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, University College London

Overview

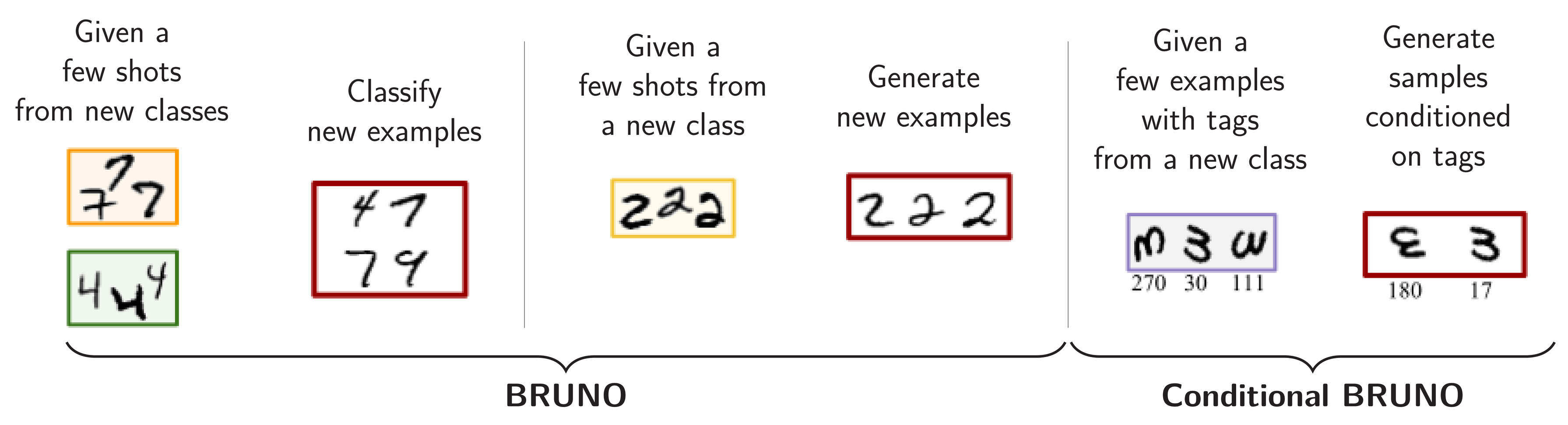
BRUNO combines the expressiveness of deep neural networks with the data-efficiency of $\mathcal{GP}$s to model exchangeable sequences of complex observations.

BRUNO can be extended to the **conditional** case so that it can model sequences of observations $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \dots$ conditionally on a set of labels or tags $\boldsymbol{h}_1, \boldsymbol{h}_2, \boldsymbol{h}_3, \dots$.

Conditional BRUNO enjoys a few properties that are desirable in practice:

- ✓ predictive distribution $p(x_n|h_n, x_{1:n-1}, h_{1:n-1})$ is fast to evaluate and to sample from
- ✓ $p(x_n|h_n, x_{1:n-1}, h_{1:n-1})$ is differentiable with respect to the model parameters
- ✓ can be trained efficiently in an RNN-like fashion

Exchangeability and meta-learning



Exchangeability and Bayesian computations

A stochastic process $x_1, x_2, x_3 \dots$ is exchangeable if for all n and all permutations π:

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

De Finetti's theorem says that every exchangeable process is a mixture of i.i.d. processes:

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

where θ is some parameter conditioned on which the data is i.i.d.

example

$x_1, \dots, x_n \sim \mathcal{N}_n(\mathbf{0}, \Sigma)$ with *compound symmetric* covariance $\Sigma = \begin{bmatrix} v & \rho & \rho & \dots & \rho \\ \rho & v & \rho & \dots & \rho \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \rho & \dots & v \end{bmatrix}$ $0 \leq \rho < v$

$x_1, \dots x_n$ are i.i.d. with $x_i \sim \mathcal{N}(\theta, v - \rho)$ conditioned on $\theta \sim \mathcal{N}(0, \rho)$

De Finetti's theorem in terms of **predictive distributions**:

$$p(x_n|x_{1:n-1}) = \int \underbrace{p(x_n|\theta)}_{\text{likelihood}} \underbrace{p(\theta|x_{1:n})}_{\text{posterior}} d\theta$$

This gives two ways for defining models of exchangeable sequences:

1) *via explicit Bayesian modelling*

2) *via exchangeable processes* $\rightarrow$ BRUNO

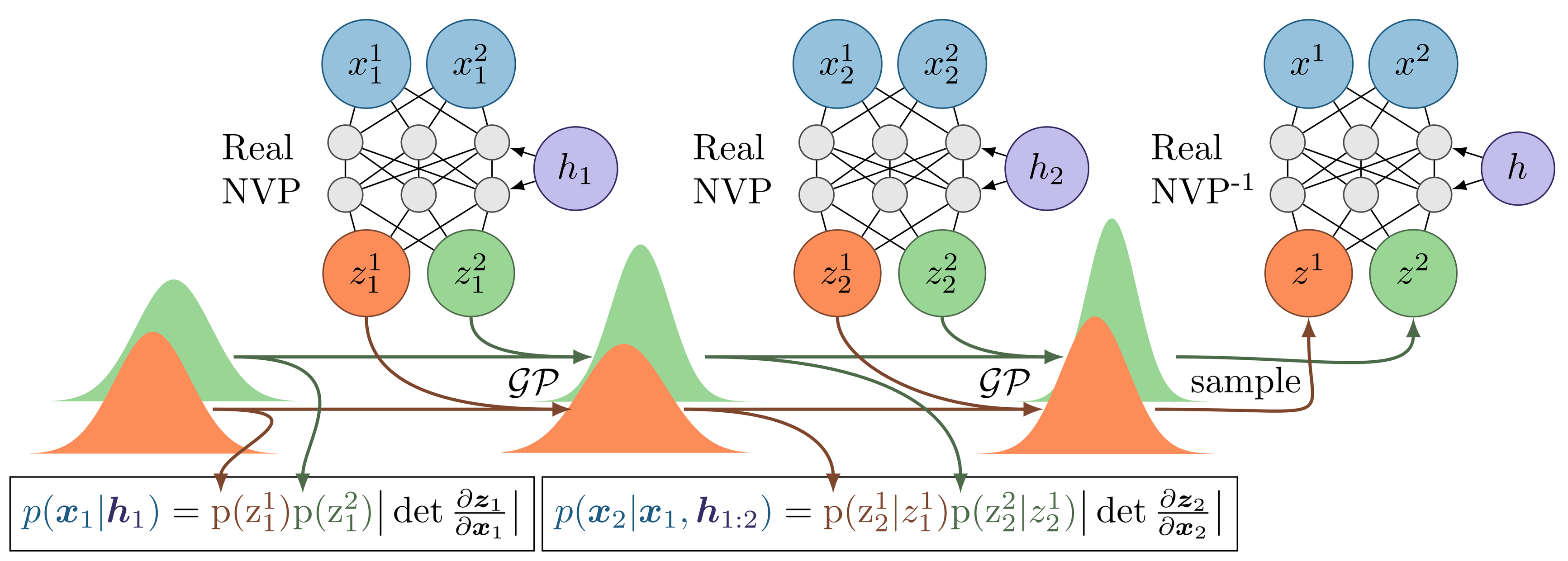
For conditional real-valued processes, where $x_1, x_2, x_3 \dots$ is associated with $h_1, h_2, h_3 \dots$, de Finetti's theorem is not proven.

conjecture

The decomposition of the form $p(x_{1:n}|h_{1:n}) = \int p(\theta) \prod_{i=1}^{n} p(x_i|h_i, \theta) d\theta$ exists if the following conditions hold:

1. $p(x_1, \dots, x_n|h_1, \dots, h_n) = p\left(x_{\pi(1)}, \dots, x_{\pi(n)}|h_{\pi(1)}, \dots, h_{\pi(n)}\right)$
2. $p(x_{1:m}|h_{1:m}) = \int p\left(x_{1:n}|h_{1:n}\right) dx_{m+1:n}$ for $1 \leq m < n$.

Conditional BRUNO



assumptions

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

A2: for each dimension d, we assume that $(z_1^d, \dots z_n^d) \sim MVN_n(\mu^d \mathbf{1}, \Sigma^d)$

- mean $\mu^d \mathbf{1}$ is a $1 \times n$ vector filled with $\mu^d \in \mathbb{R}$
- covariance $n \times n$ matrix Σ^d with $\Sigma_{ii}^d = v^d$ and $\Sigma_{ij,i \neq j}^d = \rho^d$ where $0 \leq \rho^d < v^d$

For a sequence $(\boldsymbol{x}_1, \boldsymbol{h}_1), (\boldsymbol{x}_2, \boldsymbol{h}_2), \dots (\boldsymbol{x}_N, \boldsymbol{h}_N)$ the model is trained to maximise

$$\mathcal{L} = \sum_{n=m+1}^{N} \log p(\boldsymbol{x}_n|\boldsymbol{h}_n, \boldsymbol{x}_{1:m}, \boldsymbol{h}_{1:m})$$

with respect to Real NVP parameters and Σ parameters for every latent dimension.

