PAVI: PLATE-AMORTIZED VARIATIONAL INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Given 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. 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 analyse 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 measurements per subject. For instance in 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 model parameters Θ, we want to recover the parameters 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 our applications we focus on the case where p corresponds to a Hierarchical Bayesian Model (HBM) (Gelman et al., 2004). In the 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 tackle through our method.

To perform Bayesian inference, several methods exist in the literature. Earliest works resorted to 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). The idea of amortizing a variational family's parameterization across samples is pervasive in VI (Zhang et al., 2019) and can be generalized to hierarchical setups for instance in meta-learning (Iakovleva et al., 2020; Yao et al., 2019). Agrawal & Domke (2021) studied VI in the case of 2-level HBMs, providing with theoretical guarantees for performance. In contrast, our focus is rather computational, and we study generic HBMs, aiming at tackling large population studies efficiently.

Modern VI is effective in low-dimensional settings, but does not scale up to large population studies –that can involve up to millions of random variables.. In this work we identify and tackle 2 challenges to enable this scale-up. A first challenge with scalability is a detrimental trade-off between expressivity and high-dimensionality (Rouillard & Wassermann, 2022). To reduce the bias in inference, 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. Instead of a lengthy search for a valid family, one can resort to universal density approximators: normalizing flows (Papamakarios et al., 2019). But the cost for this generality is a heavy parameterization, and normalizing flows 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. 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. While ADAVI tackled the over-parameterization of VI in population studies, it still could not perform inference in very large data regimes. This is due to a second challenge with scalability: as the size of Θ increases, the evaluation of a single gradient over the entirety of an architecture's weights quickly requires too much memory and compute. This second challenge can be overcome using stochastic VI (Hoffman et al., 2013), which subsamples the parameters Θ inferred for at each optimization step. However, using SVI, the weights for the posterior for 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. We detail two variants of our method, with different trade-offs between parameterization and inference quality. 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 Hierarchical Bayesian Models (HBMs). These HBMs feature samples from common conditional distribution at multiple levels. For instance subject measurements are i.i.d. given subject parameters, which are i.i.d. given population parameters. Figure 1 displays two toy instances of this i.i.d. sampling. Due to data being collected across hundreds of subjects (Kong et al., 2019), HBMs representing population studies feature thousands of RVs, most of which corresponding to symmetrical distributions. For instance, the distribution of a subject's measures given the subjects latent parameters is the same same across all subjects. As a result, those large HBMs $\mathcal{M}$ can be compactly represented via plate-enriched Directed Acyclic Graph (DAG) templates $\mathcal{T}$ (Gilks et al., 1994; Koller & Friedman, 2009). The plates in the template $\mathcal{T}$ translate graphically the hierarchical i.i.d. structure in the generative model $\mathcal{M}$. Our method exploits this plate-induced repeated structure to design scalable inference methods.

We denote $\mathcal{T}$'s vertices –corresponding to continuous RV templates– as X and $\Theta = \{\theta_i\}_{i=1..I}$. X corresponds to the sets of RVs observed during inference, and Θ to the parameters we want to infer: our goal is to closely approximate the posterior distribution $p(\Theta|X)$. We denote $\mathcal{T}$'s plates as $\{\mathcal{P}_p\}_{p=0..P}$, and the plates θ_i belongs to as $\mathrm{Plates}(\theta_i)$. In the toy example from 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 $p(\theta_1, \theta_2|X)$.

We build upon the interplay between the template $\mathcal{T}$ and the HBM $\mathcal{M}$. To go from one representation to the other, $\mathcal{T}$ can be *grounded* into $\mathcal{M}$ given plate cardinalities $\{\mathrm{Card}(\mathcal{P}_p)\}_{p=0..P}$ (Koller &

Friedman, 2009). $\mathrm{Card}(\mathcal{P})$ represents the number of elements in the plate $\mathcal{P}$, for instance the number of subjects in the study. The *grounding* operation instantiates the repeated structures symbolized by the plates: a given RV *template* θ_i corresponds to multiple *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 models $\mathcal{M}$. In our method, we consider the inference over the ground RVs $\theta_{i,n}$ as symmetrical problems, over which we share parameterization and learning.

Our goal is to perform inference over a target –or *full*– model denoted $\mathcal{M}^{\text{full}}$. This model typically features large plate cardinalities $\mathrm{Card}^{\text{full}}(\mathcal{P})$, and therefore many ground RVs to infer over, making it computationally intractable. Instead of tackling inference directly for this model, we will train over a smaller replica of $\mathcal{M}^{\text{full}}$. To this end, we instantiate the template $\mathcal{T}$ corresponding to $\mathcal{M}^{\text{full}}$ 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^{full} usable to approximate the posterior for the intractable target model $\mathcal{M}^{\text{full}}$.

2.2 Full variational family design

Here we define the expressive variational distribution q^{full} –corresponding to the model $\mathcal{M}^{\text{full}}$– introduced in the previous section. This is the distribution we will use for inference, but that –due to $\mathcal{M}^{\text{full}}$'s large cardinalities– we cannot directly train over. To derive q^{full}, we push forward the prior $p(\Theta)$ using trainable normalizing flows, denoted as $\mathcal{F}$ (Papamakarios et al., 2019). In the full HBM $\mathcal{M}^{\text{full}}$, the plate structure indicates that a RV template θ_i is associated to a unique conditional distribution p_i shared across all ground RVs $\theta_{i,n}$:

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

where $\pi(\theta_{i,n})$ are the parents of the RV $\theta_{i,n}$ –conditioning its distribution. We denote with a $\bullet_X$ index all variables related to the observed RVs X. To every ground RV $\theta_{i,n}$, we associate a normalizing flow $\mathcal{F}_{i,n}$. This learnable flow will be trained to approximate the posterior distribution of $\theta_{i,n}$:

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

where the distribution $q_{i,n}$ is the push-forward of the prior distribution p_i through the normalizing flow $\mathcal{F}_{i,n}$. 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. More details about the conditional dependencies modelled in the variational distribution can be found in appendix C.4.

2.3 Plate amortization

In this section we introduce the notion of plate amortization: sharing the parameterization of conditional density estimators across a model's plates to reduce the parameterization of our variational family. Traditional VI searches for a parametric distribution $q(\Theta;\phi)$ that will best approximate the posterior distribution of Θ given a value $\mathbf{X}_0$ for 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) instead performs 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*.

Similar to 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. Instead of casting every flow $\mathcal{F}_{i,n}$ –defined in eq. (2)– as an individual, fully-parameterized flow, we will share some parameters across the different $\mathcal{F}_{i,n}$. To the template θ_i, we will associate a conditional flow $\mathcal{F}_i(\,\cdot\,;\phi_i,\bullet)$ with weights ϕ_i shared across all the ground RVs $\{\theta_{i,n}\}_{n=0..N_i}$. The flow $\mathcal{F}_{i,n}$ associated to a given ground RV $\theta_{i,n}$ will be an instance of this conditional flow, conditioned by an encoding $\mathbf{E}_{i,n}$:

$$\mathcal{F}_{i,n} = \mathcal{F}_i(\,\cdot\,;\phi_i,\mathbf{E}_{i,n}) \quad \text{yielding} \quad q_{i,n} = q_{i,n}(\theta_{i,n}|\pi(\theta_{i,n});\phi_i,\mathbf{E}_{i,n}) \tag{3}$$

The 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. Using conditional normalizing flows $\mathcal{F}_i$, 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.

2.4 Encoding schemes

The distributions $q_{i,n}(\theta_{i,n}|\pi(\theta_{i,n});\phi_i,\mathbf{E}_{i,n})$ –defined in eq. (3)– with different ground index n only vary through the value of the encodings $\mathbf{E}_{i,n}$. We detail one core and one alternate scheme to derive those encodings:

Free plate encodings (PAVI-F) In our core implementation, $\mathbf{E}_{i,n}$ are free weights. We define encoding 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. This means that an additional ground RV –for instance adding a subject in a population study– requires an additional encoding vector. The associated increment in the total number of weights in the architecture is much 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 can be kept as an efficient warm start, the optimal values for the encodings $\mathbf{E}_{i,n}$ have to be searched again.

