3rd Symposium on Advances in Approximate Bayesian Inference, 2020 1–14

Learning Discrete Distributions by Dequantization

Anonymous Authors
Anonymous Institution

Abstract

Media is generally stored digitally and is therefore *discrete*. Many successful deep distribution models in deep learning learn a density, *i.e.*, the distribution of a *continuous* random variable. Naïve optimization on discrete data leads to arbitrarily high likelihoods, and instead, it has become standard practice to add noise to datapoints. In this paper, we present a general framework for dequantization via latent variable modelling. In this framework, we are able to recover existing dequantization schemes as special cases, and we are able to derive natural extensions from variational inference literature. We investigate two unexplored directions for dequantization: More sophisticated inference objectives, based on importance-weighting (*iw*) and Rényi variational inference. In addition, we analyze dequantization for different types of distributions, and show that autoregressive dequantization achieves 3.06 bits per dimension in negative log-likelihood on CIFAR10.

1. Introduction

Today, virtually all media is handled digitally. As such, it is stored in bits and is therefore *discrete*. Deep learning models Larochelle and Murray (2011); Kingma and Welling (2014) aim to learn a distribution $p_{\text{model}}(x)$ for high-dimensional data. Many of these models are *density* models Uria et al. (2013); van den Oord and Schrauwen (2014); Dinh et al. (2017); Papamakarios et al. (2017), meaning they learn a distribution of a *continuous* random variable.

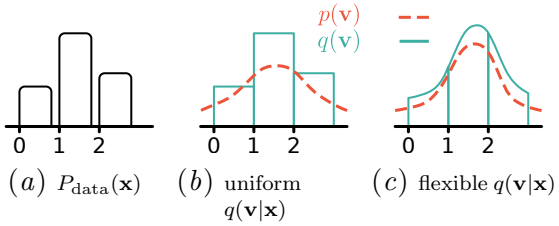


Figure 1: A discrete distribution $P_{\text{data}}(\mathbf{x})$ is dequantized by $q(\mathbf{v}|\mathbf{x})$. Here, the continuous density model $p(\mathbf{v})$ is relatively simple, and two dequantization distributions are considered: one is simple and the other is flexible. Suppose that the dequantization distribution is uniform. Then $p(\mathbf{v})$ is encouraged to have relatively high uncertainty under variational inference. In contrast, when the dequantization distribution $q(\mathbf{v}|\mathbf{x})$ is flexible it can match $p(\mathbf{v})$ which considerably improves the tightness of the variational bound.

Applying a continuous density model to discrete data, may place arbitrarily high likelihood on the discrete locations Theis et al. (2016). Since discrete and continuous spaces are topologically different, a probability density does not necessarily approximate a probability mass. The total probability at a single point under a density is always zero.

In this paper, we present a general approach for dequantization via latent variable modelling. In this framework, we are able to recover existing dequantization schemes as special cases, and we are able to derive natural extensions from variational inference literature. We investigate two directions: more sophisticated variational inference objectives, importance-weighted (*iw*) and Rényi dequantization; and autoregressive distributions, since noise does not need to be inverted. We aim to not only investigate different dequantization methods, but also to highlight its importance.

© A. Authors.

2. Related Work

A large number of distribution models learn a density, a distribution over a continuous variable (Uria et al., 2013; van den Oord and Schrauwen, 2014; Dinh et al., 2017; Papamakarios et al., 2017; Kingma and Dhariwal, 2018; Huang et al., 2018; De Cao et al., 2019; Grathwohl et al., 2019; Hoogeboom et al., 2019b; Ho et al., 2019; Chen et al., 2019; Song et al., 2019; Ma et al., 2019). A standard approach adds uniform noise to discrete values (Theis et al., 2016; Uria et al., 2013; van den Oord and Schrauwen, 2014). Recently, it was proposed to consider a learnable dequantization treated as a variational posterior over latent continuous variables (Ho et al., 2019; Winkler et al., 2019). Here, we derive a new framework for dequantization using latent variable modelling and we present two new dequantization objectives based on Burda et al. (2016); Li and Turner (2016) for VAEs.

3. Methodology

Let $\mathbf{x} \in \mathcal{X}$ denote a vector of D observable discrete random variables and $P_{\text{data}}(\mathbf{x})$ be its (unknown) distribution. We assume there is a set of data $\mathcal{D} = \{\mathbf{x}_n\}$ given, or, equivalently, an empirical distribution $\hat{P}_{\text{data}}(\mathbf{x})$ is provided. The likelihood-based approach to learning a distribution is about finding values of parameters of a model $P_{\text{model}}(\mathbf{x})$ that maximize the log-likelihood function: $\log P_{\text{model}}(\mathcal{D}) = \mathbb{E}_{\mathbf{x}\sim\hat{P}_{\text{data}}(\mathbf{x})}[\log P_{\text{model}}(\mathbf{x}_n)]$.

3.1. Dequantization as a latent variable model

Frequently, a discrete distribution models a proxy of a continuous variable in the physical world. For instance, a digital photograph of an observed scene represents the light that is reflected from observed objects, quantized to a certain precision. In other words, we can consider a latent variable model where continuous latent variables $\mathbf{v} \in \mathbb{R}^D$ correspond to a continuous representation of the world and observable discrete variables $\mathbf{x}$ are measured quantities. This suggests the following model: $P_{\text{model}}(\mathbf{x}) = \int P_\vartheta(\mathbf{x}|\mathbf{v})p_\theta(\mathbf{v})\mathrm{d}\mathbf{v}$, where $P_\vartheta(\mathbf{x}|\mathbf{v})$ is an indicator function of $\mathbf{v}$ being contained in a volume $\mathcal{B}_\vartheta(\mathbf{x}) \subseteq \mathbb{R}^D$, namely, $P_\vartheta(\mathbf{x}|\mathbf{v}) = \mathbb{1}[\mathbf{v} \in \mathcal{B}_\vartheta(\mathbf{x})]$, and $p_\theta(\mathbf{v})$ is a continuous distribution, which may be modeled using a flexible density model (MacKay and Gibbs, 1999; Dinh et al., 2017; Rippel and Adams, 2013). We refer to $P_\vartheta(\mathbf{x}|\mathbf{v})$ as a *quantizer*. Note that in principle the volumes $\mathcal{B}_\vartheta$ can be constructed to induce any type of partition of a volume space, where care should be taken that $\mathcal{B}_\vartheta$ for different $\mathbf{x}$ do not overlap. When we set $\mathcal{B}(\mathbf{x}) = \{\mathbf{x} \cdot \mathbf{u} | \mathbf{u} \in \mathbb{R}^D_+\}$ for $\mathbf{x} \in \{-1, 1\}^D$ we recover half-infinite dequantization for binary variables from Winkler et al. (2019). In this paper, since image data is often represented on a square grid we will focus on *hypercubes*, namely, $\mathcal{B}(\mathbf{x}) = \{\mathbf{x} + \mathbf{u} : \mathbf{u} \in [0, 1)^D\}$.

