

## A ARCHITECTURE AND TRAINING DETAILS

### A.1 ARCHITECTURE

In the following,  $d_z = d_s = d_h = 512$  and  $d_\psi = 32$ .

**Encoder** The embedding function for each input/output pair is

$$h_\phi(x_i, y_i) : (d_x + d_y) \xrightarrow{\text{fc+relu}} d_h \xrightarrow{\text{fc+relu}} d_h \xrightarrow{\text{fc}} d_s .$$

For SIVI models, the rest of the encoder  $\rho_\phi$  and  $\eta_\phi$  is defined as

$$\begin{aligned} \rho_\phi(s) &: (d_s + d_\epsilon) \xrightarrow{\text{fc+relu}} d_h \xrightarrow{\text{fc}} d_\psi \\ \eta_\phi(s, \psi) &: (d_s + d_\psi) \xrightarrow{\text{fc+relu}} d_h \xrightarrow{\text{fc}} 2 * d_z \end{aligned}$$

where  $\epsilon \sim \mathcal{N}(0, I)$  and  $\epsilon \in \mathbb{R}^{d_\epsilon}$ . The output of  $\eta_\phi$  is then split into two  $d_z$ -dimensional vectors  $\mu_z$  and  $\sigma'_z$  with

$$q_\phi(z|\mathbf{x}_C, \mathbf{y}_C) = \mathcal{N}(\mu_z, \text{diag}(0.9 + 0.1 * \text{sigmoid}(\sigma'_z))^2) .$$

For the NP models, there is no  $\eta_\phi$ , and  $\rho_\phi$  is defined as

$$\rho_\phi(s) : d_s \xrightarrow{\text{fc+relu}} d_h \xrightarrow{\text{fc}} 2 * d_z$$

where the output is split into two vectors  $d_z$ -dimensional vectors like in SIVI.

**Decoder**  $g_\theta(x'_j, z) : d_x + d_z \xrightarrow[\text{4 times}]{\text{fc+relu}} d_h \xrightarrow{\text{fc}} 2 * d_y$ , where the output is split into two  $d_y$ -dimensional

vectors  $\mu_y$  and  $\sigma'_y$ , and

$$q(y'_j|x'_j, z) = \mathcal{N}(\mu_y, \text{diag}(0.9 + 0.1 * \text{softplus}(\sigma'_y))^2) .$$

For the models with a fixed observation variance, the output of  $g_\theta$  is only the vector  $\mu_y$  and  $q(y'_j|x'_j, z) = \mathcal{N}(\mu_y, 0.2^2 * \mathbf{I}_{d_y})$ .

ANP model is implemented with the same specifications as above and the other components (deterministic path and attention) is the same as Kim et al. (2018).

### A.2 TRAINING

All the models were trained using Adam optimizer and a batch size of 16 for 100 epochs. Learning rate was  $5 \times 10^{-4}$  for NP+avg, NP+max and SIVI+max, and  $5 \times 10^{-5}$  for ANP. For SIVI on MNIST, a learning rate scheduler was employed as well that would multiply the learning rate by 0.1 after 20, 50 and 80 epochs.

The procedure for constructing context sets and target sets from a chosen image in the dataset was as follows. From the image,  $n + m'$  pixels, where  $n \sim [1, 200)$  and  $m' \sim [0, 200)$ , were chosen without replacement. The first  $n$  pixels constitute the context set, and all  $m = n + m'$  pixels were put into the target set.

### A.3 SIVI OBJECTIVE

As stated in the paper, since a hierarchical encoder makes the ELBO intractable, a tractable lower bound to the EBLO is often used instead. SIVI bound is a tractable lower bound to the ELBO for hierarchical variational families (c.f. Eq. (6)). This bound in the context of Neural Processes is defined as

$$\mathbb{E}_{q_\phi(z, \psi_0|\mathbf{x}, \mathbf{y})} \left[ \mathbb{E}_{q_\phi(\psi_{1:K}|\mathbf{x}, \mathbf{y})} \left[ \log \frac{p_\theta(\mathbf{y}|z, \mathbf{x}) p_\theta(z)}{\frac{1}{K+1} \sum_{k=0}^K q_\phi(z|\psi_k, \mathbf{x}, \mathbf{y})} \right] \right] \leq \text{ELBO} \leq \log p_\theta(\mathbf{y}|\mathbf{x}) \quad (7)$$

where  $q_\phi(\psi_{1:K}|\mathbf{x}, \mathbf{y}) = \prod_{i=1}^K q_\phi(\psi_i|\mathbf{x}, \mathbf{y})$  and ELBO is defined as Eq. (2). In our experiments,  $K$  is 100 and  $\psi \in \mathbb{R}^{32}$ .

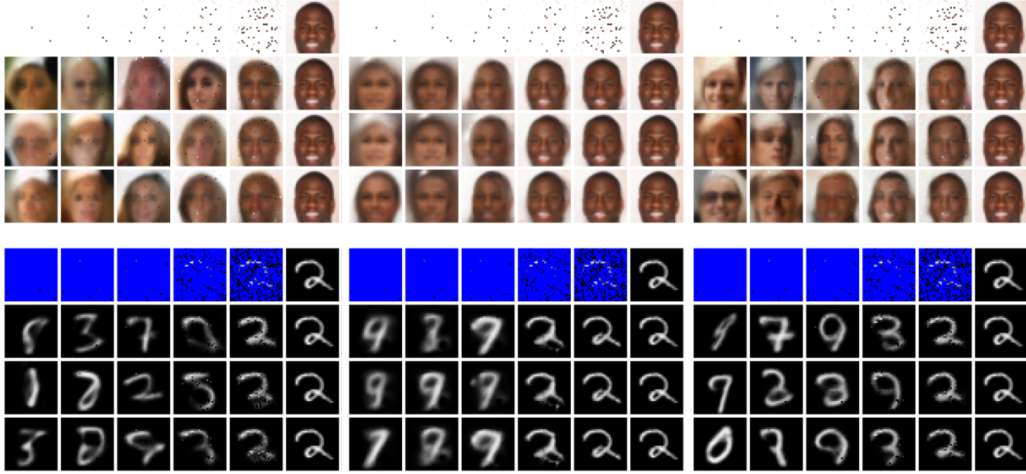


Figure 8: Image inpainting results when the context set is directly copied to the shown output and the model is queried for the rest of image pixels. From left to right, NP, ANP, and ours (NP+SIVI+max pooling). As non-attentive models are known to underfit on the context points, the final results for them are not smooth. ANP improves on this aspect.

## B IMAGE INPAINTING RESULTS

Fig. 8 shows the inpainting results from different methods when the context set is carried to the output. In other words, inference is done via Equation 1 when the target and context sets are disjoint and the given  $y_C$  is directly copied to the shown output, instead of asking the model to predict values of  $y_C$ .

## C OTHER VARIANTS OF NEURAL PROCESS MODELS

There are many variants of Neural Processes with different probabilistic modelling assumptions and network architectures. We have attempted to be as clear and fair as possible in generating the qualitative results in the main text. In this section we clarify which specific architectures were considered and why. Moreover, to make the results comparable with other publications in the literature, we include qualitative results for other popular architecture choices not considered in the main text.

