PAVI: Plate-Amortized Variational Inference

Louis Rouillard, Alexandre Le Bris, Thomas Moreau, Demian Wassermann

PAVI:
Plate-Amortized Variational Inference

Louis Rouillard
Université Paris-Saclay, Inria, CEA
Palaiseau, 91120, France
louis.rouillard-odera@inria.fr

Thomas Moreau
Université Paris-Saclay, Inria, CEA
Palaiseau, 91120, France
thomas.moreau@inria.fr

Demian Wassermann
Université Paris-Saclay, Inria, CEA
Palaiseau, 91120, France
demian.wassermann@inria.fr

Abstract

Given some observed data and a probabilistic generative model, Bayesian inference aims at obtaining the distribution of a model's latent parameters that could have yielded the data. This task is challenging for large population studies where thousands of measurements are performed over a cohort of hundreds of subjects, resulting in a massive latent parameter space. This large cardinality renders off-the-shelf Variational Inference (VI) computationally impractical. In this work, we design structured VI families that can efficiently tackle large population studies. To this end, our main idea is to share the parameterization and learning across the different i.i.d. variables in a generative model -symbolized by the model's *plates*. We name this concept *plate amortization*, and illustrate the powerful synergies it entitles, resulting in expressive, parsimoniously parameterized and orders of magnitude faster to train large scale hierarchical variational distributions. We illustrate the practical utility of PAVI through a challenging Neuroimaging example featuring a million latent parameters, demonstrating a significant step towards scalable and expressive Variational Inference.

1 Introduction

Population studies correspond to the analysis of measurements over large cohorts of human subjects. These studies are ubiquitous in health care (Fayaz et al., 2016; Towsley et al., 2011), and can typically involve hundreds of subjects and thousands of measurements per subject. For instance in the context of Neuroimaging (Kong et al., 2019), measurements X can correspond to signals measured in hundreds of locations in the brain for a thousand subjects. Given this observed data X, and a generative model that can produce data given some model parameters Θ, we want to recover the latent Θ that could have yielded the observed X. In our Neuroimaging example, Θ can be local labels for each location and subject, together with global parameters common to all subjects –such as the brain connectivity corresponding to each label. We are interested in recovering the *distribution* of the Θ that could have produced X. Following the Bayesian inference formalism (Gelman et al., 2004), we cast both Θ and X as sets of Random Variables (RVs) and our goal is to recover the *posterior* distribution: $p(\Theta|X)$. Due to the nested structure of the considered applications we will focus on the case where p corresponds to a Hierarchical Bayesian Model (HBM) (Gelman et al., 2004). In the particular context of population studies, the multitude of subjects and measurements per subject implies a large dimensionality for both Θ and X. This large dimensionality in turn creates computational hurdles that we wish to overcome through our method.

Preprint. Under review.

To tackle Bayesian inference, several methods have been proposed in the literature. Earliest works resorted Markov Chain Monte Carlo (Koller & Friedman, 2009), which tend to be slow in high dimensional settings (Blei et al., 2017). Recent approaches, coined Variational Inference (VI), cast the inference as an optimization problem (Blei et al., 2017; Zhang et al., 2019). Within this framework, inference reduces to finding the parametric distribution $q(\Theta; \phi) \in \mathcal{Q}$ closest to the unknown posterior $p(\Theta|X)$ in a variational family $\mathcal{Q}$ chosen by the experimenter. In recent years, VI has benefited from the advent of automatic differentiation (Kucukelbir et al., 2016) and the automatic derivation of the variational family $\mathcal{Q}$ based on the structure of the HBM p (Ambrogioni et al., 2021a,b).

To achieve competitive inference quality, VI requires the variational family $\mathcal{Q}$ to contain distributions closely approximating $p(\Theta|X)$ (Blei et al., 2017). Yet the form of $p(\Theta|X)$ is usually unknown to the experimenter. To forgo a lengthy search for a valid family, one can instead resort to universal density approximators, such as normalizing flows (Papamakarios et al., 2019). To achieve this generality, normalizing flows are highly parameterized and consequently scale poorly with the dimensionality of Θ. In large populations studies, as this dimensionality grows to the million, the parameterization of normalizing flows can in turn become prohibitively large. This creates a detrimental trade-off between expressivity and scalability (Rouillard & Wassermann, 2022). To tackle this challenge, Rouillard & Wassermann (2022) recently proposed –in the ADAVI architecture– to partially share the parameterization of normalizing flows across the hierarchies of a generative model. ADAVI had several limitations we improve upon in this work: removing the Mean Field approximation (Blei et al., 2017); treating arbitrary HBMs instead of pyramidal HBMs only; and introducing non-sample-amortized variants. Critically, while ADAVI tackled the over-parameterization of VI in population studies, it still could not perform inference in very large data regimes due to computational limits. Indeed, as the size of Θ increases, the evaluation of a single gradient over the entirety of the architecture's weights quickly required too much memory and compute. To overcome this second challenge, stochastic VI (Hoffman et al., 2013) subsamples the parameters Θ inferred for at each optimization step. However, using SVI, the weights for the posterior of a given local parameter $\theta \in \Theta$ are only updated when θ is visited by the algorithm. In the presence of hundreds of thousands of such local parameters, stochastic VI can become prohibitively slow.

In this work, we introduce the concept of *plate amortization* (PAVI) for fast and universal inference in large scale HBMs. Instead of considering the inference over local parameters θ as separate problems, our main idea is to share both the parameterization and learning across those local parameters –or equivalently across a model's *plates*. We first propose an algorithm to automatically derive an expressive yet parsimoniously-parameterized variational family from a plate-enriched HBM. We then propose a hierarchical stochastic optimization scheme to train this architecture efficiently, obtaining orders of magnitude faster convergence. Leveraging the repeated structure of plate-enriched HBMs, PAVI is able to perform inference over arbitrarily large population studies, with constant parameterization and training time as the cardinality of the problem augments. We illustrate this by applying PAVI to a challenging human brain cortex parcellation, featuring inference of a million parameters over a cohort of 1000 subjects, demonstrating a significant step towards scalable, expressive and fast VI.

2 PAVI architecture

2.1 Hierarchical Bayesian Models (HBMs), templates and plates

Our objective is to perform inference in the context of large population studies modelled using plate-enriched Hierarchical Bayesian Models (HBMs). These HBMs feature conditionally i.i.d. samples from a common conditional distribution at multiple levels, translating the graphical notion of *plates* (Gilks et al., 1994). fig. 1 displays 2 toy instances of this i.i.d sampling: in our target applications, the total number of samples approaches the million (Kong et al., 2019).

HBMs can be compactly represented via a Directed Acyclic Graphs (DAG) template $\mathcal{T}$ (Koller & Friedman, 2009) with vertices –corresponding to RV templates– X and $\Theta = \{\theta_i\}_{i=1..I}$ and plates $\{\mathcal{P}_p\}_{p=0..P}$. We denote as $\text{Plates}(\theta_i)$ the set of plates a given RV template θ_i belongs to. X corresponds to the sets of RVs observed during inference, and Θ to the parameters we want to infer. Our goal is therefore to approximate the posterior distribution $p(\Theta|X)$. In fig. 1, there are 2 latent RV templates: θ_1 and θ_2, two plates $\mathcal{P}_0, \mathcal{P}_1$ and we want to approximate $p(\theta_1, \theta_2|X)$.

A graph template $\mathcal{T}$ can be *grounded* into a HBM $\mathcal{M}$ given some plate cardinalities $\{\mathrm{Card}(\mathcal{P}_p)\}_{p=0..P}$. This *grounding* operation instantiates the repeated structures symbolized by the plates: a given RV *template* θ_i now corresponds to multiple similar *ground* RVs $\{\theta_{i,n}\}_{n=0..N_i}$ with the same parametric form, where $N_i = \prod_{\mathcal{P}\in\mathrm{Plates}(\theta_i)} \mathrm{Card}(\mathcal{P})$. Template grounding is illustrated in fig. 1, where $\mathcal{T}$ is instantiated into $\mathcal{M}^{\text{full}}$. We wish to exploit the repeated structure induced by plates.

Given a graph template $\mathcal{T}$, we will instantiate two HBMs. One is our target –or "full"– model denoted $\mathcal{M}^{\text{full}}$. This model typically features large plate cardinalities $\mathrm{Card}^{\text{full}}(\mathcal{P})$, making it computationally intractable. Instead of tackling inference directly for this model, we will instantiate the same template $\mathcal{T}$ into a second HBM $\mathcal{M}^{\text{redu}}$, the "reduced" model, of tractable plate cardinalities $\mathrm{Card}^{\text{redu}}(\mathcal{P}) \ll \mathrm{Card}^{\text{full}}(\mathcal{P})$. $\mathcal{M}^{\text{redu}}$ has the same template as $\mathcal{M}^{\text{full}}$, meaning the same dependency structure and the same parametric form for its conditional distributions. The only difference lies in $\mathcal{M}^{\text{redu}}$'s reduced cardinalities, resulting in fewer ground RVs, as visible in fig. 1. Our goal is to train over the tractable reduced model $\mathcal{M}^{\text{redu}}$ to obtain a variational distribution q usable to perform inference over the intractable target model $\mathcal{M}^{\text{full}}$.

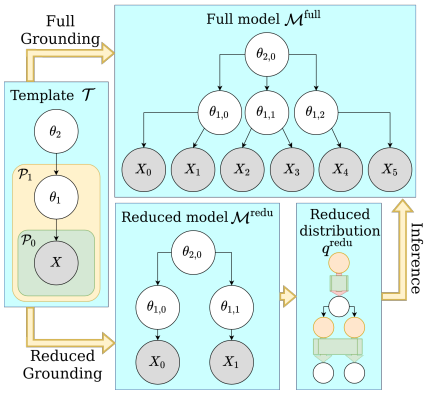

Figure 1: **Plate Amortized Variational Inference (PAVI) working principle**. Starting on the left, the graph template $\mathcal{T}$ is grounded into 2 separate HBMs: $\mathcal{M}^{\text{full}}$ (top) and $\mathcal{M}^{\text{redu}}$ (down) of respective plate cardinalities $(3, 2)$ –large– and $(2, 1)$ –small. Based on $\mathcal{M}^{\text{redu}}$, the reduced distribution q^{redu} is constructed. We train q^{redu} over data slices of small cardinality, before performing inference over the full model of large cardinality.

