Supplemental Material for: Deep Generative Stochastic Networks Trainable by Backprop

Yoshua Bengio[*] FIND.US@ON.THE.WEB
Éric Thibodeau-Laufer
Guillaume Alain
Département d'informatique et recherche opérationnelle, Université de Montréal,[*] & Canadian Inst. for Advanced Research

Jason Yosinski
Department of Computer Science, Cornell University

A. Generative denoising autoencoders with local noise

The main theorem in Bengio et al. (2013) (reproduced below as Theorem S1) requires the Markov chain to be ergodic. Sufficient conditions to guarantee ergodicity are given in the aforementioned paper, but they are somewhat restrictive, requiring $\mathcal{C}(\tilde{X}|X) > 0$ everywhere that $P(X) > 0$. Here we show how to relax these conditions and still obtain ergodicity.

Let $P_{\theta_n}(X|\tilde{X})$ be a denoising auto-encoder that has been trained on n training examples. $P_{\theta_n}(X|\tilde{X})$ assigns a probability to X, given $\tilde{X}$, when $\tilde{X} \sim \mathcal{C}(\tilde{X}|X)$. This estimator defines a Markov chain T_n obtained by sampling alternatively an $\tilde{X}$ from $\mathcal{C}(\tilde{X}|X)$ and an X from $P_\theta(X|\tilde{X})$. Let π_n be the asymptotic distribution of the chain defined by T_n, if it exists. The following theorem is proven by Bengio et al. (2013).

Theorem S1. **If** $P_{\theta_n}(X|\tilde{X})$ *is a consistent estimator of the true conditional distribution* $P(X|\tilde{X})$ **and** T_n *defines an ergodic Markov chain,* **then** *as* $n \to \infty$*, the asymptotic distribution* $\pi_n(X)$ *of the generated samples converges to the data-generating distribution* $P(X)$*.*

In order for Theorem S1 to apply, the chain must be ergodic. One set of conditions under which this occurs is given in the aforementioned paper. We slightly restate them here:

Corollary 1. **If** *the support for both the data-generating distribution and denoising model are contained in and non-zero in a finite-volume region* V *(i.e.,* $\forall \tilde{X}, \forall X \notin V,\ P(X) = 0, P_\theta(X|\tilde{X}) = 0$ *and* $\forall \tilde{X}, \forall X \in V,\ P(X) > 0, P_\theta(X|\tilde{X}) > 0, \mathcal{C}(\tilde{X}|X) > 0$*)* **and** *these statements remain true in the limit of* $n \to \infty$*,* **then** *the chain defined by* T_n *will be ergodic.*

If conditions in Corollary 1 apply, then the chain will be

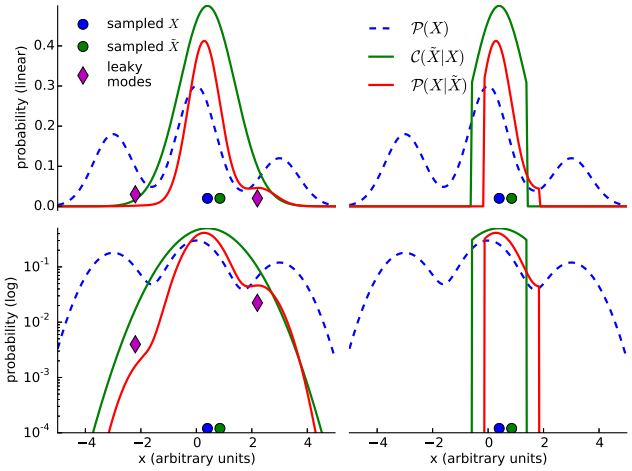


Figure 1. If $\mathcal{C}(\tilde{X}|X)$ is globally supported as required by Corollary 1 (Bengio et al., 2013), then for $P_{\theta_n}(X|\tilde{X})$ to converge to $P(X|\tilde{X})$, it will eventually have to model all of the modes in $P(X)$, even though the modes are damped (see "leaky modes" on the left). However, if we guarantee ergodicity through other means, as in Corollary 2, we can choose a local $\mathcal{C}(\tilde{X}|X)$ and allow $P_{\theta_n}(X|\tilde{X})$ to model only the local structure of $P(X)$ (see right).