ANPs (Kim et al., 2018) include a deterministic path bypassing  $z$  from the encoder to the decoder that is not found in the original NP (Garnelo et al., 2018b). Our implementation of NP follows the original model without a deterministic path. In Kim et al. (2018), a NP without attention but with a deterministic path was considered. Fig. 9 shows that adding a deterministic path to NP generally hurts sample diversity in small context sizes. An additional complicating factor is whether  $g_\theta$  produces just the mean or both the mean and the variance of the likelihood function. In line with previous results (Le et al., 2018), we found that if  $g_\theta$  is trained to produce the observation variance of  $p_\theta(y'_i|z, x'_i)$ , then models with a deterministic path (including ANP) tend to end up with a large observation variance and a low-variance task-embedding posterior  $q_\phi(z|x_C, y_C)$ , leading to poorly calibrated uncertainty and low sample diversity. Therefore, to be as fair as possible to ANP models, its results in the main text correspond to a model trained with a fixed observation variance, whereas NP and NP+SIVI results are reported with learned observation variance. Fig. 10 shows the results for ANP with learned observation variance.

Finally, our experiments show that adding a hierarchical encoder to the models with a deterministic path and SIVI objective does not mitigate their lack of sample diversity when the context set is small.

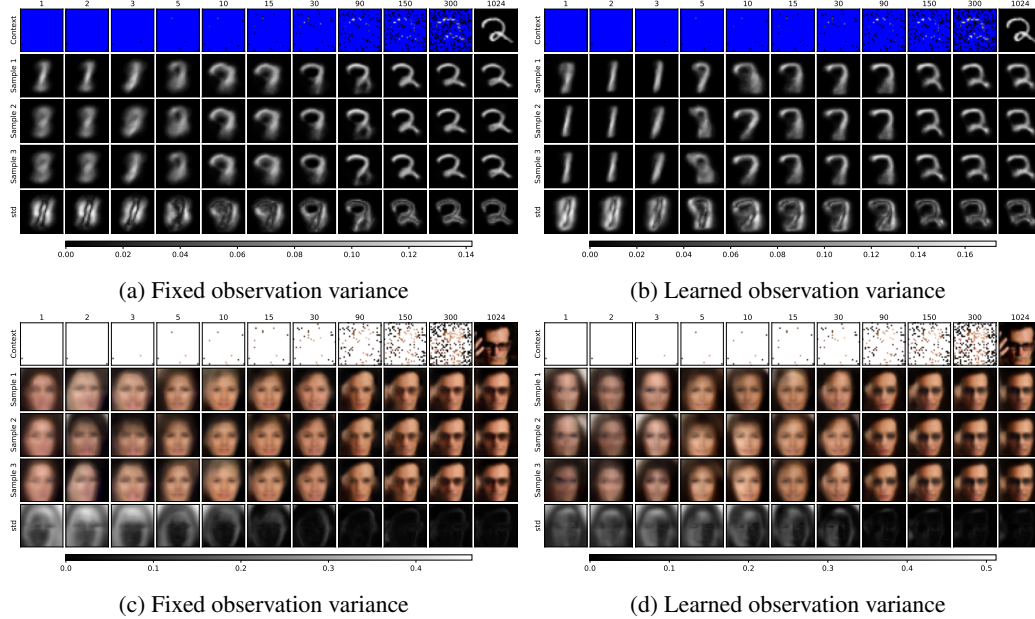


Figure 9: Qualitative results of NP+avg with deterministic path on (top) MNIST and (bottom) CelebA datasets. These plots show poor sample diversity from the model irrespective of whether the observation variance is fixed or learned.

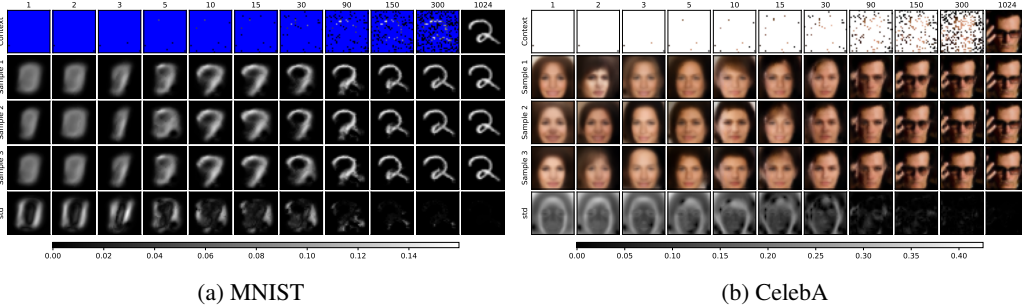


Figure 10: Qualitative results of ANP model with learned observation variance. Comparing (a) with Fig. 6c shows that learning the observation variance hurts sample diversity. It is not as easy to compare sample diversity for CelebA (see (b) and Fig. 6d). However, ANP in general performs worse than SIVI+max or NP+max on small context sets.

## D FASHIONMNIST RESULTS

Fig. 11 shows the image completion task results of the models considered in the paper trained on the FashionMNIST dataset.

## E FIG. 3 EXPERIMENTAL DETAILS

Fig. 3a shows entropy of the variational posterior, i.e.,  $q_\phi(z|x_C, y_C)$ , versus context set size ( $n$ ) for a growing context set with i.i.d. items. The plot shows an aggregation over 1000 runs of this procedure, each with a different ground truth image. The experiment verifies that the learned NP posterior follows the classical Bayesian inference results and, more interestingly, the posterior contraction even generalizes to context sets larger than the context sets seen during training. The plot was generated by an NP+max model trained on MNIST, but the observed behavior is not specific to it. We see the same behavior when average pooling is used for the CelebA dataset.

