

Natural Variational Annealing for Multimodal Optimization

Tâm Le Minh¹, Julyan Arbel¹, Thomas Möllenhoff², Mohammad Emtiyaz Khan², Florence Forbes¹

¹ *Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France*

² *RIKEN Center for Advanced Intelligence Project, 1-4-1 Nihonbashi, Chuo-ku, Tokyo 103-0027, Japan*

Abstract. We introduce a new multimodal optimization approach called Natural Variational Annealing (NVA) that combines the strengths of three foundational concepts to simultaneously search for multiple global and local modes of black-box and/or nonconvex objectives. First, it implements a simultaneous search by using variational posteriors, such as, mixture of Gaussians. Second, it applies annealing to slowly trade off exploration for exploitation. Finally, it learns the search distribution using natural-gradient ascent where updates resemble well-known and easy-to-implement algorithms. The three concepts come together in NVA giving rise to new algorithms and also allowing us to incorporate “fitness shaping”, a core concept from evolutionary algorithms. We assess the quality of search on simulations and compare them to methods using gradient ascent and evolution strategies. We also provide an application to a real-world inverse problem in planetary science.

Keywords. multimodal optimization, annealing, variational approximation, Gaussian mixtures, natural gradient, evolution strategies.

1 Introduction

Multimodal or multisolution optimization involves finding all global optima of an objective function ℓ , as well as high quality local optima. We consider $\ell : \mathbb{R}^d \rightarrow \mathbb{R}$ to be a smooth and bounded function, where d is a positive integer and assume that it has a finite number I of global modes (maxima), denoted by ξ_1^*, \dots, ξ_I^* that satisfy

$$\{\xi_1^*, \dots, \xi_I^*\} = \arg \max_{\xi \in \mathbb{R}^d} \ell(\xi). \quad (\text{P1})$$

An explicit expression for ℓ or its gradient may not always be available, that is, the objective can be a black-box, but we assume that it is possible to evaluate $\ell(\xi)$ at arbitrary locations $\xi \in \mathbb{R}^d$. Many real-world problems can be formulated as multimodal optimization (Li et al., 2016) when there are benefits obtained by finding multiple modes, as opposed to just one mode. The diversity of modes can help us improve the decision-making process and also reveal hidden characteristics of the problem.

Classic optimization methods are not specifically designed for multimodal optimization, including those used in machine learning to navigate non-concave landscapes through stochastic mini-batching (Robbins and Monro, 1951; Duchi et al., 2011; Tieleman and Hinton, 2012; Kingma and Ba, 2015). These can still be used for multimodal optimization by using multiple restarts (Rinnooy Kan and Timmer, 1987; Lee et al., 2011; Regis and Shoemaker, 2013; Custódio and Madeira,

2015; Larson and Wild, 2018), for example, by restarting with new initial conditions to avoid previously explored regions, or by running multiple parallel instances that communicate to avoid search overlaps. Unfortunately, such solutions often rely on ad hoc heuristics which can be disconnected from principles of exploration-exploitation and also often lack theoretical foundations.

A principled way to explore is through a search distribution in stochastic optimization, for example, through the use of a Gibbs distribution (Geman and Geman, 1984) and simulated annealing to target global modes by slowly decreasing the temperature (Kirkpatrick et al., 1983). However, Gibbs distribution is often infeasible to compute and sampling is expensive. Cheap alternatives rely on variational approximations (Staines and Barber, 2012), annealing of which is referred to as *variational annealing* (Katahira et al., 2008; Yoshida and West, 2010; Mandt et al., 2016; Huang et al., 2018; Tao et al., 2019; D’Angelo and Fortuin, 2021a; Sanokowski et al., 2023). However, such works often focused on Bayesian models and applications to multimodal optimization are rare, if any.

Finally, evolutionary algorithms are another mechanism to explore (Bäck et al., 1997; Hansen and Ostermeier, 2001; Wierstra et al., 2014; Ollivier et al., 2017). In particular, particle swarms (Kennedy and Eberhart, 1995; Brits et al., 2007; Liang et al., 2006; Li, 2007; Qu et al., 2012; Yamanaka and Yoshida, 2021) have been widely tested in multimodal optimization. Evolutionary algorithms iteratively generate populations of particles to explore the search space. They simulate processes of natural evolution like mutation and selection. As a result, the particles retain memories of previously explored regions and use these to explore new regions. State-of-the-art methods combine many different heuristics, most notably niching techniques, which partition the search space to maintain exploration diversity (Ahrari et al., 2017; Maree et al., 2018; de Nobel et al., 2024).

Our goal in this paper is to propose a framework that enables us to combine the strengths of various approaches discussed above. Specifically we look for three desirable properties: (1) We want an algorithm that can simultaneously search different regions, for example, by using a search distribution; (2) We also want to be able to trade off exploration vs exploitation by using mechanisms like annealing; (3) But, we want the algorithm to be cheap and easy to implement, similarly to algorithms such as SGD, Adam and Newton’s method, used in machine learning for single-mode optimization. We aim to propose a new framework that helps us achieve all the above properties.

Contributions. We introduce a new multimodal optimization approach called *Natural Variational Annealing* (NVA) that realizes the three desirable properties by combining the strengths of three foundational concepts:

1. *Variational approximation* to enable simultaneous search for multiple modes. This is achieved by using a multimodal search distribution $q_{\lambda}(\xi)$ and aiming for a relaxed objective $\mathbb{E}_{q_{\lambda}}[\ell(\xi)]$ over the variational parameters λ . Simultaneous search is made possible by using mixture distributions (for example, mixture of Gaussians) where each mixing component focuses on a different region in the search space.
2. *Entropy annealing* to introduce exploration-exploitation trade-off by optimizing

$$\mathcal{L}_{\omega}(\lambda) = \mathbb{E}_{q_{\lambda}}[\ell(\xi)] + \omega \mathcal{H}(q_{\lambda}), \quad (1)$$

where $\mathcal{H}(q_{\lambda}) = -\int q_{\lambda}(\xi) \log q_{\lambda}(\xi) d\xi$ is the entropy of the search distribution q_{λ} . Similarly to simulated annealing, larger values of the temperature parameter $\omega > 0$ facilitate more exploration.