Notice that Theis et al. (2016); Ho et al. (2019) require the definition $P(\mathbf{x}) = \int_{[0,1)^D} p(\mathbf{x}+\mathbf{u})\mathrm{d}\mathbf{u}$ to relate a discrete and continuous model. In contrast, our method is derived without this definition, and the quantizer volume $\mathcal{B}$ generalizes to any volumetric partition.

Calculating the integral in the latent variable model is troublesome, and thus, learning is infeasible especially in high dimensional cases. Therefore, in order to alleviate this issue, we introduce a new distribution $q_\phi(\mathbf{v}|\mathbf{x})$ with parameters ϕ, a *dequantizing* distribution or *dequantizer*. In fact, the dequantizer should have the same support as $P_\vartheta(\mathbf{x}|\mathbf{v})$, otherwise it would assign probability mass to regions outside the volume $\mathcal{B}(\mathbf{x})$. Therefore, we will

use $\mathbf{u}$ instead of $\mathbf{v}$ in the dequantizing distribution to highlight the fact that the support of $q_\phi(\mathbf{v}|\mathbf{x})$ equals $\mathcal{B}(\mathbf{x})$, where we define $\mathbf{v} = \mathbf{x} + \mathbf{u}$. Including the dequantizer yields:

$$P_{\text{model}}(\mathbf{x}) = \int \frac{q_\phi(\mathbf{u}|\mathbf{x}) P_\vartheta(\mathbf{x}|\mathbf{v}) p_\theta(\mathbf{v})}{q_\phi(\mathbf{u}|\mathbf{x})} \mathrm{d}\mathbf{v} = \mathbb{E}_{\mathbf{u}\sim q_\phi(\mathbf{u}|\mathbf{x})}\Big[\frac{P_\vartheta(\mathbf{x}|\mathbf{v}) p_\theta(\mathbf{v})}{q_\phi(\mathbf{u}|\mathbf{x})}\Big],$$

Introducing the dequantizer allows us to connect dequantization to the broad literature on variational inference. We propose three approaches to approximate the integral: *i)* variational inference, *ii)* weighted importance sampling and *iii)* variational Rényi approximation.

3.2. Variational Dequantization

We can interpret the dequantizing distribution as a *variational* distribution and apply Jensen's inequality to obtain the lower-bound on the log-likelihood function:

$$\log P_{\text{model}}(\mathbf{x}) \geq \mathbb{E}_{\mathbf{u}\sim q_\phi(\mathbf{u}|\mathbf{x})}\Big[\log \frac{P_\vartheta(\mathbf{x}|\mathbf{v}) p_\theta(\mathbf{v})}{q_\phi(\mathbf{u}|\mathbf{x})}\Big]. \tag{1}$$

The dequantizing distribution must be restricted to assign probability mass to $\mathcal{B}(\mathbf{x})$ only, otherwise the lower-bound is undefined for certain samples $\mathbf{u} \sim q_\phi(\mathbf{u}|\mathbf{x})$. For our choice of $\mathcal{B}(\mathbf{x})$ being a hypercube, we can apply the sigmoid function to the output of the dequantizer to ensure the lower-bound has appropriate support. Thus, we can re-write (1) as follows:

$$\log P_{\text{model}}(\mathbf{x}) \geq \mathbb{E}_{\mathbf{u}\sim q_\phi(\mathbf{u}|\mathbf{x})}\Big[\log p_\theta(\mathbf{v})\Big] + \mathbb{H}[q_\phi], \tag{2}$$

which recovers the variational dequantization (*vi dequantization*) from Ho et al. (2019).

3.3. Importance-Weighted Dequantization

Alternatively, we can interpret the dequantizing distribution as a *proposal* distribution and instead of using Jensen's inequality we sample K times from $q_\phi(\mathbf{u}|\mathbf{x})$, which directly approximates the log-likelihood:

$$\log P_{\text{model}}(\mathbf{x}) \geq \log\Big[\frac{1}{K}\sum_{k=1}^{K} \frac{P_\vartheta(\mathbf{x}|\mathbf{v}_k) p_\theta(\mathbf{v}_k)}{q_\phi(\mathbf{u}_k|\mathbf{x})}\Big] = \log\Big[\frac{1}{K}\sum_{k=1}^{K} \frac{p_\theta(\mathbf{v}_k)}{q_\phi(\mathbf{u}_k|\mathbf{x})}\Big], \tag{3}$$

where $\mathbf{u}_k \sim q_\phi(\mathbf{u}|\mathbf{x})$ and $\mathbf{v}_k = \mathbf{x} + \mathbf{u}_k$ for $k = 1, 2, \ldots, K$. The last equality follows if we constrain the proposal distribution (the dequantizer) in the same manner as we did in the case of the variational dequantization (*i.e.*, the probability mass should be assigned only to $\mathcal{B}(\mathbf{x})$). This objective was studied in the context of VAEs in (Burda et al., 2016). In general, if $K \to \infty$, then we obtain an equality in (3). But since we take a finite sample, the approximate gives a lower-bound to the log-likelihood function (*iw-bound*). Importantly, the iw-bound is tighter than the variational lower-bound Burda et al. (2016); Domke and Sheldon (2018). Hence, the importance-weighting is preferable over the variational inference and in practice it leads to a better log-likelihood performance. We refer to this dequantization scheme as *iw dequantization*.