ergodic and Theorem S1 will apply. However, these conditions are sufficient, not necessary, and in many cases they may be artificially restrictive. In particular, Corollary 1 defines a large region V containing any possible X allowed by the model and requires that we maintain the probability of jumping between any two points in a single move to be greater than 0. While this generous condition helps us easily guarantee the ergodicity of the chain, it also has the unfortunate side effect of requiring that, in order for $P_{\theta_n}(X|\tilde{X})$ to converge to the conditional distribution $P(X|\tilde{X})$, it must have the capacity to model every mode of $P(X)$, exactly the difficulty we were trying to avoid. The left two plots in Figure 1 show this difficulty: because $\mathcal{C}(\tilde{X}|X) > 0$ everywhere in V, every mode of $P(X)$ will leak, perhaps attenuated, into $P(X|\tilde{X})$.

Fortunately, we may seek ergodicity through other means. The following corollary allows us to choose a $\mathcal{C}(\tilde{X}|X)$ that only makes small jumps, which in turn only requires $P_\theta(X|\tilde{X})$ to model a small part of the space V around each $\tilde{X}$.

Let $P_{\theta_n}(X|\tilde{X})$ be a denoising auto-encoder that has been trained on n training examples and $\mathcal{C}(\tilde{X}|X)$ be some corruption distribution. $P_{\theta_n}(X|\tilde{X})$ assigns a probability to X, given $\tilde{X}$, when $\tilde{X} \sim \mathcal{C}(\tilde{X}|X)$ and $X \sim \mathcal{P}(X)$. Define a Markov chain T_n by alternately sampling an $\tilde{X}$ from $\mathcal{C}(\tilde{X}|X)$ and an X from $P_\theta(X|\tilde{X})$.

Corollary 2. If *the data-generating distribution is contained in and non-zero in a finite-volume region V (i.e., $\forall X \notin V,\ P(X) = 0$, and $\forall X \in V,\ P(X) > 0$)* **and** *all pairs of points in V can be connected by a finite-length path through V* **and** *for some $\epsilon > 0$, $\forall \tilde{X} \in V, \forall X \in V$ within ϵ of each other, $\mathcal{C}(\tilde{X}|X) > 0$ and $P_\theta(X|\tilde{X}) > 0$* **and** *these statements remain true in the limit of $n \to \infty$,* **then** *the chain defined by T_n will be ergodic.*

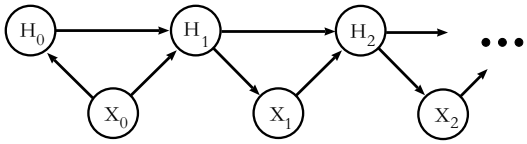
Proof. Consider any two points X_a and X_b in V. By the assumptions of Corollary 2, there exists a finite length path between X_a and X_b through V. Pick one such finite length path P. Chose a finite series of points $x = \{x_1, x_2, \ldots, x_k\}$ along P, with $x_1 = X_a$ and $x_k = X_b$ such that the distance between every pair of consecutive points (x_i, x_{i+1}) is less than ϵ as defined in Corollary 2. Then the probability of sampling $\tilde{X} = x_{i+1}$ from $\mathcal{C}(\tilde{X}|x_i))$ will be positive, because $\mathcal{C}(\tilde{X}|X)) > 0$ for all $\tilde{X}$ within ϵ of X by the assumptions of Corollary 2. Further, the probability of sampling $X = \tilde{X} = x_{i+1}$ from $P_\theta(X|\tilde{X})$ will be positive from the same assumption on P. Thus the probability of jumping along the path from x_i to x_{i+1}, $T_n(X_{t+1} = x_{i+1}|X_t = x_i)$, will be greater than zero for all jumps on the path. Because there is a positive probability finite length path between all pairs of points in V, all states commute, and the chain is irreducible. If we consider $X_a = X_b \in V$, by the same arguments $T_n(X_t = X_a|X_{t-1} = X_a) > 0$. Because there is a positive probability of remaining in the same state, the chain will be aperiodic. Because the chain is irreducible and over a finite state space, it will be positive recurrent as well. Thus, the chain defined by T_n is ergodic. □