2.2 Plate amortization

In this section we introduce the notion of plate amortization: sharing the parameterization of a conditional density estimator across a model's plates to reduce the parameterization of inference. Traditional VI aims at searching for a parametric distribution $q(\Theta;\phi)$ that will best approximate the posterior distribution of Θ given a value $\mathbf{X}_0$ for the X: $q(\Theta;\phi_0) \simeq p(\Theta|X=\mathbf{X}_0)$ where ϕ_0 are the optimal weights corresponding to $\mathbf{X}_0$. When presented with a new value $\mathbf{X}_1$ for X, optimization has to be performed again to search for the weights ϕ_1, such that $q(\Theta;\phi_1) \simeq p(\Theta|X=\mathbf{X}_1)$. *Sample amortized* inference (Zhang et al., 2019; Cremer et al., 2018) aims instead at performing inference in the general case, regressing the weights ϕ using an *encoder* f of the observed data $\mathbf{X}$: $q(\Theta;\phi = f(\mathbf{X})) \simeq p(\Theta|X=\mathbf{X})$. The cost of learning the weights of the encoder is *amortized* since inference can be performed for any new sample $\mathbf{X}$ with no additional optimization. We propose to exploit the concept of amortization, but to apply it at a different granularity, leading to our notion of *plate amortization*.

Instead of amortizing across the different samples $\mathbf{X}$ of the observed RV template X we will perform inference amortization across the different ground RVs $\{\theta_{i,n}\}_{n=0..N_i}$ corresponding to the same RV template θ_i. Specifically, to a RV template θ_i, we will associate a conditional density estimator $q_{i,\bullet}(\theta_{i,\bullet};\phi_i,\bullet)$ with weights ϕ_i shared across all the ground RVs $\{\theta_{i,n}\}_{n=0..N_i}$. The variational posterior for a given ground RV $\theta_{i,n}$ will be an instance of this conditional density estimator, conditioned by an encoding $\mathbf{E}_{i,n}$: $q_{i,n}(\theta_{i,n};\phi_i,\mathbf{E}_{i,n})$.

The resulting distributions $q_{i,n}$ thus have 2 sets of weights, ϕ_i and $\mathbf{E}_{i,n}$, creating a parameterization trade-off. Concentrating all of $q_{i,n}$'s parameterization into ϕ_i results in all the ground RVs $\theta_{i,n}$ having almost the same posterior distribution. On the contrary, concentrating all of $q_{i,n}$'s parameterization into $\mathbf{E}_{i,n}$ allows the $\theta_{i,n}$ to have completely different posterior distributions. But in a large cardinality setting, this freedom can result in a massive number of weights, proportional to the number of ground RVs times the encoding size. This double parameterization is therefore efficient when the majority of the weights for the density estimator $q_{i,n}$ is concentrated into ϕ_i. For instance, casting $q_{i,n}$ as a conditional normalizing flow (Papamakarios et al., 2019), the burden of approximating the correct parametric form for the posterior is placed onto ϕ_i, while $\mathbf{E}_{i,n}$ can be a lightweight vector of summary statistics specific to each ground RV $\theta_{i,n}$. In section 3, we will also see that this shared parameterization has synergies with stochastic training.

2.3 Variational family design

To define our variational family, we will push forward the prior $p(\Theta)$ into the variational distribution $q(\Theta)$. This push-forward will be implemented using conditional normalizing flows defined at the graph template level, conditioned by encodings defined at the ground HBM level. Consider a RV template θ_i, corresponding to the ground RVs $\theta_{i,n}$. In the full HBM $\mathcal{M}^{\text{full}}$, the plate structure indicates that θ_i is associated to a unique conditional distribution p_i shared across all ground RVs:

$$\log p^{\text{full}}(\Theta, X) = \sum_{n=0}^{N_X^{\text{full}}} \log p_X(x_n|\pi(x_n)) + \sum_{i=1}^{I} \sum_{n=0}^{N_i^{\text{full}}} \log p_i(\theta_{i,n}|\pi(\theta_{i,n})) \ , \tag{1}$$

where $\pi(\theta_i^n)$ are the parents of the RV θ_i^n, whose value condition θ_i^n's distribution. We indicate with a $\bullet_X$ index all variables related to the observed RVs X. The number of ground RVs N_i^{full} is the product of the plate cardinalities $\{\text{Card}^{\text{full}}(\mathcal{P})\}_{\mathcal{P}\in\text{Plates}(\theta_i)}$. To every parameter RV template θ_i, we associate a conditional normalizing flow $\mathcal{F}_i$, parameterized by the weights ϕ_i. Every ground RV $\theta_{i,n}$ is in turn associated to a separate encoding $\mathbf{E}_{i,n}$. In fig. 2, θ_1 is associated to the flow $\mathcal{F}_1$ pushing forward 2 different ground RVs. This results in the variational distribution:

$$\begin{aligned} \log q^{\text{full}}(\Theta) &= \sum_{i=1}^{I} \sum_{n=0}^{N_i^{\text{full}}} \log q_{i,n}(\theta_{i,n}|\pi(\theta_{i,n})) \ , \\ \log q_{i,n}(\theta_{i,n}|\pi(\theta_{i,n})) &= -\log|\det J_{\mathcal{F}_i}(u_{i,n};\phi_i,\mathbf{E}_{i,n})| + \log p_i(u_{i,n}|\pi(\theta_{i,n})) \ , \\ u_{i,n} &= \mathcal{F}_i^{-1}(\theta_{i,n};\phi_i,\mathbf{E}_{i,n}) \ , \end{aligned} \tag{2}$$

where the distribution $q_{i,n}$ is the push-forward of the prior distribution p_i through the conditional normalizing flow $\mathcal{F}_i$, conditioned by the encoding $\mathbf{E}_{i,n}$. This push-forward is illustrated in fig. 2, where flows $\mathcal{F}$ push the RVs u into the RVs θ. This "cascading" scheme was first introduced by Ambrogioni et al. (2021b), and makes q^{full} inherit the conditional dependencies of the prior p.

2.4 Encoding schemes

The distributions $q_{i,n}(\theta_{i,n}|\pi(\theta_{i,n});\phi_i,\mathbf{E}_{i,n})$ with different ground index n only vary through the value of the encodings $\mathbf{E}_{i,n}$. We detail two different schemes to derive those encodings:

Free plate encodings (PAVI-F) In this scheme, $\mathbf{E}_{i,n}$ are free weights. We define encodings arrays with the cardinality of the full model $\mathcal{M}^{\text{full}}$, one array $\mathbf{E}_i = [\mathbf{E}_{i,n}]_{n=0..N^{\text{full}}}$ per RV template θ_i. Using this scheme, the encoding values have the most flexibility, but as a result the variational family's parameterization scales linearly with the cardinalities $\text{Card}^{\text{full}}(\mathcal{P})$. Indeed, an additional ground RV in an existing plate necessitates an additional encoding vector. The resulting weights increment is nevertheless far lighter than the addition of a fully parameterized normalizing flow, as would be the case in the non-plate-amortized regime. The PAVI-F scheme cannot be sample amortized: when presented with an unseen sample $\mathbf{X}$, though the value of the weights ϕ_i could be kept as an efficient warm start, the optimal value for the encodings $\mathbf{E}_{i,n}$ would have to be searched again.

Deep set encoder (PAVI-E) In this scheme the encodings are no longer free weights but obtained processing the observed data X through an encoder f: $\mathbf{E} = f(\mathbf{X};\eta)$. As encoder f we use a *deep-set* architecture exploiting the data's plate-induced permutation invariance –detailed in our supplemental material (Zaheer et al., 2018; Lee et al., 2019). Encodings $\mathbf{E}_{i,n}$ no longer are weights for the variational family, and are replaced by the encoder's weights η. This scheme furthermore allows for *sample amortization* across different data samples $\mathbf{X}_0, \mathbf{X}_1, ...$ –see section 2.2. Note that an encoder will be used to generate the encodings whether the inference is sample amortized or not.

We have defined the architecture q^{full} to perform inference over the target model $\mathcal{M}^{\text{full}}$. Due to the large plate cardinalities $\text{Card}^{\text{full}}(\mathcal{P})$, it is however computationally impossible to optimize directly over the distribution q^{full}. In the next section we present a stochastic scheme to overcome this computational hurdle.

3 PAVI stochastic training

3.1 Reduced distribution and loss

Instead of optimizing over the computationally intractable distribution q^{full}, we will use a distribution that has the cardinalities of the reduced model $\text{Card}^{\text{redu}}(\mathcal{P})$. At each optimization step t, we will randomly select inside $\mathcal{M}^{\text{full}}$ paths of reduced cardinality, as visible in fig. 2. Selecting paths is equivalent to selecting from X a RV subset of size N_X^{redu}, denoted $X^{\text{redu}}[t]$. We subsequently select from Θ the RV set $\Theta^{\text{redu}}[t]$ of ascendants and descendants of $X^{\text{redu}}[t]$. For a given θ_i, we denote as $\mathcal{B}_i^{\text{redu}}[t]$ the resulting batch of selected ground RVs, of size N_i^{redu}. Inferring over $\Theta^{\text{redu}}[t]$, we will simulate the fact that we train over the distribution q^{full}, resulting in the distribution:

$$\log q^{\text{redu}}(\Theta^{\text{redu}}[t]) = \sum_{i=1}^{I} \frac{N_i^{\text{full}}}{N_i^{\text{redu}}} \sum_{n \in \mathcal{B}_i^{\text{redu}}[t]} \log q_{i,n}(\theta_{i,n}|\pi(\theta_{i,n})) \tag{3}$$

where the factor $N_i^{\text{full}}/N_i^{\text{redu}}$ simulates that we observe as many ground RVs as in the full HBM $\mathcal{M}^{\text{full}}$ by repeating the ground RVs from $\mathcal{M}^{\text{redu}}$ (Hoffman et al., 2013). Similarly, the loss used at the optimization step t is the reduced ELBO constructed using $X^{\text{redu}}[t]$ as observed RVs:

$$\log p^{\text{redu}}(X^{\text{redu}}[t], \Theta^{\text{redu}}[t]) = \frac{N_X^{\text{full}}}{N_X^{\text{redu}}} \sum_{n \in \mathcal{B}_X^{\text{redu}}[t]} \log p_X(x_n|\pi(x_n)) + \sum_{i=1}^{I} \frac{N_i^{\text{full}}}{N_i^{\text{redu}}} \sum_{n \in \mathcal{B}_i^{\text{redu}}[t]} \log p_i(\theta_{i,n}|\pi(\theta_{i,n}))$$

$$\text{ELBO}^{\text{redu}}[t] = \mathbb{E}_{\Theta^{\text{redu}} \sim q^{\text{redu}}} \left[\log p^{\text{redu}}(X^{\text{redu}}[t], \Theta^{\text{redu}}[t]) - \log q^{\text{redu}}(\Theta^{\text{redu}}[t]) \right] \tag{4}$$

This scheme can be viewed as the instantiation of $\mathcal{M}^{\text{redu}}$ over batches of $\mathcal{M}^{\text{redu}}$'s ground RVs. In fig. 2 we can see that q^{redu} has the cardinalities of $\mathcal{M}^{\text{redu}}$, and replicates its conditional dependencies. The resulting training is analogous to the usage of stochastic VI (Hoffman et al., 2013) over $\mathcal{M}^{\text{full}}$, generalized with multiple hierarchies and using minibatches of ground RVs. Our novelty lies in the interaction of this stochastic scheme with plate amortization, as explained in the next section.

