NP-DRAW: A Non-Parametric Structured Latent Variable Model for Image Generation (Supplementary material)

Xiaohui Zeng[1,2] **Raquel Urtasun**[1,2,4] **Richard Zemel**[1,2,4] **Sanja Fidler**[1,2,3] **Renjie Liao**[1,2]

[1] University of Toronto
[2] Vector Institute
[3] NVIDIA
[4]Canadian Institute for Advanced Research

1 EXPERIMENTAL SETUP

Details of Datasets Images in MNIST and Omniglot are of size 28×28, whereas ones in CIFAR-10 are 32×32. We center crop images in CelebA with 148×148 bounding boxes and resize them into 64×64 following [Larsen et al., 2016, Dinh et al., 2016]. Omniglot has 50 different alphabets and 2089 characters. Each alphabet has 41 characters on average and each character belongs to only one alphabet. There are 11 images per character on average. The appearance of different alphabets vary a lot while the appearance of different characters within the same alphabet are similar. The original number of training images and test images are 24345 and 8070 respectively.

Details of Baselines We use the open-source implementation of DRAW[1], VQ-DRAW[2], AIR[3], WGAN[4], and PixelCNN++[5].

For VAEs, we use an architecture with the number of parameters comparable to our model. We follow the hyperparameters used in the original paper if they are given.

For DRAW, we use the same set of hyperparameters for both MNIST and Omniglot, *i.e.*, the number of glimpses is 64, the hidden size of LSTM is 256, the latent size is 100, the read-size is 2×2, and the write-size is 5×5. For experiments on CIFAR-10 and CelebA, we use the number of glimpses as 64, the hidden size of LSTM as 400, the latent size as 200, the read-size as 5×5, and the write-size as 5×5.

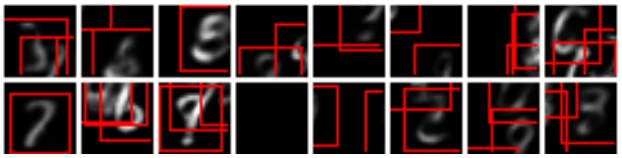
Experiments on VQ-DRAW follow the default setting as the released code. For AIR, we set the maximum number of steps as 3, following the setting in [Eslami et al., 2016]. Since AIR is originally tested on Multi-MNIST, where the number of digits, sizes of digits, and locations of the digits vary. To adapt AIR to datasets we use, we increase the present-probability in their prior as well as the box size and reduce the variance of the location in the prior. We try several groups of hyperparameters (present-probability in $\{0.01, 0.6, 0.8\}$, object size in $\{8, 12, 20\}$, std of object location in $\{1.0, 0.1, 0.001\}$, and object scale in $\{0.5, 1, 1.2, 2\}$) for the prior. The best hyperparameters we found are present-probability $= 0.8$, object size $= 20$, std of object location $= 0.001$ for all datasets. The object scale is set to 0.7 for CelebA and 1.2 for the rest. We also tune the weight of the KL term in their loss function. Specifically, we try the KL weight equal to 1 or 2 for the "what-to-draw" latent variable in their model. We found 2 is better and fix it for the rest of experiments.

We show generated images from AIR on MNIST in Fig. 1 to give more intuition on why AIR has a low FID score. From Fig. 1a, we can see that the reconstruction quality of AIR is reasonable. In terms of generation, as illustrated in Fig. 1b, AIR fails to generate realistic images which is consistent with numbers reported in Table 1, *i.e.*, AIR has worse NLLs on both MNIST and Omniglot datasets compared to other models. . One reason is that the prior of AIR is independent across time steps and is still far from being as good as the posterior (*e.g.*, comparing Fig. 1a with Fig. 1b). For example, the model chooses a location to write at each time step without depending on where it wrote previously. Therefore, during sampling, the box

[1] https://github.com/czm0/draw_pytorch
[2] https://github.com/unixpickle/vq-draw/tree/official-release
[3] https://github.com/addtt/attend-infer-repeat-pytorch
[4] https://github.com/martinarjovsky/WassersteinGAN
[5] https://github.com/pclucas14/pixel-cnn-pp

Supplement for the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI 2021).