3. *Natural-gradient ascent* method of [Khan and Rue \(2023\)](#) called the Bayesian learning rule to obtain cheap and easy-to-implement algorithms to optimize (1):

$$\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \rho_t \mathbf{F}(\boldsymbol{\lambda}_t)^{-1} \nabla_{\boldsymbol{\lambda}} \mathcal{L}_{\omega_t}(\boldsymbol{\lambda}_t). \quad (2)$$

Here, $\boldsymbol{\lambda}_t$ denotes the natural-parameters of an exponential-family distribution $q_{\boldsymbol{\lambda}_t}$ in iteration t , $\mathbf{F}(\boldsymbol{\lambda}_t)$ is the Fisher Information Matrix (FIM), and $\rho_t > 0$ is a learning rate. For mixture distributions, we use the variants proposed by [Lin et al. \(2019a\)](#).

While existing methods have combined pairs of these concepts, like variational approximation with geometry-informed optimization ([Wierstra et al., 2014](#); [Ollivier et al., 2017](#); [Lin et al., 2019a](#); [Osawa et al., 2019](#); [Khan and Rue, 2023](#)) or with entropy annealing ([Katahira et al., 2008](#); [Yoshida and West, 2010](#); [Mandt et al., 2016](#); [Huang et al., 2018](#); [Tao et al., 2019](#); [Sanokowski et al., 2023](#)), to our knowledge, the three concepts have not previously been used simultaneously. Apart from combining the strengths of three foundational concepts, an additional advantage of NVA is that it allows us to incorporate the concept of *fitness shaping* used in evolutionary algorithms ([Wierstra et al., 2014](#)) and Information-Geometric Optimization (IGO, [Ollivier et al., 2017](#)).

We propose variants of NVA with mixture distributions (referred to as NVA-M). Specifically, we use the variant with Gaussian mixture (referred to as NVA-GM) to identify high-quality local solutions. We often choose the number of components to be much larger than the expected modes in the objective, and intentionally let some components get stuck at local solutions. This helps us to identify multiple good local solutions and understand the properties of the landscape. We show several such use cases of our method through simulations and a real-world example.

Outline. In Section 2, we present variational optimization with an entropic penalty term as an approximation of the annealed Gibbs measures. Section 3 describes the natural-gradient ascent approach. Section 4 discusses how to incorporate annealing, the role of Gaussian mixtures and the annealing schedule. Section 5 shows incorporation of fitness shaping and present a variant FS-NVA-GM (fitness-shaped NVA-GM). In Section 6, we present simulation results and compare them to Stochastic Gradient Ascent (SGA) and Covariance Matrix Adaptation Evolution Strategy (CMA-ES). Section 7 shows an application to a real-world inverse problem in planetary science.

2 Gibbs measures for multimodal optimization

Assume that function $\ell : \mathbb{R}^d \rightarrow \mathbb{R}$ has I global modes at locations $\boldsymbol{\xi}_1^*, \dots, \boldsymbol{\xi}_I^*$, with the corresponding optimization problem (P1). This is a non-convex problem if $I > 1$. Further assume that ℓ is twice continuously differentiable and that $\int \exp(\ell(\boldsymbol{\xi})/\omega) d\boldsymbol{\xi} < \infty$, for all $\omega > 0$. Traditional optimization algorithms are not equipped to handle multiple solutions, typically converging to a single mode. This section connects the variational formulation of the initial optimization problem to Gibbs measures and explains how Gaussian mixture approximations can be used to capture the properties of annealed Gibbs measures.

2.1 Gibbs measures

The following probability distribution function defines a Gibbs measure

$$g_{\omega}(\boldsymbol{\xi}) = \frac{\exp(\ell(\boldsymbol{\xi})/\omega)}{\int_{\mathbb{R}^d} \exp(\ell(\boldsymbol{\xi})/\omega) d\boldsymbol{\xi}}. \quad (3)$$

Gibbs measures as the solutions to a variational problem. Due to the strict concavity of the entropy function (see Proposition A.1 in Appendix A), the optimization problem

$$\arg \max_{q \in \mathcal{P}(\mathbb{R}^d)} \mathbb{E}_q[\ell(\boldsymbol{\xi})] + \omega \mathcal{H}(q), \quad (\text{P2})$$

where $\omega > 0$ is fixed, is strictly convex and therefore admits a unique solution. The following proposition, proven in Appendix B, provides the explicit form of this solution.

Proposition 2.1 (Kullback, 1959; Donsker and Varadhan, 1976). *The solution of (P2) is given by the Gibbs measure g_ω defined by (3).*

This result shows that the entropy term can be considered as a regularizing term. Indeed, when $\omega = 0$, the solutions of the problem are degenerate as given by the probability measures of the form $q^* = \sum_{i=1}^I c_i \delta_{\boldsymbol{\xi}_i^*}$, where the coefficients c_1, \dots, c_I are non-negative, sum to 1, and $\delta_{\boldsymbol{\xi}}$ is the Dirac measure centered at $\boldsymbol{\xi}$. Therefore, ω regulates the trade-off between the term $\mathbb{E}_q[\ell(\boldsymbol{\xi})]$, favouring a degenerate solution, and the entropy term $\mathcal{H}(q)$, which has a smoothing effect on the solution. By analogy with the use of Gibbs measures in statistical physics, the coefficient ω is called the *temperature* parameter. The variational problem can be viewed as a *generalized Bayesian inference* problem (Bissiri et al., 2016; Knoblauch et al., 2022), framed as an optimization problem.

2.2 Convergence of Gibbs measures

Gibbs measures satisfy interesting properties when the temperature decreases to 0. It is important to note that the mapping $\omega \mapsto g_\omega$ is continuous on $(0, +\infty)$ with respect to the total variation distance (see Proposition C.1 in Appendix C). The limit of g_ω as $\omega \rightarrow 0$ is well-defined and is given by the following result, under regularity conditions on ℓ .