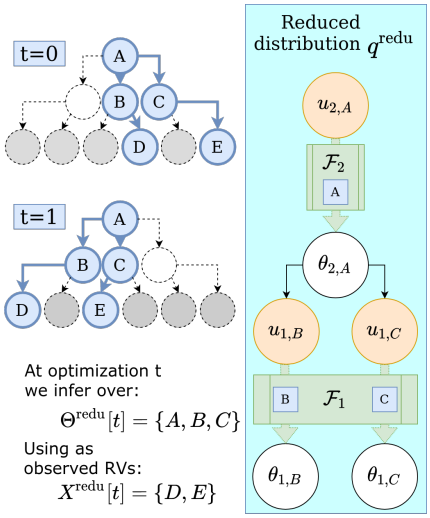

Figure 2: **PAVI stochastic training scheme** The reduced distribution q^{redu} features 2 conditional normalizing flows $\mathcal{F}_1$ and $\mathcal{F}_2$ respectively associated to the RV templates θ_1 and θ_2. During the stochastic training, q^{redu} is instantiated over different branchings of the full model $\mathcal{M}^{\text{full}}$ –highlighted in blue on the left. The branchings have the cardinalities of $\mathcal{M}^{\text{redu}}$ and change at each stochastic training step $t = 0, 1$. The branching determine the encodings $\mathbf{E}$ conditioning the flows $\mathcal{F}$ –as symbolised by the letters A, B, C– and the observed data slice –as symbolised by the letters D, E.

3.2 Sharing learning across plates

In a traditional stochastic VI training, every ground RV $\theta_{i,n}$ corresponding to the same template θ_i is associated to individual weights. Those weights are trained only when $\theta_{i,n}$ is visited by the algorithm, that is to say at an optimization step t when $n \in \mathcal{B}_i^{\text{redu}}[t]$. In the context of very large model plates, this event can become rare. If $\theta_{i,n}$ is furthermore associated to a highly-parameterized density estimator –such as a normalizing flow– many optimization steps can be required for the distribution $q_{i,n}$ to converge. The combination of those two items can lead to a slow training.

Instead, our idea is to share the learning across the ground RVs $\theta_{i,n}$. Indeed, due to the problem's plate structure, we consider the inference over those ground RVs as different instances of a common density estimation task. The precise implementation of this shared learning depends on the chosen encoding scheme –as described in section 2.4:

Conditional flow weight sharing (PAVI-F) As seen in section 2.3, a large part of the parameterization of the density estimators $q_{i,n}(\theta_{i,n}|\pi(\theta_{i,n}); \phi_i, \mathbf{E}_{i,n})$ is mutualized via the plate-wide-shared weights ϕ_i. At each optimization step t, the encodings $\mathbf{E}_{i,n}$ corresponding to $n \in \mathcal{B}_i^{\text{redu}}[t]$ are sliced from larger encoding arrays $\mathbf{E}_i = [\mathbf{E}_{i,n}]_{n=0..N^{\text{full}}}$ and are optimized for along with the

weights ϕ_i. This means that most of the weights of the flows $\mathcal{F}_i$ –concentrated in ϕ_i– are trained at every optimization step, across all the selected batches $\mathcal{B}_i^{\text{redu}}[t]$. This can result in drastically faster convergence, as demonstrated in our experiments. In fig. 2, at $t = 0$, $\mathcal{B}_1^{\text{redu}}[0] = \{1, 2\}$ and the trained encodings are therefore $\{\mathbf{E}_{1,1}, \mathbf{E}_{1,2}\}$, and at $t = 1$ $\mathcal{B}_1^{\text{redu}}[1] = \{0, 1\}$ and the used encodings are $\{\mathbf{E}_1^0, \mathbf{E}_1^1\}$. The weights ϕ_1 and ϕ_2 of the flows $\mathcal{F}_1$ and $\mathcal{F}_2$ are trained at both steps $t = 1$ and $t = 2$. At inference, instead of slicing the encoding arrays, the full arrays $\mathbf{E}_i$ are used to obtain the distribution q^{full}.

Encoder set size generalization (PAVI-E) The PAVI-E scheme also benefits from the sharing of the weights ϕ_i. In addition, it doesn't cast the encodings $\mathbf{E}_{i,n}$ as free weights, but as the output of a parametric encoder $f(\bullet; \eta)$. As a result, at training all the architecture's weights –ϕ_i and η– are trained at every optimization step t. At inference, to generate the full encoding arrays $\mathbf{E}_i = [\mathbf{E}_{i,n}]_{n=0..N^{\text{full}}}$ to plug into q^{full}, this scheme builds up on a property of the particular deep-set-like architecture we use for the encoder f: *set size generalization* (Zaheer et al., 2018). Through training, the encoder f learnt a hierarchy of permutation-invariant functions over $\text{Card}^{\text{redu}}(\mathcal{P})$-sized sets of data points. At inference, we instead apply the trained encoder to sets of size $\text{Card}^{\text{full}}(\mathcal{P})$:

$$\begin{aligned} &\textit{At training:} && \mathbf{E}_{i,n} = f_{i,n}(\mathbf{X}^{\text{redu}}[t]) && \text{for} \quad n \in \mathcal{B}_i^{\text{redu}}[t] \\ &\textit{At inference:} && \mathbf{E}_{i,n} = f_{i,n}(\mathbf{X}) && \text{for} \quad n = 0..N_i^{\text{full}} \end{aligned} \tag{5}$$

where $\mathbf{X}^{\text{redu}}[t]$ denotes the observed data corresponding to $X^{\text{redu}}[t]$. This property –learning an encoder over small sets to use it over large sets– is very strong, especially in the sample amortized context. Benefiting from set size generalization, we can effectively train a sample amortized variational family over the lightweight model $\mathcal{M}^{\text{redu}}$, and obtain "for free" a sample amortized variational family for the heavyweight model $\mathcal{M}^{\text{full}}$.

Summary In section 2 we proposed an architecture sharing its parameterization across a model's plates. In section 3 we proposed a stochastic scheme to train this architecture over batches of data of reduced cardinality. Across those data batches, we share the learning of density estimators, resulting in the fast training of a variational posterior q^{full}, as demonstrated in the following experiments.

4 Results and discussion

All experiments were performed using the Tensorflow Probability library (Dillon et al., 2017), on computational cluster nodes equipped with a Tesla V100-16Gb GPU and 4 AMD EPYC 7742 64-Core processors. VRAM intensive experiments in fig. 4 were performed on an Ampere 100 PCIE-40Gb GPU. Throughout this section we focus on the usage of the ELBO metric, as a proxy to the KL divergence between the variational posterior and the unknown true posterior. ELBO is measured across 20 different data samples $\mathbf{X}$, with 5 random seeds per sample. The ELBO allows to compare the relative performance of different architectures on a given inference problem. In our supplemental material we also provide with sanity checks to assess the quality of the obtained results.

4.1 Plate amortization and convergence speed

In this experiment, we illustrate how plate amortization results in faster convergence. We consider the following Gaussian Random Effects model (GRE):

$$\begin{aligned} &\forall n_1 = 1..\,\text{Card}(\mathcal{P}_1) \\ &\forall n_0 = 1..\,\text{Card}(\mathcal{P}_0) \end{aligned} \quad X_{n_1,n_0} | \theta_{1,n_1} \sim \mathcal{N}(\theta_{1,n_1}, \sigma_x^2)$$
$$\forall n_1 = 1..\,\text{Card}(\mathcal{P}_1) \quad \theta_{1,n_1} | \theta_{2,0} \sim \mathcal{N}(\theta_{2,0}, \sigma_1^2) \qquad \theta_{2,0} \sim \mathcal{N}(\vec{0}_D, \sigma_2^2) \ , \tag{6}$$

where D represents the feature size of the data $\mathbf{X}$, determining the dimensionality of the group means θ_1 and of the population means θ_2 as D-dimensional Gaussians. We opted in this equation for a more practical double indexing scheme instead of a simple indexing as in our methods. The GRE model features two nested plates: the group plate $\mathcal{P}_1$ and the sample plate $\mathcal{P}_0$ as in fig. 1. Performing inference over this HBM, the objective is to retrieve the posterior distribution of the group means θ_1 and the population mean θ_2 given the observed sample X.

Here we set $D = 8$, $\text{Card}^{\text{full}}(\mathcal{P}_1) = 100$ and $\text{Card}^{\text{redu}}(\mathcal{P}_1) = 2$. We compare our PAVI architecture to a stochastic non-plate-amortized baseline with the same architecture as PAVI (Hoffman et al.,

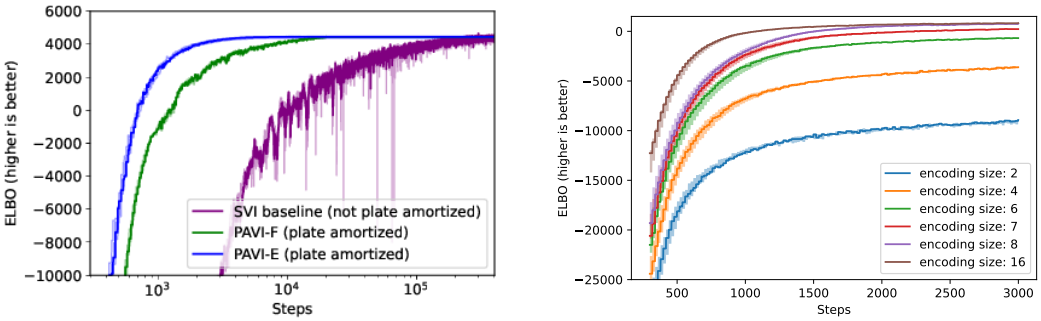

Figure 3: **Left panel: Plate amortization increases convergence speed** Plot of the ELBO (higher is better) as a function of the optimization steps (log-scale) for our methods PAVI-F (in green) and PAVI-E (in blue) versus a non-plate-amortized baseline (in purple). Due to plate amortization, our method converges orders of magnitude faster to the same asymptotic ELBO as its non-plate-amortized counterpart.; **Right panel: Encodings as ground RVs summary statistics** Plot of the ELBO (higher is better) as a function of the optimization steps for the PAVI-F architecture with increasing encoding sizes. As the encoding size augments, so does the asymptotic performance, until reaching the dimensionality of the posterior's sufficient statistics ($D = 8$), after which performance plateaus. Encoding size allows for a clear trade-off between memory footprint and inference quality.