Although this is a weaker condition that has the advantage of making the denoising distribution even easier to model (probably having less modes), we must be careful to choose the ball size ϵ large enough to guarantee that one can jump often enough between the major modes of $P(X)$ when these are separated by zones of tiny probability. ϵ must be larger than half the largest distance one would have to travel across a desert of low probability separating two nearby modes (which if not connected in this way would make V not anymore have a single connected component). Practically, there would be a trade-off between the difficulty of estimating $P(X|\tilde{X})$ and the ease of mixing between major modes separated by a very low density zone.

B. Supplemental Theorem Proofs

Theorem 2 was stated in the paper without proof. We reproduce it here, show a proof, and then discuss its robustness in a context in which we train on a finite number of samples.

Theorem 2. *Let $(H_t, X_t)_{t=0}^{\infty}$ be the Markov chain defined by the following graphical model.*



If we assume that the chain has a stationary distribution $\pi_{H,X}$, and that for every value of (x, h) we have that

- *all the $P(X_t = x|H_t = h) = g(x, h)$ share the same density for $t \geq 1$*
- *all the $P(H_{t+1} = h|H_t = h', X_t = x) = f(h, h', x)$ shared the same density for $t \geq 0$*
- $P(H_0 = h|X_0 = x) = P(H_1 = h|X_0 = x)$
- $P(X_1 = x|H_1 = h) = P(X_0 = x|H_1 = h)$

then for every value of (x, h) we get that

- $P(X_0 = x|H_0 = h) = g(x, h)$ *holds, which is something that was assumed only for $t \geq 1$*
- $P(X_t = x, H_t = h) = P(X_0 = x, H_0 = h)$ *for all $t \geq 0$*

- *the stationary distribution $\pi_{H,X}$ has a marginal distribution π_X such that $\pi(x) = P(X_0 = x)$.*

Those conclusions show that our Markov chain has the property that its samples in X are drawn from the same distribution as X_0.

Proof. The proof hinges on a few manipulations done with the first variables to show that $P(X_t = x|H_t = h) = g(x, h)$, which is assumed for $t \geq 1$, also holds for $t = 0$.

For all h we have that

$$\begin{aligned} P(H_0 = h) &= \int P(H_0 = h|X_0 = x)P(X_0 = x)dx \\ &= \int P(H_1 = h|X_0 = x)P(X_0 = x)dx \\ &= P(H_1 = h). \end{aligned}$$

The equality in distribution between (X_1, H_1) and (X_0, H_0) is obtained with

$$\begin{aligned} P(X_1 = x, H_1 = h) &= P(X_1 = x|H_1 = h)P(H_1 = h) \\ &= P(X_0 = x|H_1 = h)P(H_1 = h) \\ &\quad \text{(by hypothesis)} \\ &= P(X_0 = x, H_1 = h) \\ &= P(H_1 = h|X_0 = x)P(X_0 = x) \\ &= P(H_0 = h|X_0 = x)P(X_0 = x) \\ &\quad \text{(by hypothesis)} \\ &= P(X_0 = x, H_0 = h). \end{aligned}$$

Then we can use this to conclude that

$$\begin{aligned} & P(X_0 = x, H_0 = h) = P(X_1 = x, H_1 = h) \\ \implies \quad & P(X_0 = x|H_0 = h) = P(X_1 = x|H_1 = h) = g(x, h) \end{aligned}$$

so, despite the arrow in the graphical model being turned the other way, we have that the density of $P(X_0 = x|H_0 = h)$ is the same as for all other $P(X_t = x|H_t = h)$ with $t \geq 1$.