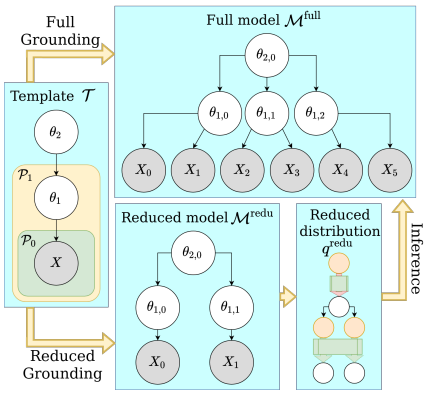

Figure 1: **Plate Amortized Variational Inference (PAVI) working principle**. Starting on the left, the graph template $\mathcal{T}$ is grounded into 2 separate HBMs: $\mathcal{M}^{\text{full}}$ (top) and $\mathcal{M}^{\text{redu}}$ (down) of respective plate cardinalities $(3,2)$ –large– and $(2,1)$ –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.

Deep set encoder (PAVI-E) The parameterization of PAVI-F scales lightly but linearly with $\text{Card}^{\text{full}}(\mathcal{P})$. This scaling could still become unaffordable in large population studies. We therefore propose an alternate scheme –PAVI-E– with a parameterization independent of the cardinalities of the model. In this more experimental scheme, the free encodings are replaced by an encoder f –with weight η– applied to the observed data $\mathbf{X}$: $\mathbf{E} = f(\mathbf{X};\eta)$. 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). This scheme allows for *sample amortization* across different data samples $\mathbf{X}_0, \mathbf{X}_1, \ldots$ –see section 2.3. Note that an encoder will be used to generate the encodings whether the inference is sample amortized or not.

Summary We have defined the distribution 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 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

Because of the large number of ground RVs in $\mathcal{M}^{\text{full}}$, optimizing over the full distribution q^{full} –defined in eq. (2)– is computationally intractable. Instead, we will optimize over a distribution that has the smaller cardinalities of the reduced model $\text{Card}^{\text{redu}}(\mathcal{P})$. At each optimization step t, we will randomly choose 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 $\Theta^{\text{redu}}[t]$ corresponding to the branching 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:

$$\log q^{\text{redu}}(\Theta^{\text{redu}}[t]) = \sum_{i=1}^{I} \frac{N_i^{\text{full}}}{N_i^{\text{redu}}} \sum_{n \in \mathcal{B}_i^{\text{redu}}[t]} \log q_{i,n}(\theta_{i,n} | \pi(\theta_{i,n})) \tag{4}$$

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:

$$\log p^{\text{redu}}(X^{\text{redu}}[t], \Theta^{\text{redu}}[t]) = \frac{N_X^{\text{full}}}{N_X^{\text{redu}}} \sum_{n \in \mathcal{B}_X^{\text{redu}}[t]} \log p_X(x_n | \pi(x_n)) + \sum_{i=1}^{I} \frac{N_i^{\text{full}}}{N_i^{\text{redu}}} \sum_{n \in \mathcal{B}_i^{\text{redu}}[t]} \log p_i(\theta_{i,n} | \pi(\theta_{i,n}))$$
$$\text{ELBO}^{\text{redu}}[t] = \mathbb{E}_{\Theta^{\text{redu}} \sim q^{\text{redu}}} \left[\log p^{\text{redu}}(X^{\text{redu}}[t], \Theta^{\text{redu}}[t]) - \log q^{\text{redu}}(\Theta^{\text{redu}}[t]) \right] \tag{5}$$

This scheme can be viewed as the instantiation of $\mathcal{M}^{\text{redu}}$ over batches of $\mathcal{M}^{\text{full}}$'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. This training is analogous to the usage of stochastic VI (Hoffman et al., 2013), 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.

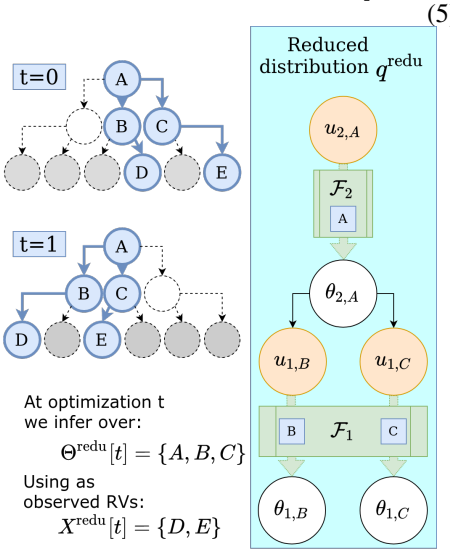

Figure 2: **PAVI stochastic training scheme** The reduced distribution q^{redu} features 2 conditional normalizing flows $\mathcal{F}_1$ and $\mathcal{F}_2$ respectively associated to the RV templates θ_1 and θ_2. During the stochastic training, q^{redu} is instantiated over different branchings of the 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. 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

Here we detail how our shared parameterization –detailed in section 2.3– combined with our stochastic training scheme, results in faster inference. 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 $\theta_{i,n}$. Due to the problem's plate structure, we consider the inference over those ground RVs as different instances of a common

density estimation task. With PAVI architectures, 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. This means that most of the weights of the flows $\mathcal{F}_{i,n}$ –concentrated in ϕ_i– are trained at every optimization step, across all the selected batches $\mathcal{B}_i^{\text{redu}}[t]$. This results in drastically faster convergence compared to SVI –as seen in exp. 4.1. PAVI-F and PAVI-E share this common property, but differ in the way they deal with the encodings $\mathbf{E}_{i,n}$:

PAVI-F The encodings $\mathbf{E}_{i,n}$ corresponding to $n \in \mathcal{B}_i^{\text{redu}}[t]$ are sliced from larger encoding arrays $\mathbf{E}_i$ and are optimized for along with the weights ϕ_i. In the toy example from fig. 2, at $t = 0$, $\mathcal{B}_1^{\text{redu}}[0] = \{1, 2\}$ and the trained encodings are $\{\mathbf{E}_{1,1}, \mathbf{E}_{1,2}\}$, and at $t = 1$ $\mathcal{B}_1^{\text{redu}}[1] = \{0, 1\}$ and the trained encodings are $\{\mathbf{E}_{1,0}, \mathbf{E}_{1,1}\}$. In appendix A.2, we show that the stochastic training of PAVI-F is unbiased: training over $\mathcal{M}^{\text{redu}}$, we converge to the same variational distribution as if training directly over $\mathcal{M}^{\text{full}}$.

PAVI-E Shared learning is further amplified as the free encodings $\mathbf{E}_i$ are replaced by the encoder $f(\,\cdot\,; \eta)$. All the architecture's weights –ϕ_i and η– are trained at every optimization step t. To collect the encodings to plug into q^{redu}, we build up on a property our encoder f: *set size generalization* (Zaheer et al., 2018). Instead of encoding the full-sized data $\mathbf{X}$, the encoder is applied to the reduced-sized slice $\mathbf{X}^{\text{redu}}[t]$. This amounts to aggregating summary statistics across a subset of the observed data (Lee et al., 2019; Agrawal & Domke, 2021). This property is intensified in the sample amortized context: we train a sample amortized variational family over the lightweight model $\mathcal{M}^{\text{redu}}$, and use it "for free" to infer over the heavyweight model $\mathcal{M}^{\text{full}}$. As detailed in appendix A.2, in this work we rely on a computationally efficient but theoretically biased training scheme for PAVI-E. The negative impact of this bias on PAVI-E's performance was seldom noticeable and always marginal throughout our experiments.

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 the variational posterior q^{full}, as demonstrated in the following experiments.

4 Results and Discussion

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 appendix B.1 we provide with sanity checks to assess the quality of the obtained results. We also evaluate the impact of the reduced model cardinalities on performance in appendix B.2, along with more comparative experiments in appendix B.3.

Experimentally, we have found the PAVI-F scheme overall faster to train and yielding better inference quality compared to the PAVI-E scheme. When its parameterization is affordable, PAVI-F should be preferred. PAVI-E nevertheless opens up promising research directions, with the potential for parameterization-constant, time-constant sample-amortized inference with respect to the cardinality of a problem. Though degraded compared to PAVI-F's performance, PAVI-E's performance is still on par or beats baselines in a variety of inference tasks –see exp. 4.3&B.3.