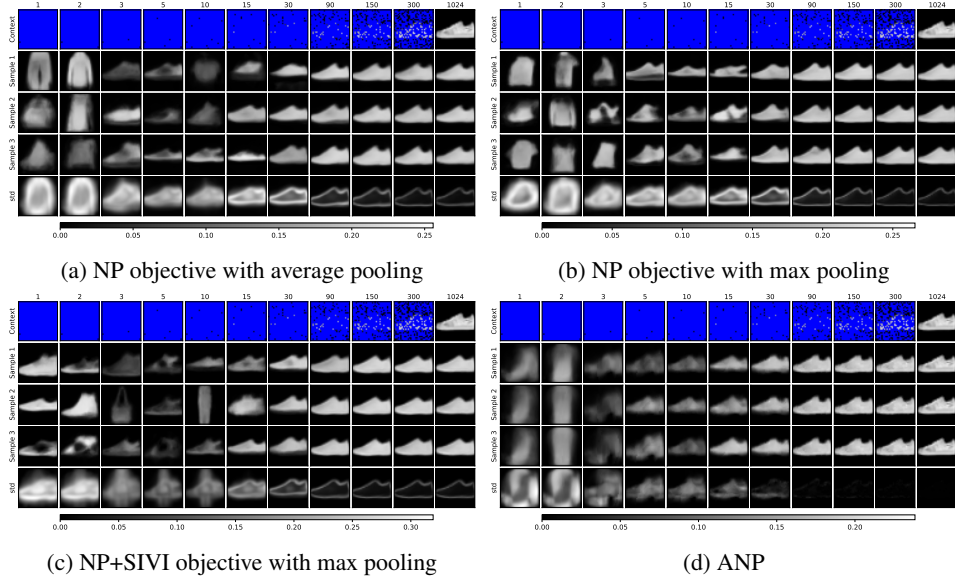
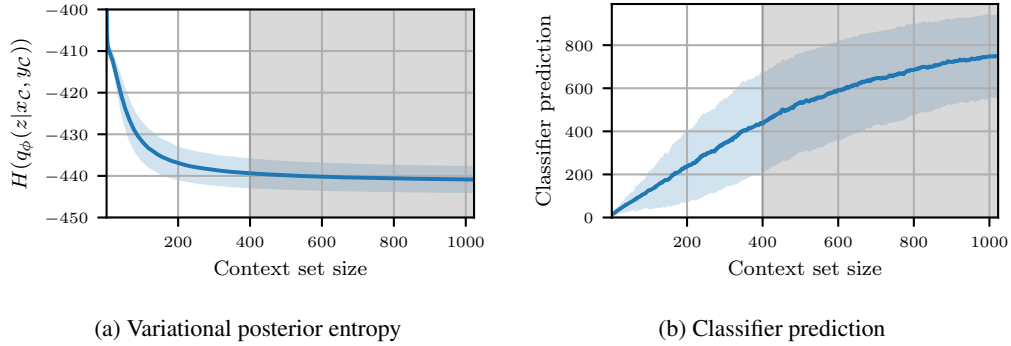


Figure 11: Qualitative results of various models trained on the FashionMNIST dataset.

Figure 12: Posterior contraction of  $q_\phi(z|\mathbf{x}_C, \mathbf{y}_C)$  in a NP+mean pooling model.

The experiment suggests that even though the context dataset is represented through an aggregated embedding that does not explicitly embed  $n$ , the training objective forces the networks and the embedding space to retain information about  $n$ . We validate this by training a classifier to predict  $n$  given the learned embeddings  $s_C$ . Fig. 3b shows the classifier performance on a held-out test set and shows a strong correlation between embeddings  $s_C$  and context set sizes.

Fig. 12 shows the same behavior with mean pooling. The plot is generated by a NP+mean model trained on MNIST dataset.

## F MAX POOLED EMBEDDINGS AND POSTERIOR ENTROPY

As discussed in the main text, a NP model with max pooling exhibits posterior contraction by learning a  $\rho_\phi$  such that the posterior entropy is a decreasing function in all dimensions of embedding space. To illustrate, Fig. 13 shows  $\|s_C\|_1$  vs context size (increasing), and the posterior entropy versus  $\|s_C\|_1$  (decreasing).



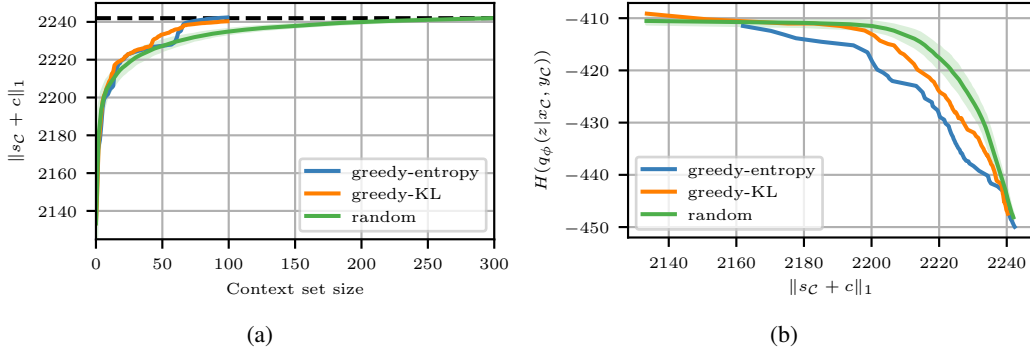


Figure 13: (a) Norm of pooled embedding,  $\|s_C\|_1$ , versus context size for three different methods of context set generation. Note that the embeddings are shifted so that the minimum embedding value in each dimension is 0. (b) Posterior entropy versus norm of (shifted) pooled embedding. Observe that the norm of the embedding is strictly increasing in context size, with large increases when the context is small; and that the posterior entropy is decreasing as a function of the norm of the embedding.

## G COMPUTING TEST DATA LOG LIKELIHOODS

The test data (normalized) log likelihoods  $\frac{1}{|\mathcal{T}|} \log p_\theta(\mathbf{y}_\mathcal{T} | \mathbf{x}_\mathcal{T}, \mathbf{x}_\mathcal{C}, \mathbf{y}_\mathcal{C})$  are computed and averaged over context/target sets sampled from held-out test sets. Context sets and target sets are *disjoint* (i.e., all the items in target set are unobserved) and have a random size in  $[1, 200)$ . As we do not have a closed form for predictive log-likelihoods, we compute the following IWAE-like lower bound (Burda et al., 2016) instead with  $K=1000$ .

$$\log p_\theta(\mathbf{y}_\mathcal{T} | \mathbf{x}_\mathcal{T}, \mathbf{x}_\mathcal{C}, \mathbf{y}_\mathcal{C}) \geq \mathbb{E}_{q_\phi(z_{1:K} | \mathbf{x}_\mathcal{C}, \mathbf{y}_\mathcal{C})} \log \frac{1}{K} \sum_{k=1}^K \frac{p_\theta(\mathbf{y}_\mathcal{T} | \mathbf{x}_\mathcal{T}, z_k) p_\theta(z_k | \mathbf{x}_\mathcal{C}, \mathbf{y}_\mathcal{C})}{q_\phi(z_k | \mathbf{x}_\mathcal{C}, \mathbf{y}_\mathcal{C})} \quad (8)$$

$$\approx \mathbb{E}_{q_\phi(z_{1:K} | \mathbf{x}_\mathcal{C}, \mathbf{y}_\mathcal{C})} \log \frac{1}{K} \sum_{k=1}^K \frac{p_\theta(\mathbf{y}_\mathcal{T} | \mathbf{x}_\mathcal{T}, z_k) q_\phi(z_k | \mathbf{x}_\mathcal{C}, \mathbf{y}_\mathcal{C})}{q_\phi(z_k | \mathbf{x}_\mathcal{C}, \mathbf{y}_\mathcal{C})} \quad (9)$$

$$= \mathbb{E}_{q_\phi(z_{1:K} | \mathbf{x}_\mathcal{C}, \mathbf{y}_\mathcal{C})} \log \frac{1}{K} \sum_{k=1}^K p_\theta(\mathbf{y}_\mathcal{T} | \mathbf{x}_\mathcal{T}, z_k). \quad (10)$$

## H COMPUTING INCEPTION SCORES IN OUR EXPERIMENTS

### H.1 DEFINITION

Inception score is defined in a way such that a high score requires the individual samples to be classifiable with high confidence and, at the same time, the marginal class distribution of samples to be diverse. More formally,

$$\log \text{IS} = \mathbb{E}_{\mathbf{x} \sim G} [D_{\text{KL}}(p(y|\mathbf{x}) || p(y))] = -\mathbb{E}_{\mathbf{x} \sim G} [H(p(y|\mathbf{x})) - H(p(y|\mathbf{x}), p(y))] \quad (11)$$

where  $G$  is a generator producing samples  $\mathbf{x}$  and  $y$  is the classification labels specified by the classifier.

