Few-shot Generative Modelling with Generative Matching Networks (Supplementary Materials)

Sergey Bartunov and **Dmitry P. Vetrov**

1 More samples from the model

We provide more samples from the model on figures 2 and 3.

2 Model architecture

In order to reduce the number of trainable parameters, we shared some functionality across different parts of our model. In particular, we set functions $f_G = f_R$ to be equal and also completely shared prototype extractors: $\psi_G = \psi_R = \psi_P$. As we mentioned, the generative part and the recognition model also shared the recurrent controller used in the full context matching procedure, but the prior had its own controller. All controllers were implemented as GRU (Chung et al., 2015) maintaining 200-dimensional hidden state.

Feature extractors $g_\cdot$ were also identical. Each function f or g used in our model is simply an affine transformation of feature encoder's output (and a hiddent state of a recurrent controller in the case of full context matching) to a 200-dimensional space followed by a non-linearity.

By default, we used a parametric rectified linear function as a non-linearity everywhere where applicable.

2.1 Conditional generator

The conditional generator network producing parameters for $p(\mathbf{x}|\mathbf{z}, \mathbf{X}, \boldsymbol{\theta})$ has concatenation of $\mathbf{z}$ and the output of the matching operation $[r, h]$ as input which is transformed to $3 \times 3 \times 32$ tensor and then passed through 3 residual blocks of transposed convolutions. Each block has the following form:

$$h = \text{conv}_1(x),$$
$$y = f(\text{conv}_2(h) + h) + \text{pool}(\text{scale}(x)),$$

where f is a non-linearity which in our architecture is always parametric rectified linear function (He et al., 2015).

The block is parametrized by size of filters used in convolutions conv_1 and conv_2, shared number of filters F and stride S.

- scale is another convolution with 1×1 filters and the shared stride S.
- In all other convolutions number of filters is the same and equals F.
- conv_1 and pool have also stride S.
- conv_2 preserve size of the input by padding and has stride 1.

Blocks used in our paper have the following parameters $(W_1 \times H_1, W_2 \times H_2, F, S)$:

1. $(2 \times 2, 2 \times 2, 32, 2)$
2. $(3 \times 3, 3 \times 3, 16, 2)$
3. $(4 \times 4, 3 \times 3, 16, 2)$

Then log-probabilities for binary pixels were obtained by summing the result of these convolutions along the channel dimension.

2.2 Feature encoder ψ

Function ψ has an architecture which is symmetric from the generator network. The only difference is that the scale scale operation is replaced by bilinear upscaling.

The residual blocks for feature encoder has following parameters:

1. $(4 \times 4, 3 \times 3, 16, 2)$
2. $(3 \times 3, 3 \times 3, 16, 2)$
3. $(2 \times 2, 2 \times 2, 32, 2)$

The result is a tensor of $3 \times 3 \times 32 = 288$ dimensions.

3 Transfer to MNIST

In this experiment we test the ability of generative matching networks to adapt not just to new concepts, but also to

a new *domain*. Since we trained our models on 28×28 resolution for Omniglot it should be possible to apply them on MNIST dataset as well. We used the test part of MNIST to which we applied a single random binarization.

Table 1 contains estimated predictive likelihood for different models. Qualitative results from the evaluation on Omniglot remain the same. Although transfer to a new domain caused significant drop in performance for all of the models, one may see that generative matching networks still demonstrate the ability to adapt to conditioning data. At the same time, average matching does not seem to efficiently re-use the conditioned data in such transfer task since relative improvements in expected conditional log-likelihood are rather small. Apparently, the model trained on a one-class datasets also learned highly dataset-dependent features as it actually performed even worse than the model with $C_{\text{train}} = 2$.

We also provide conditional samples on figure 1. Both visual quality of samples and test log-likelihoods are significantly worse comparing to Omniglot which can be caused by a visual difference of the MNIST digits from Omniglot characters. The images are bolder and less regular due to binarization. Edwards & Storkey (2016) suggest that the quality of transfer may be improved by augmentation of the training data, however for the sake of experimental simplicity and reproducibility we resisted from any augmentation.

4 Evaluation of the neural statistician model

The neural statistician model falls into the category of models with global latent variables which we describe in section 2.2. The conditional likelihood for this model has the following form:

$$p(\mathbf{x}|\mathbf{X}) = \int p(\boldsymbol{\alpha}|\mathbf{X}) \int p(\mathbf{z}|\boldsymbol{\alpha}) p(\mathbf{x}|\boldsymbol{\alpha}, \mathbf{z}) d\mathbf{z} d\boldsymbol{\alpha}. \quad (1)$$

This quantity is hard to compute since it consists of an expectation with respect to the true posterior over global variable $\boldsymbol{\alpha}$. In our evaluation we have tried three different strategies for computing the above-mentioned integral.