(a) Reconstruction of AIR on MNIST. Each pair of images consist of the original MNIST data on the left and the reconstructed image from AIR on the right. The red boxes indicate the predicted location at each time step. We can see that AIR is able to reconstruct the original MNIST data. The model keeps refining the digits at the same location across different time steps.



(b) Sampled images from AIR. The red boxes indicate the generated locations at each time step. The locations across time steps vary in a somewhat arbitrary manner. Moreover, the appearance of some generated digits per step are too dim to be visible.

Figure 1: Sampled and reconstructed images from AIR.

location tends to jump in the canvas in an arbitrary manner. In our experiments, we set the variance of the object location in the prior to be small, but parts (boxes) are still placed on the canvas somewhat randomly. This severely degrades the sample quality since parts are not well aligned.

Details of Our Architecture For our prior model, we use a eight layer Vision Transformer [Dosovitskiy et al., 2020] with hidden size 64 where each time step is one node and the sequence length is equal to the maximum time step. We use dropout with probability 0.1 for Transformer which gives better training losses compared to the prior without dropout. Given a canvas at t, we first have a shallow CNN to extract the feature. The shallow CNN consists of two layers with hidden size 16 and a ReLU unit after the first convolutional layer. The canvas feature is then added to the positional embedding to serve as input feature to Transformer. The prior model can attend to the feature of the canvas at any time step $< t$ and then predict the "what-to-draw" $\mathbf{z}_{\text{id}}$, "where-to-draw" $\mathbf{z}_{\text{loc}}$, and "whether-to-draw" $\mathbf{z}_{\text{is}}$. The positional embedding follows [Vaswani et al., 2017] and encodes the generation step index.

For experiments on Omniglot and MNIST datasets, our encoder consists of four convolutional layers, while the decoder has two convolutional layers with stride 2, two convolutional layers with stride 1 and two transposed convolutional layers. All convolutional layers are followed by Batch Normalization and a ReLU activation function. There are three independent MLP heads to predict the approximated posteriors at T time steps. In particular, we uniformly divide the 2D feature map into T parts and feed the t-th feature to get the corresponding $\mathbf{z}_{\text{id}}$, $\mathbf{z}_{\text{loc}}$, and $\mathbf{z}_{\text{is}}$ at step t. We set the canvas size equal to the image size for MNIST, Omniglot, and CIFAR-10 while use a downsampled (32×32) canvas for CelebA.

Details of Training We first train our prior model for 200 epochs with batch size 64 and learning rate $1e^{-4}$. After pre-training, we choose the prior model with the lowest validation loss and fix it during the training of the full model. The full model is trained with batch size 150 and learning rate $1e^{-3}$. We use the Adam optimizer for training the model. The patch size and the number of parts in the part bank are 5×5 and 50 for MNIST and Omniglot, and 4×4 and 200 for CIFAR-10 and CelebA. The maximum number of steps for MNIST and Omniglot is 36 and the maximum number of steps for CIFAR-10 and CelebA is 64.

Optimization Our latent space contains discrete distribution, which makes the loss difficult to optimize. We use gumbel-softmax [Maddison et al., 2016, Jang et al., 2016] for gradient estimation.