### H.2 CLASSIFIER NETWORKS

As results in the GAN literature suggest that inception score is unreliable when applied to image domains other than ImageNet, we replace inception network with classifiers trained on MNIST, FashionMNIST, and CelebA datasets. The network architecture of all classifiers is ResNet (He et al., 2016). The MNIST classifier network is trained to solve the MNIST digit classification task with 10 classes. The FashionMNIST network is similarly trained to solve its 10-way classification

task. It is more challenging for CelebA as there is no well-defined set of labelled classes for it. As CelebA images are labelled with 40 attributes, we choose the four attributes of  $\{\text{Male, Black Hair, Smiling, Young}\}$  and construct a synthetic classification task with  $2^4$  classes where each class refers to a configuration of the chosen attributes. The trained models are used in place of inception network to get calibrated scores in our experiments.

## I EFFECT OF SIVI

In this section, we investigate the effect of a hierarchical encoder and choice of objective in isolation. Fig. 14 shows inception scores achieved by different models with mean pooling trained on MNIST and CelebA. Fig. 15 shows the same for models with max pooling. Fig. 16 aggregates both plots into one to better visualize the comparison between max and mean pooling. The images and context sets used to create these plots are the same as those used in Fig. 7.

We note that both SIVI and the ELBO target a lower bound on  $p(\mathbf{y}|\mathbf{x})$  (see Eq. (2)), whereas the NP objective is an approximation of a lower bound on  $p(\mathbf{y}_\mathcal{T}|\mathbf{x}_\mathcal{T}, \mathbf{x}_\mathcal{C}, \mathbf{y}_\mathcal{C})$  (see Eq. (4)). Therefore, to see the effect of a hierarchical encoder in isolation, one should compare its performance with ELBO models. The plots in this section show that SIVI models indeed outperform their ELBO counterparts, irrespective of the pooling operation.

The NP objective approximates the lower bound by replacing  $p_\theta(z|\mathbf{x}_\mathcal{C}, \mathbf{y}_\mathcal{C})$  with  $q_\phi(z|\mathbf{x}_\mathcal{C}, \mathbf{y}_\mathcal{C})$ . Although this approximation is questionable, it empirically performs better in terms of the predictive log-likelihood (as shown in Table 1 and as previously reported by Le et al. (2018)). Unfortunately, a hierarchical proposal makes the NP objective (Eq. (4)) intractable. A tractable lower bound or approximation to the NP objective would need to be derived in order to use a hierarchical proposal in NP. Nonetheless, it seems to be a promising direction for future work according to our results.

Finally, max pooling appears to consistently improve inception score in image completion tasks regardless of the training objective (see Fig. 16).

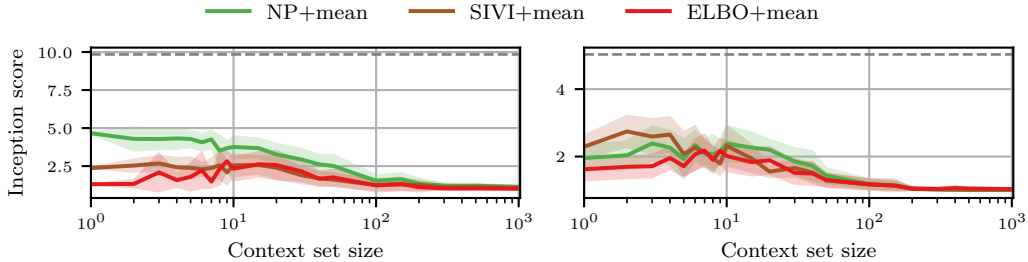


Figure 14: Inception scores of models with mean pooling. Left, MNIST; right, CelebA.

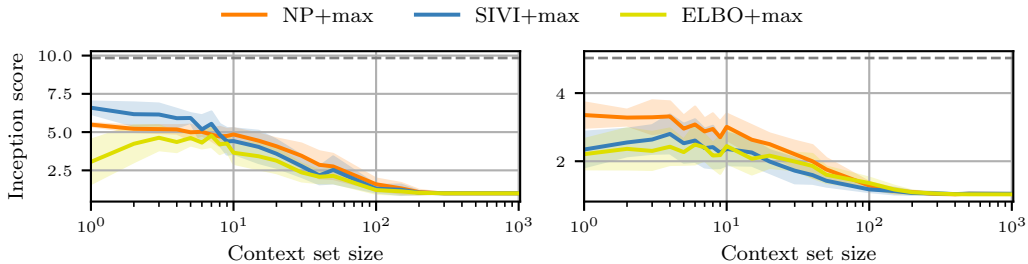


Figure 15: Inception scores of models with max pooling. Left, MNIST; right, CelebA.

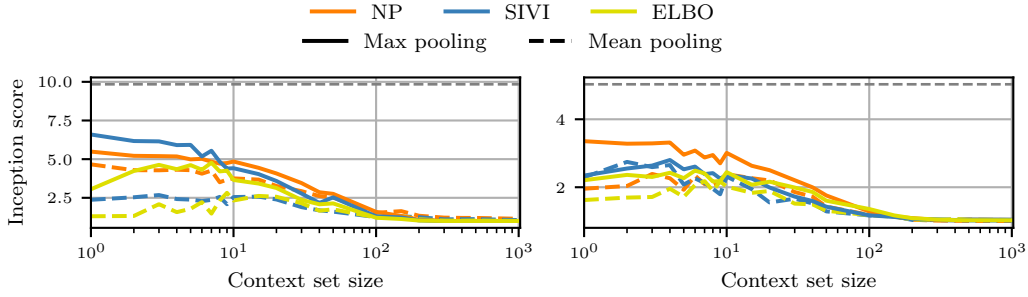


Figure 16: Inception scores of different models. Solid and dashed lines correspond to models with max and mean pooling, respectively. Variation in the inception score is not reported to avoid clutter. Left, MNIST; right, CelebA.

## J A COMBINATION OF MEAN AND MAX POOLING

In this section we present the results on models with a pooling that combines mean and max pooling, namely ”mixed pooling”. In this pooling, the first half dimensions of embeddings  $s_i$  are mean-pooled to get  $s_C^{(1)}$  and the second half are max-pooled to get  $s_C^{(2)}$ . Finally,  $s_C^{(1)}$  and  $s_C^{(2)}$  are concatenated to get the overall embedding of a context set. The idea is that the decoder network might be able to decide which pooled embedding to use and achieve better overall performance. However, the models with mixed pooling do not outperform the models with only max pooling, as shown in Fig. 17.