4.1 Plate amortization and convergence speed

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

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

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

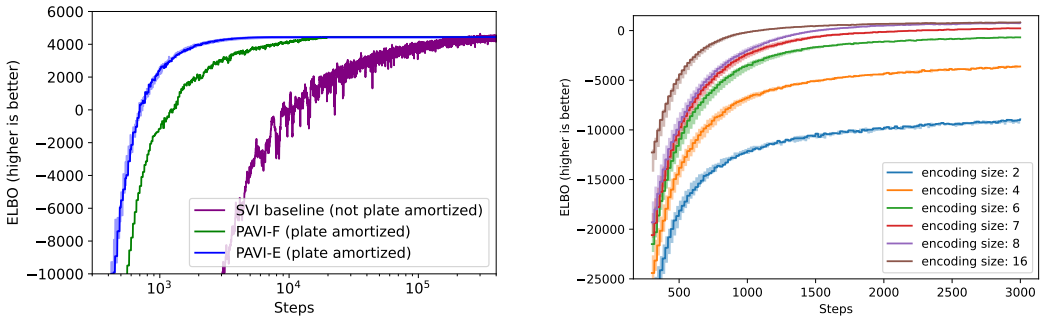

Figure 3: **Left panel: Plate amortization increases convergence speed** Plot of the ELBO (higher is better) as a function of the optimization steps (log-scale) for our methods PAVI-F (in green) and PAVI-E (in blue) versus a non-plate-amortized baseline (in purple). Due to plate amortization, our method converges ten to a hundred times 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.

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., 2013). The only difference is that every ground RV $\theta_{i,n}$ is associated in the baseline to an individual fully-parameterized flow $\mathcal{F}_{i,n}$ instead of sharing the same conditional 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 both plate amortized methods reach asymptotic ELBO equal to the one of the non-plate-amortized baseline, but with orders of magnitudes faster convergence, and more numerical stability. This stems from the individual flows $\mathcal{F}_{i,n}$ in the baseline 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 in theory a faster convergence –in terms of number of optimization steps– 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, computing the encodings results in steps of longer duration, and ultimately in slower convergence, as illustrated in section 4.3.

4.2 Impact of encoding size

Here we illustrate the role of encodings as ground RV's posterior summary statistics –as described in section 2.3. 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 2.4. 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 unequivocal 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

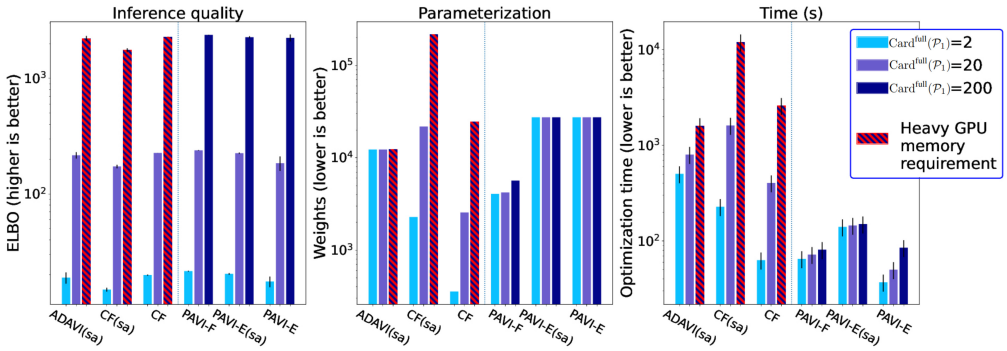

Figure 4: **PAVI provides with favorable parameterization and training time as the cardinality of the target model augments** Our architecture PAVI is displayed on the right of each panel. We augment the cardinality $\text{Card}^{\text{full}}(\mathcal{P}_1)$ of the GRE model –described in eq. (6). While 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.

to drastically reduce the memory footprint of inference while maintaining acceptable performance –choosing the encoding size approximately equal to the expected size of the sufficient statistics.

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 $\Theta : 6 \to 42 \to 402$.

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. CF push the prior p into the posterior q using *Highway Flows*. CF follows a cascading dependency structure, complemented with a backward auxiliary coupling; 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. ADAVI performs a Mean Field approximation (Blei et al., 2017), focuses on pyramidal HBMs only, and is limited to a sample-amortized variant; For all architectures, we indicate with the suffix *(sa) sample amortization*, corresponding to the classical meaning of amortization, as detailed in section 2.3. More implementation details can be found in appendix B.4.

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

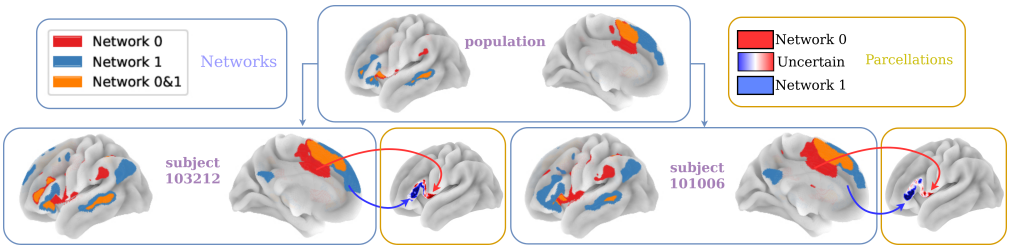

Figure 5: **Probabilistic parcellation of Brocas's area** PAVI can be applied in the challenging context of Neuroimaging population studies. For a cohort of 1000 subjects, 2 of which are represented here –in the bottom 2 items– we present 2 results. First, *connectivity networks* with the brain's left 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.

is trained over a dataset of reduced cardinality, yet performs "for free" inference over a HBM of large cardinality. Maintaining $\mathrm{Card}^{\mathrm{redu}}(\mathcal{P}_1)$ constant while $\mathrm{Card}^{\mathrm{full}}(\mathcal{P}_1)$ 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 $\mathrm{Card}^{\mathrm{full}}(\mathcal{P}_1) = 200$ 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 –$\mathrm{Card}^{\mathrm{full}}(\mathcal{P}_1) = 2000$ 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.

4.4 Neuroimaging application: 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.

REPRODUCIBILITY STATEMENT

All experiments were performed in Python using the Tensorflow Probability library (Dillon et al., 2017). All experiments were conducted 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. Appendix B.4 lists implementation details for all our synthetic experiments. Appendix B.5 is related to our Neuroimaging experiment 4.4, and details both our data pre-processing steps and our implementation. As part of our submission we furthermore packaged and release the code associated to our experiments.

ETHICS STATEMENT

Most of our experiments use synthetic data. Our Neuroimaging data come from the Human Connectome Project dataset (Van Essen et al., 2012). All data in the HCP is strongly anonymized, as per the HCP protocols. We used in this paper only Open Access imaging data data, following the HCP data use terms.

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.

Abhinav Agrawal and Justin Domke. Amortized Variational Inference for Simple Hierarchical Models. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 21388–21399. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/b28d7c6b6aec04f5525b453411ff4336-Paper.pdf.

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 and David M. Blei. Structured Stochastic Variational Inference. *arXiv:1404.4114 [cs]*, November 2014. URL http://arxiv.org/abs/1404.4114. arXiv: 1404.4114.

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.

Ekaterina Iakovleva, Jakob Verbeek, and Karteek Alahari. Meta-learning with shared amortized variational inference. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4572–4582. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/iakovleva20a.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.

Stefan Webb, Adam Golinski, Robert Zinkov, N. Siddharth, Tom Rainforth, Yee Whye Teh, and Frank Wood. Faithful Inversion of Generative Models for Effective Amortized Inference. *arXiv:1712.00287 [cs, stat]*, November 2018. URL http://arxiv.org/abs/1712.00287. arXiv: 1712.00287.

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.

Huaxiu Yao, Ying Wei, Junzhou Huang, and Zhenhui Li. Hierarchically structured meta-learning. 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. 7045–7054. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/yao19b.html.

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 $\mathcal{M}^{\text{full}}$. As an example, if a ground RV is selected as part of $\mathcal{M}^{\text{redu}}$, then its parent RV needs to be selected as well. 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. (4). The same procedure yields the observed RV subset $X^{\text{redu}}[t]$ and the data slice $\mathbf{X}^{\text{redu}}[t]$.

For instance, in the toy example from fig. 1, X_2 will be chosen iff the index 1 is selected as part of sub-sampling $\mathcal{P}_1$ and the index 0 is selected as part of sub-sampling $\mathcal{P}_0$. Less formally, this is equivalent to going *middle*, then *left* in the full graph representing $\mathcal{M}^{\text{full}}$. This stochastic choice is illustrated in fig. 2 at $t = 1$ where X_2 corresponds to the node E. This stochastic strategy also applies to the selected encoding scheme –described in section 2.4&3.2– 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 $f(\,\cdot\,,\eta)$ 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 Analysis of bias in the stochastic training

A key concern in our stochastic training scheme is its unbiasedness: we want our stochastic optimization to converge to the same variational posterior as if we trained over the full model directly –without any stochasticity. In this section we first show that the PAVI-F is unbiased. Second, we identify strategies to obtain an unbiased PAVI-E scheme, yet show how the approximations we do in practice can theoretically result in biased training. As an important note, the negative impact of this bias on the performance PAVI-E remained limited throughout our experiments –as seen in experiments 4.3, B.2, B.1 and B.3.