3.4. Rényi Dequantization

The variational inference and importance-weighting sampling for a latent variable model could be generalized by noticing that both approaches are special cases of the variational

Rényi bounds. The log-likelihood function could be lower-bounded by the Rényi divergence approximated with the sample from $q_\phi(\mathbf{u}|\mathbf{x})$ of size $K < \infty$ (Li and Turner, 2016):

$$\log P_{\text{model}}(\mathbf{x}) \geq \frac{1}{1-\alpha} \log \Big[\frac{1}{K} \sum_{k=1}^{K} \Big(\frac{P_\vartheta(\mathbf{x}|\mathbf{v}_k) p_\theta(\mathbf{v}_k)}{q_\phi(\mathbf{u}_k|\mathbf{x})} \Big)^{1-\alpha} \Big],$$

where $\alpha \in [0, 1)$ is a hyperparameter. Interestingly, for $\alpha \to 1$ we obtain the variational lower-bound and for $\alpha = 0$ we get the iw-bound. Li and Turner (2016) have further shown that it is advantageous to consider $\alpha < 0$, because it may give tighter bounds than the iw-bound when the sample size K is low.[1] Setting $\alpha = -\infty$ corresponds to picking the largest importance weight value. By restricting the domain of $q(\mathbf{u}|\mathbf{x})$ to $\mathcal{B}(\mathbf{x})$ can obtain the variational Rényi max approximation (*VR-max*):

$$\log P_{\text{model}}(\mathbf{x}) \approx \log \max_{k=1,2,\ldots,K} \Big[\frac{p_\theta(\mathbf{v}_k)}{q_\phi(\mathbf{u}_k|\mathbf{x})} \Big]. \tag{4}$$

The maximum weight dominates the contributions of all the gradients Li and Turner (2016). Thus, the VR-max approach could be seen as a fast approximation to the importance-weighting, since it speeds up computations by considering only one example instead of K in calculating gradients. We refer to this whole dequantization scheme as *Rényi dequantization*.

3.5. Dequantizing distributions

The dequantizing distribution plays an important role in the framework and its flexibility allows to obtain better log-likelihood scores. Importantly, the dequantizing distribution is a conditional distribution and we use it for sampling instead of calculating probabilities.

Uniform The special case in which $q_\phi(\mathbf{u}|\mathbf{x})$ is a uniform distribution over $\mathcal{B}(\mathbf{x})$, is the setting introduced in (Theis et al., 2016; Uria et al., 2013; van den Oord and Schrauwen, 2014), termed *uniform dequantization*.

Gaussian A more powerful dequantization scheme than the uniform dequantization is a conditional logit-normal distribution Atchison and Shen (1980), namely, $q_\phi(\mathbf{u}|\mathbf{x}) = \text{sigm}\Big(\mathcal{N}\big(\mu_\phi(\mathbf{x}), \Sigma_\phi(\mathbf{x})\big)\Big)$, where $\mu_\phi(\mathbf{x})$ and $\Sigma_\phi(\mathbf{x})$ denote the mean and the covariance matrix for given $\mathbf{x}$, respectively, and $\text{sigm}(\cdot)$ is the sigmoid function.

Flow Instead of using a certain family of distribution, we can define the quantizer by applying the change of variables formula, that is:

$$q_\phi(\mathbf{u}|\mathbf{x}) = q_\phi\big(\boldsymbol{\varepsilon} = f_\phi(\text{sigm}^{-1}(\mathbf{u}); \mathbf{x})|\mathbf{x}\big)|\boldsymbol{J}|, \tag{5}$$

where $f : \mathbb{R}^D \to \mathbb{R}^D$ is a bijective map to a simple base distribution $q_\phi(\boldsymbol{\varepsilon}|\mathbf{x})$, and $\boldsymbol{J} = \frac{\partial \boldsymbol{\varepsilon}}{\partial \mathbf{u}}$ denotes a Jacobian matrix. Notice we highlight the need of using the (inverse) sigmoid function on top of the bijective map in order to ensure correct support, *i.e.*$\mathbf{u} \in [0, 1)^D$. There are two important parts of a flow-based model, namely, a choice of a *base distribution* $q_\phi(\boldsymbol{\varepsilon}|\mathbf{x})$ and the bijective map f_ϕ. Here we decide to use a diagonal Gaussian base distribution Dinh et al. (2017). Whereas Ho et al. (2019) studied bipartite maps for f_ϕ, here we examine autoregressive bijective maps, since dequantization noise does not have to be inverted.

1. To be precise, if we consider the infinite sample for $\alpha < 0$, we get an upper-bound on the log-likelihood function. However, taking $K < \infty$ results in a tight lower-bound according to Corollary 1 in Li and Turner (2016).

Bipartite flows The bipartite bijective maps ensure invertibility by splitting an input into two parts, and processing only the second part (Dinh et al., 2017).

Autoregressive flows We can model $q_\phi(\mathbf{u}|\mathbf{x})$ with an 'expensive to invert' bijective map. We find that an autoregressive model as proposed for variational autoencoders (Kingma et al., 2016) is an appealing choice for dequantization. We utilize the following:

$$\mathbf{u} = \exp\tanh(\mathbf{s}) \odot \boldsymbol{\varepsilon} + \mathbf{m} \text{ where } [\mathbf{m}, \mathbf{s}] = \mathrm{ARM}_\phi(\boldsymbol{\varepsilon}, \mathbf{h}), \tag{6}$$

where ARM_ϕ is an autoregressive model (an autoregressive neural network), $\mathbf{h}$ is a context variable that is calculated based on the conditioning $\mathbf{x}$ using a neural network, $\mathbf{s}$. We refer to this dequantization scheme as *Autoregressive Dequantization (ARD)*.