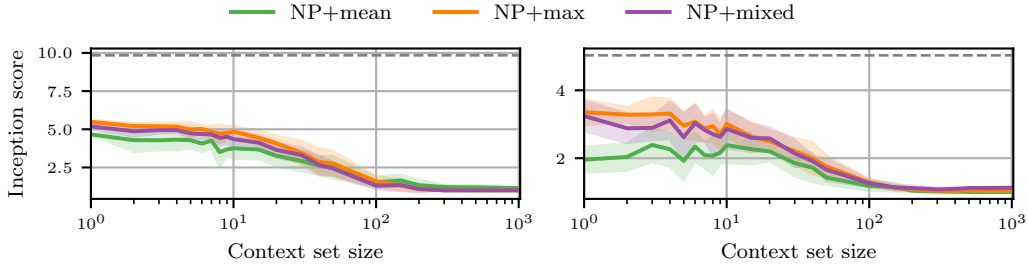


Figure 17: Inception scores of models with different pooling choices. Left, MNIST; right, CelebA.

## K MNIST CLASSIFIER RESULTS

We examine the diversity of samples generated from each model by classifying them using a MNIST classifier and looking at the distribution of the predictions. The main expectations are that (1) the models have a non-zero probability of generating the ground truth image irrespective of the context set and that (2) the models do not generate digits that are inconsistent with the context set.

As the true posterior probability of the digit given a few pixels of its image (the context set) is unknown, we report Figs. 18 and 19 as a proxy to it. These figures show the results of an experiment where a sequence of growing context sets incrementally reveals an image of a 3 and compare the prediction distribution of generated samples from different models. The final image in Fig. 18 is chosen from the test set, and the context sets are constructed to eliminate a specific digit with each step. In Fig. 19, the final image is synthetic and hand-drawn. The context set in each step is grown by adding new strokes of the digit.

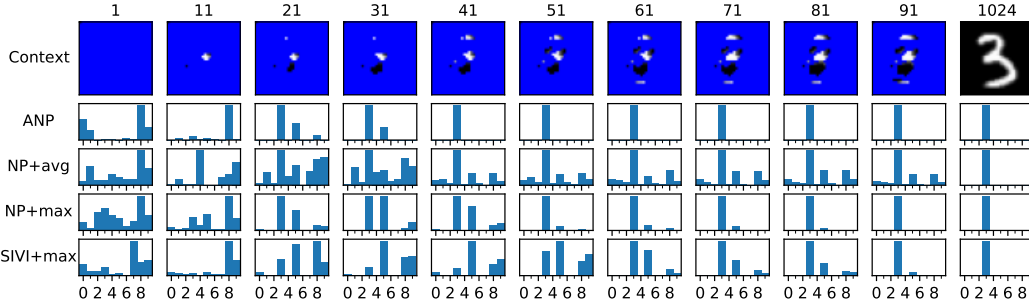


Figure 18: MNIST classification results for a sequence of growing context sets. Each column shows the results for the context set in the first row. Each histogram under a context set shows the prediction distribution of a MNIST classifier for 1000 samples from a model that was conditioned on the context set. The context set sizes are written at the top of each column. The first context set is the top left pixel, treated as an uninformative context set. Each of the following context sets add 10 new pixels that are specifically chosen to eliminate a remaining possible digit (in the order of 0 to 9). Given the digit to eliminate, the 10 chosen pixels are the ones that differ the most in pixel intensity between the mean image of all instances of 3 in the training set and the mean image of all instances of the digit to eliminate.

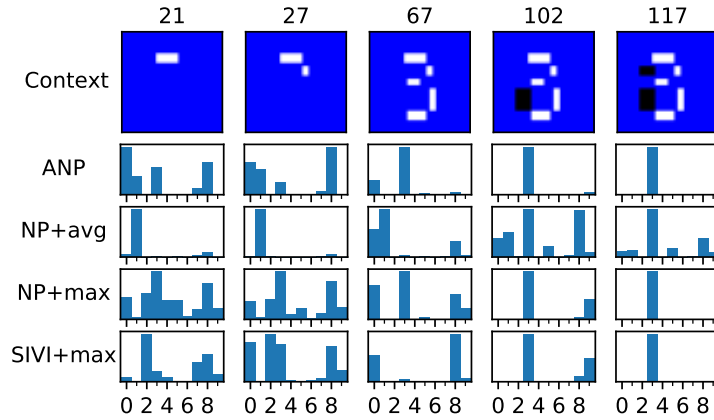


Figure 19: MNIST classification results for a synthetic sequence of context sets. The context sets are designed to hint at an image of a 3. The details of how the subplots are organized is the same as Fig. 18. All the models except NP+avg have a non-zero probability for the correct digit throughout the process. In terms of the compatibility of the generated digits with the context set, ANP works reasonably well on larger context sets while NP+max and SIVI+max generally outperform the others throughout the process.

## L ADDITIONAL QUALITATIVE RESULTS

In this section, we report additional results for MNIST and CelebA experiments. Each figure shows 15 samples per context set drawn from various methods discussed in the paper. The results from models trained on MNIST are shown in Figs. 20 and 21, FashionMNIST is shown in Figs. 22 and 23 and CelebA is shown in Figs. 24 and 25.

