LOSSLESS COMPRESSION USING CONTINUOUSLY-INDEXED NORMALIZING FLOWS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recently, a class of deep generative models known as continuously-indexed flows (CIFs) have expanding the modelling capacity of normalizing flows (NFs) in the context of both density estimation and variational inference. CIFs are provably more general and expressive than NFs, but do not induce a closed-form density model and thus require additional considerations when applying in the same contexts that NFs have shown promise. One such area is lossless compression, where NFs have been used as the density model to develop a compression scheme, known as local bits-back, with expected codelength approximately equal to the average negative log-likelihood of the NF density model. Here, we propose to extend the local bits-back scheme to CIF-based density models, as the improved expressiveness inherent in CIFs stands to reduce the expected codelength of compressed data. We also leverage recent work on compression schemes built with hierarchical variational auto-encoders – as hierarchical CIFs can themselves be seen as interpolating between these and NFs – gaining further expressiveness in our density models and effectiveness in our compression scheme.

1 INTRODUCTION

As datasets become larger, so too do the challenges in both modelling the data-generating process and providing efficient compression schemes. These two problems are linked through the framework of entropy encoding, which describes a family of lossless compression techniques making use of a model of the data probability distribution. A combination of recent advancements on both fronts have led to the development of practical lossless compression algorithms for high-dimensional data.

Perhaps most notably, Townsend et al. (2019) has demonstrated how the entropy encoding scheme known as asymmetric numeral systems (ANS) (Duda, 2009) can be integrated with bits-back coding (Frey & Hinton, 1996) to build a powerful compression method backed by variational auto-encoders (VAEs) (Kingma & Welling, 2014). Other approaches also combining ANS with bits-back coding, but making use of other deep generative models such as hierarchical VAEs (Kingma et al., 2019) and normalizing flows (NFs) (Ho et al., 2019), have emerged and shown promising empirical results, achieving codelengths close to theoretical optimum. In all cases, the expected codelength of the compression scheme is negatively correlated with the log-likelihood (or evidence lower bound) of the generative model, implying that a more expressive generative model will correspond to a better algorithm for lossless compression.

One way to directly and provably improve the expressiveness of an NF-based model is to use continuously-indexed flows (CIFs) (Cornish et al., 2020). This recently-proposed framework *augments* the generative process of any standard NF with additional indexing variables, relaxing the restrictive bijectivity constraint imposed on the baseline flow. The cost of increased modelling capacity is a now intractable density model, but this improved expressiveness has been able to overcome any obstacles stemming from this intractability in the context of both density estimation (Cornish et al., 2020) and variational inference (Caterini et al., 2020).

In this work we propose an efficient lossless compression scheme for using hierarchical CIFs as the probability model for bits-back coding. The scheme we propose builds on the ideas of BB-ANS (Townsend et al., 2019) and local bits-back coding (Ho et al., 2019) to allow for compression using a single layer CIF model, as well as Bit-Swap (Kingma et al., 2019) that allows us to decrease the

number of auxiliary bits required to utilize a hierarchical CIF model which significantly decreases the codelengths in practice, especially for short data sequences.

2 BACKGROUND

2.1 CONTINUOUSLY-INDEXED FLOWS

CIFs (Cornish et al., 2020) propose to model continuous data defined over some space $\mathcal{X}$ as the X-marginal of

$$Z \sim p_Z, \quad U \mid Z \sim p_{U|Z}(\cdot \mid Z), \quad X = F(Z; U), \tag{1}$$

where p_Z is a density defined over some latent space $\mathcal{Z}$, $p_{U|Z}$ is a parametrized conditional density defined over an *indexing* space $\mathcal{U}$, and $F : \mathcal{Z} \times \mathcal{U} \to \mathcal{X}$ is a function such that $F(\cdot; \mathbf{u})$ is a bijection for each $\mathbf{u} \in \mathcal{U}$. If we parametrize F such that, e.g. $F(\cdot; 0) = f(\cdot)$ for a standard normalizing flow transformation $f : \mathcal{Z} \to \mathcal{X}$, then it is easy to see that equation 1 strictly generalizes the typical normalizing flow density model $Z \sim p_Z, X = f(Z)$. The cost for this improvement is an intractable density model $p_X(\mathbf{x}) = \int p_{X,U}(\mathbf{x}, \mathbf{u}) \, \mathrm{d}\mathbf{u}$, although tractable $p_{X,U}$ emerges from equation 1.

