Conditional BRUNO: A Deep Recurrent Process for Exchangeable Labelled Data

Iryna Korshunova[1] **Yarin Gal**[2] **Joni Dambre**[1*] **Arthur Gretton**[3*]
[1]Ghent University [2]University of Oxford [3]Gatsby Unit, UCL
iryna.korshunova@ugent.be

1 Introduction

Exchangeability is often an implicit assumption underlying many machine learning algorithms. It implies that any re-ordering of a finite sequence of observations is equally likely. Thus, it allows to reason about the future observations based on the behaviour of the previous ones. Some problems can be explicitly formulated in terms of modelling exchangeable sequences. For instance, few-shot concept learning can be seen as learning to complete short exchangeable sequences [13]. BRUNO [12] follows this approach by modelling autoregressive distributions $p(x_n|x_{1:n-1})$ of an exchangeable process. In this work, we extend the idea of BRUNO to the conditional case, where we wish to model $p(x_n|h_n, x_{1:n-1}, h_{1:n-1})$ with h_i's being labels or tags associated with images x_i's.

Formally, a stochastic process $x_1, x_2, x_3, \dots$ is said to be exchangeable if for all n and all permutations π

$$p(x_1, \dots, x_n) = p\left(x_{\pi(1)}, \dots, x_{\pi(n)}\right), \tag{1}$$

i. e. the joint probability remains the same under any permutation of the sequence.

The concept of exchangeability is intimately related to Bayesian statistics via de Finetti's theorem, which states that every exchangeable process is a mixture of i. i. d. processes:

$$p(x_1, \dots, x_n) = \int p(\theta) \prod_{i=1}^{n} p(x_i|\theta) d\theta, \tag{2}$$

where θ is a parameter (finite or infinite-dimensional) conditioned on which, x_i's are i. i. d. [1].

This theorem gives two ways of defining models of exchangeable sequences. One is via explicit Bayesian modelling: define a prior $p(\theta)$, a likelihood $p(x_i|\theta)$ and calculate the posterior in Eq. 2 directly. Here, the difficulty is the intractability of the posterior as it requires an integration over the parameter θ. A common solution is to use a variational approximation. The neural statistician [5] implements this approach by building upon a variational autoencoder model (VAE) [11].

The second way is to *construct* an exchangeable process while modelling its autoregressive distributions $p(x_n|x_{1:n-1})$ directly without referring to the underlying Bayesian model. BRUNO [12] proposes a design for doing so. It consists of two components: **(a)** a bijective mapping that transforms an intricate input space $\mathcal{X}$ into a Gaussian latent space $\mathcal{Z}$, and **(b)** exchangeable Gaussian processes ($\mathcal{GP}$s) defined in the latent space $\mathcal{Z}$. Using deep neural networks to implement the bijection allows to model complex and high-dimensional inputs, while the whole construction of BRUNO guarantees that the process in $\mathcal{X}$ is exchangeable.

A natural extension when building exchangeable models would be to have a conditional process with two associated sequences: $x_1, x_2, x_3, \dots$ and $h_1, h_2, h_3, \dots$. For instance, $\boldsymbol{x}_i$ could be an image and $\boldsymbol{h}_i$ a vector of descriptive labels or tags. By analogy with Eq. 1, the exchangeability property becomes:

$$p(x_1, \dots, x_n|h_1, \dots, h_n) = p\left(x_{\pi(1)}, \dots, x_{\pi(n)}|h_{\pi(1)}, \dots, h_{\pi(n)}\right). \tag{3}$$

*Equal contribution

Third workshop on Bayesian Deep Learning (NeurIPS 2018), Montréal, Canada.

To have a valid stochastic process, we also need a consistency property as imposed by the Kolmogorov extension theorem [14]:

$$p(x_{1:m}|h_{1:m}) = \int p\left(x_{1:n}|h_{1:n}\right) dx_{m+1:n} \quad \text{for } 1 \leq m < n. \tag{4}$$

To our best knowledge, Bayesian theory does not have an established proof of de Finetti's theorem for conditional probabilities. Namely, that the two conditions above ensure that one can represent the process as a mixture of conditionally i.i.d. models as given in Eq. 5. For the processes where x_i and h_i take values from a finite set, this theorem is proven in the field of quantum physics [2]. However, it is yet unclear how to extend their results to continuum variables.