2013). The main difference is that every ground RV θ_i^n is associated in the baseline to an individual flow $\mathcal{F}_{i,n}$ instead of sharing the same flow $\mathcal{F}_i$ –as described in section 2.3. Figure 3 (left) displays the evolution of the ELBO for the baseline and PAVI with free encoding (PAVI-F) and with deep set encoders (PAVI-E). We see that for both plate amortized methods, the convergence speed to an asymptotic ELBO equals to the one of the non-plate-amortized baseline is orders of magnitudes faster, and numerically more stable. This stems from the individual flows $\mathcal{F}_{i,n}$ only being trained when the corresponding $\theta_{i,n}$ is visited by the stochastic training, while the shared flow $\mathcal{F}_i$ is updated at every optimization step in PAVI. We also note that the PAVI-E scheme has a faster convergence than the PAVI-F scheme, sharing not only the training of the conditional flows, but also of the encoder through the stochastic optimization steps. In practice however, the additional compute implied by the encoder results step of longer duration, and ultimately in slower convergence, as illustrated in section 4.3.

4.2 Impact of encoding size

Now we illustrate the role of encodings as ground RV's posterior summary statistics –as described in section 2.2. We use the GRE HBM detailed in eq. (6), using $D = 8$, $\text{Card}^{\text{full}}(\mathcal{P}_1) = 20$ and $\text{Card}^{\text{redu}}(\mathcal{P}_1) = 2$. We use a single PAVI-F architecture, varying the dimensionality of the encodings $\mathbf{E}_{i,n}$ –see section 2.3. Due to plate amortization, this encoding size determines how much individual information each ground RV $\theta_{i,n}$ is associated to. The size of the encodings –varying from 2 to 16– is to be compared with the dimensionality of the problem, in this case $D = 8$. Indeed, in the GRE context, $D = 8$ corresponds to the dimensionality of the sufficient statistics needed to reconstruct the posterior distribution of a given group mean –all other statistics such as the posterior variance being shared between all the group means. Figure 3 (right) shows how the asymptotic performance steadily increases when the encoding size augments, before plateauing once reaching the sufficient summary statistic size $D = 8$. Interestingly, increasing the encoding size also leads to faster convergence: redundancy in the encoding can likely be exploited in the optimization. Encoding size appears as a straightforward hyperparameter allowing to trade inference quality for computational efficiency. It is also interesting to notice that increasing the encoding size leads experimentally to diminishing returns in terms of performance. This property can be exploited in large dimensionality settings to drastically reduce the memory footprint of inference while maintaining acceptable performance.

4.3 Scaling with plate cardinalities

Now we put in perspective the gains from plate amortization when scaling up an inference problem's cardinality. We consider the GRE model in eq. (6) with $D = 2$ and augment the plate cardinalities $(\text{Card}^{\text{full}}(\mathcal{P}_1), \text{Card}^{\text{redu}}(\mathcal{P}_1)) : \ (2, 1) \to (20, 5) \to (200, 20)$. In doing so, we augment the number of estimated parameters $\Theta : \ 6 \to 42 \to 402$.

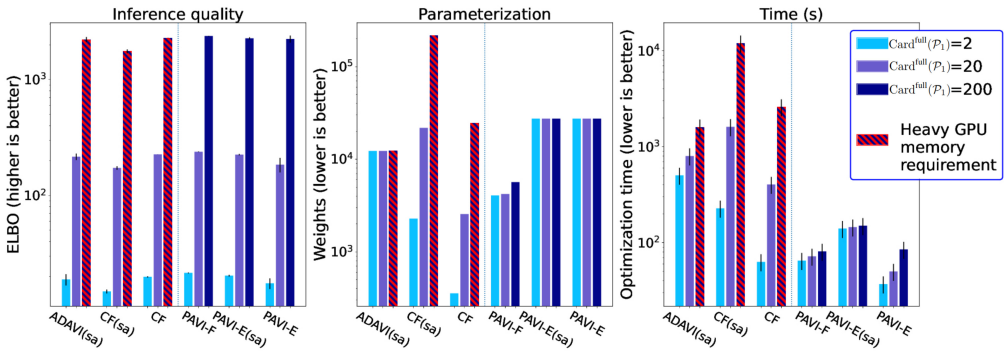

Figure 4: **PAVI provides with favorable parameterization and training time as the cardinality of the target model augments** Our architecture PAVI is displayed on the right of each panel. We augment the cardinality $\text{Card}^{\text{full}}(\mathcal{P}_1)$ of the GRE model –described in eq. (6). While doing so, we compare 3 different metrics: *In the first panel:* inference quality, as measured by the ELBO. None of the presented SOTA architecture's performance degrades as the cardinality of the problem augments. *In the second pannel:* parameterization, comparing the number of trainable weights of each architecture. PAVI –similar to ADAVI– displays a constant number of weights as the cardinality of the problem increases –or almost constant for PAVI-F. *Third panel:* GPU training time. Benefiting from learning across plates, PAVI has a short and almost constant training time as the cardinality of the problem augments. At $\text{Card}^{\text{full}}(\mathcal{P}_1) = 200$, CF and ADAVI required large GPU memory, a constraint absent from PAVI due to its stochastic training.

Baselines We compare our PAVI architecture against 2 state-of-the-art baselines: *Cascading Flows* (CF) (Ambrogioni et al., 2021b) is a non-plate-amortized structured VI architecture improving on the baseline presented in section 4.1; ADAVI (Rouillard & Wassermann, 2022) is a structured VI architecture with constant parameterization with respect to a problem's cardinality, but large training times and memory footprint. For all architectures, we indicate with the suffix *(sa) sample amortization*, corresponding to the classical meaning of amortization, as detailed in section 2.2. More details can be found in our supplemental material.

As the cardinality of the problem augments, fig. 4 shows how PAVI maintains a state-of-the-art inference quality, while being more computationally attractive. Specifically, in terms of *parameterization*, both ADAVI and PAVI-E provide with a heavyweight but constant parameterization as the cardinality $\text{Card}^{\text{full}}(\mathcal{P}_1)$ of the problem augments. Comparatively, both CF and PAVI-F's parameterization scale linearly with $\text{Card}^{\text{full}}(\mathcal{P}_1)$, but with a drastically lighter augmentation for PAVI-F. Indeed, for an additional ground RV, CF requires an additional fully parameterized normalizing flow, whereas PAVI-F only requires an additional lightweight encoding vector. In detail, PAVI-F's parameterization due to the plate-wide-shared ϕ_1 represents a constant $\approx 2k$ weights, while the part due to the encodings $\mathbf{E}_{1,n}$ grows linearly from 16 to 160 to $1.6k$ weights. Note that PAVI's stochastic training also allows for a controlled GPU memory during optimization, removing the need for a larger memory as the cardinality of the problem augments –a hardware constraint that can become unaffordable at very large cardinalities. In terms of *convergence speed*, PAVI benefits from plate amortization to have orders of magnitude faster convergence. Plate amortization is particularly significant for the PAVI-E(sa) scheme, in which a sample-amortized variational family is trained over a dataset of reduced cardinality, yet performs "for free" inference over a HBM of large cardinality. Maintaining $\text{Card}^{\text{redu}}(\mathcal{P}_1)$ constant while $\text{Card}^{\text{full}}(\mathcal{P}_1)$ augments allows for a constant parameterization *and training time* as the cardinality of the problem augments. The effect of plate amortization is particularly noticeable at $\text{Card}^{\text{full}}(\mathcal{P}_1) = 200$ between the PAVI(sa) and CF(sa) architectures, where PAVI performs amortized inference with $10\times$ fewer weights and $100\times$ lower training time. Scaling even higher the cardinality of the problem –$\text{Card}^{\text{full}}(\mathcal{P}_1) = 2000$ for instance– renders ADAVI and CF computationally intractable to use, while PAVI maintains a light memory footprint, and a short training time, as exemplified in the next experiment.

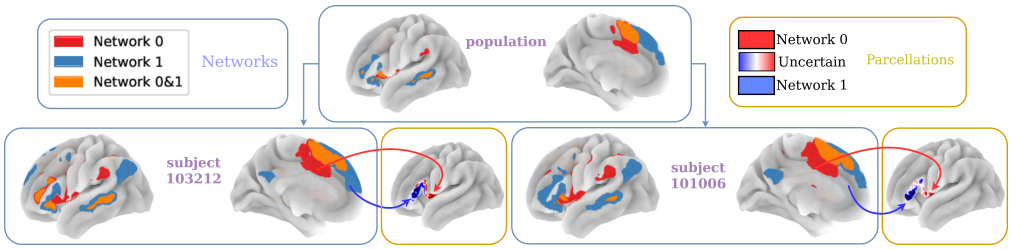

Figure 5: **Probabilistic parcellation of Brocas's area** PAVI can be applied in the challenging context of Neuroimaging population studies. For a cohort of 1000 subjects, 2 of which are represented here –in the bottom 2 items– we present 2 results. First, *connectivity networks* with the brain's left hemisphere –left purple items: this represents the zones of the brain to which the vertices with each label are "wired" to. Second, Broca's area probabilistic *parcellation* –rightmost orange items: we cluster the brain's vertices, associating them to a given *connectivity network*. Our Bayesian method features a notion of uncertainty: coloring transitions from red to an uncertain white to blue, representing the probability of a given vertex to belong to one connectivity network or the other.

4.4 Application: fMRI – parcellation of Broca's area over a large subject cohort

To illustrate the usefulness of our method, we apply PAVI to a challenging Neuroimaging example: a population study for Broca's area's functional *parcellation*. A *parcellation* of a brain region aims at clustering brain vertices into different *connectivity networks*: labels describing the vertices' co-activation with the rest of the brain –as measured using functional Magnetic Resonance Imaging (fMRI). Different subjects can exhibit a strong variability, as visible in fig. 5. However, fMRI has a costly acquisition –meaning that few noisy data is usually gathered for a given subject. It is thus essential to combine the information from different subjects and to have a notion of uncertainty in the obtained results. Those 2 points motivate our usage of Hierarchical Bayesian Models and VI in the Neuroimaging context (Kong et al., 2019): we wish to obtain the posterior distribution of connectivity networks and vertex labels, combining fMRI measurement over a large cohort of subjects. In practice, we use the HCP dataset (Van Essen et al., 2012): 2 acquisition sessions for a cohort of 1000 subjects, with thousands of measurements per subject, for a total parameter space Θ of over a million parameters. We use a model with 3 plates: subjects, measurement sessions and brain vertices. In this high plate cardinality regime, none of the state-of-the-art baselines presented in section 4.3 –CF, ADAVI– can computationally tackle inference. In terms of convergence speed, despite the massive dimensionality of the problem, thanks to plate amortization PAVI converges in a dozen epochs, under an hour of GPU time. The results of our method are visible in fig. 5, supporting the hypothesis of a functional bi-partition of Broca's area into a posterior part involved in phonology and an anterior part involved in lexical/semantic processing - following the anatomical partition between *pars opercularis* and *pars triangularis* (Heim et al., 2009; Zhang et al., 2020).