3.6. Distributions for the density model

The continuous model $p_\theta(\mathbf{v})$ is the crucial component in the presented framework since the better performance depends on the flexibility of this model. In principle, any continuous density model could be used as $p_\theta(\mathbf{v})$, *e.g.*, any model mentioned in section 3.5, with the important difference that the sigmoid function is not used as the support does not need to be confined. In practice, however, $p_\theta(\mathbf{v})$ has to be evaluated during training *and* we are interested in sampling $\mathbf{v} \sim p_\theta(\mathbf{v})$. Hence, utilizing models with autoregressive components would be prohibitively slow. Therefore, in our experiments, we consider a Gaussian distribution with diagonal covariance, full covariance, and a bipartite flow-based model (a series of coupling layers and a factored-out base distribution) as a continuous distribution.

4. Experiments

To understand and evaluate different dequantization schemes, they are tested on different data problems: *i)* a 2-dimensional binary problem, *ii)* (statically) binarized MNIST (bMNIST) (Larochelle and Murray, 2011) and centered patches of bMNIST, which is derived directly from MNIST and *iii)* CIFAR10 (Krizhevsky et al., 2009) (8 bit and 5 bit). Performance is evaluated on a held-out test-set using negative log-likelihood. This method of evaluation is common in deep distribution learning literature because it allows for an information theoretic interpretation: the negative $\log_2$-likelihood is expressed in *bits* or *bits per dimension* (bpd), where the latter is an average over dimensions and gives the lossless compression size. In the experiments we consider diagonal Gaussians, covariance Gaussians and flows as distribution models, since these models admit exact likelihood evaluation. A detailed description of experiments *(i)* and *(ii)*, and architectures and optimization can be found in the Appendix.

4.1. Image distribution modelling

In this experiment the model using ARD is compared with other methods in the literature. Experiments show that our model outperforms other methods in the literature on both variational inference objective and negative likelihood (Table 1). In general we compare to models that do not require an autoregressive inverse to sample from, where the exception is marked *. In particular, we report *vi* evaluation, also referred to as Expected Lower Bound (ELBO), and we report the approximate negative likelihood - $\log P(\mathbf{x})$ using 1000

Table 1: Comparison of negative log-likelihood, *vi* dequantization (ELBO) evaluation of our model versus literature. - $\log P(\mathbf{x})$ is approximated using 1000 importance weighted samples. $\mathrm{KL}(q_\phi|p_\theta)$ is the difference between - $\log P(\mathbf{x})$ and *vi*. In bits per dimension, lower is better.

Method	$\mathrm{KL}(q_\phi\|p_\theta)$	*vi*	$-\log P(x)$
IAF-VAE (Kingma et al., 2016)	0.04	3.15	3.11
BIVA (Maaløe et al., 2019)	0.04	3.12	3.08
Glow (Kingma and Dhariwal, 2018)	*n/a*	3.35	*n/a*
FFJORD (Grathwohl et al., 2019)	*n/a*	3.40	*n/a*
IDF (Hoogeboom et al., 2019a)	-	-	3.32
MintNet (Song et al., 2019)⋆	*n/a*	3.32	*n/a*
Residual Flow (Chen et al., 2019)†	*n/a*	3.28	*n/a*
Flow++ (Ho et al., 2019)†	0.04	3.12	3.08
ARD (ours)	**0.03**	**3.09**	**3.06**

⋆ Sampling from model requires autoregressive inverse.
† Sampling from model requires other iterative procedures.
n/a not available, this value exists but was not reported in the literature.

importance weighted samples following Maaløe et al. (2019). Note that Ho et al. (2019) use 16384 samples, which skews the experiment in their favour for *vi* and - $\log P(\mathbf{x})$, but against them for $\mathrm{KL}(q_\phi|p_\theta)$. Architecturally, the density model in ARD is most similar to IDF (Hoogeboom et al., 2019a), where 1×1 convolutions from Glow (Kingma and Dhariwal, 2018) and scale transformations from RealNVP (Dinh et al., 2017) are added. Flow++ (Ho et al., 2019) has additional attention layers and MintNet (Song et al., 2019) has autoregressive transformations instead of coupling layers in the density model. Note that even though our model utilizes autoregressive components similar to MintNet (Song et al., 2019), our model is computationally cheap to invert since it does not require the solution to autoregressive inverses. Residual Flow (Chen et al., 2019) utilizes invertible ResNets instead of coupling layers.

5. Conclusion

In this paper we presented a framework for dequantization via latent variable modeling. Using this interpretation, we derive natural extensions from variational inference literature for dequantization: importance-weighted (*iw*) and Rényi dequantization. We show that low bit-depth data combined with simple dequantizers benefit from these objectives. In addition, we analyze dequantization for different types of distributions (simple and complicated) and show that autoregressive dequantization achieves 3.06 bits per dimension in negative log-likelihood on CIFAR10.

References

J Atchison and Sheng M Shen. Logistic-normal distributions: Some properties and uses. *Biometrika*, 67(2):261–272, 1980.

Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *4th International Conference on Learning Representations, ICLR*, 2016.

Ricky TQ Chen, Jens Behrmann, David Duvenaud, and Jörn-Henrik Jacobsen. Residual flows for invertible generative modeling. *arXiv preprint arXiv:1906.02735*, 2019.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Block neural autoregressive flow. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI*, page 511, 2019.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR*, 2017.

Justin Domke and Daniel R. Sheldon. Importance weighting and variational inference. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS*, pages 4475–4484, 2018.

Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFJORD: free-form continuous dynamics for scalable reversible generative models. In *7th International Conference on Learning Representations, ICLR*, 2019.

Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *Proceedings of the 36th International Conference on Machine Learning, ICML*, pages 2722–2730, 2019.

Emiel Hoogeboom, Jorn WT Peters, Rianne van den Berg, and Max Welling. Integer Discrete Flows and Lossless Compression. *Advances in Neural Information Processing Systems 32, NeurIPS*, 2019a.