Now, since the distribution of H_1 is the same as the distribution of H_0, and the transition probability $P(H_1 = h|H_0 = h')$ is entirely defined by the (f, g) densities which are found at every step for all $t \geq 0$, then we know that (X_2, H_2) will have the same distribution as (X_1, H_1). To make this point more explicitly,

$$\begin{aligned} & P(H_1 = h|H_0 = h') \\ = \quad & \int P(H_1 = h|H_0 = h', X_0 = x)P(X_0 = x|H_0 = h')dx \\ = \quad & \int f(h, h', x)g(x, h')dx \\ = \quad & \int P(H_2 = h|H_1 = h', X_1 = x)P(X_1 = x|H_1 = h')dx \\ = \quad & P(H_2 = h|H_1 = h') \end{aligned}$$

This also holds for $P(H_3|H_2)$ and for all subsequent $P(H_{t+1}|H_t)$. This relies on the crucial step where we demonstrate that $P(X_0 = x|H_0 = h) = g(x, h)$. Once this was shown, then we know that we are using the same transitions expressed in terms of (f, g) at every step.

Since the distribution of H_0 was shown above to be the same as the distribution of H_1, this forms a recursive argument that shows that all the H_t are equal in distribution to H_0. Because $g(x, h)$ describes every $P(X_t = x|H_t = h)$, we have that all the joints (X_t, H_t) are equal in distribution to (X_0, H_0).

This implies that the stationary distribution $\pi_{X,H}$ is the same as that of (X_0, H_0). Their marginals with respect to X are thus the same. □

To apply Theorem 2 in a context where we use experimental data to learn a model, we would like to have certain guarantees concerning the robustness of the stationary density π_X. When a model lacks capacity, or when it has seen only a finite number of training examples, that model can be viewed as a perturbed version of the exact quantities found in the statement of Theorem 2.

A good overview of results from perturbation theory discussing stationary distributions in finite state Markov chains can be found in (Cho et al., 2000). We reference here only one of those results.

Theorem 3. *Adapted from (Schweitzer, 1968)*

Let K be the transition matrix of a finite state, irreducible, homogeneous Markov chain. Let π be its stationary distribution vector so that $K\pi = \pi$. Let $A = I - K$ and $Z = (A + C)^{-1}$ where C is the square matrix whose columns all contain π. Then, if $\tilde{K}$ is any transition matrix (that also satisfies the irreducible and homogeneous conditions) with stationary distribution $\tilde{\pi}$, we have that

$$\|\pi - \tilde{\pi}\|_1 \leq \|Z\|_\infty \left\|K - \tilde{K}\right\|_\infty .$$

This theorem covers the case of discrete data by showing how the stationary distribution is not disturbed by a great amount when the transition probabilities that we learn are close to their correct values. We are talking here about the transition between steps of the chain $(X_0, H_0), (X_1, H_1), \ldots, (X_t, H_t)$, which are defined in Theorem 2 through the (f, g) densities.

We avoid discussing the training criterion for a GSN. Various alternatives exist, but this analysis is for future work. Right now Theorem 2 suggests the following rules :

- Pick the transition distribution $f(h, h', x)$ to be useful (e.g. through training that maximizes reconstruction likelihood).

- Make sure that during training $P(H_0 = h|X_0 = x) \rightarrow P(H_1 = h|X_0 = x)$. One interesting way to achieve this is, for each X_0 in the training set, iteratively sample $H_1|(H_0, X_0)$ and substitute the value of H_1 as the updated value of H_0. Repeat until you have achieved a kind of "burn in". Note that, after the training is completed, when we use the chain for sampling, the samples that we get from its stationary distribution do not depend on H_0. This technique of substituting the H_1 into H_0 does not apply beyond the training step.

- Define $g(x, h)$ to be your estimator for $P(X_0 = x|H_1 = h)$, e.g. by training an estimator of this conditional distribution from the samples (X_0, H_1).

- The rest of the chain for $t \geq 1$ is defined in terms of (f, g).

As much as we would like to simply learn g from pairs (H_0, X_0), the problem is that the training samples $X_0^{(i)}$ are descendants of the corresponding values of $H_0^{(i)}$ in the original graphical model that describes the GSN. Those $H_0^{(i)}$ are hidden quantities in GSN and we have to find a way to deal with them. Setting them all to be some default value would not work because the relationship between H_0 and X_0 would not be the same as the relationship later between H_t and X_t in the chain.