$$p(x_{1:n}|h_{1:n}) = \int p(\theta) \prod_{i=1}^{n} p(x_i|h_i, \theta) d\theta. \tag{5}$$

Relying on the conditional version of de Finetti's theorem, neural processes [7] take an approach that is similar to the neural statistician's. Namely, by extending the VAE model to handle collections of (x_i, h_i) input pairs and dealing with a lower bound on $p(x_n|h_n, x_{1:n-1}, h_{1:n-1})$. Versa [8] also follows the idea of approximating the posterior predictive distribution, though it uses a training procedure that differs from the standard variational inference. Both models achieve permutation invariance of $p(\theta|x_{1:n}, h_{1:n})$ with respect to the conditioning inputs by using instance-pooling operations, e.g. the mean over representations of (x_i, h_i) pairs.

Another option is to use the idea of BRUNO and construct a process that satisfies Eq. 3 and 4. In the next section, we show this can be done by modifying the architecture of BRUNO. Namely, by conditioning the bijective transformation $f : \mathcal{X} \mapsto \mathcal{Z}$ on the tags, such that $\boldsymbol{z}_i = f_{\boldsymbol{h}_i}(\boldsymbol{x}_i)$. A schematic of our model is given in Fig. 1. While the change to the model is incremental, the fact that BRUNO can be so easily extended to the conditional case is interesting by itself. Moreover, it adds an important class into the collection of meta-learning problems that BRUNO is capable of solving.

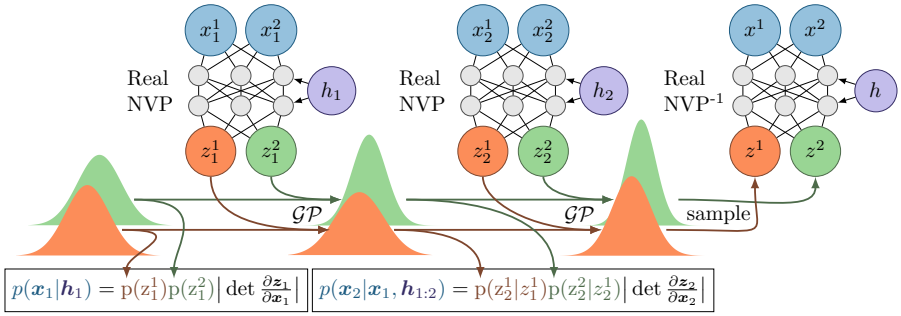

Figure 1: A schematic of the conditional BRUNO model.

2 Conditional BRUNO

The bijective transformation part of BRUNO is carried out by a Real NVP [4] – a deep, stably invertible and learnable neural network architecture that transforms some density $p(\boldsymbol{x})$ into a desired probability distribution $p(\boldsymbol{z})$. It is implemented as a sequence of alternating coupling layers, with every layer transforming a half of its input dimensions while copying the other half directly to the output. In case of modelling a conditional distribution $p(\boldsymbol{x}|\boldsymbol{h})$, we can make the transformation dependendent on $\boldsymbol{h}$, so the outputs of the coupling layer become:

$$\begin{cases} \boldsymbol{x}_{\text{out}}^{1:d} = \boldsymbol{x}_{\text{in}}^{1:d} \\ \boldsymbol{x}_{\text{out}}^{d+1:D} = \boldsymbol{x}_{\text{in}}^{d+1:D} \odot \exp(s(\boldsymbol{x}_{\text{in}}^{1:d}, \boldsymbol{h})) + t(\boldsymbol{x}_{\text{in}}^{1:d}, \boldsymbol{h}), \end{cases} \tag{6}$$

where $\odot$ is an elementwise product, and functions s (scale) and t (translation) are usually deep neural networks. We achieved the conditioning on $\boldsymbol{h}$ by adding a bias computed from the features of $\boldsymbol{h}$ to every layer inside the s and t networks.