4.5 Conclusion

In this work we present the novel PAVI architecture, combining a structured variational family and a stochastic training scheme. PAVI is based the concept of plate amortization, allowing to share parameterization and learning across a model's plates. We demonstrated the positive impact of plate amortization on training speed and scaling to large plate cardinality regimes, making a significant step towards scalable, expressive Variational Inference.

Acknowledgments and Disclosure of Funding

This work was supported by the ERC-StG NeuroLang ID:757672.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah,

Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

Alexandre Abraham, Fabian Pedregosa, Michael Eickenberg, Philippe Gervais, Andreas Mueller, Jean Kossaifi, Alexandre Gramfort, Bertrand Thirion, and Gael Varoquaux. Machine learning for neuroimaging with scikit-learn. *Frontiers in Neuroinformatics*, 8, 2014. ISSN 1662-5196. doi: 10.3389/fninf.2014.00014. URL https://www.frontiersin.org/article/10.3389/fninf.2014.00014.

Luca Ambrogioni, Kate Lin, Emily Fertig, Sharad Vikram, Max Hinne, Dave Moore, and Marcel van Gerven. Automatic structured variational inference. In Arindam Banerjee and Kenji Fukumizu (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 676–684. PMLR, 13–15 Apr 2021a. URL https://proceedings.mlr.press/v130/ambrogioni21a.html.

Luca Ambrogioni, Gianluigi Silvestri, and Marcel van Gerven. Automatic variational inference with cascading flows. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 254–263. PMLR, 18–24 Jul 2021b. URL https://proceedings.mlr.press/v139/ambrogioni21a.html.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518):859–877, April 2017. ISSN 0162-1459, 1537-274X. doi: 10.1080/01621459.2017.1285773. URL http://arxiv.org/abs/1601.00670. arXiv: 1601.00670.

Léon Bottou and Olivier Bousquet. The Tradeoffs of Large Scale Learning. In J. Platt, D. Koller, Y. Singer, and S. Roweis (eds.), *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper/2007/file/0d3180d672e08b4c5312dcdafdf6ef36-Paper.pdf.

Chris Cremer, Xuechen Li, and David Duvenaud. Inference Suboptimality in Variational Autoencoders. *arXiv:1801.03558 [cs, stat]*, May 2018. URL http://arxiv.org/abs/1801.03558. arXiv: 1801.03558.

Kamalaker Dadi, Gaël Varoquaux, Antonia Machlouzarides-Shalit, Krzysztof J. Gorgolewski, Demian Wassermann, Bertrand Thirion, and Arthur Mensch. Fine-grain atlases of functional modes for fMRI analysis. *NeuroImage*, 221:117126, 2020. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2020.117126. URL https://www.sciencedirect.com/science/article/pii/S1053811920306121.

Joshua V. Dillon, Ian Langmore, Dustin Tran, Eugene Brevdo, Srinivas Vasudevan, Dave Moore, Brian Patton, Alex Alemi, Matt Hoffman, and Rif A. Saurous. TensorFlow Distributions. *arXiv:1711.10604 [cs, stat]*, November 2017. URL http://arxiv.org/abs/1711.10604. arXiv: 1711.10604.

A Fayaz, P Croft, RM Langford, LJ Donaldson, and GT Jones. Prevalence of chronic pain in the uk: a systematic review and meta-analysis of population studies. *BMJ open*, 6(6):e010364, 2016.

Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 2nd ed. edition, 2004.

W. R. Gilks, A. Thomas, and D. J. Spiegelhalter. A Language and Program for Complex Bayesian Modelling. *The Statistician*, 43(1):169, 1994. ISSN 00390526. doi: 10.2307/2348941. URL https://www.jstor.org/stable/10.2307/2348941?origin=crossref.

Stefan Heim, Simon B. Eickhoff, Anja K. Ischebeck, Angela D. Friederici, Klaas E. Stephan, and Katrin Amunts. Effective connectivity of the left BA 44, BA 45, and inferior temporal gyrus during lexical and phonological decisions identified with DCM. *Human Brain Mapping*, 30(2):392–402, February 2009. ISSN 10659471. doi: 10.1002/hbm.20512.

Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(4):1303–1347, 2013. URL http://jmlr.org/papers/v14/hoffman13a.html.

Eric Jang, Shixiang Gu, and Ben Poole. CATEGORICAL REPARAMETERIZATION WITH GUMBEL-SOFTMAX. pp. 12, 2017.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. Adaptive computation and machine learning. MIT Press, Cambridge, MA, 2009. ISBN 978-0-262-01319-2.

Ru Kong, Jingwei Li, Csaba Orban, Mert Rory Sabuncu, Hesheng Liu, Alexander Schaefer, Nanbo Sun, Xi-Nian Zuo, Avram J. Holmes, Simon B. Eickhoff, and B. T. Thomas Yeo. Spatial topography of individual-specific cortical networks predicts human cognition, personality, and emotion. *Cerebral cortex*, 29 6:2533–2551, 2019.

Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M. Blei. Automatic Differentiation Variational Inference. *arXiv:1603.00788 [cs, stat]*, March 2016. URL http://arxiv.org/abs/1603.00788. arXiv: 1603.00788.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3744–3753. PMLR, 09–15 Jun 2019. URL http://proceedings.mlr.press/v97/lee19d.html.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *CoRR*, abs/1611.00712, 2016. URL http://arxiv.org/abs/1611.00712.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked Autoregressive Flow for Density Estimation. *arXiv:1705.07057 [cs, stat]*, June 2018. URL http://arxiv.org/abs/1705.07057. arXiv: 1705.07057.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing Flows for Probabilistic Modeling and Inference. *arXiv:1912.02762 [cs, stat]*, December 2019. URL http://arxiv.org/abs/1912.02762. arXiv: 1912.02762.

Louis Rouillard and Demian Wassermann. ADAVI: Automatic Dual Amortized Variational Inference Applied To Pyramidal Bayesian Models. In *ICLR 2022*, Virtual, France, April 2022. URL https://hal.archives-ouvertes.fr/hal-03267956.

Hojjat Salehinejad, Julianne Baarbe, Sharan Sankar, Joseph Barfett, Errol Colak, and Shahrokh Valaee. Recent advances in recurrent neural networks. *CoRR*, abs/1801.01078, 2018. URL http://arxiv.org/abs/1801.01078.

Stephen M Smith, Christian F Beckmann, Jesper Andersson, Edward J Auerbach, Janine Bijsterbosch, Gwenaëlle Douaud, Eugene Duff, David A Feinberg, Ludovica Griffanti, Michael P Harms, et al. Resting-state fmri in the human connectome project. *Neuroimage*, 80:144–168, 2013.

Kayle Towsley, Michael I Shevell, Lynn Dagenais, Repacq Consortium, et al. Population-based study of neuroimaging findings in children with cerebral palsy. *European journal of paediatric neurology*, 15(1):29–35, 2011.

D. C. Van Essen, K. Ugurbil, E. Auerbach, D. Barch, T. E. Behrens, R. Bucholz, A. Chang, L. Chen, M. Corbetta, S. W. Curtiss, S. Della Penna, D. Feinberg, M. F. Glasser, N. Harel, A. C. Heath, L. Larson-Prior, D. Marcus, G. Michalareas, S. Moeller, R. Oostenveld, S. E. Petersen, F. Prior, B. L. Schlaggar, S. M. Smith, A. Z. Snyder, J. Xu, and E. Yacoub. The Human Connectome Project: a data acquisition perspective. *Neuroimage*, 62(4):2222–2231, Oct 2012.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2020. ISSN 2162-237X, 2162-2388. doi: 10.1109/TNNLS.2020.2978386. URL http://arxiv.org/abs/1901.00596. arXiv: 1901.00596.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep Sets. *arXiv:1703.06114 [cs, stat]*, April 2018. URL http://arxiv.org/abs/1703.06114. arXiv: 1703.06114.

Cheng Zhang, Judith Butepage, Hedvig Kjellstrom, and Stephan Mandt. Advances in Variational Inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):2008–2026, August 2019. ISSN 0162-8828, 2160-9292, 1939-3539. doi: 10.1109/TPAMI.2018.2889774. URL https://ieeexplore.ieee.org/document/8588399/.

Yizhen Zhang, Kuan Han, Robert Worth, and Zhongming Liu. Connecting concepts in the brain by mapping cortical representations of semantic relations. *Nature Communications*, 11(1):1877, April 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-15804-w.

Supplemental Material

A Supplemental methods

A.1 PAVI implementation details

A.1.1 Plate branchings and stochastic training

As exposed in section 3.1, at each optimization step t we randomly select branchings inside the full model $\mathcal{M}^{\text{full}}$, branchings over which we "instantiate" the reduced model $\mathcal{M}^{\text{redu}}$. In doing so, we define batches $\mathcal{B}_i[t]$ for the RV templates θ_i. Those batches have to be "coherent" with one another: they have to respect the conditional dependencies of the original model $\mathcal{M}^{\text{full}}$. To ensure this, during the stochastic training we do not sample RVs directly but plates:

1. For every plate $\mathcal{P}_p$, we sample without replacement $\text{Card}^{\text{redu}}(\mathcal{P}_p)$ indices amongst the $\text{Card}^{\text{full}}(\mathcal{P}_p)$ possible indices.
2. Then, for every RV template θ_i, we select the ground RVs $\theta_{i,n}$ corresponding to the sampled indices for the plates $\text{Plates}(\theta_i)$.
3. The selected ground RVs $\theta_{i,n}$ constitute the set $\Theta^{\text{redu}}[t]$ of parameters appearing in eq. (3). The same procedure yields the RV subset $X^{\text{redu}}[t]$ and the data slice $\mathbf{X}^{\text{redu}}[t]$.

This stochastic strategy also applies to the selected encoding scheme –described in section 2.4– as detailed in the next sections.

