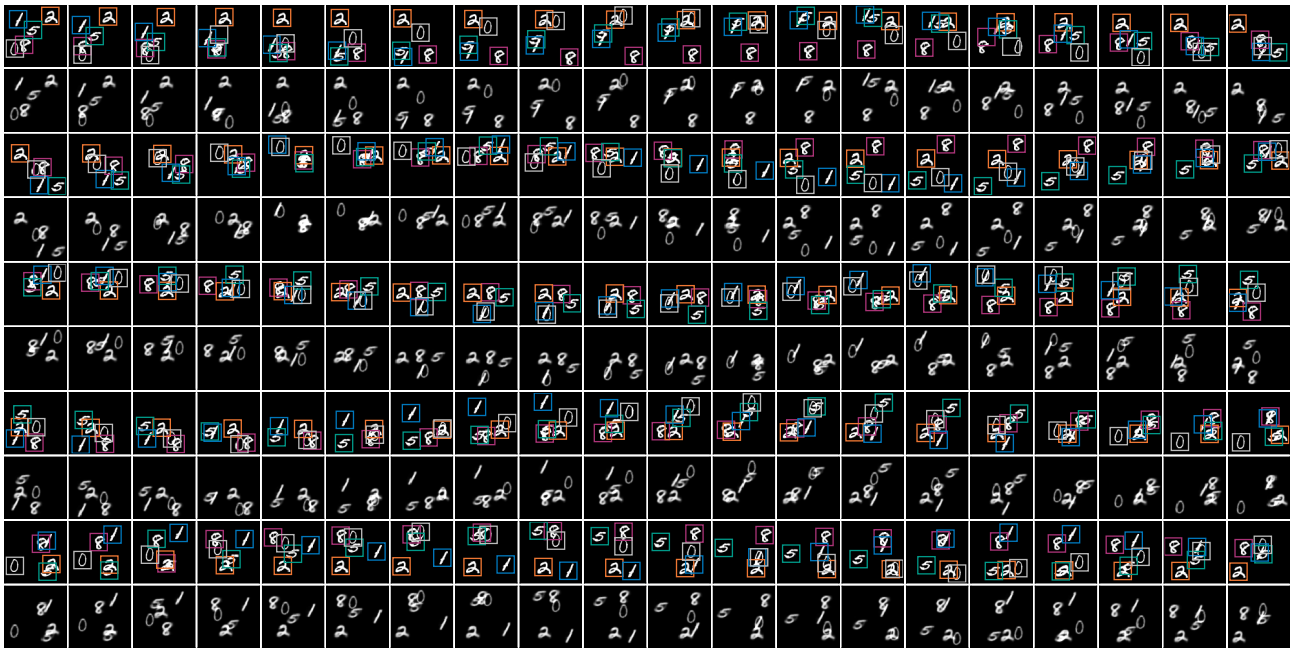

Figure 12. Full reconstruction for a video where $T = 100$, $D = 5$.