

BRUNO: A Deep Recurrent Model for Exchangeable Data

Iryna Korshunova¹♥ Jonas Degraeve¹* Ferenc Huszár² Yarin Gal³ Arthur Gretton⁴△ Joni Dambre¹△

¹Ghent University ²Twitter ³University of Oxford ⁴Gatsby Unit, University College London

♥△Equal contribution *Now at DeepMind

Overview

BRUNO is a versatile meta-learning model that combines the expressiveness of deep neural networks with the data-efficiency of \mathcal{GP} s to model exchangeable sequences of high-dimensional, complex observations like images.

BRUNO is exchangeable by construction, meaning that its joint distribution $p(x_1, \dots, x_n)$ is permutation-invariant. As a consequence, BRUNO carries out an exact Bayesian inference, albeit implicitly.

BRUNO enjoys some properties that are desirable in practice:

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

Exchangeability and Bayesian computations

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

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

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

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

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

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

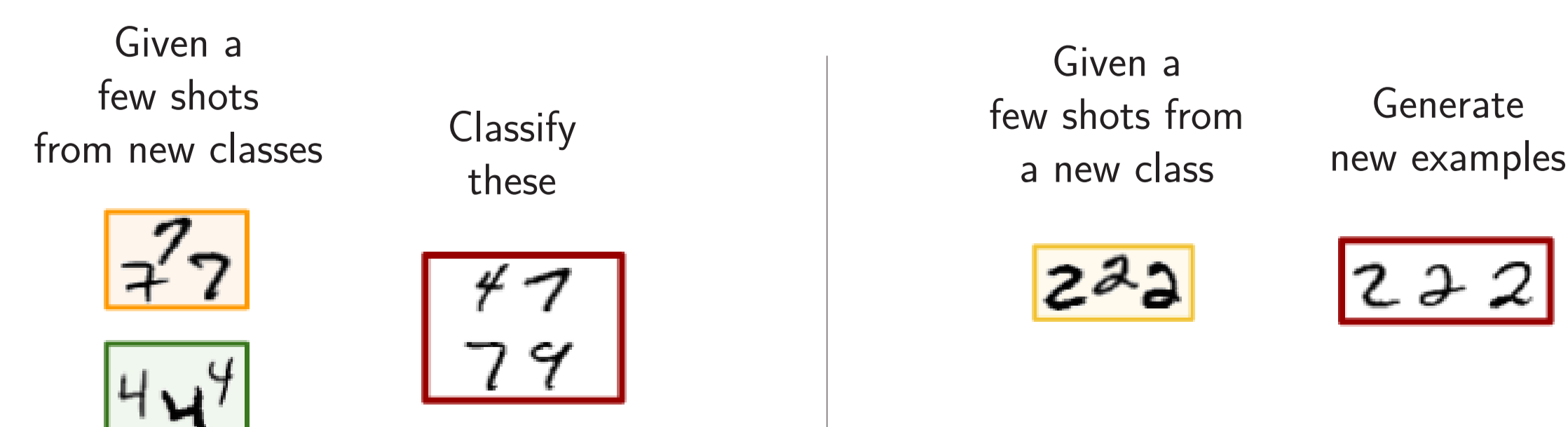
$$p(x_n|x_{1:n-1}) = \int \underbrace{p(x_n|\theta)}_{\text{likelihood}} \underbrace{p(\theta|x_{1:n-1})}_{\text{posterior}} d\theta$$

This gives two ways for defining models of exchangeable sequences:

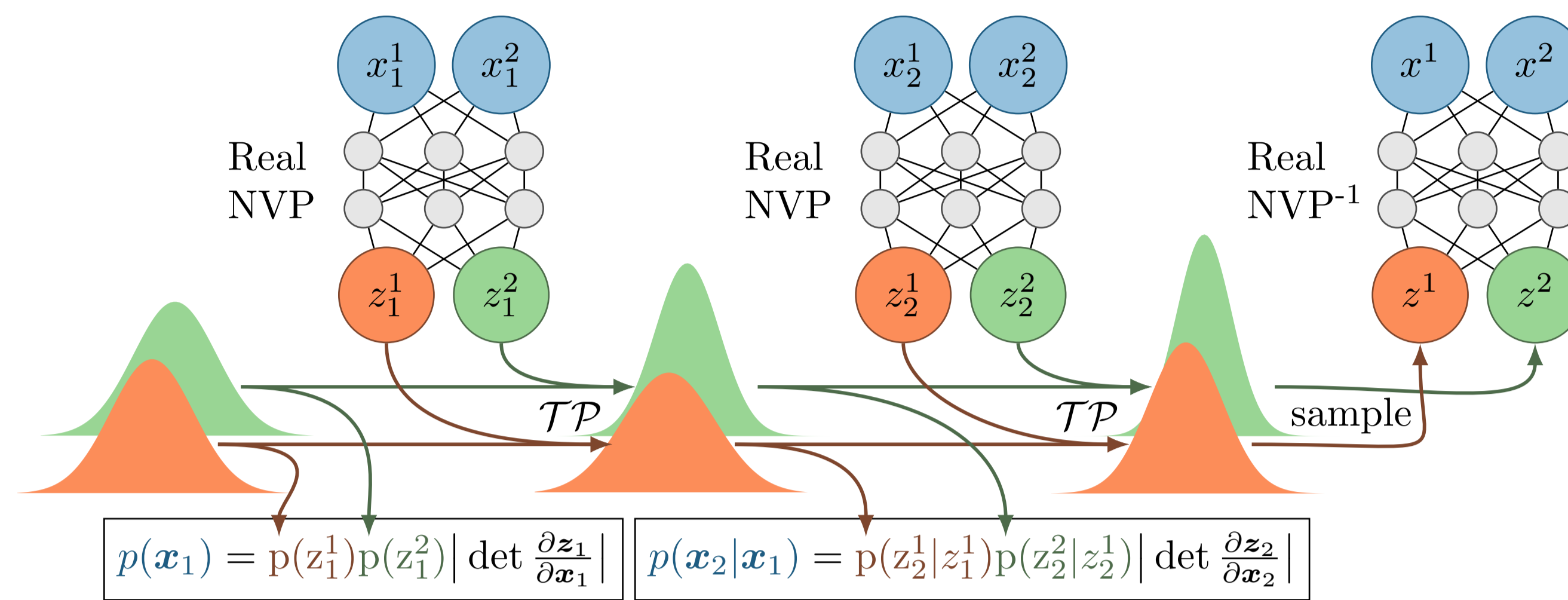
- 1) via explicit Bayesian modelling, e.g. like in the neural statistician [1]
- 2) via exchangeable processes, e.g. BRUNO

Exchangeability and meta-learning

Exchangeability is to meta-learning as convolutions are to vision.



BRUNO: Bayesian Recurrent Neural mOdel



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

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

- degrees of freedom $\nu^d \in \mathbb{R}_+ \setminus [0, 2]$
- mean $\mu^d \mathbf{1}$ is a $1 \times n$ vector filled with $\mu^d \in \mathbb{R}$
- covariance $n \times n$ matrix \mathbf{K}^d with $\mathbf{K}_{ii}^d = \nu^d$ and $\mathbf{K}_{ij, i \neq j}^d = \rho^d$ where $0 \leq \rho^d < \nu^d$

Real NVP [2]

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

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

Real NVP assumes a simple distribution for $p(\mathbf{z})$, so we can use the *change of variables* formula to evaluate $p(\mathbf{x})$:

$$p(\mathbf{x}) = p(\mathbf{z}) \left| \det \left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right) \right|$$