As in case of the original Real NVP model, we can assume a fixed distribution for the latents $\boldsymbol{z}$ due to the fact that dependence of $\boldsymbol{x}$ on $\boldsymbol{h}$ is introduced via the Jacobian of the transformation. The

latter is used in the change of variables formula: $p(\boldsymbol{x}|\boldsymbol{h}) = p(\boldsymbol{z})\left|\det \boldsymbol{J}_{\boldsymbol{h}}\right|$. For the same reasons, the conditional BRUNO can use the identical assumptions as its unconditional counterpart:

A1: dimensions $\{z^d\}_{d=1,\dots,D}$ are independent, so $p(\boldsymbol{z}) = \prod_{d=1}^{D} p(z^d)$

A2: for every dimension d, we assume that $(z_1^d, \dots z_n^d) \sim MVN_n(\mathbf{0}, \boldsymbol{\Sigma}^d)$, where $\boldsymbol{\Sigma}^d$ is a $n \times n$ covariance matrix with $\boldsymbol{\Sigma}^d_{ii} = v^d$ and $\boldsymbol{\Sigma}^d_{ij,i\neq j} = \rho^d$, $0 \leq \rho^d < v^d$.

3 Experiments

We consider a task of few-shot image reconstruction, where the model is required to infer how an object looks from various angles based on a small set of observed views [8]. In our model, this problem can be framed as generating samples from a predictive conditional distribution $p(\boldsymbol{x}_n|\boldsymbol{h}_n, \boldsymbol{x}_{1:n-1}, \boldsymbol{h}_{1:n-1})$, where $\boldsymbol{h}_n$ is a desired angle and $\boldsymbol{x}_{1:n-1}$ is a set of observed views associated with angles $\boldsymbol{h}_{1:n-1}$. We use airplanes and chairs from the ShapeNetCore v2 [3] dataset as constructed by Gordon et al. [8], and train the conditional BRUNO on one-shot tasks. We give a single random view $\boldsymbol{x}_1$ and its angle $\boldsymbol{h}_1$ and the goal is to predict N views of the same object under angles $\boldsymbol{h}_1, \dots, \boldsymbol{h}_N$. Namely, the objective in a single task is to maximise $\mathcal{L} = \sum_{n=1}^{N} \log p(\boldsymbol{x}_n|\boldsymbol{h}_n, \boldsymbol{x}_1, \boldsymbol{h}_1)$ with respect to the Real NVP parameters and variance-covariance parameters of the latent $\mathcal{GP}$s. Note, that unlike in Gordon et al. [8], we train a single model on a combined set of chairs and airplanes.

In Figure 2, we show samples from a conditional BRUNO when the model was given a single viewpoint from an object not seen during training. Note that conditioned on a chair, it never samples an airplane or vice versa. Moreover, from Figure 3 one can see better how the model's uncertainty about the object is reflected in the samples. Specifically, in the case when a single shot it is conditioned upon gives insufficient information about the object, conditional BRUNO generates diverse objects which are quite consistent with the given shot. Also, our samples always have a correct orientation and their quality is superior to the one from Versa [8] as our samples are sharp.

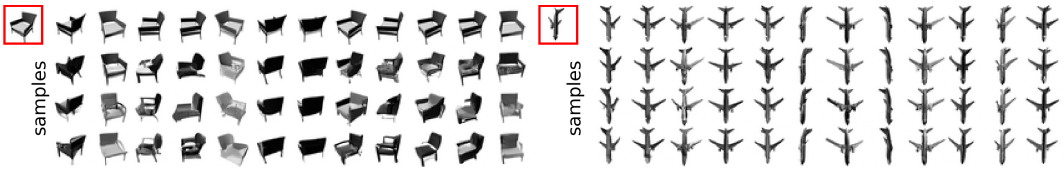

Figure 2: One-shot BRUNO samples for the unseen test objects. Here, the model is given a single view $(\boldsymbol{x}_1, \boldsymbol{h}_1)$ of a chair or an airplane. This input shot is marked in red. On the top row is the ground truth, whereas the three rows underneath contain samples from the model conditioned on the input shot and a desired angle.