A.2.1 General derivation (applicable to both the PAVI-F and PAVI-E schemes)

We first formalize the *plate sampling* strategy described in appendix A.1.1. To every plate $\mathcal{P}$ we associate the RV $I_{\mathcal{P}}$ corresponding to the $\text{Card}^{\text{redu}}(\mathcal{P})$-sized set of indices sampled without replacement from the $\text{Card}^{\text{full}}(\mathcal{P})$ possible index values. As an example, with a plate $\mathcal{P}_0$ with $\text{Card}^{\text{full}}(\mathcal{P}_0) = 4$ and $\text{Card}^{\text{redu}}(\mathcal{P}_0) = 2$, $\{0, 2\}$ or $\{2, 3\}$ can be 2 different samples from $I_{\mathcal{P}_0}$. At a given optimization step t, we sample independently from the RVs $\{I_{\mathcal{P}_p}\}_{p=0..P}$. This defines the batches $\mathcal{B}_i^{\text{redu}}[t]$ and the distribution q^{redu} in eq. (4).

To check the unbiasedness of our stochastic training, we need to show that:

$$\mathbb{E}_{I_{\mathcal{P}_0}} \dots \mathbb{E}_{I_{\mathcal{P}_P}} \left[\text{ELBO}^{\text{redu}}[t]\right] = \text{ELBO}^{\text{full}} \tag{A.6}$$

Where:

$$\text{ELBO}^{\text{full}} = \mathbb{E}_{\Theta \sim q^{\text{full}}} \left[\log p^{\text{full}}(X, \Theta) - \log q^{\text{full}}(\Theta)\right] \tag{A.7}$$

And $\text{ELBO}^{\text{redu}}[t]$ is defined in eq. (5). In that expression, q and p have symmetrical roles. As the ELBO amounts to the difference between the logarithms of distributions p and q, we can prove the equality in eq. (A.6) if we prove that the expectation of each reduced distribution is equal to the corresponding full distribution. To prove the equality in eq. (A.6), a sufficient condition is therefore to prove that:

$$\mathbb{E}_{I_{\mathcal{P}_0}} \dots \mathbb{E}_{I_{\mathcal{P}_P}} \left[\log q^{\text{redu}}(\Theta^{\text{redu}}[t])\right] = \mathbb{E}_{I_{\mathcal{P}}} \left[\log q^{\text{redu}}(\Theta^{\text{redu}}[t])\right] = \log q^{\text{full}}(\Theta) \tag{A.8}$$

where to de-clutter the notations we denote the expectation over the collection of RVs $\{I_{\mathcal{P}_p}\}_{p=0..P}$ as $\mathbb{E}_{I_{\mathcal{P}}}$.

Consider a given ground RV $\theta_{i,n}$ corresponding to the RV template θ_i and to the plates $\text{Plates}(\theta_i)$. At a given stochastic step t, $\theta_{i,n}$ will be chosen if and only if its corresponding *branching* is chosen. Recall that when sampling equiprobably without replacement a set of k elements from a population of n elements, a given element will be present in the set with probability k/n. We can apply this reasoning to the choice of *branching* corresponding to a given ground RV. For instance, in fig. 1, X_2 will be chosen iff the index 1 is selected as part of sub-sampling $\mathcal{P}_1$ and the index 0 is selected as part of sub-sampling $\mathcal{P}_0$. As $\text{Card}^{\text{redu}}(\mathcal{P}_1) = 2$ indices are chosen inside the plate $\mathcal{P}_1$ of full cardinality 3, and $\text{Card}^{\text{redu}}(\mathcal{P}_0) = 1$ indices are chosen inside the plate $\mathcal{P}_0$ of full cardinality 2, X_2 is therefore chosen with probability $2/3 \times 1/2$. More formally, for a ground RV $\theta_{i,n}$ we have:

$$\begin{aligned}\forall n = 0..N_i^{\text{full}} \ : \ \mathbb{P}(\theta_{i,n} \in \mathcal{B}_i^{\text{redu}}[t]) &= \prod_{\mathcal{P} \in \text{Plates}(\theta_i)} \frac{\text{Card}^{\text{redu}}(\mathcal{P})}{\text{Card}^{\text{full}}(\mathcal{P})} \\ &= \frac{N_i^{\text{redu}}}{N_i^{\text{full}}}\end{aligned} \tag{A.9}$$

Applying this reasoning to every RV template θ_i, we have that:

$$\begin{aligned}\mathbb{E}_{I_{\mathcal{P}}}\left[\log q^{\text{redu}}(\Theta^{\text{redu}}[t])\right] &= \sum_{i=1}^{I} \frac{N_i^{\text{full}}}{N_i^{\text{redu}}} \mathbb{E}_{I_{\mathcal{P}}}\left[\sum_{n \in \mathcal{B}_i^{\text{redu}}[t]} \log q_{i,n}(\theta_{i,n}|\pi(\theta_{i,n}))\right] \\ &= \sum_{i=1}^{I} \frac{N_i^{\text{full}}}{N_i^{\text{redu}}} \mathbb{E}_{I_{\mathcal{P}}}\left[\sum_{n=0}^{N_i^{\text{full}}} \mathbb{1}_{n \in \mathcal{B}_i^{\text{redu}}[t]} \log q_{i,n}(\theta_{i,n}|\pi(\theta_{i,n}))\right] \\ &= \sum_{i=1}^{I} \frac{N_i^{\text{full}}}{N_i^{\text{redu}}} \sum_{n=0}^{N_i^{\text{full}}} \mathbb{E}_{I_{\mathcal{P}}}\left[\mathbb{1}_{n \in \mathcal{B}_i^{\text{redu}}[t]} \log q_{i,n}(\theta_{i,n}|\pi(\theta_{i,n}); \phi_i, \mathbf{E}_{i,n})\right]\end{aligned} \tag{A.10}$$

where we exploited the fact that the expectation of the sum of RVs is the sum of the expectations, even in the case of dependent RVs. The term $\mathbb{1}_{n \in \mathcal{B}_i^{\text{redu}}[t]} \times \log q_{i,n}(\theta_{i,n}|\pi(\theta_{i,n}); \phi_i, \mathbf{E}_{i,n})$ is the product of 2 RVs –related to the stochastic choice of plate indices:

- the RV $\mathbb{1}_{n \in \mathcal{B}_i^{\text{redu}}[t]}$ is an indicator that $\theta_{i,n}$'s *branching* has been chosen via the stochastic sampling of plate indices. By construction, this RV depends only on the indices of the plates $\mathcal{P} \in \text{Plates}(\theta_i)$.
- the RV $\log q_{i,n}(\theta_{i,n}|\pi(\theta_{i,n}); \phi_i, \mathbf{E}_{i,n})$ depends on $\mathbf{E}_{i,n}$, whose construction depends on the encoding scheme:
  - In the PAVI-F scheme, $\mathbf{E}_{i,n}$ is a constant.
  - In the PAVI-E scheme, $\mathbf{E}_{i,n}$ results of the application of an encoder to the observed data of a subset of $\theta_{i,n}$'s descendants. By construction, this subset will only depend on the indices of plates containing θ_i's descendants, but not containing θ_i. The value of $\mathbf{E}_{i,n}$ therefore only depends on the indices of plates $\mathcal{P} \notin \text{Plates}(\theta_i)$

As a consequence –in both PAVI-F and PAVI-E– the terms $\mathbb{1}_{n \in \mathcal{B}_i^{\text{redu}}[t]}$ and $\log q_{i,n}(\theta_{i,n}|\pi(\theta_{i,n}); \phi_i, \mathbf{E}_{i,n})$ are independent, meaning that the expectation of their product can be rewritten as the product of their expectations:

$$\begin{aligned}\mathbb{E}_{I_{\mathcal{P}}}\left[\log q^{\text{redu}}(\Theta^{\text{redu}}[t])\right] &= \sum_{i=1}^{I} \frac{N_i^{\text{full}}}{N_i^{\text{redu}}} \sum_{n=0}^{N_i^{\text{full}}} \mathbb{E}_{I_{\mathcal{P}}}\left[\mathbb{1}_{n \in \mathcal{B}_i^{\text{redu}}[t]}\right] \mathbb{E}_{I_{\mathcal{P}}}\left[\log q_{i,n}(\theta_{i,n}|\pi(\theta_{i,n}))\right] \\ &= \sum_{i=1}^{I} \frac{N_i^{\text{full}}}{N_i^{\text{redu}}} \sum_{n=0}^{N_i^{\text{full}}} \frac{N_i^{\text{redu}}}{N_i^{\text{full}}} \mathbb{E}_{I_{\mathcal{P}}}\left[\log q_{i,n}(\theta_{i,n}|\pi(\theta_{i,n}))\right] \\ &= \sum_{i=1}^{I} \sum_{n=0}^{N_i^{\text{full}}} \mathbb{E}_{I_{\mathcal{P}}}\left[\log q_{i,n}(\theta_{i,n}|\pi(\theta_{i,n}); \mathbf{E}_{i,n})\right]\end{aligned} \tag{A.11}$$

This equality can be further simplified in the PAVI-F case –proving its unbiasedness– but not in the PAVI-E case, as detailed in the sections below.