A.1.2 PAVI-F details

In section 2.4 we refer to encodings $\mathbf{E}_i = [\mathbf{E}_{i,n}]_{n=0..N^{\text{full}}}$ corresponding to RV templates θ_i. In practice, we have some amount of sharing for those encodings: instead of defining separate encodings for every RV template, we define encodings for every *plate level*. A plate level is a combination of plates with at least one parameter RV template θ_i belonging to it:

$$\text{PlateLevels} = \{(\mathcal{P}_k..\mathcal{P}_l) = \text{Plates}(\theta_i)\}_{\theta_i \in \Theta} \tag{A.1}$$

For every plate level, we construct a large encoding array with the cardinalities of the full model $\mathcal{M}^{\text{full}}$:

$$\begin{aligned} \text{Encodings} &= \{(\mathcal{P}_k..\mathcal{P}_l) \mapsto \mathbb{R}^{\text{Card}^{\text{full}}(\mathcal{P}_k)\times..\times\text{Card}^{\text{full}}(\mathcal{P}_l)\times D}\}_{(\mathcal{P}_k..\mathcal{P}_l)\in\text{PlateLevels}} \\ \mathbf{E}_i &= \text{Encodings}(\text{Plates}(\theta_i)) \end{aligned} \tag{A.2}$$

Where D is an encoding size that we kept constant to de-clutter the notation but can vary between plate levels. The encodings for a given ground RV $\theta_{i,n}$ then correspond to an element from the encoding array $\mathbf{E}_i$.

A.1.3 PAVI-E details

In the PAVI-E scheme, encodings are not free weights but the output of en encoder $f(\bullet, \eta)$ applied to the observed data $\mathbf{X}$. In this section we detail the design of this encoder.

As in the previous section, the role of the encoder will be to produce one encoding per plate level. We start from a dependency structure for the plate levels:

$$\begin{aligned} \forall(\mathcal{P}_a..\mathcal{P}_b) \in \text{PlateLevels}\ , \\ \forall(\mathcal{P}_c..\mathcal{P}_d) \in \text{PlateLevels}\ , \\ (\mathcal{P}_a..\mathcal{P}_b) \in \pi((\mathcal{P}_c..\mathcal{P}_d)) \Leftrightarrow {}^{\exists\theta_i/\,\text{Plates}(\theta_i)=(\mathcal{P}_a..\mathcal{P}_b)}_{\exists\theta_j/\,\text{Plates}(\theta_j)=(\mathcal{P}_c..\mathcal{P}_d)} \theta_j \in \pi(\theta_i) \end{aligned} \tag{A.3}$$

note that this dependency structure is in the "backward" direction: a plate level will be the parent of another plate level, if the former contains a RV who has a child in the latter. We therefore obtain a plate level dependency structure that "reverts" the conditional dependency structure of the graph template $\mathcal{T}$. To avoid redundant paths in this dependency structure, we take the maximum branching of the obtained graph.

Given the plate level dependency structure, we will recursively construct the encodings, starting from the observed data:

$$\forall x \in X : \quad \text{Encodings}(\text{Plates}(x)) = \rho(\mathbf{x}) \tag{A.4}$$

where $\mathbf{x}$ is the observed data for the RV x, and ρ is a simple encoder that processes every observed ground RV's value independently through an identical multi-layer perceptron. Then, until we have exhausted all plate levels, we process existing encodings to produce new encodings:

$$\begin{aligned} &\forall (\mathcal{P}_k..\mathcal{P}_l) \in \text{PlateLevels} \,/\, \nexists x \in X, \text{Plates}(x) = (\mathcal{P}_k..\mathcal{P}_l) : \\ &\text{Encodings}((\mathcal{P}_k..\mathcal{P}_l)) = g(\text{Encodings}(\pi(\mathcal{P}_k..\mathcal{P}_l))) \end{aligned} \tag{A.5}$$

where g is the composition of attention-based deep-set networks called *Set Transformers* (Lee et al., 2019; Zaheer et al., 2018). For every plate $\mathcal{P}_p$ present in the parent plate level but absent in the child plate level, g will compute summary statistics *across* that plate, effectively contracting the corresponding batch dimensionality in the parent encoding (Rouillard & Wassermann, 2022).

In the case of multiple observed RVs, we run this "backward pass" independently for each observed data –with one encoder per observed RV. We then concatenate the resulting encodings corresponding to the same plate level.

For more precise implementation details, we invite the reader to consult the codebase released with this supplemental material.

A.2 PAVI algorithms

More technical details can be found in the codebase provided with this supplemental material.

A.2.1 Architecture build

Algorithm 1: PAVI architecture build

Input: Graph template $\mathcal{T}$, plate cardinalities $\{(\text{Card}^{\text{full}}(\mathcal{P}_p), \text{Card}^{\text{redu}}(\mathcal{P}_p))\}_{p=0..P}$, encoding scheme
Output: q^{full} distribution
for $i = 1..I$ **do**
  Construct conditional flow $\mathcal{F}_i$;
  Define conditional posterior distributions $q_{i,n}$ as the push-forward of the prior via $\mathcal{F}_i$, following eq. (2);
Combine the $q_{i,n}$ distributions following the cascading flows scheme, as in section 2.3 (Ambrogioni et al., 2021b) ;
if *PAVI-F encoding scheme* **then**
  Construct encoding arrays $\{\mathbf{E}_i = [\mathbf{E}_{i,n}]_{n=0..N_i^{\text{full}}}\}_{i=1..I}$ as in appendix A.1.2 ;
else if *PAVI-E encoding scheme* **then**
  Construct encoder f as in appendix A.1.3 ;

A.2.2 Stochastic training

Algorithm 2: PAVI stochastic training

Input: Untrained architecture q^{full}, observed data $\mathbf{X}$, encoding scheme, number of steps T
Output: trained architecture q^{full}
for $t = 0..T$ **do**
  Sample plate indices to define the batches $\mathcal{B}_i[t]$, the latent $\Theta^{\text{redu}}[t]$ and the observed $X^{\text{redu}}[t]$ and $\mathbf{X}^{\text{redu}}[t]$, following appendix A.1.1 ;
  Define reduced distribution p^{redu} ;
  if *PAVI-F encoding scheme* **then**
    Collect encodings $\mathbf{E}_{i,n}$ by slicing from the arrays $\mathbf{E}_i$ the elements corresponding to the batches $\mathcal{B}_i[t]$;
  else if *PAVI-E encoding scheme* **then**
    Compute encodings as $\mathbf{E} = f(\mathbf{X}^{\text{redu}}[t])$;
  Feed obtained encodings into q^{redu} ;
  Compute reduced ELBO as in eq. (4), back-propagate its gradient ;
  Update conditional flow weights $\{\phi_i\}_{i=1..I}$;
  if *PAVI-F encoding scheme* **then**
    Update encodings $\{\mathbf{E}_{i,n}\}_{i=1..I, n\in\mathcal{B}_{i,t}}$;
  else if *PAVI-E encoding scheme* **then**
    Update encoder weights η;

A.2.3 Inference

Algorithm 3: PAVI inference

Input: trained architecture q^{full}, observed data $\mathbf{X}$, encoding scheme
Output: approximate posterior distribution
if *PAVI-F encoding scheme* **then**
  Collect full encoding arrays $\mathbf{E}_i$;
else if *PAVI-E encoding scheme* **then**
  Compute encodings as $\mathbf{E} = f(\mathbf{X})$ using set size generalization ;
Feed obtained encodings into q^{full} ;

A.3 Inference gaps

In terms of inference quality, the impact of our architecture can be formalized following the *gaps* terminology (Cremer et al., 2018). Consider a joint distribution $p(\Theta, X)$, and a value $\mathbf{X}$ for the RV template X. We pick a variational family $\mathcal{Q}$, and in this family look for the parametric distribution $q(\Theta;\phi)$ that best approximates $p(\Theta|X=\mathbf{X})$. Specifically, we want to minimize the Kulback-Leibler divergence (Blei et al., 2017) between our variational posterior and the true posterior, that Cremer et al. (2018) refer to as the *gap* $\mathcal{G}$:

$$\begin{aligned}\mathcal{G} &= \mathrm{KL}(q(\Theta;\phi)||p(\Theta|X)) \\ &= \log p(X) - \mathrm{ELBO}(q;\phi)\end{aligned} \tag{A.6}$$

We denote $q^*(\Theta;\phi^*)$ the optimal distribution inside $\mathcal{Q}$ that minimizes the KL divergence with the true posterior:

$$\begin{aligned}\mathcal{G}_{\text{approx}}(\mathcal{Q};\phi^*) &= \log p(X) - \mathrm{ELBO}(q^*;\phi^*) \\ &\geq 0 \\ \mathcal{G}_{\text{vanilla VI}} &= \mathcal{G}_{\text{approx}}\end{aligned} \tag{A.7}$$

The *approximation gap* $\mathcal{G}_{\text{approx}}$ depends on the expressivity of the variational family $\mathcal{Q}$, specifically whether $\mathcal{Q}$ contains distributions arbitrarily close to the posterior –in the KL sense. Cremer et al. (2018) demonstrate that, in the case of sample amortized inference, when the weights ϕ no longer are free but the output of an encoder $f \in \mathcal{F}$, inference cannot be better than in the non-sample-amortized

case, and a positive *amortization gap* is introduced:

$$\begin{aligned} \mathcal{G}_{\text{sa}}(\mathcal{Q}, \mathcal{F}; \eta^*) &= \mathcal{G}_{\text{approx}}(\mathcal{Q}; f(\mathbf{X}, \eta^*)) - \mathcal{G}_{\text{approx}}(\mathcal{Q}; \phi^*) \\ &\geq 0 \\ \mathcal{G}_{\text{sample amortized VI}} &= \mathcal{G}_{\text{approx}} + \mathcal{G}_{\text{sa}} \end{aligned} \tag{A.8}$$

Where we denote as η^* the optimal weights for the encoder f inside the function family $\mathcal{F}$. The gap terminology can be interpreted as follow: "theoretically, sample amortization cannot be beneficial in terms of KL divergence for the inference over a given sample $\mathbf{X}$."

Using the same gap terminology, we can define gaps implied by our PAVI architecture. Instead of picking the distribution q inside the family $\mathcal{Q}$, consider picking q from the *plate-amortized* family $\mathcal{Q}_{\text{pa}}$ corresponding to $\mathcal{Q}$. Distributions in $\mathcal{Q}_{\text{pa}}$ are distributions from $\mathcal{Q}$ with the additional constraints that some weights have to be equal. Consequently, $\mathcal{Q}_{\text{pa}}$ is a subset of $\mathcal{Q}$:

$$\mathcal{Q}_{\text{pa}} \subset \mathcal{Q} \tag{A.9}$$

As such, looking for the optimal distribution inside $\mathcal{Q}_{\text{pa}}$ instead of inside $\mathcal{Q}$ cannot result in better performance, leading to a *plate amortization gap*:

$$\begin{aligned} \mathcal{G}_{\text{pa}}(\mathcal{Q}, \mathcal{Q}_{\text{pa}}; \psi^*, \phi^*) &= \mathcal{G}_{\text{approx}}(\mathcal{Q}_{\text{pa}}; \psi^*) - \mathcal{G}_{\text{approx}}(\mathcal{Q}; \phi^*) \\ &\geq 0 \\ \mathcal{G}_{\text{PAVI-F}} &= \mathcal{G}_{\text{approx}} + \mathcal{G}_{\text{pa}} \end{aligned} \tag{A.10}$$

Where we denote as ψ^* the optimal weights for a variational distribution q inside $\mathcal{Q}_{\text{pa}}$ –in the KL sense. The equation A.10 is valid for the PAVI-F scheme –see section 2.4. We can interpret it as follow: "theoretically, plate amortization cannot be beneficial in terms of KL divergence for the inference over a given sample $\mathbf{X}$".