We can expand the KL term in our objective as follow:

$$D_{\text{KL}}(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\log \frac{q(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}\right] \tag{1}$$

$$= \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\sum_{t=1}^{T} \log \frac{q_\phi(\mathbf{z}_{\text{id}}^t|\mathbf{x}) q_\phi(\mathbf{z}_{\text{loc}}^t|\mathbf{x}) q_\phi(\mathbf{z}_{\text{is}}^t|\mathbf{x})}{p_\theta(\mathbf{z}_{\text{id}}^t|\mathbf{c}^{<t}, \mathbf{z}_{\text{loc}}^{<t}) p_\theta(\mathbf{z}_{\text{loc}}^t|\mathbf{c}^{<t}, \mathbf{z}_{\text{loc}}^{<t}) p_\theta(\mathbf{z}_{\text{is}}^t|\mathbf{c}^{<t}, \mathbf{z}_{\text{loc}}^{<t})}\right], \tag{2}$$

where we can apply the Monte Carlo estimators.

2 MORE RESULTS

Method	MNIST	Omniglot	CIFAR-10	CelebA
	28×28	28 × 28	32×32	64×64
VAE	$\leq$ 87.00	$\leq$ 107.90	$\leq$ 5.70	$\leq$ 5.76
NVAE	$\leq$ 79.60	$\leq$ 92.80	$\leq$ **2.91**	$\leq$ **2.06**
BIVA	$\leq$ **78.41**	$\leq$ **91.34**	$\leq$ 3.08	$\leq$ 2.48
Real NVP	80.34	99.60	3.49	3.07
PixelCNN++	79.92	**90.82**	**3.10**	**2.26**
AIR	$\leq$ 128.40	$\leq$ 116.08	*	*
DRAW	$\leq$ **66.07**	$\leq$ 96.54	*	*
Ours	$\leq$ 98.92	$\leq$ 129.73	$\leq$ 5.48	$\leq$ 5.89

Table 1: Comparison against the state-of-the-art likelihood-based generative models. The performance is measured in bits/dimension (bpd) for all the datasets but MNIST and Omniglot in which negative log-likelihood in nats is reported (the lower the better in all cases). * entries are incomparable since they are using continues likelihood for data.

2.1 MORE RESULTS ON GENERATION

In Fig 2, we show more visualization of sampled images from different models on all datasets. From the figure, we can see that our method clearly outperforms VAE and VQ-DRAW and is comparable to WGAN on all datasets (even better than WGAN on Omniglot). We also compare our model against state-of-the-art likelihood based generative models in Table 1. We again approximate the likelihood of VAE-family of models using importance sampling with 50 samples. From the table, we can see that ours is comparable to other structured image models but worse than other generic models including the vanilla VAE. This is likely caused by two facts. First, the construction and the dimension of the latent space between ours and other VAEs are very different which would affect the numerical values of ELBO. Second, likelihood (or ELBO) in general is not a faithful metric that well captures the visual quality of images.

2.2 MORE RESULTS ON LOW-DATA LEARNING

In Fig 3, we show more visualization of sampled images from our model under different percentages of training data. As one can see from the figure, the visual quality of our samples improves as more training data is included. Even with 0.1% training data, our model could still generate plausible faces on CelebA dataset.