Theorem 2.2 (Annealed Gibbs measure). *Suppose that ℓ is twice continuously differentiable and all the global modes of ℓ are non-degenerate, meaning that for all $i \in [I]$, the Hessian matrix $\nabla^2 \ell(\boldsymbol{\xi}_i^*)$ is negative definite. Then, we have*

$$g_\omega \xrightarrow[\omega \rightarrow 0]{w} g_0 := \sum_{i=1}^I \tilde{c}_i \delta_{\boldsymbol{\xi}_i^*}, \quad \text{with } \forall i \in [I], \tilde{c}_i = \frac{\det(-\nabla^2 \ell(\boldsymbol{\xi}_i^*))^{-1/2}}{\sum_{i'=1}^I \det(-\nabla^2 \ell(\boldsymbol{\xi}_{i'}^*))^{-1/2}},$$

where \xrightarrow{w} denotes weak convergence and $\delta_{\boldsymbol{\xi}}$ is the Dirac measure centered at $\boldsymbol{\xi}$.

This result is similar to the one of Hwang (1980), which applies to Gibbs posteriors (see Appendix F.2), but in our case, there is no prior distribution. The proof relies on Laplace’s method and is provided in Appendix D. This result provides a characterization of the weak limit g_0 , under the assumption of non-degenerate modes of ℓ . The distribution g_0 is a mixture of Dirac distributions located on the global modes of ℓ , with specific weights depending on the Hessian of ℓ . We call g_0 the *annealed Gibbs measure*.

Furthermore, it is interesting to note that the weights $(\tilde{c}_i)_{i \in [I]}$ of the annealed Dirac mixture g_0 provide valuable insight on the curvature of the function ℓ at its modes. Specifically, these weights evaluate the “flatness” of the modes, with larger values corresponding to flatter modes. More detailed explanations are given in Appendix E.

2.3 Annealed Gibbs measures for multimodal optimization

Theorem 2.2 may seem counterintuitive as we have established that problem (P2) admits an infinite number of solutions when $\omega = 0$, whereas there is a unique solution g_ω when $\omega > 0$. Theorem 2.2 can be explained as the sequence $(g_\omega)_\omega$ follows a specific, continuous path in terms of total variation distance when $\omega \rightarrow 0$ (Appendix C, Proposition C.1), eventually reaching the degenerate solution set. The limit of the solutions $(g_\omega)_\omega$ is unique when $\omega \rightarrow 0$ and corresponds to the annealed Gibbs measure g_0 . Thus, we can approximate g_0 as closely as desired (in the sense of weak convergence) by solving successive versions of (P2) with decreasing ω .

In our approach to multimodal optimization, we exploit the relation between Gibbs measures and the variational problem (P2) to identify the annealed Gibbs measure g_0 . This is not the first time that Gibbs measures have been employed as a tool to solve optimization problems. Notably, Gibbs measures are a key concept in simulated annealing (Kirkpatrick et al., 1983; Geman and Geman, 1984), and they have been used in global optimization as a special case of *nascent* probability distributions (Luo, 2018; Zhang and Ghosh, 2023; Serré et al., 2024). However, to our knowledge, they have not been applied to multimodal optimization. The annealed Gibbs measure is crucial in multimodal optimization because it is characterized by the locations of all the global modes, guaranteeing that they are well-represented.

The variational problem can also be viewed as a *generalized Bayesian inference* problem (Bissiri et al., 2016), framed as an optimization problem. The connection between the variational formulation and approximate Bayesian inference has been extensively explored by Knoblauch et al. (2022). Appendix F extends the analogy between Bayesian inference and annealing by giving a Bayesian interpretation to Gibbs measures, which play a fundamental role in our method.

In the following sections, we propose a strategy to extract the features of g_0 using Gaussian mixture approximations. More specifically, we introduce variational annealing as a deterministic approximation alternative to simulated annealing. In simulated annealing, g_0 is reached through samples obtained by approximating the temperature-evolving Gibbs measures with samples simulated using an MCMC procedure. Instead, we propose to consider variational approximations of the Gibbs measures and follow their behavior over iterations. The variational approach is based on the choice of a so-called variational family of distributions to approximate the target distribution. We believe this is more adapted to our multimodal optimization objective as the choice of such a family provides an additional degree of flexibility. In particular, we will later show that Gaussian mixtures represent an interesting approximating family to handle multimodal optimization as a multi-task problem, with the hope that each mixture component will take responsibility for one of the task.

3 Natural variational annealing with mixtures

In the previous section, we demonstrated that solving problem (P2) with $\omega = 0$ alone is insufficient for capturing all the modes of ℓ . This is because many solutions to this problem do not guarantee that every mode is identified. Inversely, as established by Theorem 2.2, the annealed solution g_0 is directly defined by the locations of all global modes of ℓ , under the non-degeneracy assumption, and by their flatness (see Appendix E). The core idea of our method is to extract the key properties of the annealed solution g_0 described in Theorem 2.2, specifically the locations and the weights of this Dirac mixture. Therefore, this approach addresses the challenge of finding all global modes of ℓ and identifying the most robust ones.

Our approach uses concepts from the *Bayesian Learning Rule* framework (Khan and Rue, 2023), which employs natural gradients to solve variational problems, with variational families

contained in the exponential family. Natural gradients allow faster convergence (see Appendix G for a justification). However, for multimodal optimization, the use of annealing, combined with mixture distributions, is also necessary.

The initial step in reaching g_0 involves finding a way to approximate the Gibbs measure g_ω with a mixture for a fixed $\omega > 0$. Its explicit form remains unknown due to the general lack of knowledge about ℓ . This section focuses on this crucial step. We show how natural gradient ascent can be employed to approximate g_ω with a mixture and discuss its practical implementation. Annealing can then be implemented by defining an annealing schedule $(\omega_t)_{t \geq 1}$ modifying the value of ω at each iteration.