Coupling layer - the main building block of Real NVP:

$$\begin{cases} \mathbf{y}^{1:d} = \mathbf{x}^{1:d} \\ \mathbf{y}^{d+1:D} = \mathbf{x}^{d+1:D} \odot \exp(s(\mathbf{x}^{1:d})) + \mathbf{t}(\mathbf{x}^{1:d}) \end{cases} \quad \begin{array}{l} \text{scales and translates} \\ \text{only half of the input} \\ \text{dimensions at a time} \end{array}$$

Exchangeable Gaussian and Student-t 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 $p(z_{n+1}|z_{1:n}) = \mathcal{N}(\mu_{n+1}, \nu_{n+1})$ are:

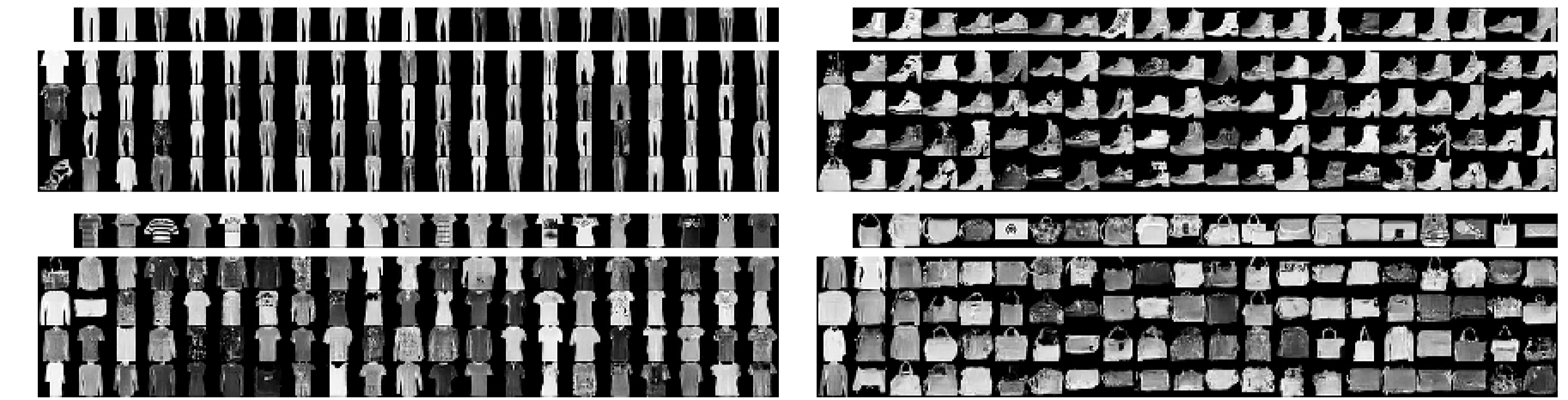
$$\begin{aligned} \mu_{n+1} &= (1 - d_n)\mu_n + d_n z_n & \text{with} & & d_n &= \rho / (\nu + \rho(n-1)) \\ \nu_{n+1} &= (1 - d_n)\nu_n + d_n(\nu - \rho) & & & \mu_1 &= \mu, \nu_1 = \nu \end{aligned}$$

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

\mathcal{TP} [3] is a generalisation of a \mathcal{GP} that can be derived by placing an inverse Wishart process prior on the covariance of a \mathcal{GP} . \mathcal{TP} s tend to be more robust when training BRUNO at negligible additional costs compared to \mathcal{GP} s.