We can also stack the final two steps of equation 1 to gain further expressiveness as per Cornish et al. (2020). Specifically, model X now as the Z_L-marginal of the following L-layered model:

$$Z_0 \sim p_{Z_0}, \quad U_\ell \mid Z_{\ell-1} \sim p_{U_\ell|Z_{\ell-1}}(\cdot \mid Z_{\ell-1}), \quad Z_\ell = F_\ell(Z_{\ell-1}; U_\ell), \tag{2}$$

where the final two steps are repeated for $\ell \in \{1, \ldots, L\}$. We can write the joint density over $(X, U_{1:L})$ recursively, but p_X remains intractable, cf. equation 12 of Cornish et al. (2020). However, we can learn the parameters of p_X using variational inference, constructing an inference model

$$q_{U_{1:L}|X}(\mathbf{u}_{1:L} \mid \mathbf{x}) \coloneqq \prod_{\ell=1}^{L} q_{U_\ell|Z_\ell}(\mathbf{u}_\ell \mid \mathbf{z}_\ell), \tag{3}$$

where $\mathbf{z}_L \coloneqq \mathbf{x}$, $\mathbf{z}_\ell \coloneqq F_{\ell+1}^{-1}(\mathbf{z}_{\ell+1}; \mathbf{u}_{\ell+1})$ recursively for $\ell \in \{1, \ldots, L\}$, and each $q_{U_\ell|Z_\ell}$ is a parametrized conditional density. We can then train a CIF density model by maximizing the evidence lower bound (ELBO) objective, given for a single point $\mathbf{x} \in \mathcal{X}$ as:

$$\mathcal{L}(\mathbf{x}) \coloneqq \mathbb{E}_{\mathbf{u}_{1:L} \sim q_{U_{1:L}|X}(\cdot|\mathbf{x})} \left[\log p_{X,U_{1:L}}(\mathbf{x}, \mathbf{u}_{1:L}) - \log q_{U_{1:L}|X}(\mathbf{u}_{1:L} \mid \mathbf{x}) \right] \leq \log p_X(\mathbf{x}). \tag{4}$$

It is important to note that the factorization of the true posterior over indexing variables $p_{U_{1:L}|X}$ matches that of equation 3 (Cornish et al., 2020, Appendix B.6), which helps to increase the value of $\mathcal{L}(\mathbf{x})$ as equation 4 is maximized in q when $q_{U_{1:L|X}} = p_{U_{1:L}|X}$.

2.2 BITS-BACK CODING

Bits-back coding (Frey & Hinton, 1996) is a method that allows for lossless compression using density models with latent variables, which Townsend et al. (2019) recently combined with asymmetric numeral systems (ANS) (Duda, 2009) to devise a practical compression scheme based on variational auto-encoders (VAEs). Specifically, suppose we have access to a VAE with generative model $p_{X,Z}(\mathbf{x}, \mathbf{z}) \coloneqq p_{X|Z}(\mathbf{x} \mid \mathbf{z}) \cdot p_Z(\mathbf{z})$ and encoder $q_{Z|X}(\mathbf{z} \mid \mathbf{x})$ trained over a dataset $\mathcal{D} \coloneqq \{\mathbf{x}_i\}_i$. Given a stack-like entropy encoding scheme such as ANS, we can iteratively encode each point $\mathbf{x}_i$ in the dataset onto a pre-existing bit stack – here denoted m – by performing the following steps: (i) decode $\mathbf{z}_i$ from m using $q_{Z|X}(\cdot \mid \mathbf{x}_i)$; (ii) encode $\mathbf{x}_i$ onto m using $p_{X|Z}(\cdot \mid \mathbf{z}_i)$; (iii) encode $\mathbf{z}_i$ onto m using p_Z. We can reverse this procedure to then decode the dataset from m once encoding is complete. The expected codelength for a point $\mathbf{x}$ is approximately equal to the negative evidence lower bound $\mathbb{E}_{\mathbf{z} \sim q_{Z|X}(\cdot|\mathbf{x})}[\log q_{Z|X}(\mathbf{z} \mid \mathbf{x}) - \log p_{X,Z}(\mathbf{x}, \mathbf{z})]$, which conveniently also serves as the VAE objective function, underscoring the relationship between performant generative models and efficient compression schemes.

2.3 LOCAL BITS-BACK CODING

Local bits-back coding is a method allowing for lossless compression using normalizing flows (Ho et al., 2019). The key idea behind this scheme is reinterpreting $\mathbf{z}$ and $\mathbf{x}$, which in the context of

standard NFs are related by a bijection $\mathbf{x} = f(\mathbf{z})$, as random variables which have instead a fuzzy relationship centred on a sharply peaked normal distribution. In the context of CIFs, the flow F is a now a bijection between $\mathbf{z}$ and $\mathbf{x}$ *conditioned* on $\mathbf{u}$ as noted above. Thus, given $\mathbf{u} \in \mathcal{U}$, the fuzzy relationship between $\mathbf{z}$ and $\mathbf{x}$ is given by