3.1 Natural parameterization of mixtures

We adopt the following parameterization for mixtures of exponential family distributions, expressed as:

$$q_{\mathbf{\Lambda}}(\boldsymbol{\xi}) = \sum_{k=1}^K \pi_k q_{\boldsymbol{\lambda}_k}(\boldsymbol{\xi}),$$

where $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K)$ are the mixture weights and $q_{\boldsymbol{\lambda}_k}$ are the components' density functions of the form

$$q_{\boldsymbol{\lambda}_k}(\boldsymbol{\xi}) = h_k(\boldsymbol{\xi}) \exp(\langle \boldsymbol{\lambda}_k, \mathbf{T}_k(\boldsymbol{\xi}) \rangle - A(\boldsymbol{\lambda}_k)). \quad (4)$$

The exponential family natural parameters are $\boldsymbol{\lambda}_k$, while $\mathbf{T}_k(\boldsymbol{\xi})$ are their sufficient statistics, and h_k and A_k are respectively called their base measure and log partition function. Finally, the mixture parameters are denoted $\mathbf{\Lambda} := (\log(\pi_1/\pi_K), \dots, \log(\pi_{K-1}/\pi_K), \boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_K)$. Although mixtures of exponential family distributions do not belong to the exponential family itself, $\mathbf{\Lambda}$ can still be considered as the *natural parameters* of the mixture $q_{\mathbf{\Lambda}}$, viewed as a *minimal conditional exponential-family distribution* (MCEF, Lin et al., 2019a). Indeed, a mixture $\boldsymbol{\xi}$ can be represented hierarchically with a mixing random variable Z that takes values in $[K]$ such that

$$Z \sim \mathcal{M}(1, \boldsymbol{\pi}), \quad \boldsymbol{\xi} \mid Z \sim q_{\boldsymbol{\lambda}_Z}, \quad (5)$$

where $\mathcal{M}(1, \boldsymbol{\pi})$ is the multinoulli distribution (multinomial distribution with 1 trial) with probability vector $\boldsymbol{\pi}$. In this form, $\mathbf{\Lambda}$ combines the traditional natural parameters of the multinomial distribution governing Z with the traditional natural parameters of each component in the conditional distribution $\boldsymbol{\xi} \mid Z = k$ (see Lin et al., 2019a for further details).

Thus, the restriction of (P2) to mixtures of exponential family distributions can be reformulated as

$$\mathbf{\Lambda}^{*,\omega} = \arg \max_{\mathbf{\Lambda}} \mathbb{E}_{q_{\mathbf{\Lambda}}}[\ell(\boldsymbol{\xi})] + \omega \mathcal{H}(q_{\mathbf{\Lambda}}). \quad (\text{P3})$$

3.2 Natural gradient update rules

Now, we consider solving (P3). For clarity, let us define $f_\omega(\boldsymbol{\xi}; \mathbf{\Lambda}) := \ell(\boldsymbol{\xi}) + \omega \log(q_{\mathbf{\Lambda}}(\boldsymbol{\xi}))$ and $\mathcal{L}_\omega(\mathbf{\Lambda}) := \mathbb{E}_{q_{\mathbf{\Lambda}}}[f_\omega(\boldsymbol{\xi}; \mathbf{\Lambda})]$. With these notations, the optimization problem (P3) can also be written

$$\mathbf{\Lambda}^{*,\omega} = \arg \max_{\mathbf{\Lambda}} \mathbb{E}_{q_{\mathbf{\Lambda}}}[f_\omega(\boldsymbol{\xi}; \mathbf{\Lambda})] = \arg \max_{\mathbf{\Lambda}} \mathcal{L}_\omega(\mathbf{\Lambda}). \quad (\text{P3b})$$

Appendix G provides sufficient motivation to consider natural gradients (Amari, 1998), instead of (vanilla) gradients, for solving this problem. Natural gradients for a mixture of exponential family distribution are defined by

$$\tilde{\nabla}_{\mathbf{\Lambda}} \mathcal{L}_\omega(\mathbf{\Lambda})|_{\mathbf{\Lambda}_t} = \mathbf{F}(\mathbf{\Lambda}_t)^{-1} \nabla_{\mathbf{\Lambda}} \mathcal{L}_\omega(\mathbf{\Lambda})|_{\mathbf{\Lambda}_t}, \quad (6)$$

where $\mathbf{F}(\boldsymbol{\Lambda}) := \mathbb{E}[\nabla_{\boldsymbol{\Lambda}} \log q_{\boldsymbol{\Lambda}}(\boldsymbol{\xi}, Z) \nabla_{\boldsymbol{\Lambda}} \log q_{\boldsymbol{\Lambda}}(\boldsymbol{\xi}, Z)^T]$ is the Fisher Information Matrix (FIM) of $q_{\boldsymbol{\Lambda}}(\boldsymbol{\xi}, Z)$, which is non-singular (Lin et al., 2019a).

A common challenge in applying natural gradients is the computation of the inverse FIM, which is typically required in each iteration. For certain distributions, such as those within the exponential family, the FIM has a simple closed form. However, in many cases, estimating, storing, and inverting this matrix can be computationally expensive, especially for models like Gaussian mixtures, where the FIM may contain $O(K^2 d^4)$ elements. Moreover, there is no guarantee that the estimated matrix will be invertible.

Fortunately, Lin et al. (2019a) provided a method to avoid directly estimating the FIM for mixtures, based on the fact that the FIM of the joint MCEF distribution is invertible. They derived natural gradient updates for mixtures by exploiting the following relation between natural and expectation parameters:

$$\tilde{\nabla}_{\boldsymbol{\Lambda}} \mathcal{L}_{\omega}(\boldsymbol{\Lambda}) = \nabla_{\mathbf{M}} \mathcal{L}_{\omega}(\boldsymbol{\Lambda}), \quad (7)$$

where $\mathbf{M} = (\pi_1, \dots, \pi_{K-1}, \mathbf{M}_1, \dots, \mathbf{M}_K)$ are the expectation parameters of the mixture, with $\mathbf{M}_k = \mathbb{E}_{q_{\lambda_k}}[\mathbf{T}_k(\boldsymbol{\xi})]$ and $\mathbf{T}_k(\boldsymbol{\xi})$ are the sufficient statistics of component k , for all $k \in [K]$.