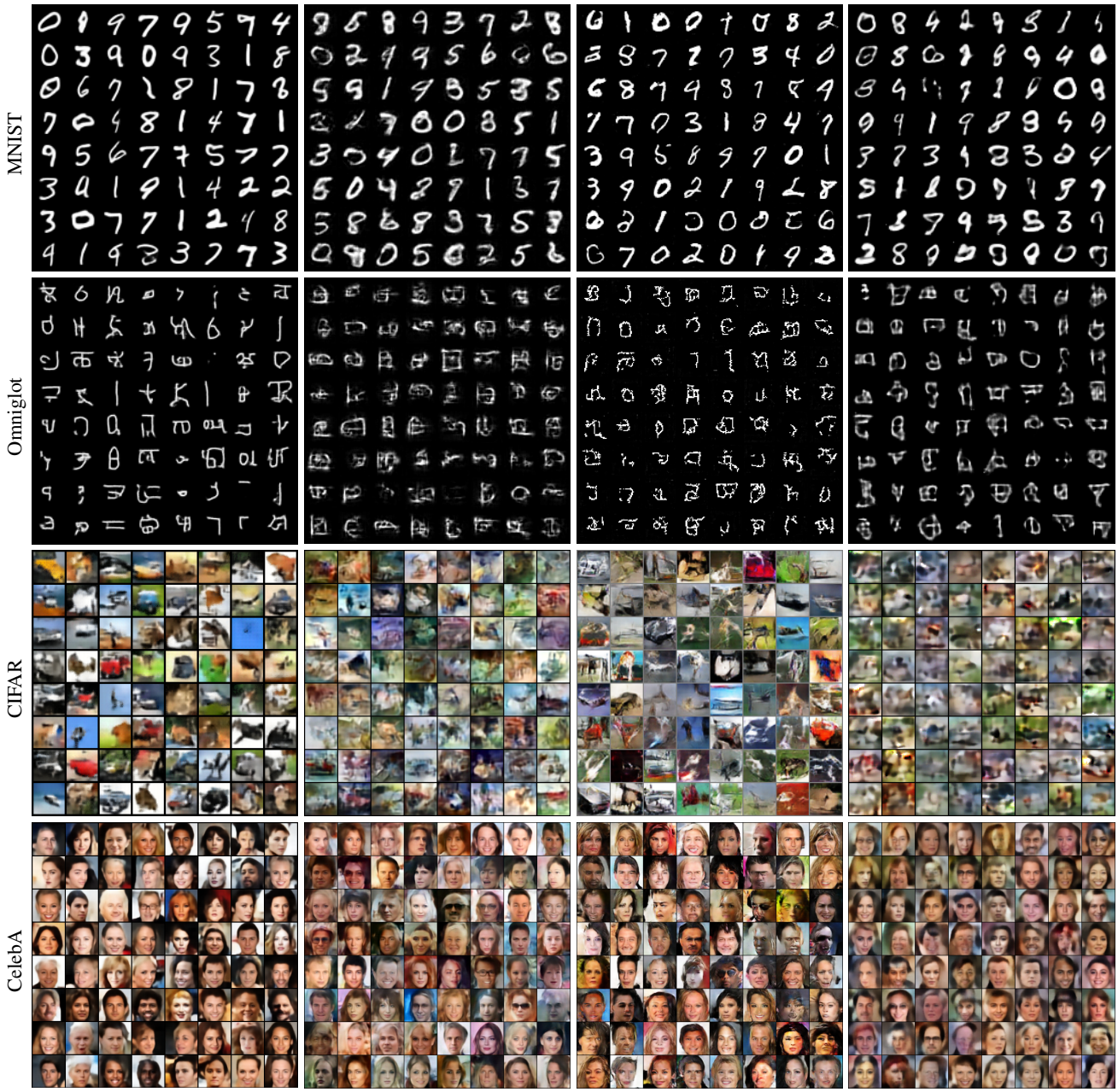
2.3 ABLATION STUDY

To figure out a reasonable range of values fro the patch size K and the size of part bank B, we measure the visual quality of the output of our heuristic parsing algorithm using the peak signal-to-noise ratio (PSNR). In particular, we paste all the selected parts returned by our heuristic parsing on a blank canvas and compute the PSNR between the original image with the created canvas. PSNR is computed as PSNR=$20\log_{10}(\text{MAX}_I) - 10\log_{10}(\text{MSE})$, where MAX_I is the maximum value of the pixel intensity and MSE is the mean squared error between the pasted canvas and the original image. For the threshold ϵ used in our heuristic parsing algorithm, we set its value as 0.01 based on the best PSNR.

References

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.



(a) Ours (b) VQ-DRAW (c) WGAN (d) VAE

Figure 2: Visualization of generated images from different models on different dataset.