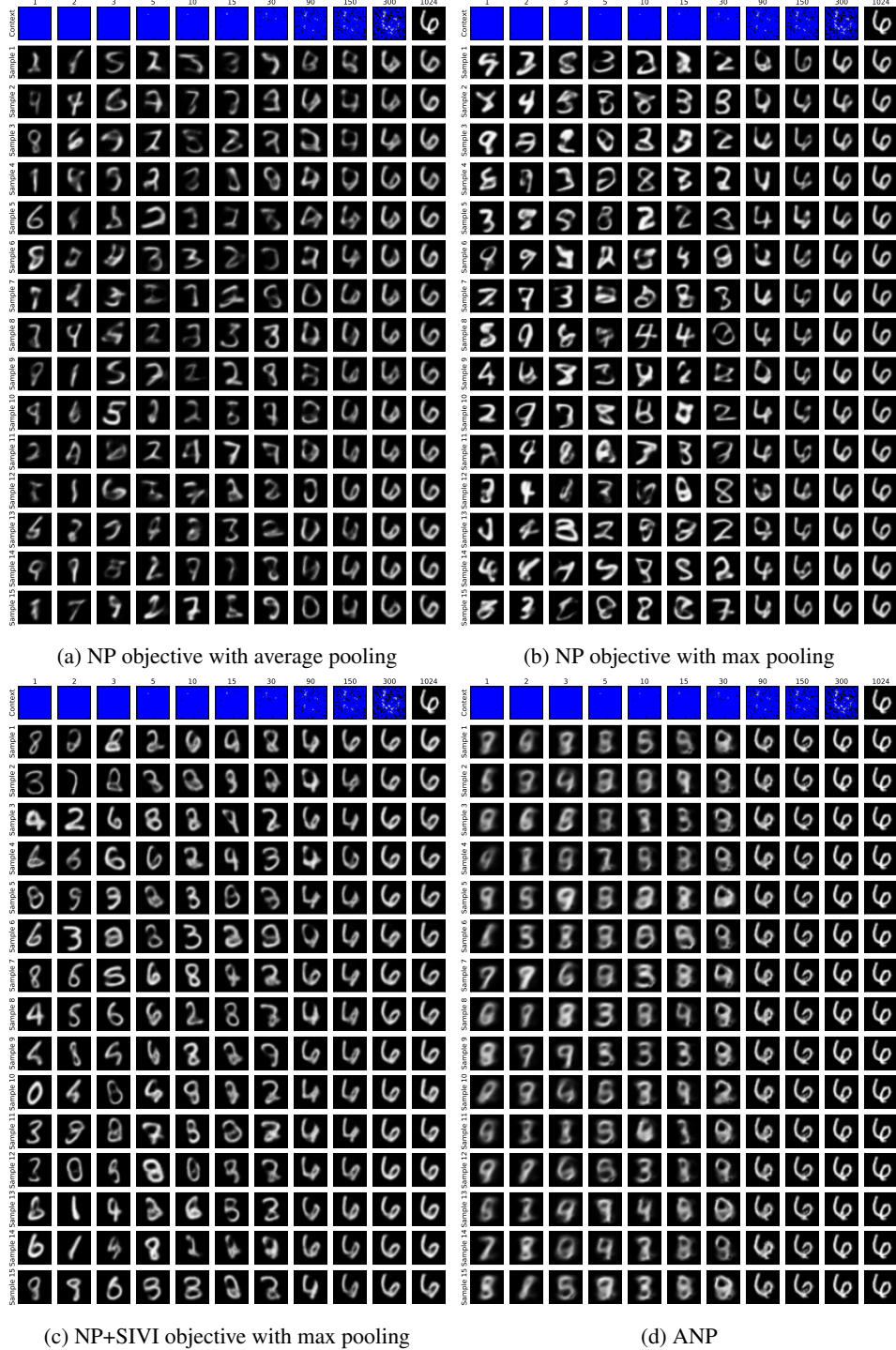


Figure 20: Samples from models trained on the MNIST dataset.

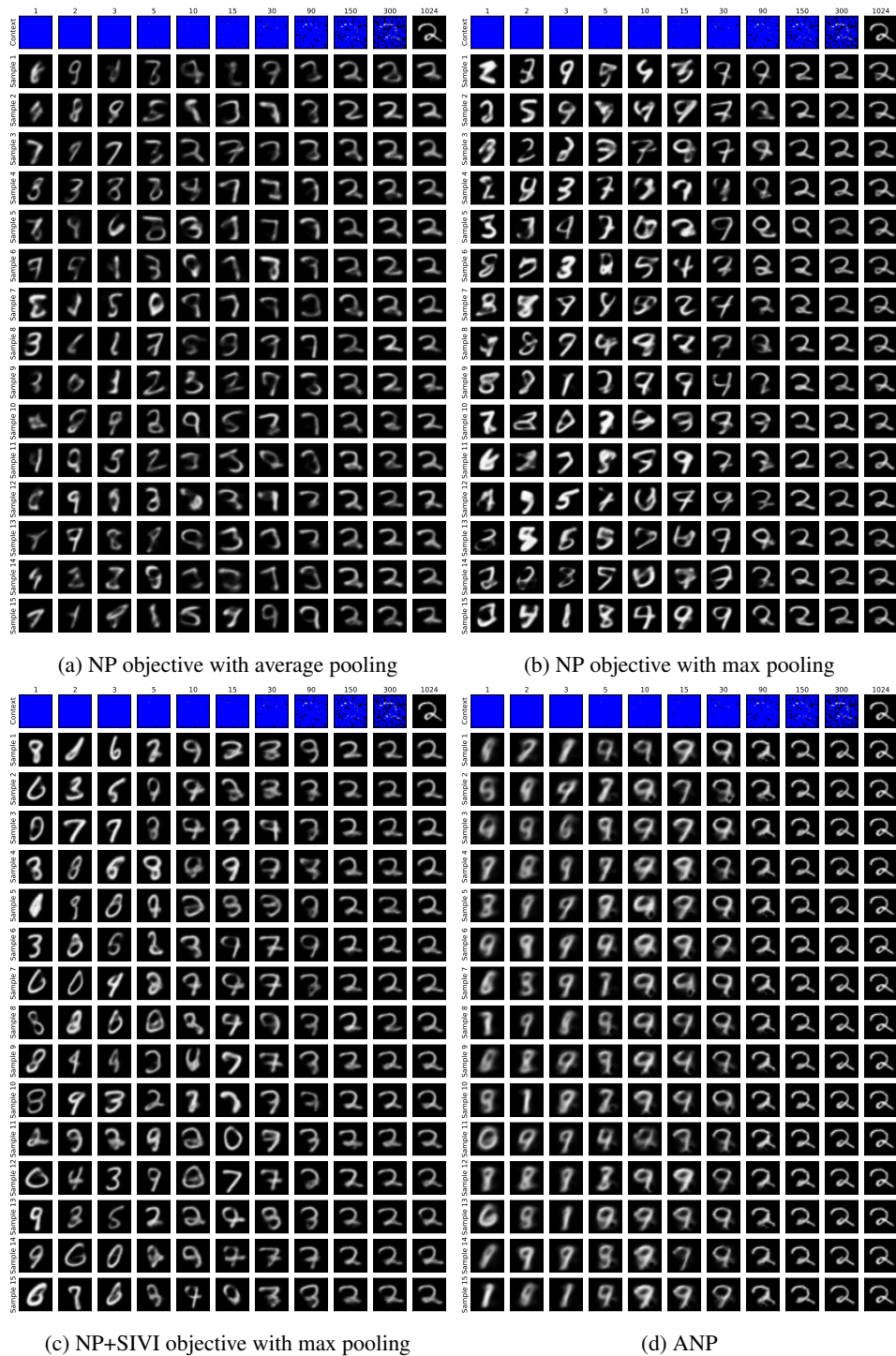


Figure 21: Samples from models trained on the MNIST dataset.