Using (7), the natural gradient ascent update rule

$$\boldsymbol{\Lambda}_{t+1} = \boldsymbol{\Lambda}_t + \rho_t \tilde{\nabla}_{\boldsymbol{\Lambda}} \mathcal{L}_{\omega}(\boldsymbol{\Lambda})|_{\boldsymbol{\Lambda}_t}, \quad (8)$$

becomes

$$v_{k,t+1} = v_{k,t} + \rho_t \nabla_{\pi_k} \mathcal{L}_{\omega}(\boldsymbol{\Lambda})|_{\boldsymbol{\Lambda}_t}, \quad (9)$$

$$\boldsymbol{\lambda}_{k,t+1} = \boldsymbol{\lambda}_{k,t} + \rho_t \nabla_{\mathbf{M}_k} \mathcal{L}_{\omega}(\boldsymbol{\Lambda})|_{\boldsymbol{\Lambda}_t}, \quad (10)$$

where $v_{k,t} := \log(\pi_{k,t}/\pi_{K,t})$. Therefore, in order to use these update rules, at each iteration t , we need to estimate the gradients $\nabla_{\pi_k} \mathcal{L}_{\omega}(\boldsymbol{\Lambda})|_{\boldsymbol{\Lambda}_t}$ and $\nabla_{\mathbf{M}_k} \mathcal{L}_{\omega}(\boldsymbol{\Lambda})|_{\boldsymbol{\Lambda}_t}$.

3.3 Estimation of the natural gradients

The gradient expression for the weight update (9) is given by equation (24) of Appendix J.2, for which a Monte Carlo estimator $\hat{\gamma}_{\omega, \boldsymbol{\Lambda}_t, B}^{(\pi_k)} := \widehat{\nabla_{\pi_k} \mathcal{L}_{\omega}(\boldsymbol{\Lambda})|_{\boldsymbol{\Lambda}_t}}$ is defined by (30) of Appendix J.4.

For the update of the individual component natural parameters (10), the gradient expressions depend on the distribution considered. Since the exact form of ℓ is typically unknown, it is not possible to derive closed-form expressions for these gradients. However, these gradients can be expressed as expectations of functions of $\boldsymbol{\xi}$ when $\boldsymbol{\xi}$ follows q_{λ_k} , for some $k \in [K]$. Indeed, from the chain rule, Proposition H.1, and equation (4), we have

$$\begin{aligned} \tilde{\nabla}_{\lambda_k} \mathcal{L}_{\omega}(\boldsymbol{\Lambda})|_{\boldsymbol{\Lambda}_t} &= \nabla_{\mathbf{M}_k} \mathcal{L}_{\omega}(\boldsymbol{\Lambda})|_{\boldsymbol{\Lambda}_t} = \mathbf{J}_{\lambda_k}(\mathbf{M}_k)^T \nabla_{\lambda_k} \mathbb{E}_{q_{\boldsymbol{\Lambda}}} [f_{\omega_t}(\boldsymbol{\xi}; \boldsymbol{\Lambda})]|_{\boldsymbol{\Lambda}_t} \\ &= \mathbb{E}_{q_{\lambda_k}} [\mathbf{J}_{\lambda_k}(\mathbf{M}_k)^T (\mathbf{T}_k(\boldsymbol{\xi}) - \nabla_{\lambda_k} A_k(\lambda_{k,t})) f_{\omega_t}(\boldsymbol{\xi}; \boldsymbol{\Lambda}_t)], \end{aligned} \quad (11)$$

where $\mathbf{J}_{\lambda_k}(\mathbf{M}_k)$ is the Jacobian matrix of λ_k with respect to \mathbf{M}_k , as a function of \mathbf{M}_k . These functions can be evaluated at any $\boldsymbol{\xi} \in \mathbb{R}^d$, so Monte Carlo approximations can be used to estimate the expectations. Reparameterization tricks can give alternative expressions for these gradients (see Section 4.3 for the example of Gaussian mixtures).

3.4 Stochastic natural gradient ascent

As discussed above, using the estimator $\hat{\gamma}_{\omega, \boldsymbol{\Lambda}_t, B}^{(\pi_k)}$ alongside with some estimator $\widehat{\nabla_{\mathbf{M}_k} \mathcal{L}_{\omega}(\boldsymbol{\Lambda})|_{\boldsymbol{\Lambda}_t}}$ for $\tilde{\nabla}_{\lambda_k} \mathcal{L}_{\omega}(\boldsymbol{\Lambda})|_{\boldsymbol{\Lambda}_t} = \nabla_{\mathbf{M}_k} \mathcal{L}_{\omega}(\boldsymbol{\Lambda})|_{\boldsymbol{\Lambda}_t}$, we can derive a mini-batch stochastic natural gradient ascent algorithm (SNGA), as described by Algorithm A in Appendix K.

The hyperparameters include the maximum number of iterations T (additionally, other stopping criteria can be defined), the mini-batch size B determining the number of samples used in the Monte Carlo gradient approximations, the number of components K in the mixture, the initial mixture parameters $\mathbf{\Lambda}_0$, and the learning rate sequence $(\rho_t)_{1 \leq t \leq T}$ associated with the natural gradient ascent updates.

SNGA is the basis for the NVA-M algorithms, which incorporate annealing, an essential feature for solving the multimodal optimization problem. Given the non-convex nature of the variational problem within the constrained search space of these mixtures, there is a risk that the algorithm may converge to a local solution of problem (P3) if ω is too small. SNGA does not provide any mechanism to remediate this issue, while the annealing process in NVA-M leverages Theorem 2.2, mitigating this problem by approximating increasingly sharper Gibbs measures g_ω converging to the annealed Gibbs measure g_0 .