Proposition 1 was stated in the paper without proof. We reproduce it here and then show a proof:

Proposition 1. If *a subset $x^{(s)}$ of the elements of X is kept fixed (not resampled) while the remainder $X^{(-s)}$ is updated stochastically during the Markov chain of Theorem 2, but using $P(X_t|H_t, X_t^{(s)} = x^{(s)})$,* **then** *the asymptotic distribution π_n of the Markov chain produces samples of $X^{(-s)}$ from the conditional distribution $\pi_n(X^{(-s)}|X^{(s)} = x^{(s)})$.*

Proof. Without constraint, we know that at convergence, the chain produces samples of π_n. A subset of these samples satisfies the condition $X = x^{(s)}$, and these constrained samples could equally have been produced by sampling X_t from $P_{\theta_2}(X_t|f_{\theta_1}(X_{t-1}, Z_{t-1}, H_{t-1}), X_t^{(s)} = X^{(s)})$, by definition of conditional distribution. Therefore, at convergence of the chain, we have that using the constrained distribution $P(X_t|f(X_{t-1}, Z_{t-1}, H_{t-1}), X_t^{(s)} = x^{(s)})$ produces a sample from π_n under the condition $X^{(s)} = x^{(s)}$. □

C. Supplemental Experimental Results

Experiments evaluating the ability of the GSN models to generate good samples were performed on the MNIST and TFD datasets, following the setup in Bengio et al. (2013c). Theorem 2 requires H_0 to have the same distribution as H_1 (given X_0) during training, and the main paper suggests a way to achieve this by initializing each training chain with H_0 set to the previous value of H_1 when the same example X_0 was shown. However, we did not implement that procedure in the experiments below, so that is left for future work to explore.

Networks with 2 and 3 hidden layers were evaluated and compared to regular denoising auto-encoders (just 1 hidden layer, i.e., the computational graph separates into separate ones for each reconstruction step in the walkback algorithm). They all have $\tanh$ hidden units and pre- and post-activation Gaussian noise of standard deviation 2, applied to all hidden layers except the first. In addition, at each step in the chain, the input (or the resampled X_t) is corrupted with salt-and-pepper noise of 40% (i.e., 40% of the pixels are corrupted, and replaced with a 0 or a 1 with probability 0.5). Training is over 100 to 600 epochs at most, with good results obtained after around 100 epochs, using stochastic gradient descent (minibatch size = 1). Hidden layer sizes vary between 1000 and 1500 depending on the experiments, and a learning rate of 0.25 and momentum of 0.5 were selected to approximately minimize the reconstruction negative log-likelihood. The learning rate is reduced multiplicatively by 0.99 after each epoch. Following Breuleux et al. (2011), the quality of the samples was also estimated quantitatively by measuring the log-likelihood of the test set under a Parzen density estimator constructed from 10000 consecutively generated samples (using the real-valued mean-field reconstructions as the training data for the Parzen density estimator). This can be seen as an *lower bound on the true log-likelihood*, with the bound converging to the true likelihood as we consider more samples and appropriately set the smoothing parameter of the Parzen estimator[1]. Results are summarized in Table 1. The test set Parzen log-likelihood bound was not used to select among model architectures, but visual inspection of samples generated did guide the preliminary search reported here. Optimization hyper-parameters (learning rate, momentum, and learning rate reduction schedule) were selected based on the reconstruction log-likelihood training objective. The Parzen log-likelihood bound

[1]However, in this paper, to be consistent with the numbers given in Bengio et al. (2013c) we used a Gaussian Parzen density, which (in addition to being lower rather than upper bounds) makes the numbers not comparable with the AIS log-likelihood upper bounds for binarized images reported in some papers for the same data.