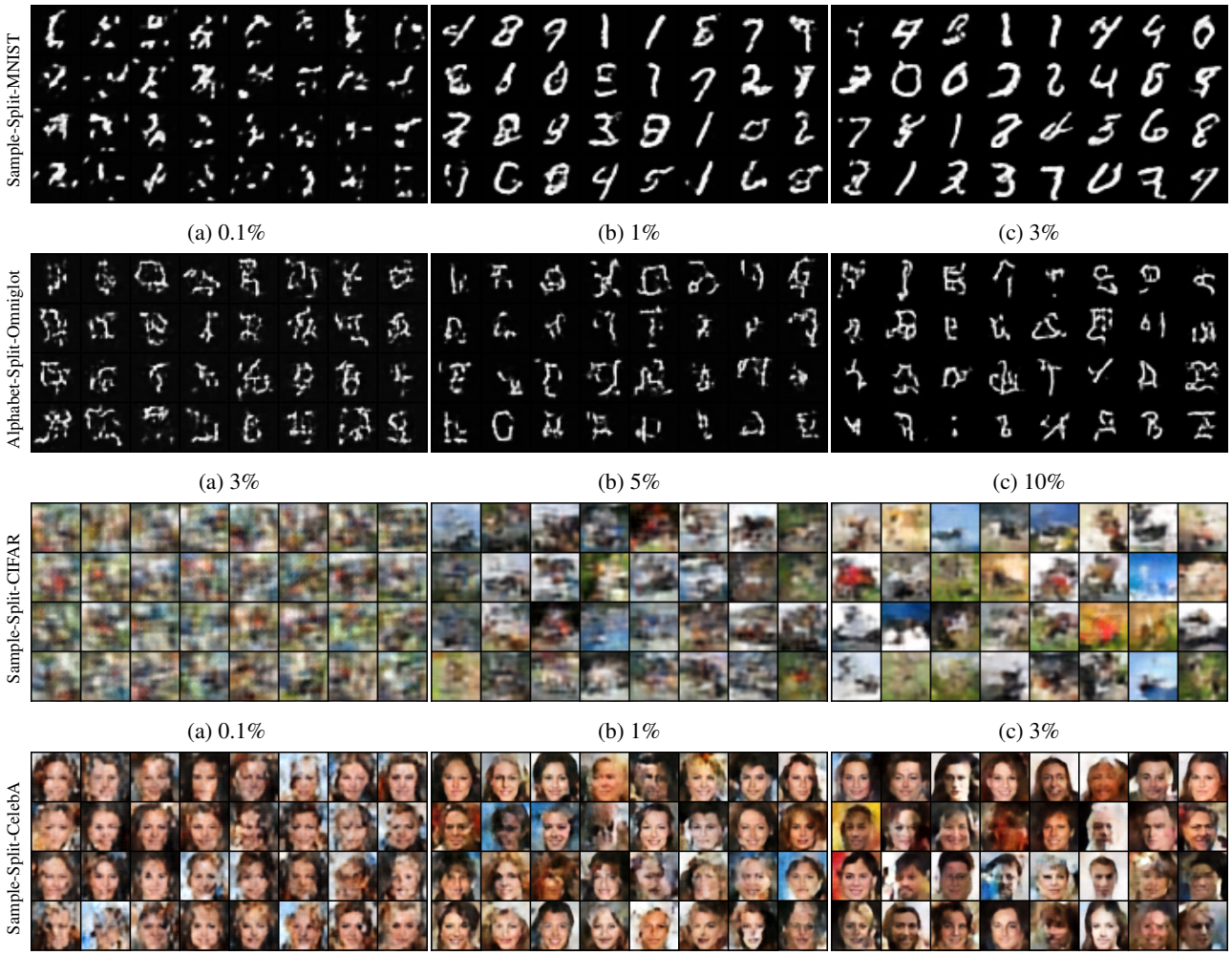


Figure 3: Visualization of generated images from our model on all datasets under different percentages of training data.

SM Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Koray Kavukcuoglu, and Geoffrey E Hinton. Attend, infer, repeat: Fast scene understanding with generative models. *arXiv preprint arXiv:1603.08575*, 2016.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *ICML*, 2016.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.