3.5 The NVA-M algorithm

An annealing schedule, represented by a sequence of values $(\omega_t)_{t \geq 1}$, sets the temperature ω at each iteration t , thus changing the objective function $\mathcal{L}_{\omega_t}(\mathbf{\Lambda})$ and affecting the gradients in the update rule (8). Algorithm 1 gives the NVA-M algorithm, where changes compared to the basic SNGA with fixed ω (Algorithm A) are colored in blue.

Algorithm 1: NVA-M

```

1  GIVEN a function  $\ell$ .
2  SET  $T, B, K, \mathbf{\Lambda}_0, (\rho_t)_{1 \leq t \leq T}, (\omega_t)_{1 \leq t \leq T}$ .
3  COMPUTE  $(v_{k,0})_{k \in [K]} = (\log(\pi_{k,0}/\pi_{K,t}))_{k \in [K]}$ .
4  for  $t = 0:(T-1)$  do
5      for  $k = 1:K$  do
6          SAMPLE  $\xi_b^{(k)} \stackrel{\text{i.i.d.}}{\sim} q_{\lambda_{k,t}}, \text{ for } b = 1:B$ .
7          COMPUTE  $\widehat{\nabla_{M_k} \mathcal{L}_{\omega_t}}(\mathbf{\Lambda}_t)$ .
8          UPDATE  $\lambda_{k,t+1} = \lambda_{k,t} + \rho_t \widehat{\nabla_{M_k} \mathcal{L}_{\omega_t}}(\mathbf{\Lambda}_t)$ .
9      end
10     for  $k = 1:(K-1)$  do
11         COMPUTE  $\widehat{\gamma}_{\omega_t, \mathbf{\Lambda}_t, B}^{(\pi_k)}$ .
12         UPDATE  $v_{k,t+1} = v_{k,t} + \rho_t \widehat{\gamma}_{\omega_t, \mathbf{\Lambda}_t, B}^{(\pi_k)}$ .
13     end
14 end
15 COMPUTE  $(\pi_{k,T})_{k \in [K]}$  from  $(v_{k,T})_{k \in [K]}$ .
16 return  $\mathbf{\Lambda}_T$ .
```

While the NVA-M algorithm is flexible and can accomodate any mixture of exponential family distribution, in the rest of the paper, we will be considering Gaussian mixtures. The next section shows how optimizing Gaussian mixture parameters with NVA-M directly solves the initial multimodal optimization problem.

4 Natural variational annealing with Gaussian mixtures (NVA-GM)

In this section, we investigate the use of NVA framework with Gaussian mixtures (NVA-GM). First, we show how Gaussian mixtures allow to track the location of the modes of ℓ as ω goes to 0, providing motivation for NVA-GM. Then, we write the NVA-GM algorithm and we discuss the importance of the annealing schedule.

4.1 Annealing properties of Gaussian mixtures variational approximation

Here, for convenience, we parameterize Gaussian distributions by their means and covariance matrices instead of their natural parameters. Thus, a Gaussian mixture with K components is expressed as

$$q_{\boldsymbol{\theta}} = \sum_{k=1}^K \pi_k q_{\boldsymbol{\theta}_k},$$

where the weights $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K)$ sum to 1, and each component is a Gaussian $q_{\boldsymbol{\theta}_k} = \mathcal{N}(\boldsymbol{\mu}_k, \mathbf{S}_k^{-1})$. We aggregate these parameters into a vector, for all $k \in [K]$, $\boldsymbol{\theta}_k := (\boldsymbol{\mu}_k, \mathbf{S}_k^{-1})$ and

$$\boldsymbol{\theta} := (\pi_1, \dots, \pi_K, \boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K, \mathbf{S}_1^{-1}, \dots, \mathbf{S}_K^{-1}). \quad (12)$$

Note that although $\boldsymbol{\theta}$ includes all the elements of $\boldsymbol{\pi}$ for clarity, only $K-1$ weights are free parameters due to the summation constraint.

The variational approximation problem (P3) can be alternatively formulated as

$$\boldsymbol{\theta}^{*,\omega} \in \arg \max_{\boldsymbol{\theta}} \mathbb{E}_{q_{\boldsymbol{\theta}}}[\ell(\boldsymbol{\xi})] + \omega \mathcal{H}(q_{\boldsymbol{\theta}}). \quad (\text{P4})$$

Although the solution $\boldsymbol{\theta}^{*,\omega}$ is not necessarily unique, we can analyze the behavior of $\boldsymbol{\theta}^{*,\omega}$ as $\omega \rightarrow 0$. Consider a sequence $(\omega_t)_{t \geq 1}$ such that $\omega_t \rightarrow 0$ as $t \rightarrow \infty$, and a corresponding sequence $(\boldsymbol{\theta}^{*,\omega_t})_{t \geq 1}$ where, $\boldsymbol{\theta}^{*,\omega_t}$ is a solution of (P4) for each $\omega = \omega_t$. Our primary interest lies in the behavior of the component means $\boldsymbol{\mu}_k^{*,\omega_t}$ and precision matrices $\mathbf{S}_k^{*,\omega_t}$ as $\omega \rightarrow 0$, under the assumption that ℓ has $I = K$ non-degenerate modes (see Assumption D.2, for a precise statement).

Single Gaussian behavior. First, we consider the simpler case where ℓ has only one mode, implying that the variational approximation is performed with a single Gaussian ($I = K = 1$), solving

$$\boldsymbol{\theta}_1^{*,\omega} \in \arg \max_{\boldsymbol{\theta}_1} \mathbb{E}_{q_{\boldsymbol{\theta}_1}}[\ell(\boldsymbol{\xi})] + \omega \mathcal{H}(q_{\boldsymbol{\theta}_1}). \quad (\text{P5})$$

We have the following result.