A.2.2 UNBIASEDNESS OF THE PAVI-F SCHEME

In the PAVI-F scheme, detailed in section 2.4, the encodings $\mathbf{E}_{i,n}$ are constants with respect to the branching choice, therefore we have:

$$\begin{aligned}\mathbb{E}_{I_{\mathcal{P}}}\left[\log q^{\text{redu}}(\Theta^{\text{redu}}[t])\right] &= \sum_{i=1}^{I}\sum_{n=0}^{N_i^{\text{full}}} \mathbb{E}_{I_{\mathcal{P}}}\left[\log q_{i,n}(\theta_{i,n}|\pi(\theta_{i,n});\mathbf{E}_{i,n})\right] \\ &= \sum_{i=1}^{I}\sum_{n=0}^{N_i^{\text{full}}} \log q_{i,n}(\theta_{i,n}|\pi(\theta_{i,n});\mathbf{E}_{i,n}) \\ &= \log q^{\text{full}}(\Theta)\end{aligned} \tag{A.12}$$

which proves eq. (A.8) and eq. (A.6). This demonstrates that the PAVI-F scheme is unbiased: training over stochastically chosen sub-graphs for q^{redu} is in expectation equal to training over the full graph of q^{full}.

A.2.3 APPROXIMATIONS IN THE PAVI-E SCHEME

In the PAVI-E scheme, detailed in section 3.2, the encodings $\mathbf{E}_{i,n}$ are computed from the observed data X. Specifically, considering the ground RV $\theta_{i,n}$, we have $\mathbf{E}_{i,n} = f(\mathbf{X}_{i,n}^{\text{redu}}[t])$ where $\mathbf{X}_{i,n}^{\text{redu}}[t]$ corresponds to the observed data of a subset of $\theta_{i,n}$'s descendants. Depending on the chosen branching *downstream* of $\theta_{i,n}$, the value of $\mathbf{E}_{i,n}$ can therefore vary. This means we cannot further simplify eq. (A.11): the terms $\log q_{i,n}(\theta_{i,n}|\pi(\theta_{i,n});\mathbf{E}_{i,n})$ are not constants with respect to the RVs $I_{\mathcal{P}}$.

How could the PAVI-E scheme be made unbiased? Specifically, by making the value of $\mathbf{E}_{i,n}$ independent of the choice of downstream branching. A possibility would be to parameterize $\mathbf{E}_{i,n}$ as an average –an expectation– over all the possible sub-branchings downstream of $\theta_{i,n}$. Yet, in practical cases, the cardinalities of the reduced model are much inferior to the ones of the full model: $\text{Card}^{\text{redu}}(\mathcal{P}) \ll \text{Card}^{\text{full}}(\mathcal{P})$. This means that numerous $\text{Card}^{\text{redu}}(\mathcal{P})$-sized subsets can be chosen inside the $\text{Card}^{\text{full}}(\mathcal{P})$ possible descendants. In order to average over all those subset choices to compute $\mathbf{E}_{i,n}$, numerous encoding calculations would be required at each stochastic training step. For large-scale cases, we deemed this possibility impractical. Other possibilities could exist, all revolving around the problematic of aggregating collections of stochastic estimators into one general estimator –in an unbiased and efficient manner. To our knowledge, this is a complex and still open research question, whose advancement could much benefit our applications.

Practical approximation for the PAVI-E scheme In practice, we compute the encoding $\mathbf{E}_{i,n}$ based on the single downstream branching corresponding to the sampling of the RVs $I_{\mathcal{P}}$. Compared to the previous paragraph, this amounts to estimating the expectation of $\mathbf{E}_{i,n}$ –over all downstream branchings– using a single one of those branchings. Note that, even if this encoding estimate was unbiased, $\log q_{i,n}$ would remain an highly non-linear function of $\mathbf{E}_{i,n}$. As a consequence, we need to rely on the approximation:

$$\mathbb{E}_{I_{\mathcal{P}}}\left[\log q_{i,n}(\theta_{i,n}|\pi(\theta_{i,n});\phi_i, f(\mathbf{X}_{i,n}^{\text{redu}}[t]))\right] \simeq \log q_{i,n}(\theta_{i,n}|\pi(\theta_{i,n});\phi_i, f(\mathbf{X}_{i,n})) \tag{A.13}$$

which can theoretically introduce some bias in our gradients. The approximation eq. (A.13) can be interpreted as follow: "the expectation of the density of $\theta_{i,n}$ when collecting summary statistics over a stochastic subset of $\theta_{i,n}$'s descendants is approximately equal to the density of $\theta_{i,n}$ when collecting summary statistics over the entirety of $\theta_{i,n}$'s descendants". Another interpretation is that the distribution associated to the summary of the full data can be approximated by an annealing of the distributions associated to summaries of subsets of this data. In practice, this approximation did not yield significantly worse performance for the PAVI-E scheme over the generative models we tested. At the same time, computing the encodings over a single branching allows to compute all the $\mathbf{E}_{i,n}$ encodings in a single lightweight pass over the data $\mathbf{X}^{\text{redu}}[t]$. This simple solution therefore provided with a substantial increase in training speed with seldom noticeable bias. Yet, we do not bar the existence of pathological generative HBMs where this approximation would become coarse.

Experimenters should bear in mind this possibility when using the PAVI-E scheme. In practice, using the PAVI-F scheme as a sanity check over synthetic, toy-dimension implementations of the considered generative models is a good way to validate the PAVI-E scheme –before moving onto the real problem instantiating the same generative model with a larger dimensionality.

A.3 PAVI ALGORITHMS

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

A.3.1 ARCHITECTURE BUILD

Algorithm 1: PAVI architecture build

Input: Graph template $\mathcal{T}$, plate cardinalities $\{(\mathrm{Card}^{\text{full}}(\mathcal{P}_p), \mathrm{Card}^{\text{redu}}(\mathcal{P}_p))\}_{p=0..P}$, 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 2.2 (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.3.2 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. (5), 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.3.3 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.4 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* $\mathcal{G}$:

$$\begin{aligned} \mathcal{G} &= \mathrm{KL}(q(\Theta; \phi)||p(\Theta|X)) \\ &= \log p(X) - \mathrm{ELBO}(q; \phi) \end{aligned} \tag{A.14}$$

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

$$\begin{aligned} \mathcal{G}_{\text{approx}}(\mathcal{Q}; \phi^*) &= \log p(X) - \mathrm{ELBO}(q^*; \phi^*) \\ &\geq 0 \\ \mathcal{G}_{\text{vanilla VI}} &= \mathcal{G}_{\text{approx}} \end{aligned} \tag{A.15}$$

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:

$$\begin{aligned} \mathcal{G}_{\text{sa}}(\mathcal{Q}, \mathcal{F}; \eta^*) &= \mathcal{G}_{\text{approx}}(\mathcal{Q}; f(\mathbf{X}, \eta^*)) - \mathcal{G}_{\text{approx}}(\mathcal{Q}; \phi^*) \\ &\geq 0 \\ \mathcal{G}_{\text{sample amortized VI}} &= \mathcal{G}_{\text{approx}} + \mathcal{G}_{\text{sa}} \end{aligned} \tag{A.16}$$