$$\tilde{p}_{Z|X,U}(\mathbf{z} \mid \mathbf{x}, \mathbf{u}) \coloneqq \mathcal{N}(\mathbf{z} \mid F^{-1}(\mathbf{x}; \mathbf{u}), \sigma^2 \mathbf{J}(\mathbf{x}, \mathbf{u})\mathbf{J}(\mathbf{x}, \mathbf{u})^\top) \tag{5}$$

$$\tilde{p}_{X|Z,U}(\mathbf{x} \mid \mathbf{z}, \mathbf{u}) \coloneqq \mathcal{N}(\mathbf{x} \mid F(\mathbf{z}; \mathbf{u}), \sigma^2 \mathbf{I}) \tag{6}$$

where, as per Ho et al. (2019), $\sigma > 0$ is a small scalar parameter and $\bar{\mathbf{x}}$ are the centres of hypercube bins of volume δ_x for the discretization of continuous data $\mathbf{x}$. Furthermore, $\mathbf{J}(\mathbf{x}, \mathbf{u})$ denotes the Jacobian of F^{-1}, with respect to its first argument (keeping the second fixed), evaluated at $(\mathbf{x}, \mathbf{u})$.

3 METHOD

We first consider using just a single layer CIF as the density model for bits back coding, and then extend it to the hierarchical CIF case in the second part of this section. Note that throughout this section the notation $\mathbf{u}$ is intended as per CIF model notation of Cornish et al. (2020), rather than the dequantization notation of Ho et al. (2019). We assume we discretize the continuous space $\mathcal{X}$ into bins of volume δ_x, and we denote the centres of the bins the data was quantized to as $\bar{\mathbf{x}}$. Similarly, $\mathcal{U}$ and $\mathcal{Z}$ are also discretized into bins of volume δ_u and δ_z, respectively, and discretized values are denoted as $\bar{\mathbf{u}}$ and $\bar{\mathbf{z}}$. NB: As is standard, each distribution that we are decoding/encoding with respect to is defined over continuous data; these can be approximately converted into discrete distributions by taking the previous continuous density value at the respective bin centre and multiplying by the discretization volume, e.g. $Q(\bar{\mathbf{u}}|\bar{\mathbf{x}}) = q(\bar{\mathbf{u}}|\bar{\mathbf{x}})\delta_u$.

The CIF-based lossless coding scheme requires one additional constraint on the CIF architecture: the neural networks parameterizing the distribution $p_{U|Z}(\mathbf{u}|\mathbf{z})$ must be constrained to be Lipschitz continuous. The reasons for that are given in Appendix A.2.

3.1 SINGLE LAYER

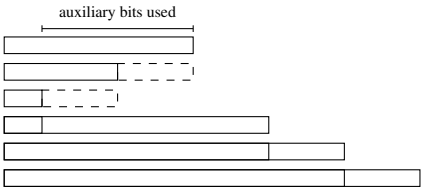
The encoding procedure is outlined in Algorithm 1 below, while the decoding procedure is presented in Algorithm 3 in Appendix A.1.

Algorithm 1 Encoding – Single Layer

Input: data $\bar{\mathbf{x}}$, auxiliary random bits on the ANS message stack m, flow F, CIF distributions $q_{U|X}$, $p_{U|Z}$, discretization volumes δ_x, δ_z, δ_u, noise level σ

Output: updated ANS message stack m

Current state of the ANS message stack m



Initialize m with auxiliary bits
1: Decode $\bar{\mathbf{u}} \sim q_{U|X}(\cdot \mid \bar{\mathbf{x}})\delta_u$
2: Decode $\bar{\mathbf{z}} \sim \tilde{p}_{Z|X,U}(\cdot \mid \bar{\mathbf{x}}, \bar{\mathbf{u}})\delta_z$
3: Encode $\bar{\mathbf{x}}$ using $\tilde{p}_{X|Z,U}(\cdot \mid \bar{\mathbf{z}}, \bar{\mathbf{u}})\delta_x$
4: Encode $\bar{\mathbf{u}}$ using $p_{U|Z}(\cdot \mid \bar{\mathbf{z}})\delta_u$
5: Encode $\bar{\mathbf{z}}$ using $p_Z(\cdot)\delta_z$

The resulting expected asymptotic codelength of such a scheme, i.e. ignoring the impact of the initial auxiliary random bits used, is approximately equal to the sum of the CIF negative ELBO, a constant depending on the discretization precision, and second order term in σ stemming from the use of Local Bits-Back coding (derivation in the Appendix A.2):