Proposition 4.1. *Suppose that ℓ is strictly concave with one global maximum located at $\boldsymbol{\xi}_1^*$. Let $\omega_t \rightarrow 0$ as $t \rightarrow \infty$. For all $t \geq 1$, let $\boldsymbol{\theta}_1^{*,\omega_t} = (\boldsymbol{\mu}_1^{*,\omega_t}, (\mathbf{S}_1^{*,\omega_t})^{-1})$ be a solution of (P5) for $\omega = \omega_t$. Then, we have*

$$\begin{aligned} \boldsymbol{\mu}_1^{*,\omega_t} &\xrightarrow[t \rightarrow \infty]{} \boldsymbol{\xi}_1^*, \\ \omega_t \mathbf{S}_1^{*,\omega_t} &\xrightarrow[t \rightarrow \infty]{} -\nabla_{\boldsymbol{\xi}}^2 \ell(\boldsymbol{\xi}_1^*). \end{aligned}$$

The proof for this proposition is given in Appendix L. This result ensures that when ℓ has only one mode, the Gaussian approximation converges weakly to a Dirac distribution centered at the mode. Furthermore, it establishes a convergence rate for the covariance matrix of the Gaussian distribution, showing that $(\mathbf{S}_1^{*,\omega})^{-1} \underset{\omega \rightarrow 0}{\sim} \omega(-\nabla_{\boldsymbol{\xi}}^2 \ell(\boldsymbol{\xi}_1^*))^{-1}$. This notably implies that the eigenvalues of the covariance matrix decrease linearly with ω , tending to 0 as $\omega \rightarrow 0$.

Mixture behavior. Let us generalize this to the case where $K = I > 1$. According to Theorem 2.2, g_ω converges to g_0 , with g_0 being a mixture of Dirac measures located at the modes $(\boldsymbol{\xi}_1^*, \dots, \boldsymbol{\xi}_K^*)$. Since g_0 is the limit of a sequence of Gaussian mixtures, where eigenvalues of the covariance matrices vanish, it is reasonable to assume that the sequence $\boldsymbol{\theta}^{*,\omega}$ converges to, up to a permutation of the component labels, to

$$\boldsymbol{\theta}^{*,0} = (\tilde{c}_1, \dots, \tilde{c}_K, \boldsymbol{\xi}_1^*, \dots, \boldsymbol{\xi}_K^*, 0, \dots, 0),$$

so that in g_0 , the locations and the weights of the Dirac mixture are respectively retrieved as the limits of the means and the weights of the components of the Gaussian mixture. Proving such a result is beyond the scope of our paper. In Appendix M, we may also risk this conjecture to demonstrate how $\mathbf{S}_k^{*,\omega} \underset{\omega \rightarrow 0}{\sim} -\omega^{-1} \nabla_{\boldsymbol{\xi}}^2 \ell(\boldsymbol{\xi}_k^*)$ for all $k \in [K]$.

4.2 Update rules and gradient estimators for Gaussian mixtures

In the previous section, we have shown that the component means of a Gaussian mixture variational approximation of the Gibbs measure g_ω converge to the global modes of the objective function ℓ , as g_ω becomes closer to the annealed Gibbs measure g_0 . This provides motivation in using Gaussian mixtures in NVA-M to extract the modes of ℓ , as a direct output of the algorithm. More specifically, if $K = I$, up to a permutation of the component labels, we would expect the means $\boldsymbol{\mu}_{k,t}$ to converge to the mode $\boldsymbol{\xi}_k^*$, the precision matrices $\mathbf{S}_{k,t}$ to have eigenvalues diverging to $+\infty$, and the mixture weights $\pi_{k,t}$ to converge to the weight \tilde{c}_k . Here, we give more detailed expressions applying NVA-M to Gaussian mixtures, giving rise to the NVA-GM algorithm which directly updates the means and the precision matrices of the Gaussian components.

Natural gradient update rules. For a more practical use, the chain rule can be applied to express the gradient with respect to the expectation parameters \mathbf{M}_k as a combination of gradients with respect to the Gaussian parameters.

For Gaussian mixtures with means $\boldsymbol{\mu}_k$ and precision matrices \mathbf{S}_k , for $k \in [K]$, the update (10) becomes (see derivation in Appendix J.1)

$$\mathbf{S}_{k,t+1} = \mathbf{S}_{k,t} - \frac{2\rho_t}{\pi_{k,t}} \nabla_{\mathbf{S}_k^{-1} \mathcal{L}_{\omega_t}(\boldsymbol{\Lambda})|_{\boldsymbol{\Lambda}_t}}, \quad (13)$$

$$\boldsymbol{\mu}_{k,t+1} = \boldsymbol{\mu}_{k,t} + \frac{\rho_t}{\pi_{k,t}} \mathbf{S}_{k,t+1}^{-1} \nabla_{\boldsymbol{\mu}_k} \mathcal{L}_{\omega_t}(\boldsymbol{\Lambda})|_{\boldsymbol{\Lambda}_t}. \quad (14)$$

In this case, at each iteration t , we need to estimate the gradients $\nabla_{\pi_k} \mathcal{L}_{\omega_t}(\boldsymbol{\Lambda})|_{\boldsymbol{\Lambda}_t}$, $\nabla_{\mathbf{S}_k^{-1} \mathcal{L}_{\omega_t}(\boldsymbol{\Lambda})|_{\boldsymbol{\Lambda}_t}}$ and $\nabla_{\boldsymbol{\mu}_k} \mathcal{L}_{\omega_t}(\boldsymbol{\Lambda})|_{\boldsymbol{\Lambda}_t}$.

iBLR update rules. The *improved Bayesian Learning Rule* (iBLR) of Lin et al. (2020) can be used instead of natural gradient update rule by adding a correction term in the update (10). In