obtained with a two-layer model on MNIST is 214 ($\pm$ standard error of 1.1), while the log-likelihood bound obtained by a single-layer model (regular denoising auto-encoder, DAE in the table) is substantially worse, at -152$\pm$2.2. In comparison, Bengio et al. (2013c) report a log-likelihood bound of -244$\pm$54 for RBMs and 138$\pm$2 for a 2-hidden layer DBN, using the same setup. We have also evaluated a 3-hidden layer DBM (Salakhutdinov & Hinton, 2009), using the weights provided by the author, and obtained a Parzen log-likelihood bound of 32$\pm$2. See `http://www.mit.edu/~rsalakhu/DBM.html` for details. Figure 6 shows two runs of consecutive samples from this trained model, illustrating that it mixes quite well (better than RBMs) and produces rather sharp digit images. The figure shows that it can also stochastically complete missing values: the left half of the image was initialized to random pixels and the right side was clamped to an MNIST image. The Markov chain explores plausible variations of the completion according to the trained conditional distribution.

A smaller set of experiments was also run on TFD, yielding for a GSN a test set Parzen log-likelihood bound of 1890 $\pm$29. The setup is exactly the same and was not tuned after the MNIST experiments. A DBN-2 yields a Parzen log-likelihood bound of 1908 $\pm$66, which is undistinguishable statistically, while an RBM yields 604 $\pm$ 15. A run of consecutive samples from the GSN-3 model are shown in Figure 8.

References

Bengio, Yoshua, Yao, Li, Alain, Guillaume, and Vincent, Pascal. Generalized denoising auto-encoders as generative models. In *NIPS26*. Nips Foundation, 2013.

Cho, Grace E., Meyer, Carl D., Carl, and Meyer, D. Comparison of perturbation bounds for the stationary distribution of a markov chain. *Linear Algebra Appl*, 335:137–150, 2000.

Schweitzer, Paul J. Perturbation theory and finite markov chains. *Journal of Applied Probability*, pp. 401–413, 1968.