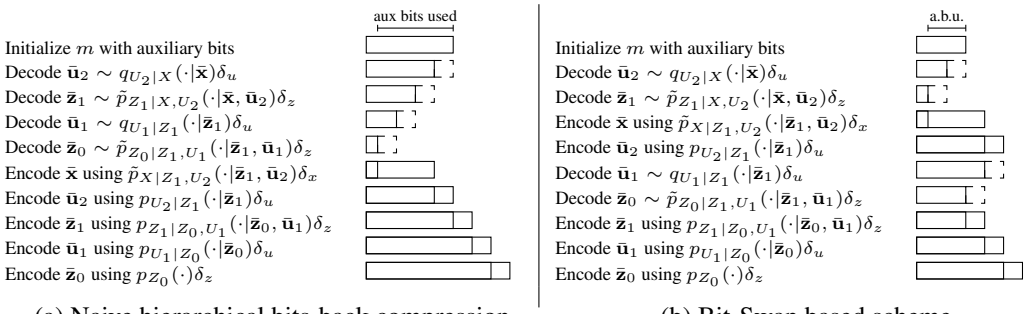
$$\mathbb{E}_{\mathbf{u} \sim q_{U|X}(\cdot|\mathbf{x})} \left[-\log \frac{p_{X,U}(\mathbf{x}, \mathbf{u})}{q_{U|X}(\mathbf{u}|\mathbf{x})} \right] - \log \delta_x + O(\sigma^2). \tag{7}$$

The expected number of initial auxiliary bits required for this scheme to operate is

$$\mathbb{E}_{\bar{\mathbf{x}} \sim p_D(\cdot)} \left[\mathbb{E}_{\bar{\mathbf{u}} \sim q_{U|X}(\cdot|\bar{\mathbf{x}})\delta_u} \Big[\underbrace{-\log q_{U|X}(\bar{\mathbf{u}}|\bar{\mathbf{x}})\delta_u}_{\text{decoding } \bar{\mathbf{u}}} + \mathbb{E}_{\bar{\mathbf{z}} \sim \tilde{p}_{Z|X,U}(\cdot|\bar{\mathbf{x}},\bar{\mathbf{u}})\delta_z} \underbrace{\big[-\log \tilde{p}_{Z|X,U}(\bar{\mathbf{z}}|\bar{\mathbf{x}}, \bar{\mathbf{u}})\delta_z \big]}_{\text{decoding } \bar{\mathbf{z}}} \Big] \right],$$

where p_D is the true marginal data distribution over $\mathbf{x} \in \mathcal{X}$.

Figure 1: Naive hierarchical bits-back scheme [left] vs Bit-Swap based scheme [right] for a hierarchical CIF model with $L = 2$ layers. Notice that the naive scheme uses a larger number of auxiliary random bits as compared to for Bit-Swap based scheme. "a.b.u." stands for "auxiliary bits used".



(a) Naive hierarchical bits-back compression

(b) Bit-Swap based scheme

3.2 Hierarchical Model

Practically, the only way CIF models can be expressive enough to achieve competitive compression rates is via multi-layer stacking. However, a naïve application of bits-back coding to multiple layers of CIF would lead to a linear growth in the required number of auxiliary random bits with respect to the number of flow layers L. We overcome this limitation and reduce the required number of auxiliary bits by applying the Bit-Swap algorithm (Kingma et al., 2019), as the stacking of CIF layers is much like the hierarchical VAEs for which this was originally designed.

Algorithm 2 Hierarchical Encoding

Input: data $\bar{\mathbf{x}}$, auxiliary random bits on the ANS message stack m, flows F_ℓ, CIF distributions $q_{U_\ell|Z_\ell}, p_{U_\ell|Z_{\ell-1}}$, discretization volumes δ_x, δ_z, δ_u, noise level σ
Output: updated ANS message stack m

1: $\bar{\mathbf{z}}_L \leftarrow \bar{\mathbf{x}}$
2: **for** $\ell = L, \ldots, 1$ **do**
3: Decode $\bar{\mathbf{u}}_\ell \sim q_{U_\ell|Z_\ell}(\cdot \mid \bar{\mathbf{z}}_\ell)\delta_u$
4: Decode $\bar{\mathbf{z}}_{\ell-1} \sim \tilde{p}_{Z_{\ell-1}|Z_\ell,U_\ell}(\cdot \mid \bar{\mathbf{z}}_\ell, \bar{\mathbf{u}}_\ell)\delta_z$
5: Encode $\bar{\mathbf{z}}_\ell$ using $\tilde{p}_{Z_\ell|Z_{\ell-1},U_\ell}(\cdot \mid \bar{\mathbf{z}}_\ell, \bar{\mathbf{u}}_\ell)\delta_x$
6: Encode $\bar{\mathbf{u}}_\ell$ using $p_{U_\ell|Z_{\ell-1}}(\cdot \mid \bar{\mathbf{z}}_{\ell-1})\delta_u$ ▷ Left with $\bar{\mathbf{z}}_{\ell-1}$ when loop restarts
7: Encode $\bar{\mathbf{z}}_0$ using $p_{Z_0}(\cdot)\delta_z$