Experiments

Fashion MNIST generation



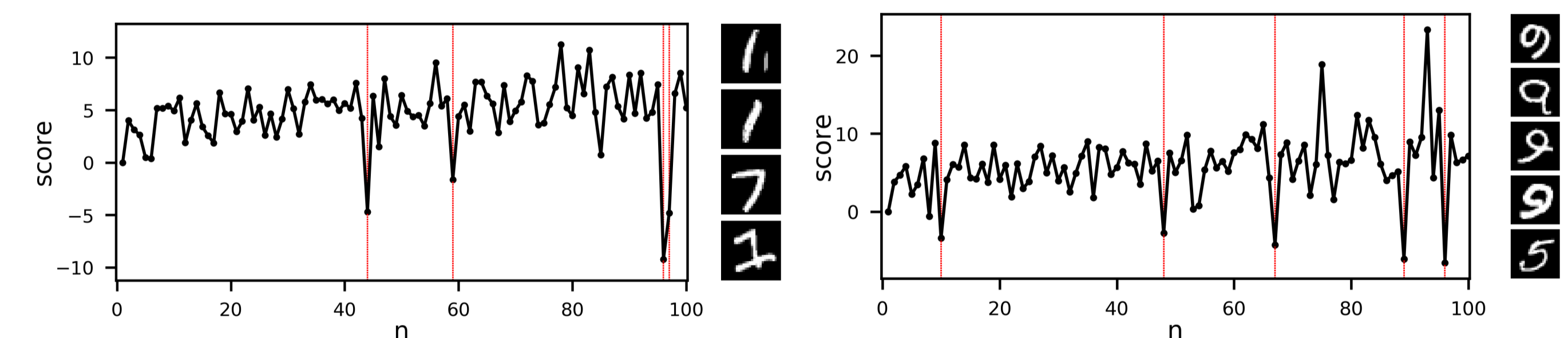
OMNIGLOT few-shot generation



OMNIGLOT few-shot classification

Model	5-way		20-way	
	1-shot	5-shot	1-shot	5-shot
Baseline Classifier [4]	80.0	95.0	69.5	89.1
Matching Nets [4]	98.1	98.9	93.8	98.5
BRUNO	86.3	95.6	69.2	87.7
BRUNO (discriminative fine-tuning)	97.1	99.4	91.3	97.8

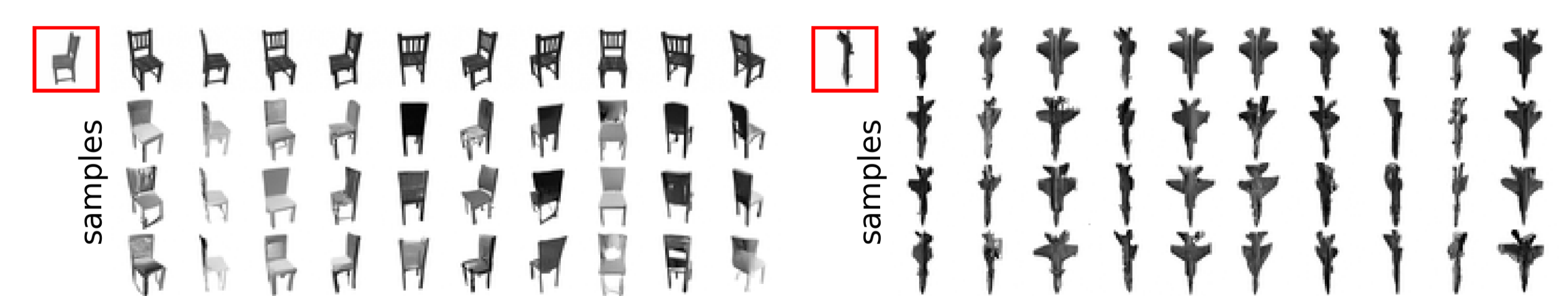
Online set anomaly detection



Extra: conditional BRUNO

BRUNO can be easily extended to handle exchangeable sequences where every x_i is associated with a vector of labels or tags h_i . Here, we model $p(x_n|h_n, x_{1:n-1}, h_{1:n-1})$.

ShapeNet 1-shot BRUNO samples conditioned on the camera angle



Bibliography

- [1] H. Edwards and A. Storkey. Towards a neural statistician. In *ICLR'17*.
- [2] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using Real NVP. In *ICLR'17*.
- [3] A. Shah, A. G. Wilson, and Z. Ghahramani. Student-t processes as alternatives to gaussian processes. In *AISTATS'14*.
- [4] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In *NIPS'16*.