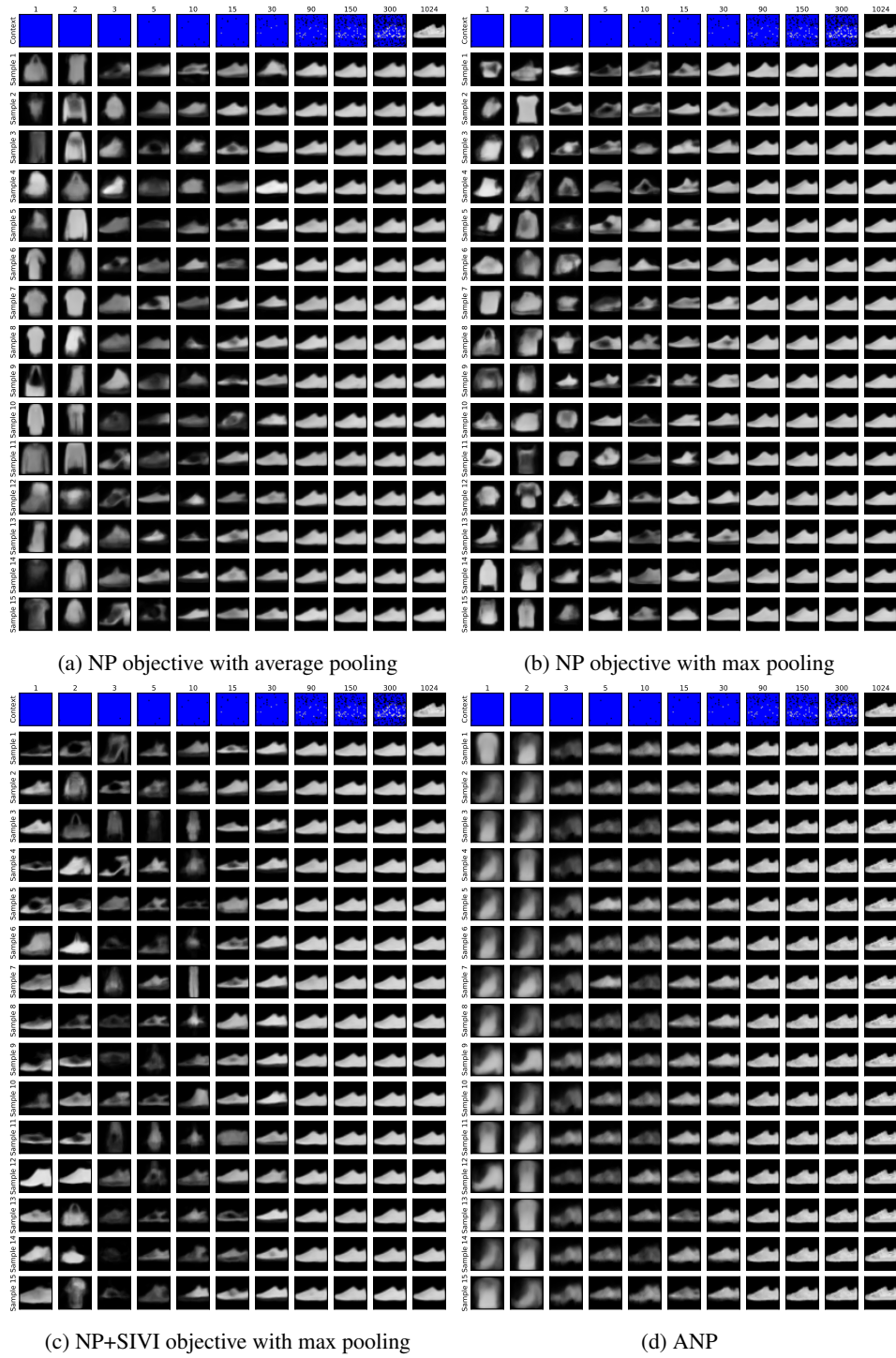


Figure 22: Samples from models trained on the FashionMNIST dataset.

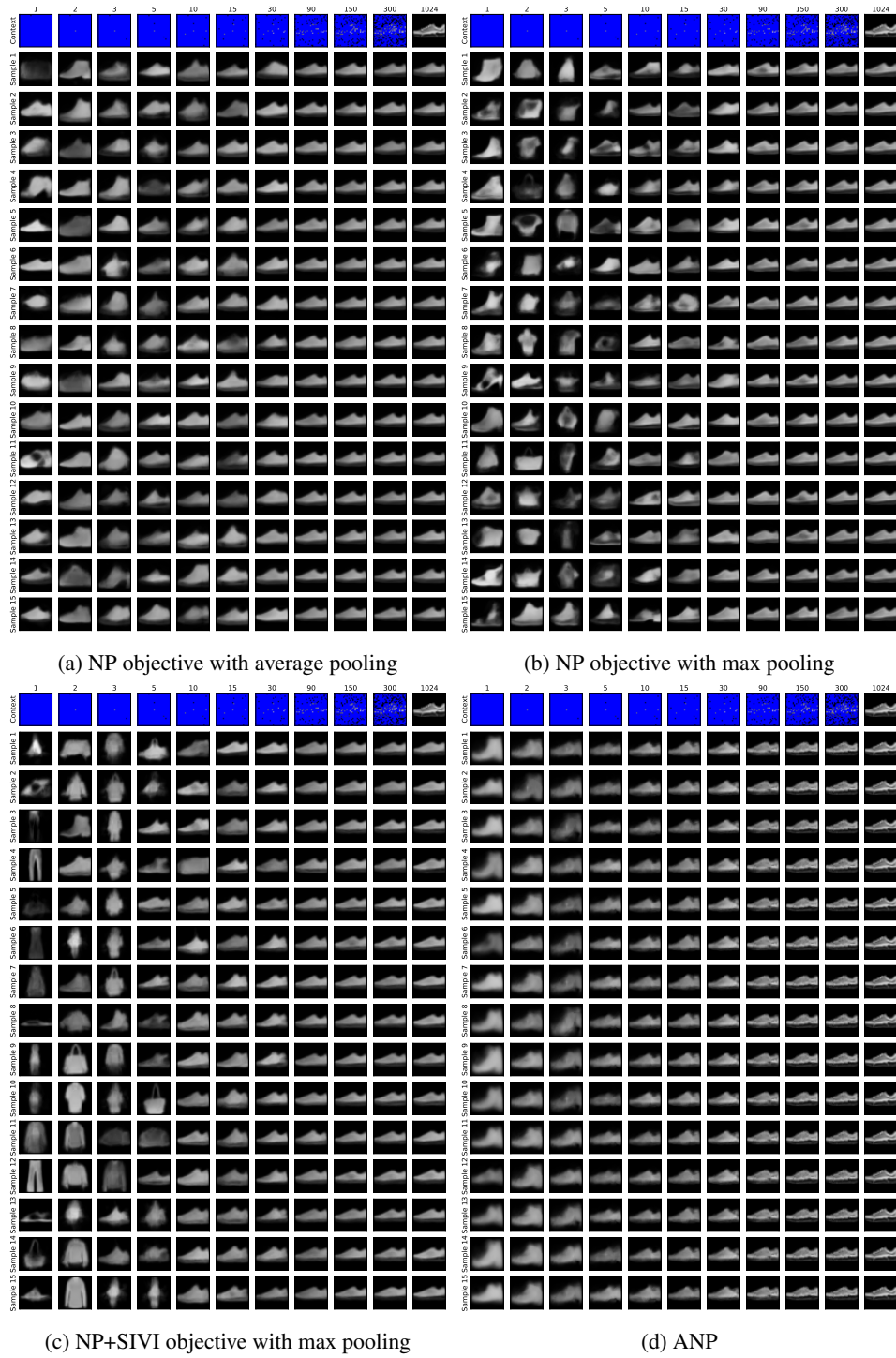


Figure 23: Samples from models trained on the FashionMNIST dataset.