Emiel Hoogeboom, Rianne van den Berg, and Max Welling. Emerging convolutions for generative normalizing flows. In *Proceedings of the 36th International Conference on Machine Learning, ICML*, pages 2771–2780, 2019b.

Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron C. Courville. Neural autoregressive flows. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, pages 2083–2092, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR*, 2015.

Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 10236–10245, 2018.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR*, 2014.

Diederik P. Kingma, Tim Salimans, Rafal Józefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improving variational autoencoders with inverse autoregressive flow. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems*, 2016.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Hugo Larochelle and Iain Murray. The neural autoregressive distribution estimator. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS*, pages 29–37, 2011.

Yingzhen Li and Richard E. Turner. Rényi divergence variational inference. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems*, pages 1073–1081, 2016.

Xuezhe Ma, Xiang Kong, Shanghang Zhang, and Eduard H. Hovy. Macow: Masked convolutional generative flow. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing (NeurIPS)*, pages 5891–5900, 2019.

Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. BIVA: A very deep hierarchy of latent variables for generative modeling. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems NeurIPS*, pages 6548–6558, 2019.

David JC MacKay and Mark N Gibbs. Density networks. *Statistics and neural networks: advances at the interface. Oxford University Press, Oxford*, pages 129–144, 1999.

George Papamakarios, Iain Murray, and Theo Pavlakou. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 2338–2347, 2017.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NeurIPS Autodiff Workshop*, 2017.

Oren Rippel and Ryan Prescott Adams. High-dimensional probability estimation with deep density models. *arXiv preprint arXiv:1302.5125*, 2013.

Yang Song, Chenlin Meng, and Stefano Ermon. Mintnet: Building invertible neural networks with masked convolutions. *CoRR*, abs/1907.07945, 2019.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. In *4th International Conference on Learning Representations, ICLR*, 2016.

Benigno Uria, Iain Murray, and Hugo Larochelle. RNADE: the real-valued neural autoregressive density-estimator. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems.*, pages 2175–2183, 2013.

Aäron van den Oord and Benjamin Schrauwen. Factoring variations in natural images with deep gaussian mixture models. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems*, pages 3518–3526, 2014.

Christina Winkler, Daniel Worrall, Emiel Hoogeboom, and Max Welling. Learning Likelihoods with Conditional Normalizing Flows. *CoRR*, abs/1912.00042, 2019.

Appendix A. Experiments

A.1. Analysis in 2d

The different dequantization methods and objectives are analyzed in two dimensions using the *binary checkerboard*, which places equal probability over two of the four states in the binary space $\{0,1\}^2$, such that $P_{\text{data}}(x) = 0.5$ if $x = (1,0)$ or if $x = (0,1)$, and zero otherwise. Although the theoretical likelihood limit of a dataset is typically unknown, for the binary checkerboard this is exactly 1 bit, because there is an equal probability over two events. Since the problem is two dimensional, the learned distributions can be visualized. Figure 2 depicts the probability density of the dequantizer $q(\mathbf{v}|\mathbf{x})$ and the density model $p(\mathbf{v})$, for models trained using *vi*-dequantization. Since by construction $q(\mathbf{v}|\mathbf{x})$ only places density on a bin corresponding to $\mathbf{x}$, the distribution $q(\mathbf{v}|\mathbf{x})$ can be visualized without overlap in the marginal distribution $q(\mathbf{v}) = \mathbb{E}_{\mathbf{x}\sim P_{\text{data}}(\mathbf{x})}[q(\mathbf{v}|\mathbf{x})]$.

When the model $p(\mathbf{v})$ is a flow and the dequantizer $q(\mathbf{v}|\mathbf{x})$ is uniform (Figure 2a), the model $p(\mathbf{v})$ is struggling to adequately model the boundaries of the dequantized density regions. When the model $p(\mathbf{v})$ is a simple diagonal Gaussian and the dequantizer $q(\mathbf{v}|\mathbf{x})$ is a flow (Figure 2b), the flexible dequantizer compensates the limitations of the density model by shaping itself to the limitations of the simple distribution. An interesting variant we would like to highlight is when the density model $p(\mathbf{v})$ is a Gaussian with covariance, and $q(\mathbf{v}|\mathbf{x})$ is a flow (Figure 2c). Aided by the dequantizer, the model $p(\mathbf{v})$ aims to place density on the diagonal line which improves the performance to 1.08 bits, which is already close to the theoretical limit. Surprisingly, when the density model is relatively simple, a flexible dequantizer can compensate a lot. The best performance is achieved when both $q(\mathbf{v}|\mathbf{x})$ and $p(\mathbf{v})$ are flexible (Figure 2d). For this problem we observe the density contracts somewhat away from boundaries, and the center has relatively high density.

The effects seen in Figure 2 are also confirmed quantitatively with the likelihood performance of these models (Table 2). Note that the more flexible the distributions, the better the performance. An interesting observation is that when $p(\mathbf{v})$ is a flow distribution, a Gaussian $q(\mathbf{v}|\mathbf{x})$ and a flow $q(\mathbf{v}|\mathbf{x})$ have equal performance. Presumably, the flexibility of $p(\mathbf{v})$ does not require a more complicated dequantizer for this relatively simple problem. In addition the effects *iw* and *Rényi* dequantization are shown in Table 3. Uniformly dequantized models that are trained using *Rényi* or *iw* dequantization are considerably