First, we used self-normalizing importance sampling to directly estimate $p(\mathbf{x}|\mathbf{X}, \boldsymbol{\theta})$ as

$$\hat{p}(\mathbf{x}|\mathbf{X}, \boldsymbol{\theta}) = \frac{\sum_{s=1}^{S} w_s p(\mathbf{x}, \mathbf{z}^{(s)}|\boldsymbol{\alpha}^{(s)}, \boldsymbol{\theta})}{\sum_{s=1}^{S} w_s},$$
$$w_s = \frac{p(\boldsymbol{\alpha}^{(s)}, \mathbf{X}, \mathbf{Z}^{(s)}|\boldsymbol{\theta})}{q(\boldsymbol{\alpha}^{(s)}|\mathbf{X}, \boldsymbol{\phi}) q(\mathbf{Z}^{(s)}, \mathbf{z}^{(s)}|\mathbf{X}, \mathbf{x}, \boldsymbol{\alpha}^{(s)}, \boldsymbol{\phi})},$$

with samples $\boldsymbol{\alpha}^{(s)}$ and $\mathbf{z}^{(s)}$ obtained from the recognition model. We observed somewhat contradictory results such as non-monotonic dependency of the estimate on the size of conditioning dataset. The diagnostic of the effective sample size suggested that the recognition model is not well suited as proposal for the task.

Another strategy was to sequentially estimate $p(\mathbf{X}_{<t}, \boldsymbol{\theta})$ and then use the equation

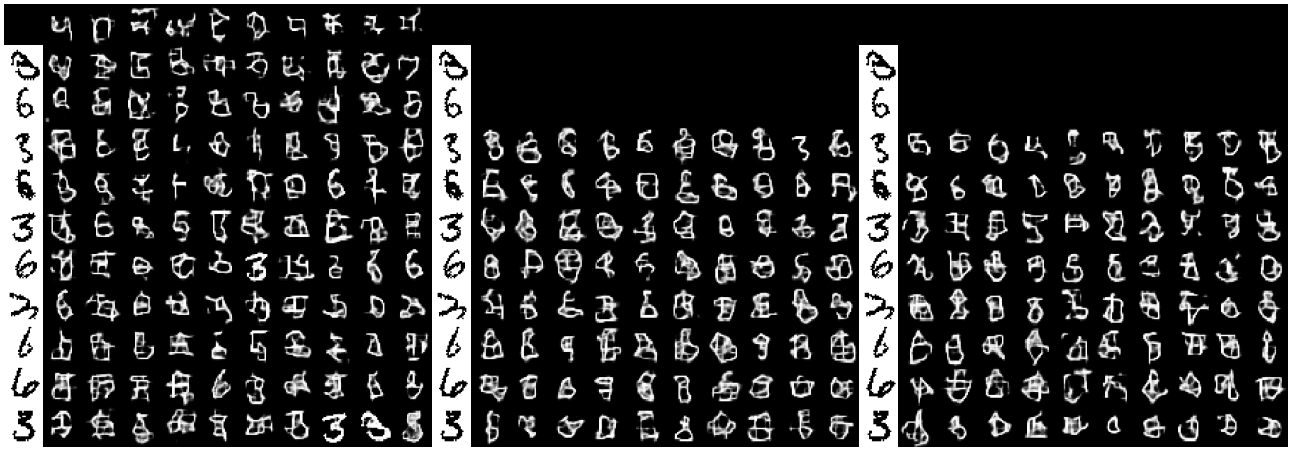
$$p(\mathbf{x}_t|\mathbf{X}_{<t}, \boldsymbol{\theta}) = \frac{p(\mathbf{x}_t, \mathbf{X}_{<t}|\boldsymbol{\theta})}{p(\mathbf{X}_{<t}|\boldsymbol{\theta})},$$

which appeared to as unreliable as the previous strategy.

Finally, we decided to use the approximate posterior $q(\boldsymbol{\alpha}|\mathbf{X})$ in equation (1) that was learned together with the model instead of the exact one. Practically, one can almost never access the true posterior and when using the model would rather resort to the recognition model. Hence, the resulting approximate predictive distribution and the corresponding estimate is aligned with practical usage of the model and is often considered in the literature (Snelson & Ghahramani, 2005; Jaakkola & Jordan, 2000). Fortunately, the approximate posterior is easy to sample from by construction, being a multivariate Normal distribution. The final estimate was computed as:

$$\hat{p}(\mathbf{x}|\mathbf{X}, \boldsymbol{\theta}) = \sum_{s=1}^{S} \frac{w_s p(\mathbf{x}, \mathbf{z}^{(s)}|\boldsymbol{\alpha}^{(s)}, \boldsymbol{\theta})}{S},$$
$$w_s = \frac{p(\mathbf{z}^{(s)}|\boldsymbol{\alpha}^{(s)}, \boldsymbol{\theta})}{q(\mathbf{z}^{(s)}|\mathbf{x}, \boldsymbol{\alpha}^{(s)}, \boldsymbol{\phi})},$$

where samples are, again, obtained from the recognition model.