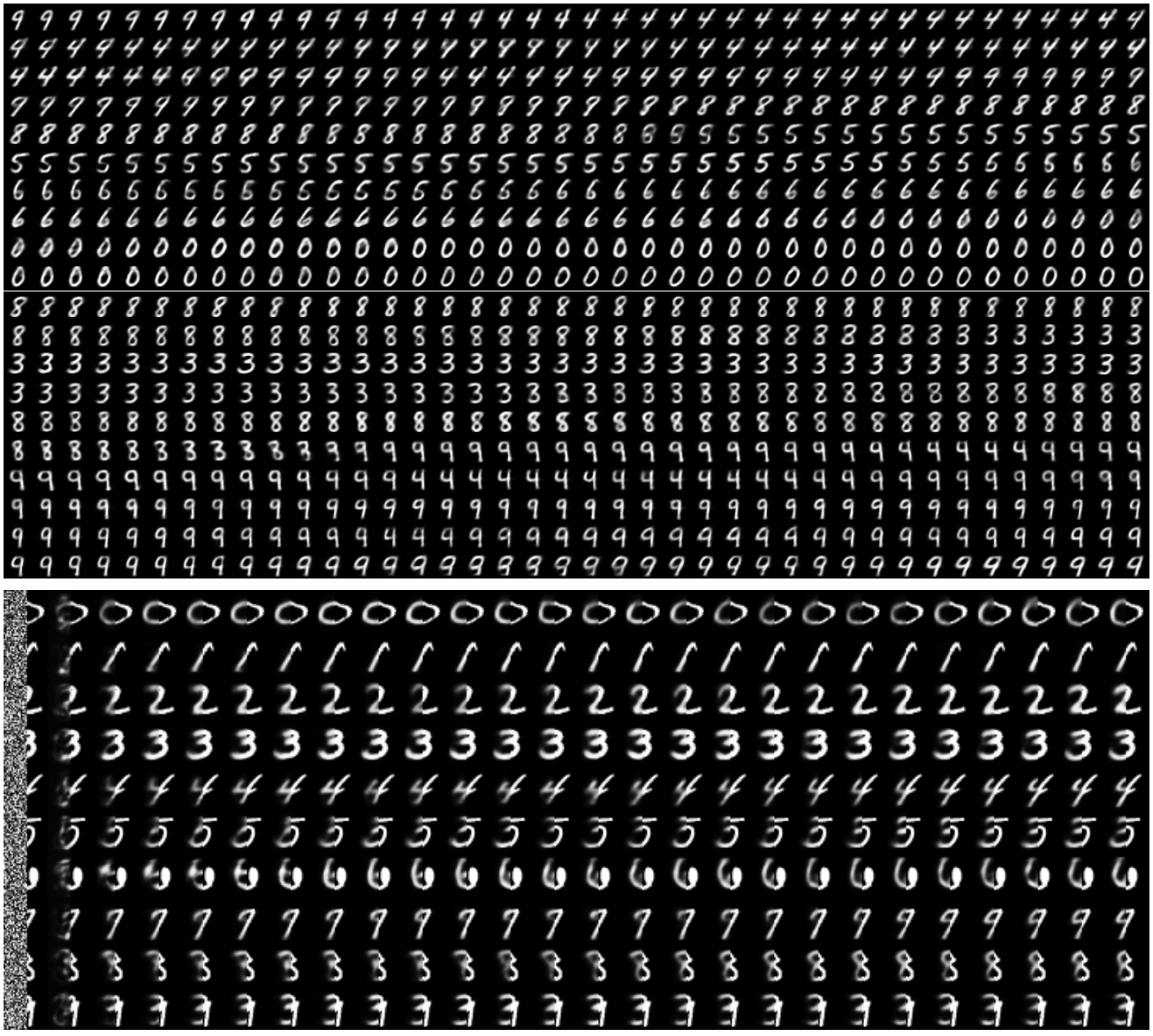


Figure 6. These are expanded plots of those in Figure 3. *Top:* two runs of consecutive samples (one row after the other) generated from a 2-layer GSN model, showing that it mixes well between classes and produces nice and sharp images. Figure 3 contained only one in every four samples, whereas here we show every sample. *Bottom:* conditional Markov chain, with the right half of the image clamped to one of the MNIST digit images and the left half successively resampled, illustrating the power of the trained generative model to stochastically fill-in missing inputs. Figure 3 showed only 13 samples in each chain; here we show 26.