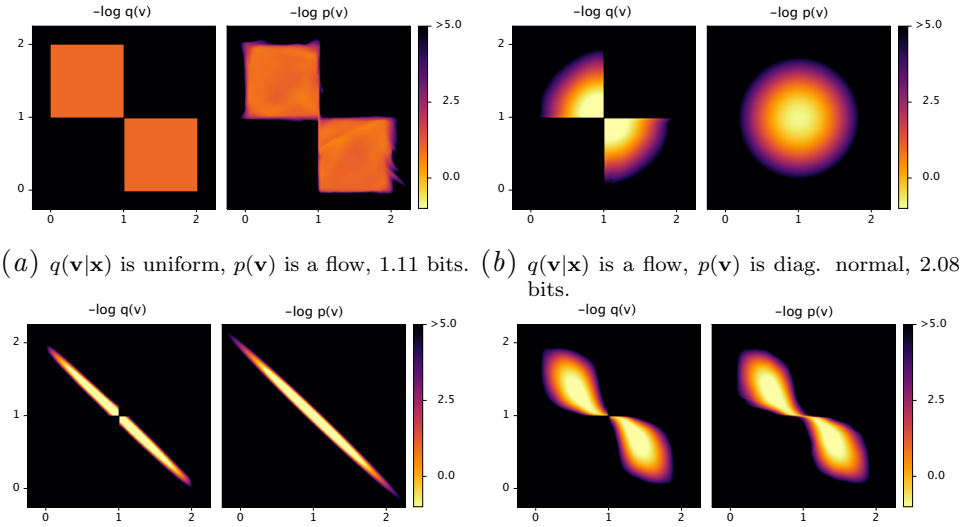
Table 2: Binary checkerboard *vi*-dequantization performance for different dequantizer $q(\mathbf{v}|\mathbf{x})$ and density model $p(\mathbf{v})$ pairs in bits. Lower is better.

		$q(\mathbf{v}\|\mathbf{x})$		
		Uniform	Diag.	Flow
$p(\mathbf{v})$	Diag.	2.51	2.08	2.01
	Cov.	1.91	1.66	1.08
	Flow	1.11	1.02	1.02

Table 3: Likelihood performance on binary checkerboard when trained with *iw* or *Rényi* dequantization in bits per dimension (bpd). Lower is better.

	$q(\mathbf{v}\|\mathbf{x})$		
	Uniform	Normal	Flow
vi⋆	1.05	1.00	1.00
iw ($K = 16$)	1.00	1.00	1.00
Rényi ($K = 2$)	1.02	1.00	1.01

⋆ *vi* is equivalent to *iw* or *Rényi* with $K = 1$.



(a) $q(\mathbf{v}|\mathbf{x})$ is uniform, $p(\mathbf{v})$ is a flow, 1.11 bits. (b) $q(\mathbf{v}|\mathbf{x})$ is a flow, $p(\mathbf{v})$ is diag. normal, 2.08 bits.

(c) $q(\mathbf{v}|\mathbf{x})$ is a flow, $p(\mathbf{v})$ is a cov. normal, 1.08 bits. (d) $q(\mathbf{v}|\mathbf{x})$ is a flow, $p(\mathbf{v})$ is a flow, 1.02 bits.

Figure 2: Density visualization of different density models $p(\mathbf{v})$ and dequantizer $q(\mathbf{v}|\mathbf{x})$ pairs. This figure considers a selection, for all different pairs we tested please see the Appendix. The dequantizing distributions is visualized in the marginal distribution $q(\mathbf{v}) = \mathbb{E}_{\mathbf{x}\sim P_{\text{data}}(\mathbf{x})}[q(\mathbf{v}|\mathbf{x})]$. Models are trained using *vi*-dequantization and the values reported are *vi* evaluation.

better than *vi* in terms of likelihood. For more complicated dequantizers though, we find that improvements are negligible. Therefore, these sophisticated objectives appear to be particularly useful when the dequantization distribution is simple. Note that due to the low dimensionality of the problem *Rényi* dequantization will typically result in loose bounds for larger values than $K = 2$.

A.2. Analysis in high-dim: Image modelling

Similar to the 2d example, more sophisticated training objectives are most advantageous when dequantization distributions are simple, which can be seen in Table 6. For binary MNIST and patches of binary MNIST, training using *iw* dequantization improves negative likelihood performance consistently. However, for more expressive learnable dequantizers the added benefit of these objectives becomes smaller. For CIFAR10 we train only last 100 epochs with the sophisticated objectives and the first with *vi* to reduce the computational cost. In high-bit depth settings such as CIFAR10, we find that the performance gains of importance weighting are minimal. Hence, for simple dequantizers on data with low bit depth, *iw*-dequantization may considerably improve performance. *Rényi* dequantization achieves similar but slightly worse performance, which is acceptable since it is a faster approximation.

Autoregressive Dequantization Experiments show that ARD outperforms all other dequantization distributions, when trained using comparable architectures (see Table 4 and 5). Two results are particularly striking: Firstly, even when binary MNIST is learned using a simple density model $p(\mathbf{v})$ (covariance normal), ARD achieves a negative log-likelihood of 0.183 bpd. In constrast, the uniformly dequantized covariance model achieves only

Table 4: Performance of *vi* dequantization on binary MNIST for different density model $p(\mathbf{v})$ and dequantizer distributions $q(\mathbf{v}|\mathbf{x})$ pairs. In bits per dimension, lower is better.

$p(\mathbf{v})$		$q(\mathbf{v}\|\mathbf{x})$ Unif.	Normal	Bipart.	*ARD*
Cov.	$\mathrm{KL}(q_\phi\|p_\theta)$	0.061	0.046	0.010	**0.007**
	vi	0.533	0.268	0.196	**0.190**
	$-\log P(x)$	0.472	0.242	0.186	**0.183**
Flow	$\mathrm{KL}(q_\phi\|p_\theta)$	0.014	0.007	**0.005**	**0.005**
	vi	0.176	0.156	0.153	**0.152**
	$-\log P(x)$	0.162	0.149	0.148	**0.147**

Table 5: Performance of *vi* dequantization on CIFAR10 8 and 5 bit for a flow density model $p(\mathbf{v})$ and different dequantizer distributions $q(\mathbf{v}|\mathbf{x})$. In bits per dimension.