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 $\mathcal{Q}$:

$$\mathcal{Q}_{\text{pa}} \subset \mathcal{Q} \tag{A.17}$$

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*:

$$\begin{aligned} \mathcal{G}_{\text{pa}}(\mathcal{Q}, \mathcal{Q}_{\text{pa}}; \psi^*, \phi^*) &= \mathcal{G}_{\text{approx}}(\mathcal{Q}_{\text{pa}}; \psi^*) - \mathcal{G}_{\text{approx}}(\mathcal{Q}; \phi^*) \\ &\geq 0 \\ \mathcal{G}_{\text{PAVI-F}} &= \mathcal{G}_{\text{approx}} + \mathcal{G}_{\text{pa}} \end{aligned} \tag{A.18}$$

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.18 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.16), using an encoder cannot result in better performance, leading to an *encoder gap*:

$$\begin{aligned} \mathcal{G}_{\text{encoder}}(\mathcal{Q}_{\text{pa}}, \mathcal{F}; \psi^*, \eta^*) &= \mathcal{G}_{\text{approx}}(\mathcal{Q}_{\text{pa}}; f(\mathbf{X}, \eta^*)) - \mathcal{G}_{\text{approx}}(\mathcal{Q}_{\text{pa}}; \psi^*) \\ &\geq 0 \\ \mathcal{G}_{\text{PAVI-E}} &= \mathcal{G}_{\text{approx}} + \mathcal{G}_{\text{pa}} + \mathcal{G}_{\text{encoder}} \end{aligned} \tag{A.19}$$

The equation eq. (A.19) 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:

$$\mathcal{G}_{\text{vanilla VI}} \leq \mathcal{G}_{\text{PAVI-F}} \leq \mathcal{G}_{\text{PAVI-E}} \tag{A.20}$$

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 or experiments B.3. 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):

$$\text{ELBO}(q) = \log p(X) - \text{KL}(q(\Theta)||p(\Theta|X)) \tag{B.21}$$

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 $\text{Card}^{\text{full}}(\mathcal{P}_1) = 20$ 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 Effect of the reduced model cardinalities on the training efficiency

In fig. B.2 we show the impact of the augmentation of $\text{Card}^{\text{redu}}(\mathcal{P}_1)$ on the efficiency of the variational posterior's optimization.

In practice, we noticed that the most efficient choice in the case of the PAVI-F scheme was to maximize the cardinalities of the reduced model given the memory constraints of the GPU. Indeed, training over larger cardinalities does make each optimization step slightly slower but also make the ELBO gradient estimates less noisy and allows to train more encoding vectors $\mathbf{E}_{i,n}$ at a given optimization step.

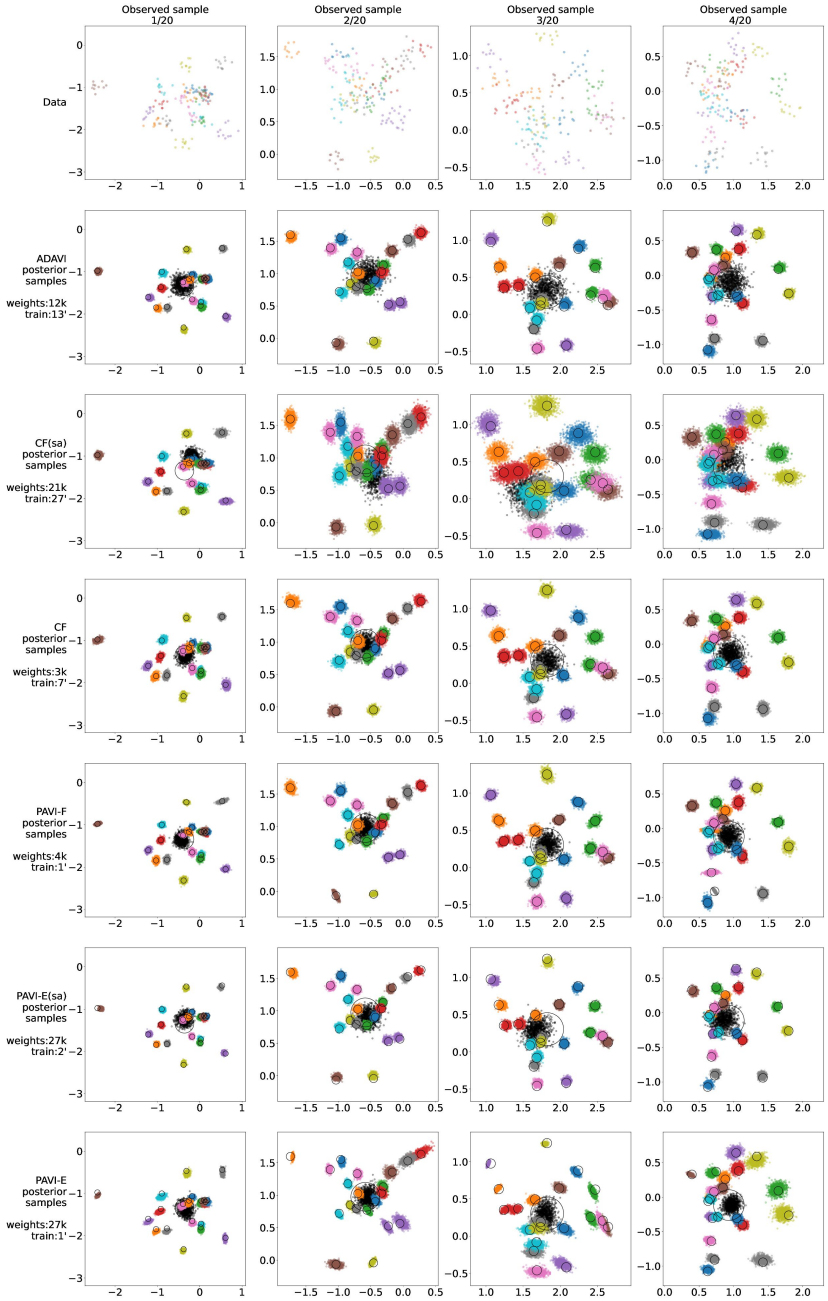

Figure B.1: **GRE Sanity check** Inference methods present qualitatively correct results, making ELBO comparisons relevant in our experiments. *On the topmost line*, we represent 4 different $\mathbf{X}$ samples for the GRE model described in eq. (6) with $\mathrm{Card}^{\mathrm{full}}(\mathcal{P}_1) = 20$. 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. Some minor bias can be observed for the PAVI-E scheme, but this approximation error is marginal compared to the optimization error that can be observed for unbiased methods, such as CF(sa) (Bottou & Bousquet, 2007). 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.