Real NVP*

$$f: \mathcal{X} \mapsto \mathcal{Z} \text{ with } \mathcal{X} = \mathbb{R}^D \text{ and } \mathcal{Z} = \mathbb{R}^D$$

- f is *bijective*
- forward $\boldsymbol{z} = f(\boldsymbol{x})$ and inverse $\boldsymbol{x} = f^{-1}(\boldsymbol{z})$ mappings are equally expensive
- computing the Jacobian takes $\mathcal{O}(D)$

Coupling layer - the main building block of Real NVP:

$$\begin{cases} \boldsymbol{y}^{1:d} = \boldsymbol{x}^{1:d} \\ \boldsymbol{y}^{d+1:D} = \boldsymbol{x}^{d+1:D} \odot \exp(s(\boldsymbol{x}^{1:d})) + t(\boldsymbol{x}^{1:d}) \end{cases}$$

scales and translates only half of the input dimensions at a time; **s** and **t** are deep neural nets

For a **conditional Real NVP** mapping $\boldsymbol{z} = f_{\boldsymbol{h}}(\boldsymbol{x})$, we can make $\boldsymbol{s}$ and $\boldsymbol{t}$ depend on $\boldsymbol{h}$ by adding a bias computed from the features of $\boldsymbol{h}$ to every layer inside $\boldsymbol{s}$ and $\boldsymbol{t}$.

Given a distribution $p(\boldsymbol{z})$, we can evaluate $p(\boldsymbol{x}|\boldsymbol{h})$ using the *change of variables* formula:

$$p(\boldsymbol{x}|\boldsymbol{h}) = p(\boldsymbol{z}) \left| \det \left(\frac{\partial f_{\boldsymbol{h}}(\boldsymbol{x})}{\partial \boldsymbol{x}} \right) \right|$$

*L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using Real NVP. In ICLR'17

Exchangeable Gaussian processes

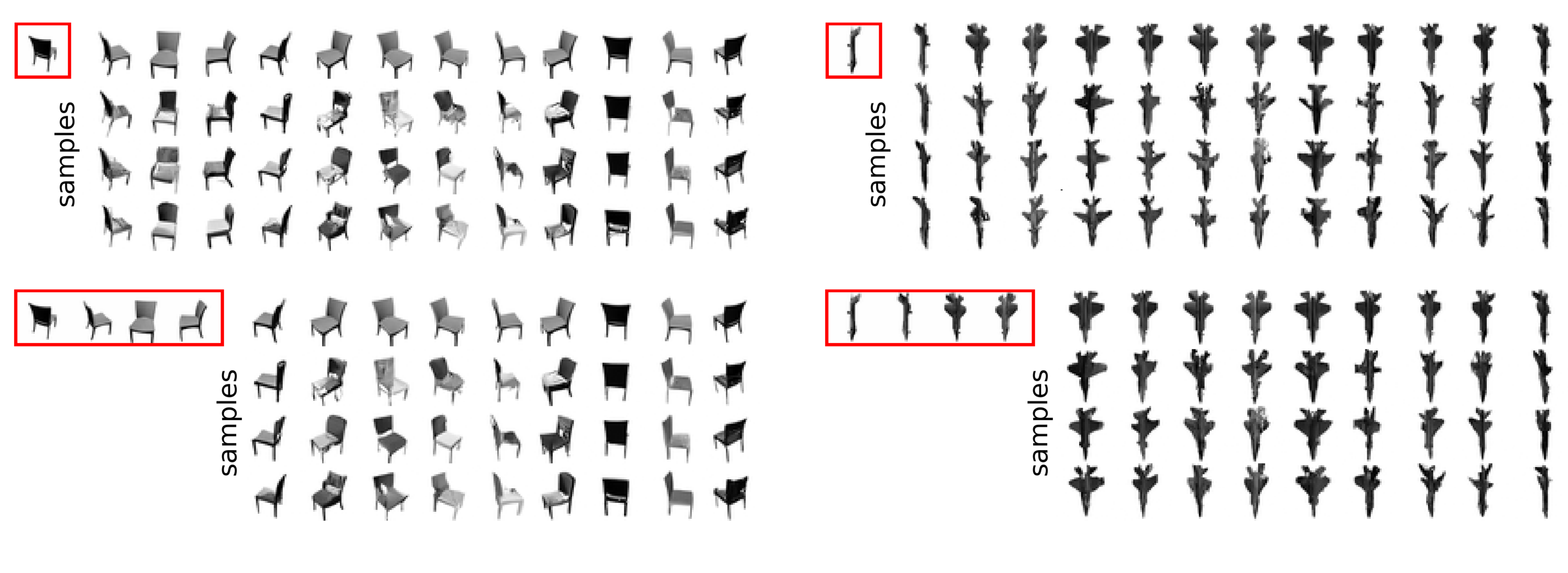
In a $\mathcal{GP}$, where any finite collection $(z_1, \dots z_n) \sim MVN_n(\mu \mathbf{1}, \Sigma)$ with a compound symmetric Σ, recurrent updates for the params of $p(z_{n+1}|z_{1:n}) = \mathcal{N}(\mu_{n+1}, v_{n+1})$ are:

$$\begin{aligned} \mu_{n+1} &= (1 - d_n)\mu_n + d_n z_n \\ v_{n+1} &= (1 - d_n)v_n + d_n(v - \rho) \end{aligned} \quad \text{with} \quad \begin{aligned} d_n &= \rho / v + \rho(n-1) \\ \mu_1 &= \mu, \ v_1 = v \end{aligned}$$

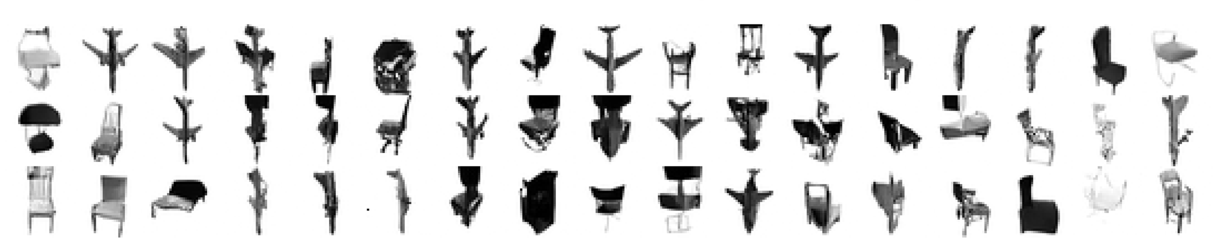
$\mathcal{O}(n)$ runtime and $\mathcal{O}(1)$ memory complexity!

Experiments

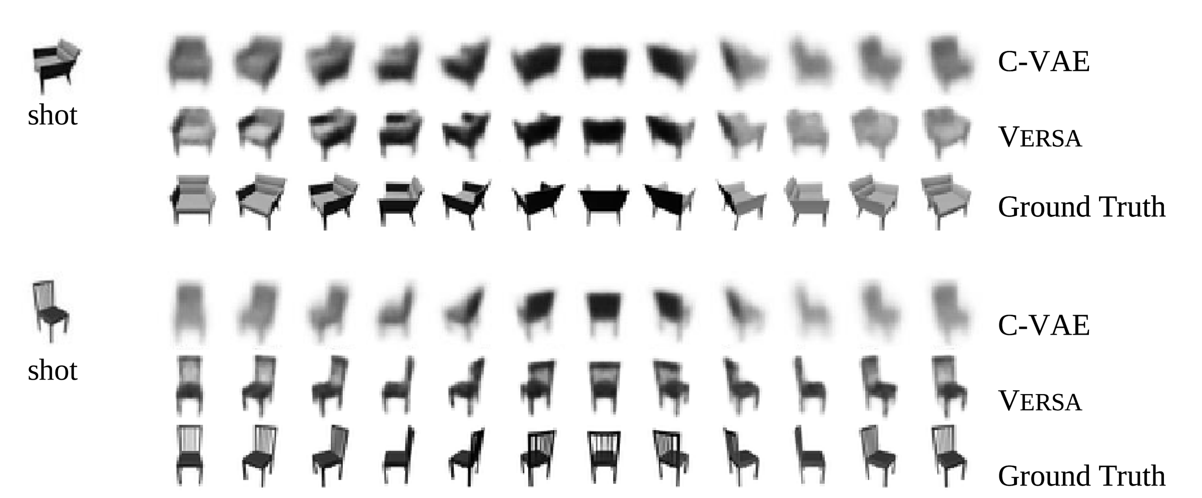
Conditional BRUNO trained on ShapeNet chairs and airplanes



Conditional BRUNO prior samples



VERSA* trained on ShapeNet chairs



*J. Gordon, J. Bronskill, M. Bauer, S. Nowozin, R. E. Turner. Decision-Theoretic Meta-Learning: Versatile and Efficient Amortization of Few-Shot Learning. arXiv:1805.09921

Contact: irene.korshunova@gmail.com Code: github.com/IraKorshunova/bruno