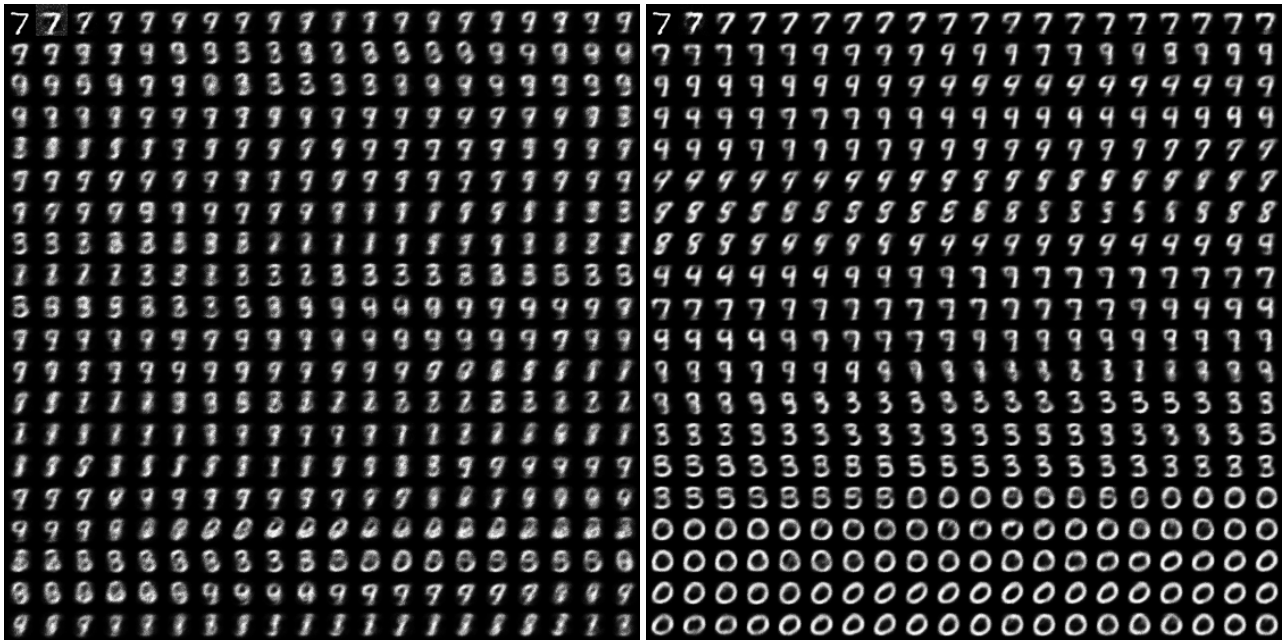


Figure 7. Left: consecutive GSN samples obtained after 10 training epochs. Right: GSN samples obtained after 25 training epochs. This shows quick convergence to a model that samples well. The samples in Figure 6 are obtained after 600 training epochs.

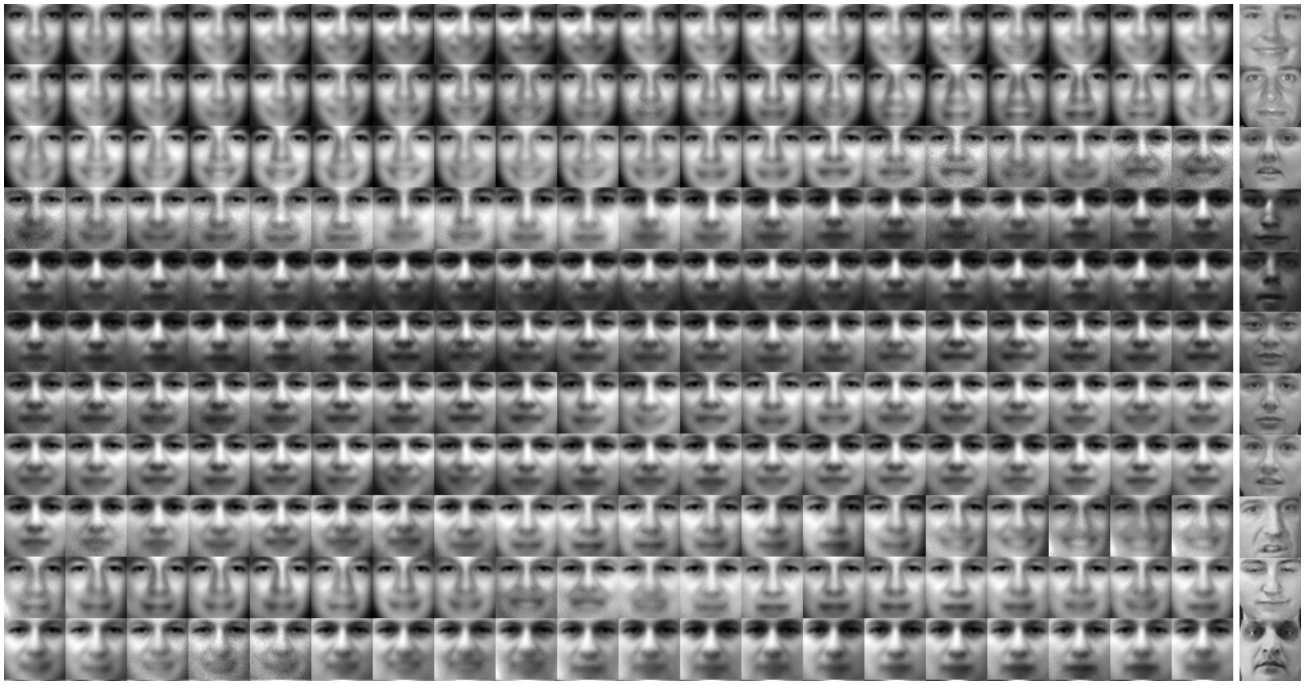


Figure 8. Consecutive GSN samples from a 3-layer model trained on the TFD dataset. At the end of each row, we show the nearest example from the training set to the last sample on that row to illustrate that the distribution is not merely copying the training set.