	$q(\mathbf{v}\|\mathbf{x})$ Unif.	Normal	Bipart.	*ARD*
8 bit $\mathrm{KL}(q_\phi\|p_\theta)$	0.03	**0.02**	**0.02**	**0.02**
8 bit *vi*	3.29	3.21	3.18	**3.16**
8 bit $-\log P(x)$	3.26	3.19	3.16	**3.14**
5 bit $\mathrm{KL}(q_\phi\|p_\theta)$	0.04	0.02	**0.01**	0.02
5 bit *vi*	1.65	1.50	1.43	**1.41**
5 bit $-\log P(x)$	1.61	1.48	1.42	**1.39**

Table 6: Likelihood performance for models trained with *iw* or *Rényi* objectives and uniform dequantization on binary MNIST and CIFAR10 in bits per dimension (bpd). The reported values are a (bounded) approximations of - $\log P(\mathbf{x})$ using *iw*-dequantization with 256 samples. Lower is better.

Dataset	bMNIST 2×2			bMNIST 4×4			bMNIST	CIFAR10
$q(\mathbf{v}\|\mathbf{x})$	Unif.	Normal	Bipart.	Unif.	Normal	Bipart.	Unif.	Unif.
vi	0.747	0.724	0.723	0.633	0.603	0.601	0.162	3.26
iw ($K = 4$)	0.726	**0.722**	**0.721**	**0.610**	**0.600**	**0.599**	**0.159**	3.25
Rényi ($K = 4$)	**0.724**	**0.722**	0.723	**0.610**	0.602	0.600	0.160	3.25

0.472 bpd. Secondly, notice that dequantization seems to matter more when bit-depths are smaller. To see this, consider the log-likelihood improvement when comparing uniform dequantization and ARD: For the 8 bit data the improvement is 0.12 bpd, which is about 3.7% relative to the total bpd. However, for 5 bit data the improvement is already 0.20 bpd which is about 12% relatively. Hence, log-likelihood modelling of lower bit depth data may especially benefit from more expressive dequantizers.

Recommendations This section aims to give the reader recommendations on what dequantization methods to use and what gains are to be expected. When bit-depths are small and dequantization distributions are fixed, we find that *iw* or *Rényi* dequantization objectives improve log-likelihood performance. Note that by design of the objectives, the approximate posterior $q(\mathbf{v}|\mathbf{x})$ will diverge more from the (unknown) true posterior $p(\mathbf{v}|\mathbf{x})$. Therefore, a downside of these objectives is that a single sample *iw* dequantization (equivalent to *vi*) will be a poor approximation to the log-likelihood, and instead multiple samples are required to obtain accurate estimates.

In contrast, when dequantization noise can be learned or bit-depths are higher, it may be better to simply use *vi* dequantization and better performance is obtained by increasing the complexity of the dequantizer. Note that at the cost of some small performance losses, Gaussian dequantization might be a good simple alternative to flow-based dequantization.

Appendix B. Architecture and Optimization details

In the experiments we consider diagonal Gaussian, covariance Gaussian and flows as distribution models, since these models admit exact likelihood evaluation. The diagonal Gaussian is parametrized straightforwardly using parameters for mean and log scale. The covariance Gaussian is parametrized using a Cholesky decomposition, *i.e.*, the precision $\Lambda = \mathrm{LL}^{\mathrm{T}}$ where L is the learnable parameter. The diagonal of L is modelled separately using a log diagonal parameter, which ensures positive-definiteness of Λ. The covariance matrix is defined then as $\Sigma = \Lambda^{-1}$. Further, flows have an architecture as described in Kingma and Dhariwal (2018) using the densenet coupling networks from Hoogeboom et al. (2019a). For MNIST data we use the given split of 40000 train, 10000 validation and 10000 test images. In the bMNIST patches experiments, center patches of the relevant size are taken. For CIFAR10 we split the 50000 training images into the first 40000 for train and the last 10000 for validation, we use the 10000 test images as provided.

Models were all optimized using Kingma and Ba (2015) with a learning rate of 0.0005 and standard β parameters. Furthermore, during initial 10 epochs the learning rate is multiplied by epoch divided by 10, referred to as *warmup* Kingma and Dhariwal (2018). All our code was implemented in PyTorch Paszke et al. (2017). Since experiments are computationally expensive, architectures were trained once. Models were trained using two Nvidia Tesla V100 GPUs, with Nvidia driver 410.104, CUDA 10.0, and cuDNN v7.5.1. In this setting, smaller models (Binary MNIST) take approximately three days to complete, and larger models (CIFAR10) require five to six days to complete. Results are obtained by running models a single time, due to the large computational power that normalizing flows require. The basic architecture was built following Kingma and Dhariwal (2018): The flow is divided in multiple levels with a decreasing number of dimensions. At the end of every level, half of the representation is modelled using a factor out layer (splitprior) Dinh et al. (2017); Kingma and Dhariwal (2018). Every level consists of subflows, *i.e.* a coupling layer followed by a 1×1 convolution Kingma and Dhariwal (2018). The coupling layers utilize neural networks as described in Hoogeboom et al. (2019a). For the autoregressive transformation, we utilize the masking as described in Song et al. (2019). In terms of autoregressive order, this is equivalent to reshaping a $\mathrm{C} \times \mathrm{H} \times \mathrm{W}$ image to a vector and applying the autoregressive mask. This is opposed to masking in Kingma et al. (2016), which is equivalent to a mask on a reshaped $\mathrm{H} \times \mathrm{W} \times \mathrm{C}$ image. In practice, the autoregressive transformation is obtained by masking convolutions.

Table 7: Optimization details.

Experiment	levels	subflows	net. depth	net. channels	context channels	q levels	q subflows	batch size
Binary checkerboard	1	8	12	192	16	1	4	128
Binary MNIST patches	1	8	12	192	16	1	4	128
Binary MNIST	2	8	12	192	16	1	4	128
CIFAR10 5bit	2	10	12	768	16	1	2	256
CIFAR10	2	10	12	768	16	1	2	256
CIFAR10 (Literature comparison)	2	18	12	768	16	1	2	128

Appendix C. Additional results

Visualization of samples $\mathbf{v} \sim p(\mathbf{v})$ from a density model, and the quantizer $\mathbf{x} \sim P(\mathbf{x}|\mathbf{v})$ are depicted in Figure 3. The quantizer is simply a Kronecker delta peak and amounts to applying a floor function in the case of hypercube partitioning. The density model is a flow trained with autoregressive dequantization on standard 8 bit CIFAR10. Notice that although the method is trained using autoregressive dequantization, the density model $p(\mathbf{v})$ uses bipartite transformations and does not require the solution to autoregressive inverses.