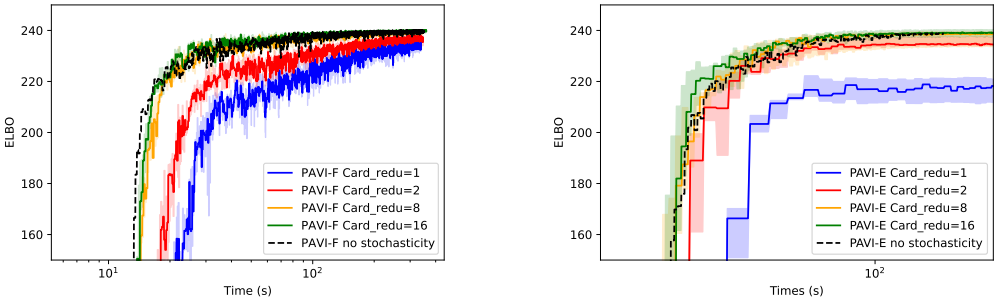

Figure B.2: **Both panels: Effect of $\mathrm{Card}^{\mathbf{redu}}(\mathcal{P}_1)$ on the training efficiency** Experiment performed on the GRE model (see eq. (6)). We increase the cardinality of the reduced model $\mathrm{Card}^{\mathrm{redu}}(\mathcal{P}_1)$ from 1 to 20 while keeping $\mathrm{Card}^{\mathrm{full}}(\mathcal{P}_1) = 20$ fixed. At $\mathrm{Card}^{\mathrm{redu}}(\mathcal{P}_1) = 20$, there is no stochasticity in the training, meaning we train directly on $\mathcal{M}^{\mathrm{full}}$. This experiment is interesting to evaluate the bias introduced in the stochastic training, as detailed in appendix A.2. **Left panel: PAVI-F scheme** As $\mathrm{Card}^{\mathrm{redu}}(\mathcal{P}_1)$ augments, the training gets faster and less noisy, likely due to less stochasticity in the gradient estimates and more encodings vectors $\mathbf{E}_{i,n}$ being trained at once. This increase in speed quickly caps, and the speed is approximately the same between $\mathrm{Card}^{\mathrm{redu}}(\mathcal{P}_1) = 8$ and $\mathrm{Card}^{\mathrm{redu}}(\mathcal{P}_1) = 16$. This experiment also illustrates the unbiasedness of the stochastic training: at $\mathrm{Card}^{\mathrm{redu}}(\mathcal{P}_1) = 20$ there is no stochasticity in the training, and the asymptotic performance is the same as for the stochastic training (see appendix A.2). **Right panel: PAVI-E scheme** The case of $\mathrm{Card}^{\mathrm{redu}}(\mathcal{P}_1) = 1$ is pathological: the encoder "learns" to collect summary statistics across a set of 1 element, and –not surprisingly– the learnt function doesn't generalize well on sets of size 20. In all of the other cases, the ELBO converges approximately to the same values as with $\mathrm{Card}^{\mathrm{redu}}(\mathcal{P}_1) = 20$, that is to say when there is no stochasticity in the training (black curve). We furthermore observe a slight reduction of the inference bias as $\mathrm{Card}^{\mathrm{redu}}(\mathcal{P}_1)$ augments. This illustrates how the theoretical bias of the PAVI-E scheme identified in appendix A.2 does not translate in significantly worse empirical results –albeit in the pathological case of $\mathrm{Card}^{\mathrm{redu}}(\mathcal{P}_1) = 1$. Interestingly, the training speed is approximately constant across all values of $\mathrm{Card}^{\mathrm{redu}}(\mathcal{P}_1)$. This feature is essential in the PAVI-E scheme: we can train over a reduced version of our model –with a light memory footprint– and apply the obtained architecture to the full model. Note that this feature wouldn't necessarily be present when training on CPU: computing summary statistics over larger sets would make the training *slower* when $\mathrm{Card}^{\mathrm{redu}}(\mathcal{P}_1)$ augments.

Architecture	ELBO (10^1)	Optimization time (s)
CF	- 16.2 ($\pm$ 2.9)	3,100
ADAVI	- 20.0 ($\pm$ 2.3)	3,000
PAVI-E (ours)	-15.2 ($\pm$2.8)	470
PAVI-F (ours)	**- 11.1 ($\pm$1.1)**	**300**

Table 1: **Analytical performance over a Gaussian mixture HBM** PAVI shows slightly superior performance compared to CF, in a fraction of CF's optimization time.

In the case of the PAVI-E scheme, the training speed is constant with respect to $\mathrm{Card}^{\text{redu}}(\mathcal{P}_1)$. This is due to the compute of the encodings being vectorized across plates in our deep-set encoder –see appendix A.1.3 (Lee et al., 2019). We observe a slight if barley noticeable reduction of the inference bias when augmenting $\mathrm{Card}^{\text{redu}}(\mathcal{P}_1)$. Similar to the PAVI-F scheme, $\mathrm{Card}^{\text{redu}}(\mathcal{P}_1)$ should be maximized with respect to the GPU memory constraint. The intuition behind this choice is that the generalization of the learning of summary statistics across sets of data points is easier the closer the reduced set size is to the full set size. This constant training speed allows for a controlled memory footprint of the stochastic training: contrary to the PAVI-F scheme, $\mathrm{Card}^{\text{redu}}(\mathcal{P}_1)$ could be maintained at a value manageable by the GPU without reducing the training speed. This example also displays a pathological choice for $\mathrm{Card}^{\text{redu}}(\mathcal{P}_1)$: the encoder fails to learn to compute the correct summary statistics over sets of size 1. However trivial, this example underlines that $\mathrm{Card}^{\text{redu}}(\mathcal{P}_1)$ should be chosen at a value logical with respect to the posterior's sufficient statistics. When generalizing to moments of higher order –such as the variance– there is therefore a theoretical lower bound to consider when fixating the value of $\mathrm{Card}^{\text{redu}}(\mathcal{P}_1)$. Our intuition is that the more complex to estimate the statistic is, the larger $\mathrm{Card}^{\text{redu}}(\mathcal{P}_1)$ should be.

Note that in practice non-stochastic VI is intractable for large-scale models, because of its memory requirement. In large scale experiments such as the one presented in fig. 5, stochastic training is necessary. Our claim is to be faster than non-plate-amortized stochastic VI in those large-scale contexts –but not necessarily to be faster then non-stochastic VI in the small-scale regime of this experiment. For PAVI-F we nonetheless obtain similar training speed compared to non-stochastic VI with $\mathrm{Card}^{\text{redu}}(\mathcal{P}_1)$ as small as 8. For PAVI-E, the stochastic training is as fast as the non-stochastic one.

B.3 Additional comparative experiments

B.3.1 Gaussian Mixture model

In this experiment we test out various baselines over a challenging model: a Gaussian mixture with random effects.

$$
\begin{aligned}
&D, \quad \mathrm{Card}(\mathcal{P}_1), \quad \mathrm{Card}(\mathcal{P}_0) = 2, \quad 20, \quad 10 \\
&\forall n_1 = 1..\mathrm{Card}(\mathcal{P}_1),\ \forall n_0 = 1..\mathrm{Card}(\mathcal{P}_0) \quad X_{n_1,n_0} | \theta_{1,n_1}, \pi_{n_1} \sim \mathrm{Mixture}(\left[\mathcal{N}(\theta^1_{1,n_1}, \sigma_x^2), \dots, \mathcal{N}(\theta^L_{1,n_1}, \sigma_x^2)\right], \pi_{n_1}) \\
&\forall n_1 = 1..\mathrm{Card}(\mathcal{P}_1) \quad \pi_{n_1} \sim \mathrm{Dirichlet}(1 \times \vec{1}_L) \\
&\forall n_1 = 1..\mathrm{Card}(\mathcal{P}_1),\ \forall l = 1..L \quad \theta^l_{1,n_1} | \theta^l_{2,0} \sim \mathcal{N}(\theta^l_{2,0}, \sigma_1^2) \\
&\forall l = 1..L \quad \theta^l_{2,0} \sim \mathcal{N}(\vec{0}_D, \sigma_2^2) \ ,
\end{aligned}
\tag{B.22}
$$

where $\mathrm{Mixture}([\mathcal{D}_1, \dots, \mathcal{D}_L], \pi)$ denotes a mixture between the distributions $[\mathcal{D}_1, \dots, \mathcal{D}_L]$ with mixture weights π. Results are visible in table 1, where all compared architectures show similar performances, but PAVI converges in a fraction of the optimization time required by the baseline architectures. On that note, we underline that stochastic training over a mixture distribution is challenging, as –due to the sub-sampling of points– only a fraction of the mixture components could be expressed at a given step, requiring the architecture to dynamically cluster the data points across time.

Architecture	ELBO (10^2)	Optimization time (s)
CF	- 11.2 ($\pm$ 2.9)	500
PAVI-E (ours)	-6.7 ($\pm$1.0)[1]	90
PAVI-F (ours)	**- 6.7 ($\pm$0.8)**	**20**

Table 2: **Analytical performance over a Hierarchical Variance HBM** In this setting with hard-to-estimate summary statistics, PAVI-E doesn't show any empirical bias. [1]: PAVI-E suffers from numerical instability on this example, with runs degenerating into NaN results.

Architecture	ELBO (10^4)	Optimization time (s)
CF	- 139.7 ($\pm$ 21)	3,200
ADAVI	—	—
PAVI-F (ours)	**- 8.2** ($\pm$ **0.1**)	**1,600**

Table 3: **Analytical performance over our Neuroimaging HBM** This HBM is not pyramidal and consequently cannot be processed by the ADAVI architecture. PAVI shows superior compared to CF, in a comparable optimization time.