In the hierarchical case, the resulting expected codelength is analogous to the single layer case:

$$\mathbb{E}_{u_{1:L} \sim q_{U_{1:L}|X}(\cdot|x)} \left[-\log \frac{p_{X,U_{1:L}}(x, u_{1:L})}{q_{U_{1:L}|X}(u_{1:L}|x)} \right] - \log \delta_x + O(\sigma^2). \tag{8}$$

4 Discussion

Previous works (Townsend et al., 2019; Ho et al., 2019; Kingma et al., 2019) have empirically shown that the codelengths they achieve are close to the theoretical optimum for the generative models they have investigated. Hence we anticipate our compression scheme to achieve codelengths as predicted by the (negative) log-likelihood performance of CIFs (Cornish et al., 2020). The results reported by Cornish et al. (2020) are obtained using an importance sampling estimator with $K = 100$ samples, which implies that in order to achieve codelengths equal to those we would have to combine our method with the recently-introduced McBits method, which supports the application of a range of Monte Carlo inference algorithms as part of the bits-back coding scheme (Ruan et al., 2021).

In future work we intend to extend and formalize our lossless compression scheme for the entire family of models covered by the SurVAE framework (Nielsen et al., 2020), along with providing a practical implementation of our proposed method.

REFERENCES

Anthony Caterini, Rob Cornish, Dino Sejdinovic, and Arnaud Doucet. Variational inference with continuously-indexed normalizing flows. *arXiv preprint arXiv:2007.05426*, 2020.

Rob Cornish, Anthony L. Caterini, George Deligiannidis, and Arnaud Doucet. Relaxing Bijectivity Constraints with Continuously Indexed Normalising Flows. *ICML*, 2020.

Jarek Duda. Asymmetric numeral systems. *arXiv preprint arXiv:0902.0271*, 2009.

Brendan J Frey and Geoffrey E Hinton. Free energy coding. In *Proceedings of Data Compression Conference-DCC'96*, pp. 73–81. IEEE, 1996.

Jonathan Ho, Evan Lohn, and Pieter Abbeel. Compression with Flows via Local Bits-Back Coding. *NeurIPS*, 2019.

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

Friso H. Kingma, Pieter Abbeel, and Jonathan Ho. Bit-Swap: Recursive Bits-Back Coding for Lossless Compression with Hierarchical Latent Variables. *NeurIPS*, 2019.

Didrik Nielsen, Priyank Jaini, Emiel Hoogeboom, Ole Winther, and Max Welling. SurVAE Flows: Surjections to Bridge the Gap between VAEs and Flows. *ICML*, 2020.

Yangjun Ruan, Karen Ullrich, Daniel Severo, James Townsend, Ashish Khisti, Arnaud Doucet, Alireza Makhzani, and Chris J. Maddison. Improving lossless compression rates via Monte Carlo bits-back coding, 2021.

James Townsend, Thomas Bird, and David Barber. Practical lossless compression with latent variables using bits back coding. *ICLR*, 2019.

A APPENDIX

A.1 DECODING ALGORITHMS

Algorithm 3 Decoding – Single Layer

Input: ANS message stack m, flow F, CIF distributions $q_{U|X}$, $p_{U|Z}$, discretization volumes δ_x, δ_z, δ_u, noise level σ
Output: quantized data $\bar{\mathbf{x}}$, auxiliary random bits on the ANS message stack m

1: Decode $\bar{\mathbf{z}} \sim p_Z(\cdot)\delta_z$
2: Decode $\bar{\mathbf{u}} \sim p_{U|Z}(\cdot \mid \bar{\mathbf{z}})\delta_u$
3: Decode $\bar{\mathbf{x}} \sim \tilde{p}_{X|Z,U}(\cdot \mid \bar{\mathbf{z}}, \bar{\mathbf{u}})\delta_x$
4: Encode $\bar{\mathbf{z}}$ using $\tilde{p}_{Z|X,U}(\cdot \mid \bar{\mathbf{x}}, \bar{\mathbf{u}})\delta_z$
5: Encode $\bar{\mathbf{u}}$ using $q_{U|X}(\cdot \mid \bar{\mathbf{x}})\delta_u$

Algorithm 4 Hierarchical Decoding

Input: ANS message stack m, flows F_ℓ, CIF distributions $q_{U_\ell|Z_\ell}$, $p_{U_\ell|Z_{\ell-1}}$, discretization volumes δ_x, δ_z, δ_u, noise level σ
Output: quantized data $\bar{\mathbf{x}}$, auxiliary random bits on the ANS message stack m