(a) GMN (b) GMN, no pseudo-input (c) GMN, no attention, no pseudo-input

Figure 1: Conditionally generated samples on MNIST. Models were trained on the train part of Omniglot. Format of the figure is similar to fig. 2 in the main paper.

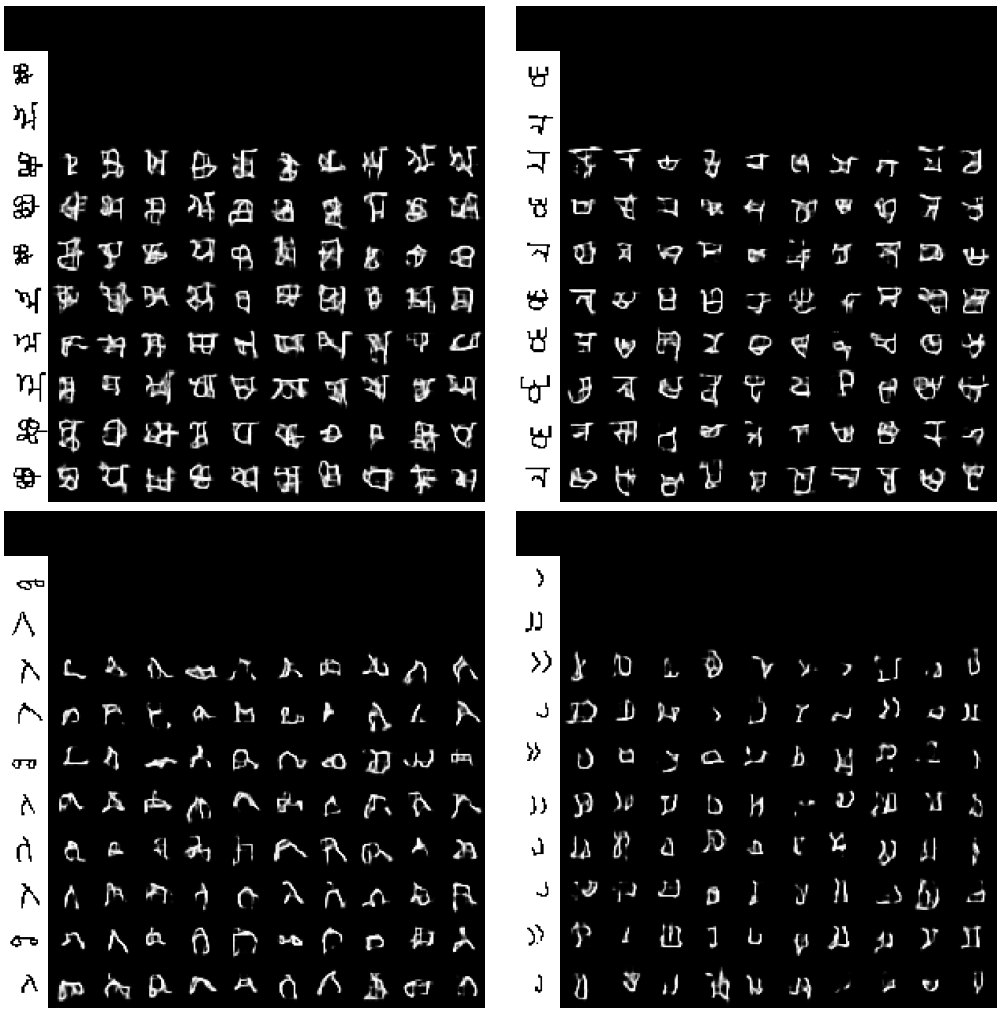


Figure 2: Additional conditionally-generated samples from the GMN, no pseudo-inputs used.