B.3.2 Hierarchical Variance model

In this experiment we test out our architectures over a non-canonical model, in which parent RVs play the role of variance for the distribution of their children. Our goal is in particular to evaluate a potential empirical bias for the PAVI-E scheme –as studied in appendix A.2. We will refer the following HBM as the Hierarchical Variance model:

$$\begin{aligned} & D, \quad \mathrm{Card}(\mathcal{P}_1), \quad \mathrm{Card}(\mathcal{P}_0) = 2, \quad 15, \quad 15 \\ \forall n_1 = 1..\mathrm{Card}(\mathcal{P}_1) \; \forall n_0 = 1..\mathrm{Card}(\mathcal{P}_0) \quad & \log X_{(n_1,n_0)} | \theta_{1,n_1} \sim \mathcal{N}(0, \theta_{1,n_1}) \\ \forall n_1 = 1..\mathrm{Card}(\mathcal{P}_1) \quad & \log \theta_{1,(n_1)} | \theta_{2,0} \sim \mathcal{N}(0, \theta_{0,0}) \\ & \log \theta_{2,0} \sim \mathcal{N}(\vec{0}_D, 1) \end{aligned} \tag{B.23}$$

In this model, the encoder –used in the PAVI-E scheme– has to collect non-trivial summary statistics: empirical variances across variable subsets of the observed data. This is the context in which we would expect to observe the most bias due to the approximation in eq. (A.13). The performance of PAVI-E remains nonetheless competitive. This illustrates how PAVI-E's theoretical bias –introduced in the stochastic training– does not result in significantly worse inference compared to state-of-the-art architectures.

B.3.3 Small dimension version of our Neuroimaging model

In this experiment we validate the performance of our architecture on synthetic data generated using a small dimension version of the model presented in eq. (B.24). To allow the use of the comparative baselines, we fixate $S^{\text{full}}, S^{\text{redu}} = 5, 3$, $T^{\text{full}}, T^{\text{redu}} = 6, 3$, $N^{\text{full}}, N^{\text{redu}} = 12, 3$, $D = 4$, $L = 7$. Results are visible in table 3: PAVI can deal with HBMs unavailable to the ADAVI architecture (Rouillard & Wassermann, 2022), and so so with a performance superior to the CF baseline (Ambrogioni et al., 2021b).

B.4 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.4.1 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.4.2 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.4.3 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
- 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.4.4 GAUSSIAN MIXTURE (B.3.1)

ADAVI (Rouillard & Wassermann, 2022) we used:

- for the flows $\mathcal{F}_i$, a MAF with $[32]$ hidden units, after an affine block with diagonal 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 8 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 16
- as flows, *Highway Flows* as designed by the Cascading Flows authors

PAVI-F we used:

- for the flows $\mathcal{F}_i$, a MAF with $[32]$ hidden units, after an affine block with diagonal scaling matrix.
- an encoding size of 16
- $\mathrm{Card}^{\text{redu}}(\mathcal{P}_1) = 5$

PAVI-E we used:

- for the flows $\mathcal{F}_i$, a MAF with $[128, 128]$ hidden units, after an affine block with traingular scaling matrix.
- for the encoder, an encoding size of 128 with a multi-head architecture with 4 heads of 32 units each, 2 ISAB blocks with 128 inducing points.
- $\mathrm{Card}^{\text{redu}}(\mathcal{P}_1) = 5$

B.4.5 Hierarchical Variances (B.3.2)

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

- a mean-field distribution over the auxiliary variables r
- as auxiliary size, a fixed value of 32
- 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 diagonal scaling matrix.
- an encoding size of 32
- $\mathrm{Card}^{\text{redu}}(\mathcal{P}_1) = 3$
- $\mathrm{Card}^{\text{redu}}(\mathcal{P}_0) = 3$

PAVI-E we used:

- for the flows $\mathcal{F}_i$, a MAF with $[128, 128]$ hidden units, after an affine block with diagonal scaling matrix.
- for the encoder, an encoding size of 128 with a multi-head architecture with 4 heads of 32 units each, 2 ISAB blocks with 128 inducing points.
- $\mathrm{Card}^{\text{redu}}(\mathcal{P}_1) = 3$
- $\mathrm{Card}^{\text{redu}}(\mathcal{P}_0) = 3$

B.4.6 Small Neuroimaging example (B.3.3)

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

- a mean-field distribution over the auxiliary variables r
- as auxiliary size, a fixed value of 32
- 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 diagonal scaling matrix.
- an combination of encoding sizes of 32 and 8

B.5 Details about our Neuroimaging experiment (4.4)

B.5.1 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.5.2 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:

$$
\begin{aligned}
S^{\text{full}}, T^{\text{full}}, N^{\text{full}}, D, L &= 1000, 2, 314, 64, 2 \\
s^-, s^+ &= -6, 0 \\
\forall l{=}1..L: \quad \mu_l &\sim \text{Uniform}(-4 \times \vec{1}_D, 4 \times \vec{1}_D) \\
\forall l{=}1..L: \quad \log \epsilon_l &\sim \text{Uniform}(s^- \times \vec{1}_D, s^+ \times \vec{1}_D) \\
\forall l{=}1..L, \forall s{=}1..S: \quad \mu_{l,s} | \mu_l, \epsilon_l &\sim \mathcal{N}(\mu_l, \epsilon_l) \\
\forall l{=}1..L: \quad \log \sigma_l &\sim \text{Uniform}(s^- \times \vec{1}_D, s^+ \times \vec{1}_D) \\
\forall l{=}1..L, \forall s{=}1..S, \forall t{=}1..T: \quad \mu_{l,s,t} | \mu_{l,s}, \sigma_l &\sim \mathcal{N}(\mu_{l,s}, \sigma_l) \\
\forall l{=}1..L: \quad \log \kappa_l &\sim \text{Uniform}(s^- \times \vec{1}_D, s^+ \times \vec{1}_D) \\
\forall s{=}1..S, \forall n{=}1..N: \quad \text{probs}_{s,n} &\sim \text{Dirichlet}(1 \times \vec{1}_L) \\
\forall s{=}1..S, \forall n{=}1..N: \quad \text{label}_{s,n} \,|\, \text{probs}_{s,n} &\sim \text{Categorical}(\text{probs}_{s,n}) \\
\forall s{=}1..S, \forall t{=}1..T, \forall n{=}1..N: \quad X_{s,t,n} | [\mu_{l,s,t}]_{l=1..L}, [\kappa_l]_{l=1..L}, \text{label}_{s,n} &\sim \mathcal{N}(\mu_{\text{label}_{s,n},s,t}, \kappa_{\text{label}_{s,n}})
\end{aligned}
\tag{B.24}
$$

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.5.1.

B.5.3 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 2.3 we introduced plate amortization as the application of the generic concept of amortization to the granularity of plates. 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 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 2.2&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

Our contributions can be abstracted through the concept of plate amortization -see section 2.3. 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 CONDITIONAL DEPENDENCIES MODELLED IN THE VARIATIONAL FAMILY

The Mean-Field approximation was originally introduced in VI to facilitate computation, allowing dedicated optimization schemes (Blei et al., 2017). Nonetheless, this approximation assumes independence between RVs in the posterior, and ultimately limits the expressivity of the variational family. To remove this approximation, modelling complex conditional dependencies in the variational family is an open research subject (Ambrogioni et al., 2021b; Webb et al., 2018). In the PAVI design, we inherit our statistical dependency structure from the prior distribution p, as detailed in eq. (2). This choice of dependencies follows the line of research from Hoffman & Blei (2014) and Ambrogioni et al. (2021a). As pointed out by Ambrogioni et al. (2021b), when modelling only those *forward* dependencies, the modelling of colliders can be an issue.

In the case of a single plate –the 2-level case– Agrawal & Domke (2021) demonstrate that modelling only the *forward* dependencies does not result in reduced expressivity compared to the modelling of the full dependencies. Yet this result does not hold in the n-level case, as Webb et al. (2018) show that faithful inversion features conditional dependencies in the posterior between ground RVs of the same template. We can dub those dependencies as *horizontal* dependencies, across RVs in the same plate. Similar to Structured SVI (Hoffman & Blei, 2014) or Automatic SSVI (Ambrogioni et al., 2021a), the PAVI design therefore results in reduced expressivity when stacking multiple plates in the model p. This issue can be partially alleviated with the usage of a *backward* encoding scheme –going in the reverse direction compared to the prior's dependencies– as in the PAVI-E design (see section 3.2) or in Cascading Flows (Ambrogioni et al., 2021b).

In practice, though limiting our expressivity, *horizontal* dependencies are difficult to inject back into our architecture. Critically, in the PAVI design, the use of a common density estimator across the ground RVs of the same template (see section 2), and the stochastic training over batches of those RVs (see section 3.1) prevent the direct modeling *horizontal* dependencies. Put differently, the fact that we consider the inference over different ground RVs as conditionally independent inference problems is central to our design, and adverse to the modeling of *horizontal* dependencies.

Injecting *horizontal* dependencies back into our variational family is therefore a non-trivial research direction, that is not at the core of this paper. This opens up promising research directions: how could arbitrary conditional dependencies be modelled in the variational posterior in the context of stochastic training?

C.5 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 disentangling 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.