Now consider that encodings are no longer free parameters but the output of an encoder f. Similar to the case presented in eq. (A.8), using an encoder cannot result in better performance, leading to an *encoder gap*:

$$\begin{aligned} \mathcal{G}_{\text{encoder}}(\mathcal{Q}_{\text{pa}}, \mathcal{F}; \psi^*, \eta^*) &= \mathcal{G}_{\text{approx}}(\mathcal{Q}_{\text{pa}}; f(\mathbf{X}, \eta^*)) - \mathcal{G}_{\text{approx}}(\mathcal{Q}_{\text{pa}}; \psi^*) \\ &\geq 0 \\ \mathcal{G}_{\text{PAVI-E}} &= \mathcal{G}_{\text{approx}} + \mathcal{G}_{\text{pa}} + \mathcal{G}_{\text{encoder}} \end{aligned} \tag{A.11}$$

The equation $eq.$ (A.11) is valid for the PAVI-E scheme –see section 2.4.

The most complex case is the PAVI-E(sa) scheme, where we combine both plate and sample amortization. Our argument cannot account for the resulting $\mathcal{G}_{\text{PAVI-E(sa)}}$ gap: both the PAVI-E and PAVI-E(sa) schemes rely upon the same encoder f. In the PAVI-E scheme, f is overfit over a dataset composed of the slices of a given data sample $\mathbf{X}$. In the PAVI-E(sa) scheme, the encoder is trained over the whole distribution of the samples of the reduced model $\mathcal{M}^{\text{redu}}$. Intuitively, it is likely that the performance of PAVI-E(sa) will always be dominated by the performance of PAVI-E, but –as far as we understand it– the gap terminology cannot account for this discrepancy.

Comparing previous equations, we therefore have:

$$\mathcal{G}_{\text{vanilla VI}} \leq \mathcal{G}_{\text{PAVI-F}} \leq \mathcal{G}_{\text{PAVI-E}} \tag{A.12}$$

Note that those are *theoretical* results, that do not necessarily pertain to optimization in practice. In particular, in section 4.1&4.3, this theoretical performance loss is not observed empirically over the studied examples. On the contrary, in practice our results can actually be better than non-amortized baselines, as is the case for the PAVI-F scheme in fig. 4. We interpret this as a result of a simplified optimization problem due to plate amortization –with fewer parameters to optimize for, and mini-batching effects across different ground RVs. A better framework to explain those discrepancies could be the one from Bottou & Bousquet (2007): performance in practice is not only the reflection of an *approximation error*, but also of an *optimization error*. A less expressive architecture –using plate amortization– may in practice yield better performance. Furthermore, for the experimenter, the theoretical gaps $\mathcal{G}_{\text{pa}}, \mathcal{G}_{\text{encoder}}$ are likely to be well "compensated for" by the lighter parameterization and faster convergence entitled by plate amortization.

B Supplemental results

B.1 GRE results sanity check

As exposed in the introduction of section 4, in this work we focused on the usage of the ELBO as an inference performance metric (Blei et al., 2017):

$$\mathrm{ELBO}(q) = \log p(X) - \mathrm{KL}(q(\Theta)||p(\Theta|X)) \tag{B.13}$$

Given that the likelihood term $\log p(X)$ does not depend on the variational family q, differences in ELBOs directly transfer in differences in KL divergence, and provide with a straightforward metric to compare different variational posteriors. Nonetheless, the ELBO doesn't provide with an absolute metric of quality. As a sanity check, we want to assert the quality of the results presented in section 4.3 –that are transferable to section 4.1&4.2, based on the same model. In fig. B.1 we plot the posterior samples of various methods against analytical ground truths, using the $\mathrm{Card}^{\mathrm{full}}(\mathcal{P}_1) = 20$ case. All the method's results are aligned with the analytical ground truth, with differences in ELBO translating meaningful qualitative differences in terms of inference quality.

B.2 Experimental details - analytical examples

All experiments were performed in Python, using the *Tensorflow Probability* library (Dillon et al., 2017). Throughout this section we refer to *Masked Autoregressive Flows* (Papamakarios et al., 2018) as *MAF*. All experiments are performed using the Adam optimizer (Kingma & Ba, 2015). At training, the ELBO was estimated using a Monte Carlo procedure with 8 samples. All architectures were evaluated over a fixed set of 20 samples $\mathbf{X}$, with 5 seeds per sample. Non-sample-amortized architectures were trained and evaluated on each of those points. Sample amortized architectures were trained over a dataset of $20,000$ samples separate from the 20 validation samples, then evaluated over the 20 validation samples.

B.2.1 Plate amortization and convergence speed (4.1)

All 3 architectures (baseline, PAVI-F, PAVI-E) used:

- for the flows $\mathcal{F}_i$, a MAF with $[32, 32]$ hidden units;
- as encoding size, 128

For the encoder f in the PAVI-E scheme, we used a multi-head architecture with 4 heads of 32 units each, 2 ISAB blocks with 64 inducing points.

B.2.2 Impact of encoding size (4.2)

All architectures used:

- for the flows $\mathcal{F}_i$, a MAF with $[32, 32]$ hidden units, after an affine block with triangular scaling matrix.
- as encoding size, a value varying from 2 to 16

B.2.3 Scaling with plate cardinalities (4.3)

ADAVI (Rouillard & Wassermann, 2022) we used:

- for the flows $\mathcal{F}_i$, a MAF with $[32, 32]$ hidden units, after an affine block with triangular scaling matrix.
- for the encoder, an encoding size of 8 with a multi-head architecture with 2 heads of 4 units each, 2 ISAB blocks with 32 inducing points.

Cascading Flows (Ambrogioni et al., 2021b) we used:

- a mean-field distribution over the auxiliary variables r
- as auxiliary size, a fixed value of 8

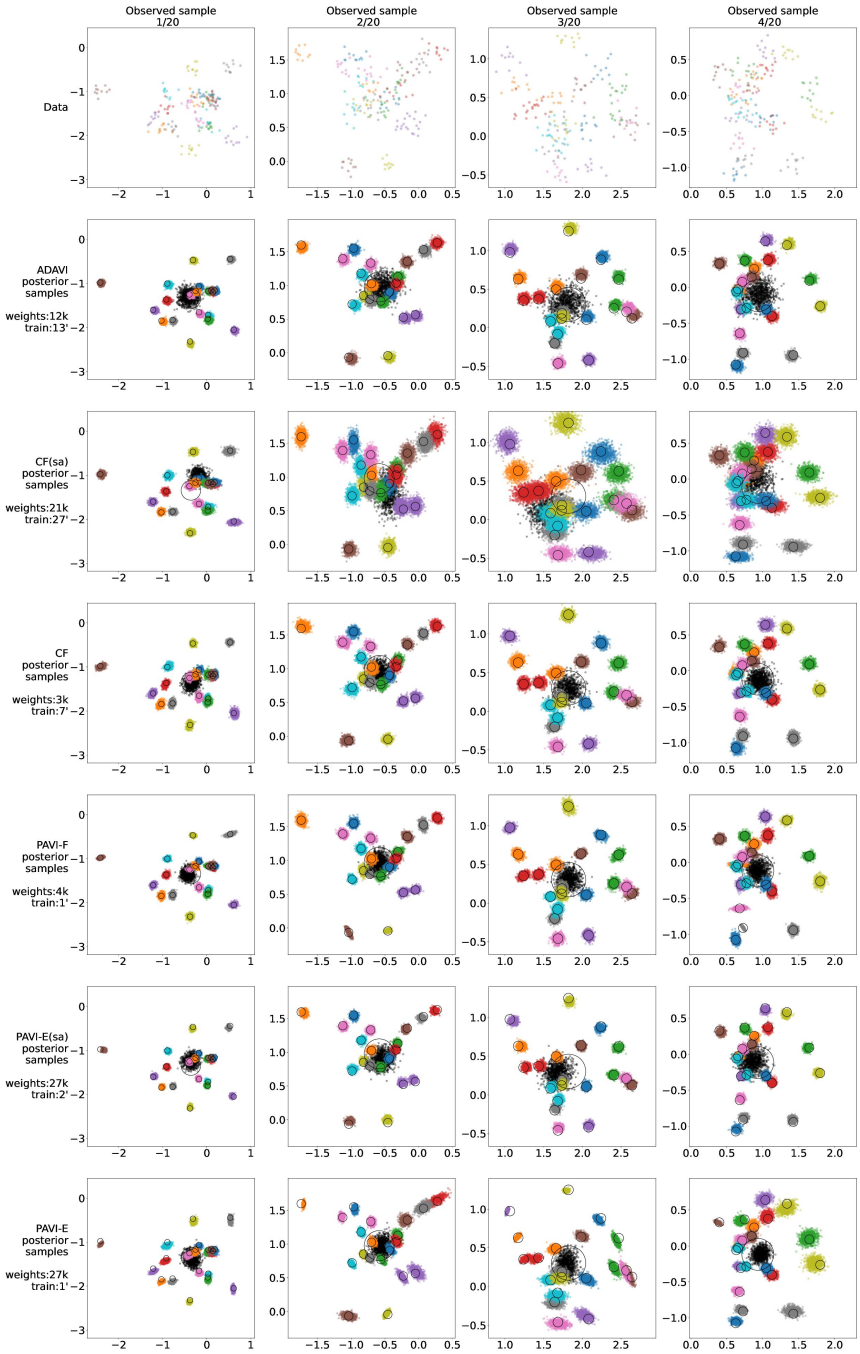

Figure B.1: **GRE Sanity check** Inference methods present qualitatively correct results, making ELBO comparisons relevant in our experiments. *On the topmost line*, we represent 4 different $\mathbf{X}$ samples for the GRE model described in eq. (6) with $\mathrm{Card}^{\mathrm{full}}(\mathcal{P}_1) = 20$. Each set of colored points represent the $\mathbf{X}_{n_1,\bullet}$ points belonging to one of the 20 groups. *Bottom lines* represent the posterior samples for the methods used in section 4.3. Colored points are sampled from the posterior of the groups means θ_1, whereas black points are samples from the population mean θ_2. We represent as black circles an analytical ground truth, centered on the correct posterior mean, and with a radius equal to 2 times the analytical posterior's standard deviation. **Correct posterior samples should be centered on the same point as the corresponding black circle, and 95% of the points should fall within the black circle**. PAVI is represented on the 3 last lines, where we can observed a superior quality for the PAVI-F scheme, rivaling ADAVI and CF's performance with orders of magnitude less parameters and training time, as visible in fig. 4.