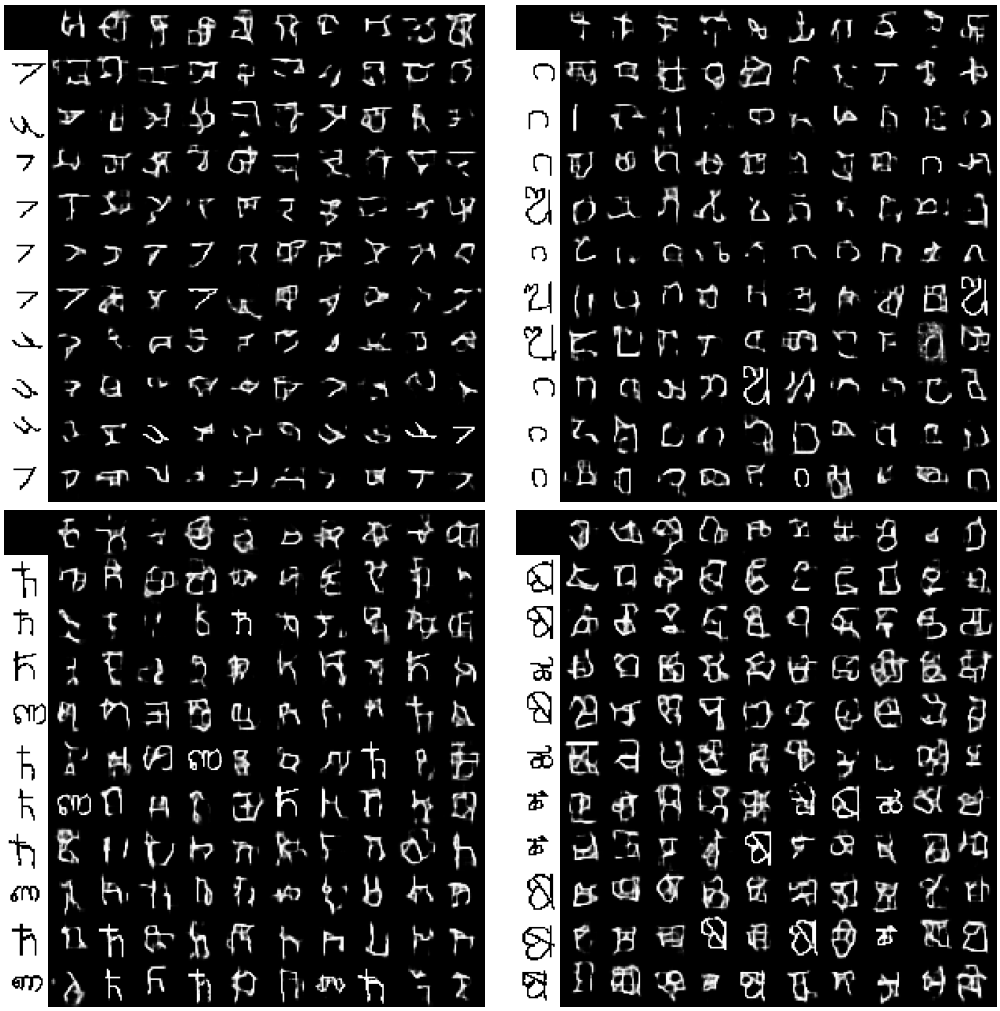


Figure 3: Additional conditionally-generated samples from the GMN, one pseudo-input used

Table 1: Conditional negative log-likelihoods for the test part of MNIST. Models were trained on the train part of Omniglot.

		Number of conditioning examples							
Model	C_{test}	**0**	**1**	**2**	**3**	**4**	**5**	**10**	**19**
GMN, $C_{\text{train}} = 2$	1	126.7	121.1	118.4	117.6	117.1	117.1	117.1	118.5
GMN, $C_{\text{train}} = 2$	2	126.2	123.1	121.3	120.1	119.4	118.9	118.3	119.6
GMN, $C_{\text{train}} = 2$, no pseudo-input	1		135.1	120.9	117.5	115.7	114.4	111.7	109.8
GMN, $C_{\text{train}} = 2$, no pseudo-input	2			123.1	121.9	119.4	118.8	115.2	113.2
GMN, $C_{\text{train}} = 1$, avg	1	131.5	126.5	123.3	121.9	121.0	120.2	118.6	117.5
GMN, $C_{\text{train}} = 2$, avg	2	126.2	122.8	121.0	119.9	118.9	118.7	117.8	116.8
GMN, $C_{\text{train}} = 1$, avg, no pseudo-input	1		132.1	126.9	125.0	124.8	123.9	121.7	120.9
GMN, $C_{\text{train}} = 2$, avg, no pseudo-input	2			118.4	117.9	117.4	117.1	116.6	115.8

References

Chung, Junyoung, Gülçehre, Çalar, Cho, Kyunghyun, and Bengio, Yoshua. Gated feedback recurrent neural networks. In *Proceedings of the 32nd International Conference on Machine Learning (ICML'15)*, 2015.

Edwards, Harrison and Storkey, Amos. Towards a neural statistician. *arXiv preprint arXiv:1606.02185*, 2016.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034, 2015.

Jaakkola, Tommi S and Jordan, Michael I. Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10(1):25–37, 2000.

Snelson, Edward and Ghahramani, Zoubin. Compact approximations to bayesian predictive distributions. In *Proceedings of the 22nd international conference on Machine learning*, pp. 840–847. ACM, 2005.