1: Decode $\bar{\mathbf{z}}_0 \sim p_{Z_0}(\cdot)\delta_z$
2: **for** $\ell = 1, \ldots, L$ **do**
3: Decode $\bar{\mathbf{u}}_\ell \sim p_{U_\ell|Z_{\ell-1}}(\cdot \mid \bar{\mathbf{z}}_{\ell-1})\delta_u$
4: Decode $\bar{\mathbf{z}}_\ell \sim \tilde{p}_{Z_\ell|Z_{\ell-1},U_\ell}(\cdot \mid \bar{\mathbf{z}}_\ell, \bar{\mathbf{u}}_\ell)\delta_x$
5: Encode $\bar{\mathbf{z}}_{\ell-1}$ using $\tilde{p}_{Z_{\ell-1}|Z_\ell,U_\ell}(\cdot \mid \bar{\mathbf{z}}_\ell, \bar{\mathbf{u}}_\ell)\delta_z$
6: Encode $\bar{\mathbf{u}}_\ell$ using $q_{U_\ell|Z_\ell}(\cdot \mid \bar{\mathbf{z}}_\ell)\delta_u$
7: $\bar{\mathbf{x}} \leftarrow \bar{\mathbf{z}}_L$

A.2 EXPECTED CODELENGTH

To compute the expected codelength let us sum the expected codelengths of the individual terms coded in Algorithm 1 in their coding order:

$$
\begin{aligned}
&\mathbb{E}_{\bar{\mathbf{u}}\sim q_{U|X}(\cdot|\bar{\mathbf{x}})\delta_u}\left[\mathbb{E}_{\bar{\mathbf{z}}\sim\tilde{p}_{Z|X,U}(\cdot|\bar{\mathbf{x}},\bar{\mathbf{u}})\delta_z}\left[\Sigma\cdot\right]\right] \\
=&\mathbb{E}_{\bar{\mathbf{u}},\bar{\mathbf{z}}}\left[\log q_{U|X}(\bar{\mathbf{u}}|\bar{\mathbf{x}})\delta_u + \log\tilde{p}_{Z|X,U}(\bar{\mathbf{z}}|\bar{\mathbf{x}},\bar{\mathbf{u}})\delta_z - \log\tilde{p}_{X|Z,U}(\bar{\mathbf{x}}|\bar{\mathbf{z}},\bar{\mathbf{u}})\delta_x - \log p_{U|Z}(\bar{\mathbf{u}}|\bar{\mathbf{z}})\delta_u - \log p_Z(\bar{\mathbf{z}})\delta_z\right] \\
=&\mathbb{E}_{\bar{\mathbf{u}},\bar{\mathbf{z}}}\left[\log q_{U|X}(\bar{\mathbf{u}}|\bar{\mathbf{x}}) + \underbrace{\log\tilde{p}_{Z|X,U}(\bar{\mathbf{z}}|\bar{\mathbf{x}},\bar{\mathbf{u}}) - \log\tilde{p}_{X|Z,U}(\bar{\mathbf{x}}|\bar{\mathbf{z}},\bar{\mathbf{u}})}_{\substack{-\log|\det\mathbf{J}(\bar{\mathbf{x}},\bar{\mathbf{u}})|+O(\sigma^2) \\ \text{as per Ho et al. (2019) eq. (7)}}} - \log p_{U|Z}(\bar{\mathbf{u}}|\bar{\mathbf{z}}) - \log p_Z(\bar{\mathbf{z}})\right] - \log\delta_x \\
=&\mathbb{E}_{\bar{\mathbf{u}},\bar{\mathbf{z}}}\left[\log q_{U|X}(\bar{\mathbf{u}}|\bar{\mathbf{x}}) \underbrace{- \log|\det\mathbf{J}(\bar{\mathbf{x}},\bar{\mathbf{u}})| - \log p_{U|Z}(\bar{\mathbf{u}}|\bar{\mathbf{z}}) - \log p_Z(\bar{\mathbf{z}})}\right] - \log\delta_x + O(\sigma^2) \qquad (9)
\end{aligned}
$$

Now, we notice that the second, third, and fourth terms are very close to the CIF joint distribution

$$\log p_{X,U}(\mathbf{x},\mathbf{u}) = \log|\det\mathbf{J}(\mathbf{x},\mathbf{u})| + \log p_{U|Z}(\mathbf{u} \mid F^{-1}(\mathbf{x};\mathbf{u})) + \log p_Z(F^{-1}(\mathbf{x};\mathbf{u})), \qquad (10)$$

but now $\mathbf{z}$ is not exactly equal to $F^{-1}(\mathbf{x};\mathbf{u})$, as the two are related according to the local bits-back model

$$\bar{\mathbf{z}} = F^{-1}(\bar{\mathbf{x}};\bar{\mathbf{u}}) + \sigma\mathbf{J}(\bar{\mathbf{x}},\bar{\mathbf{u}})\epsilon, \qquad (11)$$

for some $\epsilon \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ (with $\bar{\mathbf{z}}$ representing the quantized version of this relationship). However, since $\sigma \ll 1$, we expect equation 11 to approximately hold and thus (like in Ho et al. (2019)) we consider the asymptotics of equation 9 in σ.