Table 8: Likelihood performance for models trained with *iw* or *Rényi* objectives and uniform dequantization on binary MNIST and CIFAR10 in bits per dimension (bpd). The reported values are a (bounded) approximations of - $\log P(\mathbf{x})$ using *iw*-dequantization with 256 samples. Lower is better.

Dataset	bMNIST 4×4				
	$K = 1$	$K = 2$	$K = 4$	$K = 8$	$K = 16$
iw	0.633	0.619	0.610	0.607	0.604
Rényi	0.633	0.621	0.610	0.608	0.609

Appendix D. Visualizations on Binary Checkerboard

In this section a comprehensive overview of the distributions dequantizer and density model pairs is visualized. The models trained using variational inference are displayed in Table 9. In general, the dequantizer $q(\mathbf{v}|\mathbf{x})$ and density model $p(\mathbf{v})$ try to compensate for each other where they are lacking flexibility. This effect can be seen when $q(\mathbf{v}|\mathbf{x})$ is a flow and $p(\mathbf{v})$ is a diagonal Gaussian, a covariance Gaussian and lastly a flow. When $p(\mathbf{v})$ is a flow, it is generally difficult to capture the edges of the squares when dequantization noise is uniform. However, both Gaussian and flow dequantization perform equally when the model $p(\mathbf{v})$ is a flow. In this simple problem, Gaussian dequantization is sufficiently flexible when combined with a flow.

The models trained using *iw* and *Rényi* dequantization objectives are depicted in Table 10. An important difference with *vi*-dequantization is that it is much less important for $q(\mathbf{v}|\mathbf{x})$ and $p(\mathbf{v})$ to match completely. Rather, more emphasis is placed so that $p(\mathbf{v})$ places distribution somewhere in the appropriate bin, where the exact location in the bin matters less. As a result, when $q(\mathbf{v}|\mathbf{x})$ is uniform the model $p(\mathbf{v})$ is not forced to learn the uniform square and retracts somewhat away from the edges.

Figure 3: Samples from the flow model in the literature comparison, trained using *ARD*.

Table 9: Different dequantizer $q(\mathbf{v}|\mathbf{x})$ and density model $p(\mathbf{v})$ pairs trained using *vi*-dequantization. The depicted values are computed using *vi*-dequantization (ELBO).

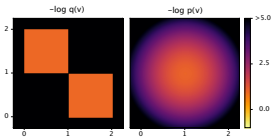
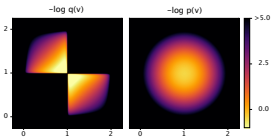
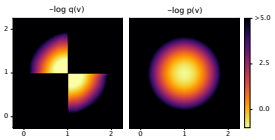
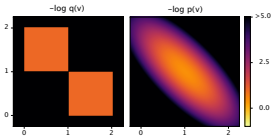
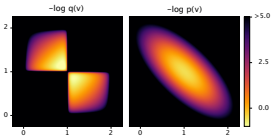
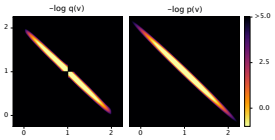
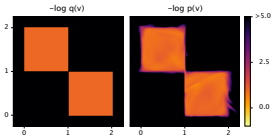
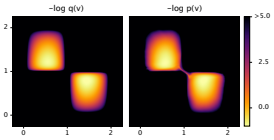
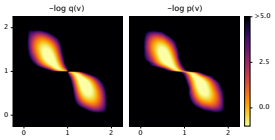
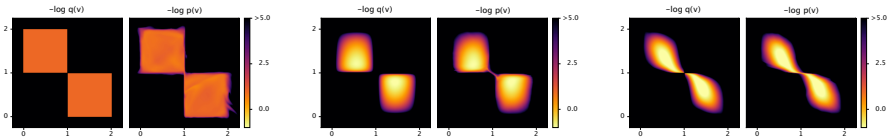
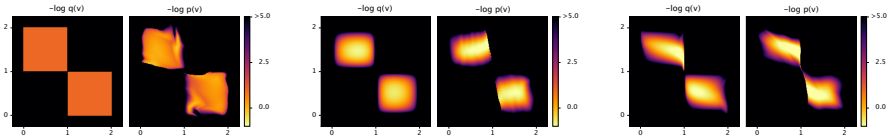
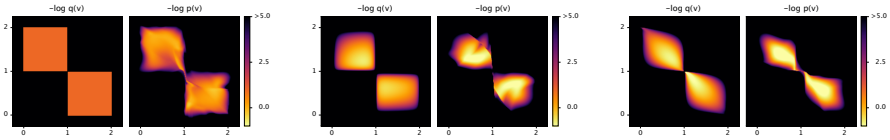
		$q(\mathbf{v}\|\mathbf{x})$		
		Uniform	Normal	Flow
$p(\mathbf{v})$	Normal diag.	 2.51	 2.08	 2.01
	Normal cov.	 1.91	 1.66	 1.08
	Flow	 1.11	 1.02	 1.02

Table 10: Models trained using *iw* and *Rényi* dequantization with different dequantizing distributions $q(\mathbf{v}|\mathbf{x})$, and a flow $p(\mathbf{v})$. The values are an approximation of - $\log P(x)$ using importance-weighted dequantization with $K_{\text{test}} = 256$ samples.

	$q(\mathbf{v}\|\mathbf{x})$ Uniform	Normal	Flow
vi	 1.05	1.00	1.00
iw $(K = 16)$	 1.00	1.00	1.00
Rényi $(K = 2)$	 1.02	1.00	1.01