- as flows, *Highway Flows* as designed by the Cascading Flows authors

PAVI-F we used:

- for the flows $\mathcal{F}_i$, a MAF with $[32, 32]$ hidden units, after an affine block with triangular scaling matrix.
- an encoding size of 8

PAVI-E we used:

- for the flows $\mathcal{F}_i$, a MAF with $[32, 32]$ hidden units, after an affine block with triangular scaling matrix.
- for the encoder, an encoding size of 16 with a multi-head architecture with 2 heads of 8 units each, 2 ISAB blocks with 64 inducing points.

B.3 Details about our Neuroimaging experiment (4.4)

B.3.1 Data description

In this experiment we use data from the *Human Connectome Project (HCP)* dataset (Van Essen et al., 2012). We randomly select a cohort of $S = 1,000$ subjects from this dataset, each subject being associated with $T = 2$ resting state fMRI sessions (Smith et al., 2013). We minimally pre-process the signal using the nilearn python library (Abraham et al., 2014):

1. removing high variance confounds
2. detrending the data
3. band-filtering the data (0.01 to 0.1 Hz), with a repetition time of 0.74 seconds
4. spatially smoothing the data with a 4mm Full-Width at Half Maximum

For every subject, we extract the surface Blood Oxygenation Level Dependent (BOLD) signal of $N = 314$ vertices corresponding to an average Broca's area (Heim et al., 2009). We compare this signal with the extracted signal of $D = 64$ DiFuMo components: a dictionary of brain spatial maps allowing for an effective fMRI dimensionality reduction (Dadi et al., 2020). Specifically, we compute the one-to-one Pearson's correlation coefficient of every vertex with every DiFuMo component. The resulting connectome, with S subjects, T sessions, N vertices and a connectivity signal with D dimensions, is of shape $(S \times T \times N \times D)$. We project this data –correlation coefficients lying in $]-1; 1[$– in an unbounded space using an inverse sigmoid function.

B.3.2 Model description

We use a model inspired from the work of Kong et al. (2019). We hypothesize that every vertex in Broca's area belongs to either one of $L = 2$ functional networks. This functional bi-partition would reflect the anatomical partition between *pars opercularis* and *pars triangularis* (Heim et al., 2009; Zhang et al., 2020).

Each network is a pattern of connectivity with the brain cortex, represented as a the correlation of the BOLD signal with the signal from the $D = 64$ DiFuMo components. We define $L = 2$ such functional networks at the population level, that correspond to some "average" across the cohort of subjects. Every subject has an individual connectivity, and therefore individual $L = 2$ networks, that are considered as a Gaussian perturbation of the population networks, with variance ϵ. The connectivity of a given subject also evolves through time, giving rise to session-specific networks, that are a Gaussian perturbation of the subject networks with variance σ. Finally, every vertex in Broca's area has its individual connectivity, and is a perturbation of one network's connectivity or the other's. We model this last step as a Gaussian mixture distribution with variance κ. We explicitly model the label label of a given vertex, and we consider this label constant across sessions.

The resulting model can be described as:

$$
\begin{aligned}
S^{\text{full}}, T^{\text{full}}, N^{\text{full}}, D, L &= 1000, 2, 314, 64, 2 \\
s^-, s^+ &= -6, 0 \\
\forall l{=}1..L: \quad \mu_l &\sim \text{Uniform}(-4 \times \vec{1}_D, 4 \times \vec{1}_D) \\
\forall l{=}1..L: \quad \log \epsilon_l &\sim \text{Uniform}(s^- \times \vec{1}_D, s^+ \times \vec{1}_D) \\
\forall l{=}1..L, \forall s{=}1..S: \quad \mu_{l,s} | \mu_l, \epsilon_l &\sim \mathcal{N}(\mu_l, \epsilon_l) \\
\forall l{=}1..L: \quad \log \sigma_l &\sim \text{Uniform}(s^- \times \vec{1}_D, s^+ \times \vec{1}_D) \\
\forall l{=}1..L, \forall s{=}1..S, \forall t{=}1..T: \quad \mu_{l,s,t} | \mu_{l,s}, \sigma_l &\sim \mathcal{N}(\mu_{l,s}, \sigma_l) \\
\forall l{=}1..L: \quad \log \kappa_l &\sim \text{Uniform}(s^- \times \vec{1}_D, s^+ \times \vec{1}_D) \\
\forall s{=}1..S, \forall n{=}1..N: \quad \text{probs}_{s,n} &\sim \text{Dirichlet}(1 \times \vec{1}_L) \\
\forall s{=}1..S, \forall n{=}1..N: \quad \text{label}_{s,n} \,|\, \text{probs}_{s,n} &\sim \text{Categorical}(\text{probs}_{s,n}) \\
\forall s{=}1..S, \forall t{=}1..T, \forall n{=}1..N: \quad X_{s,t,n} | [\mu_{l,s,t}]_{l=1..L}, [\kappa_l]_{l=1..L}, \text{label}_{s,n} &\sim \mathcal{N}(\mu_{\text{label}_{s,n},s,t}, \kappa_{\text{label}_{s,n}})
\end{aligned}
\tag{B.14}
$$

The model contains 4 plates: the *network* plate of full cardinality L (that we did not exploit in our implementation), the *subject* plate of full cardinality S^{full}, the *session* plate of full cardinality T^{full} and the *vertex* plate of full cardinality N^{full}.

Our goal is to recover the posterior distribution of the networks μ –represented as networks in fig. 5– and the labels label –represented as the parcellation in fig. 5– given the observed connectome described in appendix B.3.1.

B.3.3 PAVI implementation

We used in this experiment the PAVI-F scheme, using:

- for the RVs $\mu_l, \mu_{l,s}, \mu_{l,s,t}$:
  - for the flows $\mathcal{F}_i$, a MAF with $[128, 128]$ hidden units, following an affine block with diagonal scale
  - for the encoding size: 128
- for the RVs $\epsilon_l, \sigma_l, \kappa_l, \text{probs}_{s,n}, \text{labels}_{s,n}$:
  - for the flows $\mathcal{F}_i$, a MAF with $[8, 8]$ hidden units, following an affine block with diagonal scale
  - for the encoding size: 8
- for the reduced model, we used $S^{\text{redu}} = 30$, $T^{\text{redu}} = 1$ and $N^{\text{redu}} = 32$.

To allow for the optimization over the discrete $\text{label}_{s,n}$ RV, we used the Gumbell-Softmax trick, using a fixed temperature of 1.0 (Jang et al., 2017; Maddison et al., 2016).

C Supplemental discussion

C.1 Plate amortization as a generalization of sample amortization

In section 2.2 we introduced plate amortization as the application of the generic concept of amortization to the granularity of plates. Taking a step back, there is actually an even stronger connection between sample amortization and plate amortization.

A HBM p models the distribution of a given observed RV X –jointly with the parameters Θ. Different samples $\mathbf{X}_0, \mathbf{X}_1, ...$ of the model p are i.i.d. draws from the distribution $p(X)$. p can thus be considered as the model for "one sample". Consider, instead of p, a "macro" model for the whole *population* of samples one could draw from p. The observed RV of that macro model would be the infinite collection of samples drawn from the same distribution $p(X)$. In that light, the i.i.d. sampling of different X values from p could be interpreted as a plate of the macro model. Thus, we could consider

sample amortization as a instance of plate amortization for the "sample plate". Or equivalently: plate amortization can be seen as the natural generalization of amortization beyond the particular case of sample amortization.

C.2 Alternate formalism for SVI – PAVI-E(sa) scheme

In this work, we propose a different formalism for SVI, based around the concept of full HBM $\mathcal{M}^{\text{full}}$ versus reduced HBM $\mathcal{M}^{\text{redu}}$ sharing the same template $\mathcal{T}$. This formalism is helpful to set up GPU-accelerated stochastic VI (Dillon et al., 2017), as it entitles a fixed computation graph -with the cardinality of the reduced model $\mathcal{M}^{\text{redu}}$- in which encodings are "plugged in" -either sliced from larger encoding arrays or as the output of an encoder applied to a data slice, see section 2.4&3.2. Particularly, our formalism doesn't entitle a control flow over models and distributions, which can be hurtful in the context of *compiled* computation graphs such as in *Tensorflow* (Abadi et al., 2015).

The reduced model formalism is also meaningful in the PAVI-E(sa), where we train and amortized variational posterior over $\mathcal{M}^{\text{redu}}$ and obtain "for free" a variational posterior for the full model $\mathcal{M}^{\text{full}}$ –see section 3.2. In this context, our scheme is no longer a different take on hierarchical, batched SVI: the cardinality of the full model is truly independent from the cardinality of the training, and is only simulated as a scaling factor in the stochastic training –see section 3.1. We have the intuition that fruitful research directions could stem from this concept.

C.3 Benefiting from structure in inference

Conceptually, all our contributions can be abstracted through the notion of plate amortization -see section 2.2. Plate amortization is particularly useful in the context of heavily parameterized density approximators such as normalizing flows, but is not tied to it: plate-amortized Mean Field (Blei et al., 2017) or ASVI (Ambrogioni et al., 2021a) schemes are also possible to use. Plate amortization can be viewed as the amortization of common density approximators across different sub-structures of a problem. This general concept could have applications in other highly-structured problem classes such as graphs or sequences (Wu et al., 2020; Salehinejad et al., 2018).

C.4 Towards user-friendly Variational Inference

By re-purposing the concept of amortization at the plate level, our goal is to propose clear computation versus precision trade-offs in VI. Hyper-parameters such as the encoding size –as illustrated in fig. 3 (right)– allow to clearly trade inference quality in exchanged for a reduced memory footprint. On the contrary, in classical VI, changing $\mathcal{Q}$'s parametric form –for instance switching from Gaussian to Student distributions– can have a strong and complex impact both on number of weights and inference quality (Blei et al., 2017). By allowing the usage of normalizing flows in very large cardinality regimes, our contribution aims at de-correlating approximation power and computational feasibility. In particular, having access to expressive density approximators for the posterior can help experimenters diversify the proposed HBMs, removing the need of properties such as conjugacy to obtain meaningful inference (Gelman et al., 2004). Combining clear hyper-parameters and scalable yet universal density approximators, we tend towards a user-friendly methodology in the context of large population studies VI.