Consider first $\log p_Z$, which will typically be a standard normal distribution. If we assume this to be the case, then for any $\xi \in \mathbb{R}^d$, with $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$:

$$\begin{aligned}\mathbb{E}_\epsilon \log p_Z\left(\xi + \sigma\epsilon\right) &= C - \frac{1}{2}\mathbb{E}_\epsilon(\xi+\sigma\epsilon)^T(\xi+\sigma\epsilon) \\ &= C - \frac{1}{2}\mathbb{E}_\epsilon \xi^T\xi - \mathbb{E}_\epsilon \sigma\epsilon^T\xi - \sigma^2\frac{1}{2}\mathbb{E}_\epsilon\epsilon^T\epsilon \\ &= \log p_Z(\xi) + \mathcal{O}(\sigma^2),\end{aligned}$$

where the $\mathcal{O}(\sigma)$ term disappears because ϵ has mean $\mathbf{0}$.

Therefore,

$$\mathbb{E}_{\bar{\mathbf{z}}} \log p_Z(\bar{\mathbf{z}}) \approx \log p_Z(F^{-1}(\bar{\mathbf{x}}; \bar{\mathbf{u}})) + \mathcal{O}(\sigma^2),$$

with the approximation error stemming from quantization.

As for the $\log p_{U|Z}$ term, we have a bit more work to do. We will often assume this is a conditional Gaussian defined as

$$\log p_{U|Z}(\mathbf{u} \mid \mathbf{z}) = \log \mathcal{N}(\mathbf{u} \mid \mu(\mathbf{z}), \Sigma(\mathbf{z})) = -\frac{1}{2}\log\det(2\pi\Sigma(\mathbf{z})) - \frac{1}{2}(\mathbf{u}-\mu(\mathbf{z}))^T\Sigma(\mathbf{z})^{-1}(\mathbf{u}-\mu(\mathbf{z})),$$

where μ and Σ are two different outputs of the same neural network, and Σ is diagonal. We see that $\log p_{U|Z}$ depends quadratically on $\mu(\mathbf{z})$ and $\sqrt{\Sigma(\mathbf{z})}^{-1}$ (as we typically parametrize the standard deviations and not the variances). Thus, if μ and Σ are parametrized such that small changes in their inputs can only induce small changes in their outputs – say if they are **1-Lipschitz functions** – then we can say that (refer to discussion starting from equation 12)

$$\mathbb{E}_{\bar{\mathbf{z}}} \log p_{U|Z}(\bar{\mathbf{u}} \mid \bar{\mathbf{z}}) \approx \log p_{U|Z}(\bar{\mathbf{u}} \mid F^{-1}(\bar{\mathbf{x}}; \bar{\mathbf{u}})) + \mathcal{O}(\sigma^2),$$

where again the error originates from the discretization.

Thus, we can finally rewrite equation 9 as:

$$\begin{aligned}&\mathbb{E}_{\bar{\mathbf{u}}\sim q_{U|X}(\cdot|\bar{\mathbf{x}})\delta_u}\left[\mathbb{E}_{\bar{\mathbf{z}}\sim\tilde{p}_{Z|X,U}(\cdot|\bar{\mathbf{x}},\bar{\mathbf{u}})\delta_z}\left[\Sigma\cdot\right]\right] \\ =&\mathbb{E}_{\bar{\mathbf{u}}\sim q_{U|X}(\cdot|\bar{\mathbf{x}})\delta_u}\left[-\log\frac{p_{X,U}(\bar{\mathbf{x}},\bar{\mathbf{u}})}{q_{U|X}(\bar{\mathbf{u}}|\bar{\mathbf{x}})}\right] - \log\delta_x + O(\sigma^2)\end{aligned}$$

Now, assuming δ_u is small enough that the probability mass function $q_{U|X}(\bar{\mathbf{u}}|\bar{\mathbf{x}})\delta_u$ is approximately equivalent to the probability distribution function $q_{U|X}(\mathbf{u}|\bar{\mathbf{x}})$, then

$$\approx\mathbb{E}_{\mathbf{u}\sim q_{U|X}(\cdot|\bar{\mathbf{x}})}\left[-\log\frac{p_{X,U}(\bar{\mathbf{x}},\mathbf{u})}{q_{U|X}(\mathbf{u}|\bar{\mathbf{x}})}\right] - \log\delta_x + O(\sigma^2).$$