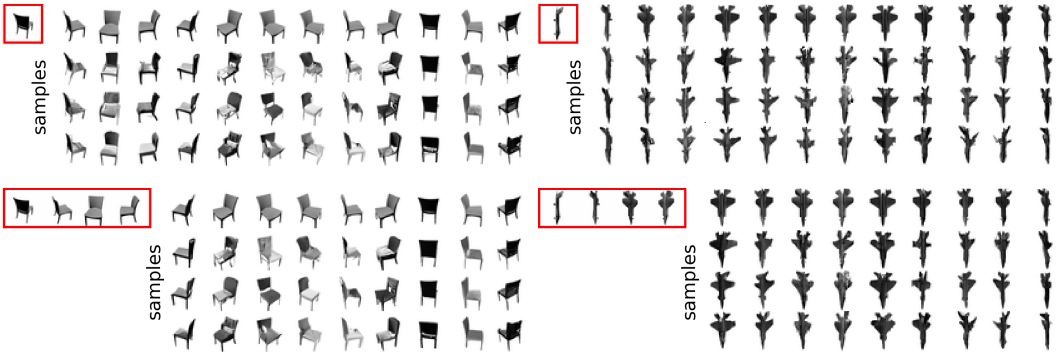

Figure 3: BRUNO samples for the test objects conditioned on 1 or 4 shots and an angle. From a single shot, it is difficult to infer the exact appearance, e.g. the front of the chair or the aiplane type. Thus, the model increases the variability of the samples along these dimensions. With more input shots, samples become more consistent. This is most noticeable for the shape of the airplane's wings.

A more complex version of the few-shot image reconstruction, is learning to render scenes as done by Generative Query Networks [6], which are similar to the aforementioned neural processes [7] in their

core idea. We suppose that conditional BRUNO, when scaled to complex datasets, might become a viable alternative to the VAE-based types of models.

4 Conclusion

We showed that BRUNO [12] can be easily extended to the conditional case while maintaining its appealing properties such as provable exchangeability, exact posterior computation, fast sampling and recurrent formulation of the Bayesian updates. These features make BRUNO a simple yet an effective and flexible model for meta-learning.

BRUNO combines the data-efficiency of $\mathcal{GP}$s with a power of deep learning to model complex data types, and while the former is unlikely to be improved, we expect BRUNO to greatly benefit from the recent advances in building normalising flows, which is currently an active area of research [9, 10].

Acknowledgements

We would like to thank Jonas Degrave for insightful discussions and Figure 1, Thu Nguyen Phuoc for an overview of ShapeNet papers, Jonathan Gordon and John Bronskill for answering questions about Versa and their code on github, and Ferenc Huszár for the whole idea of exchangeability via RNNs.

References

[1] Aldous, D., Hennequin, P., Ibragimov, I., and Jacod, J. (1985). *Ecole d'Ete de Probabilites de Saint-Flour XIII, 1983*. Lecture Notes in Mathematics. Springer Berlin Heidelberg.

[2] Barrett, J. and Leifer, M. (2009). The de Finetti theorem for test spaces. *New Journal of Physics*, 11(3).

[3] Chang, A. X., Funkhouser, T. A., Guibas, L. J., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., and Yu, F. (2015). ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*.

[4] Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017). Density estimation using Real NVP. In *Proceedings of the 5th International Conference on Learning Representations*.

[5] Edwards, H. and Storkey, A. (2017). Towards a neural statistician. In *Proceedings of the 5th International Conference on Learning Representations*.

[6] Eslami, S. M. A., Jimenez Rezende, D., Besse, F., Viola, F., Morcos, A. S., Garnelo, M., Ruderman, A., Rusu, A. A., Danihelka, I., Gregor, K., Reichert, D. P., Buesing, L., Weber, T., Vinyals, O., Rosenbaum, D., Rabinowitz, N., King, H., Hillier, C., Botvinick, M., Wierstra, D., Kavukcuoglu, K., and Hassabis, D. (2018). Neural scene representation and rendering. *Science*, 360(6394):1204–1210.

[7] Garnelo, M., Schwarz, J., Rosenbaum, D., Viola, F., Rezende, D. J., Eslami, S. M. A., and Teh, Y. W. (2018). Neural processes. *Theoretical Foundations and Applications of Deep Generative Models, ICML workshop*.

[8] Gordon, J., Bronskill, J., Bauer, M., Nowozin, S., and Turner, R. E. (2018). Decision-theoretic meta-learning: Versatile and efficient amortization of few-shot learning. *arXiv preprint arXiv:1805.09921*.

[9] Grathwohl, W., Chen, R. T. Q., Bettencourt, J., Sutskever, I., and Duvenaud, D. (2018). Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*.

[10] Kingma, D. P. and Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*.

[11] Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations*.

[12] Korshunova, I., Degrave, J., Huszár, F., Gal, Y., Gretton, A., and Dambre, J. (2018). BRUNO: A deep recurrent model for exchangeable data. In *Proceedings of the 32th International Conference on Neural Information Processing Systems*.

[13] Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science*.

[14] Øksendal, B. (2003). *Stochastic Differential Equations: An Introduction with Applications*. Hochschultext / Universitext. Springer.