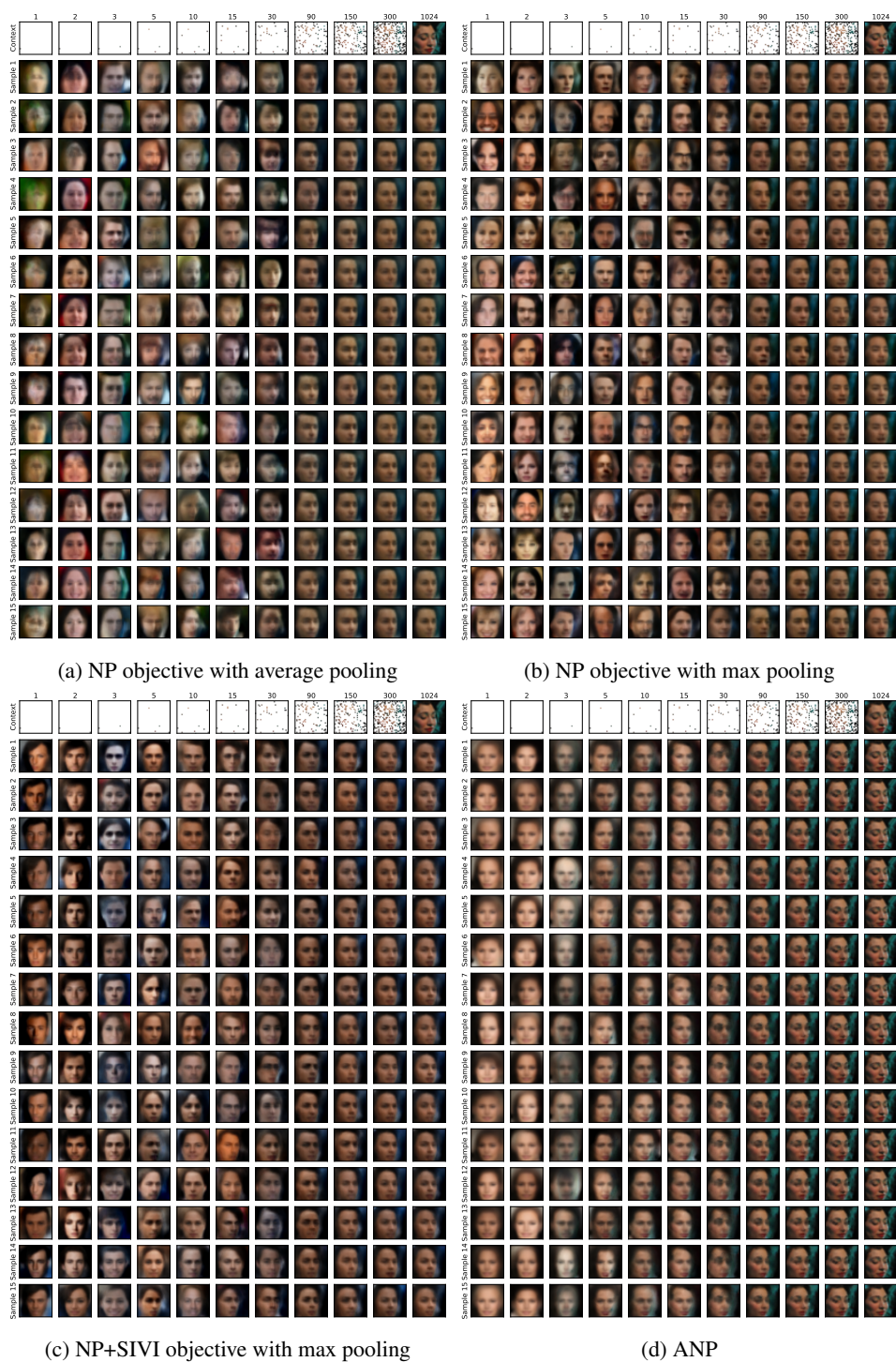


Figure 24: Samples from models trained on the CelebA dataset.



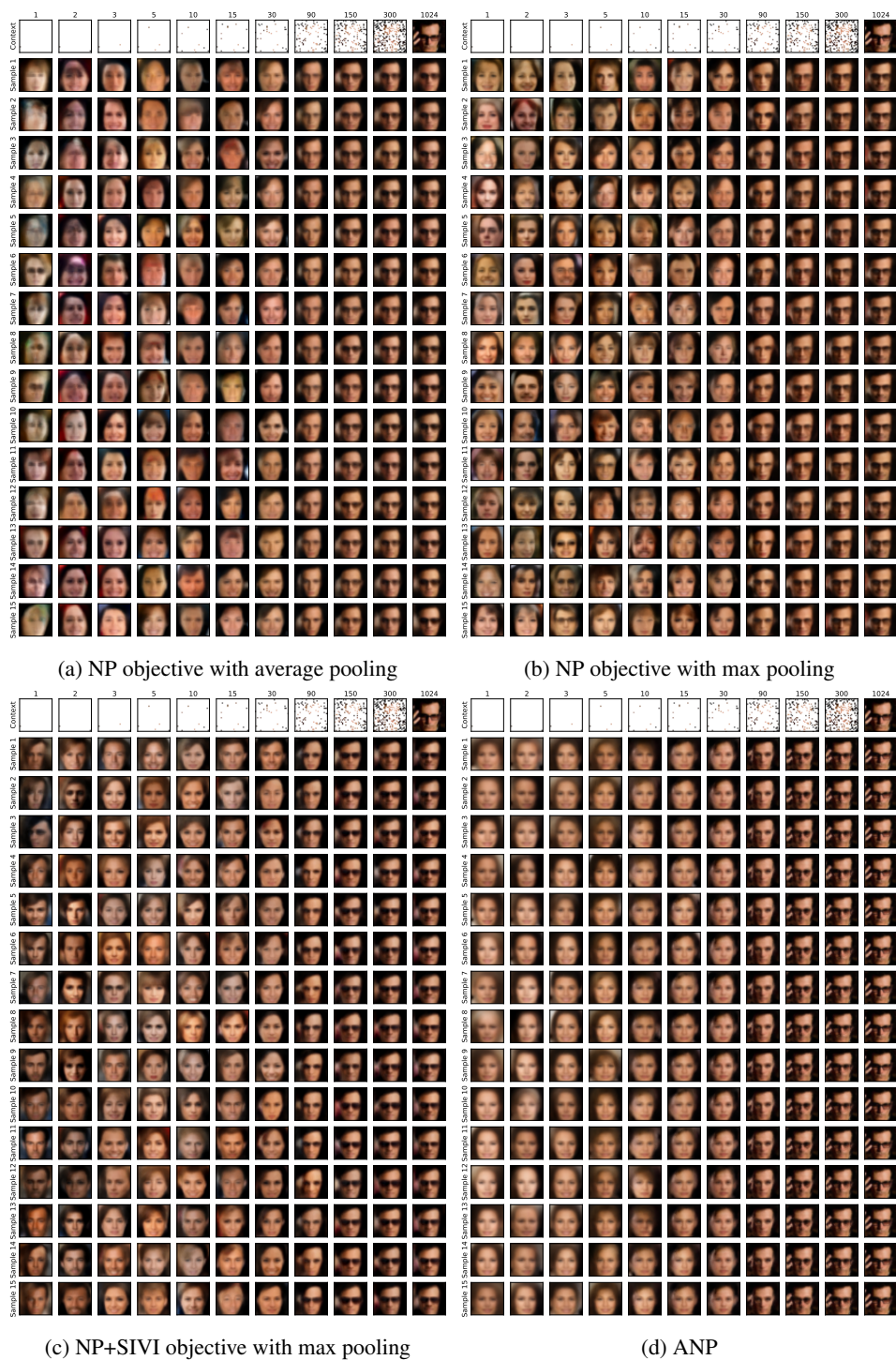


Figure 25: Samples from models trained on the CelebA dataset.