Now, we give further details on the analysis of $p_{U|Z}(\bar{\mathbf{u}} \mid \bar{\mathbf{z}})$: First recall that (up to discretization error)

$$\bar{\mathbf{z}} = F^{-1}(\bar{\mathbf{x}}; \bar{\mathbf{u}}) + \sigma\mathbf{J}(\bar{\mathbf{x}}, \bar{\mathbf{u}})\epsilon$$

for $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Now let's write out $p_{U|Z}$:

$$\log p_{U|Z}(\bar{\mathbf{u}} \mid \bar{\mathbf{z}}) = \log\mathcal{N}(\bar{\mathbf{u}} \mid \mu(\bar{\mathbf{z}}), \bar{\sigma}(\bar{\mathbf{z}})) \tag{12}$$

$$= -\frac{1}{2}\log\det(2\pi\bar{\sigma}(\bar{\mathbf{z}})) - \frac{1}{2}\bar{\sigma}(\bar{\mathbf{z}})^{-1}(\bar{\mathbf{u}}-\mu(\bar{\mathbf{z}}))^T(\bar{\mathbf{u}}-\mu(\bar{\mathbf{z}})) \tag{13}$$

$$= -\frac{1}{2}\log 2\pi \underbrace{-\frac{1}{2}\sum_i \log\sigma(\bar{\mathbf{z}})_i}_{\text{(a)}} \underbrace{-\frac{1}{2}\bar{\sigma}(\bar{\mathbf{z}})^{-1}(\bar{\mathbf{u}}-\mu(\bar{\mathbf{z}}))^T(\bar{\mathbf{u}}-\mu(\bar{\mathbf{z}}))}_{\text{(b)}} \tag{14}$$

$$\mathbb{E}_\epsilon\left[\text{(b)}\right] = \mathbb{E}_\epsilon\left[-\frac{1}{2}\bar{\sigma}(\bar{\mathbf{z}})^{-1}(\bar{\mathbf{u}}-\mu(\bar{\mathbf{z}}))^T(\bar{\mathbf{u}}-\mu(\bar{\mathbf{z}}))\right]. \tag{15}$$

Taking Taylor expansion of $\bar{\sigma}(\bar{\mathbf{z}})^{-1}$ and $\mu(\bar{\mathbf{z}})$ around $\mathbf{y} = F^{-1}(\bar{\mathbf{x}}; \bar{\mathbf{u}})$

$$= \mathbb{E}_\epsilon \Bigg[-\frac{1}{2}\left(\bar{\sigma}(\mathbf{y})^{-1} - \bar{\sigma}(\mathbf{y})^{-2}(\nabla_{\mathbf{y}}\bar{\sigma}(\mathbf{y}))^T\sigma\epsilon + O(\sigma^2)\right)$$

$$(\mathbf{u} - (\mu(\mathbf{y}) + (\nabla_{\mathbf{y}}\mu(\mathbf{y}))^T\sigma\epsilon + O(\sigma^2)))^T(\mathbf{u} - (\mu(\mathbf{y}) + (\nabla_{\mathbf{y}}\mu(\mathbf{y}))^T\sigma\epsilon + O(\sigma^2)))\Bigg]$$

Since $\mathbb{E}_\epsilon[\epsilon] = 0$

$$= \mathbb{E}_\epsilon\left[-\frac{1}{2}\bar{\sigma}(\mathbf{y})^{-1}(\mathbf{u} - \mu(\mathbf{y}))^T(\mathbf{u} - \mu(\mathbf{y}))\right] + O(\sigma^2)$$

Now, term (a)

$$\mathbb{E}_\epsilon\left[\text{(a)}\right] = \mathbb{E}_\epsilon\left[-\frac{1}{2}\sum_i \log\bar{\sigma}(\bar{\mathbf{z}})_i\right] \tag{16}$$

Taking Taylor expansion of $\log\bar{\sigma}(\bar{\mathbf{z}})^{-1}$ around $\mathbf{y} = F^{-1}(\bar{\mathbf{x}}; \bar{\mathbf{u}})$

$$= \mathbb{E}_\epsilon\left[-\frac{1}{2}\sum_i \left(\log\bar{\sigma}(\mathbf{y})_i + (\bar{\sigma}(\mathbf{y})_i)^{-1}(\nabla_{\mathbf{y}}\bar{\sigma}(\mathbf{y}))^T\sigma\epsilon + O(\sigma^2)\right)\right] \tag{17}$$

Again, by the fact that $\mathbb{E}_\epsilon[\epsilon] = 0$

$$= -\frac{1}{2}\sum_i \log\bar{\sigma}(\mathbf{y})_i \tag{18}$$