Possible choices for:	$\hat{\gamma}_{\omega_t, \Lambda_t, B}^{(\pi_k)}$	$\hat{\gamma}_{\omega_t, \Lambda_t, B}^{(\mu_k)}$	$\hat{\gamma}_{\omega_t, \Lambda_t, B}^{(S_k^{-1})}$
If ℓ available	$\hat{\gamma}_{\omega_t, \Lambda_t, B}^{(\pi_k)}$	$\hat{\gamma}_{\omega_t, \Lambda_t, B}^{(\mu_k, 0)}$	$\hat{\gamma}_{\omega_t, \Lambda_t, B}^{(S_k^{-1}, 0)}$
If $\nabla_{\xi} \ell$ available	-	$\hat{\gamma}_{\omega_t, \Lambda_t, B}^{(\mu_k, 1)}$	$\hat{\gamma}_{\omega_t, \Lambda_t, B}^{(S_k^{-1}, 1)}$
If $\nabla_{\xi}^2 \ell$ available	-	-	$\hat{\gamma}_{\omega_t, \Lambda_t, B}^{(S_k^{-1}, 2)}$

Table 1: Possible choices for the gradient estimators used in the update of Gaussian mixture parameters, depending on the availability of $\nabla_{\xi} \ell$ and $\nabla_{\xi}^2 \ell$. The expressions for these estimators are given by equations (30)-(35) in Appendix J.4.

the case of Gaussian mixtures, the iBLR is simply obtained as it only affects the precision matrix update (13):

$$S_{k,t+1} = S_{k,t} - \frac{2\rho_t}{\pi_{k,t}} \nabla_{S_k^{-1}} \mathcal{L}_{\omega_t}(\Lambda)|_{\Lambda_t} + \frac{2\rho_t^2}{\pi_{k,t}^2} \nabla_{S_k^{-1}} \mathcal{L}_{\omega_t}(\Lambda)|_{\Lambda_t} S_{k,t}^{-1} \nabla_{S_k^{-1}} \mathcal{L}_{\omega_t}(\Lambda)|_{\Lambda_t}.$$

For Gaussian mixtures, this update rule has two advantages: it does not only improve convergence speed, but it also ensures that the updated precision matrix $S_{k,t+1}$ remains positive-definite (Lin et al., 2020). In the rest of the paper, we focus on the natural gradient update rule for simplicity, while keeping in mind that the iBLR rule can be used instead.

Estimation of the natural gradients. Typically, for Gaussian mixtures, the gradient needed in (13) and (14) can be expressed under several forms using Bonnet and Price’s theorems (Appendix I), which are given by (25)-(29) in Appendix J.3. Several Monte Carlo estimators are available for these quantities, given by equations (30)-(35) in Appendix J.4. Although all these estimators are unbiased and consistent, they may perform differently since they do not have the same variance. Usually, it is hard to analytically determine which estimators have the lowest variances, as the variance expressions are highly dependent on ℓ .

The main factor to consider in selecting these estimators is the computational availability of the gradient $\nabla_{\xi} \ell(\xi)$ and the Hessian $\nabla_{\xi}^2 \ell(\xi)$. Table 1 lists the possible choices in the case of Gaussian mixtures. If $\nabla_{\xi}^2 \ell(\xi)$ cannot be computed efficiently, then $\hat{\gamma}_{\omega_t, \Lambda_t, B}^{(S_k^{-1}, 2)}$ cannot be used. If $\nabla_{\xi} \ell(\xi)$ cannot be computed efficiently, then $\hat{\gamma}_{\omega_t, \Lambda_t, B}^{(\mu_k, 1)}$ and $\hat{\gamma}_{\omega_t, \Lambda_t, B}^{(S_k^{-1}, 1)}$ cannot be used. In contrast, the estimators $\hat{\gamma}_{\omega_t, \Lambda_t, B}^{(\mu_k, 0)}$ and $\hat{\gamma}_{\omega_t, \Lambda_t, B}^{(S_k^{-1}, 0)}$ can always be computed as long as ℓ can be evaluated at all points. However, in some experiments, estimators using the higher-order derivatives of ℓ tend to perform better, often having the lowest variances.

4.3 The NVA-GM algorithm

Algorithm 2 implements NVA-M with Gaussian mixtures (NVA-GM). For convenience, instead of updating the natural parameters of the components, the updates are performed on the parameterization θ defined by (12). The use of θ instead of Λ does not affect the behavior of the algorithm and is more user-friendly.

The behavior of the mixture parameters in the NVA-GM algorithm can be interpreted in different ways. The component means can be viewed as different particles searching for the modes,

but the component themselves can be viewed as separate search distributions in an evolutionary algorithm. These two interpretation are discussed in Appendices N.1 and N.2.

Algorithm 2: NVA-GM

```

1  GIVEN a function  $\ell$ .
2  SET  $T, B, K, \theta_0, (\rho_t)_{1 \leq t \leq T}, (\omega_t)_{1 \leq t \leq T}$ .
3  COMPUTE  $(v_{k,0})_{k \in [K]} = (\log(\pi_{k,0}/\pi_{K,t}))_{k \in [K]}$ .
4  for  $t = 0:(T-1)$  do
5      for  $k = 1:K$  do
6          SAMPLE  $\xi_b^{(k)} \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(\mu_{k,t}, \mathbf{S}_{k,t}^{-1})$ , for  $b = 1:B$ .
7          COMPUTE  $\hat{\gamma}_{\omega_t, \mathbf{\Lambda}_{t,B}}^{(\mu_k)}, \hat{\gamma}_{\omega_t, \mathbf{\Lambda}_{t,B}}^{(\mathbf{S}_k^{-1})}$ .
8          UPDATE  $\mathbf{S}_{k,t+1} = \mathbf{S}_{k,t} - \rho_t \hat{\gamma}_{\omega_t, \mathbf{\Lambda}_{t,B}}^{(\mathbf{S}_k^{-1})}$ , AND  $\mu_{k,t+1} = \mu_{k,t} + \rho_t \mathbf{S}_{k,t+1}^{-1} \hat{\gamma}_{\omega_t, \mathbf{\Lambda}_{t,B}}^{(\mu_k)}$ .
9      end
10     for  $k = 1:(K-1)$  do
11         COMPUTE  $\hat{\gamma}_{\omega_t, \mathbf{\Lambda}_{t,B}}^{(\pi_k)}$ .
12         UPDATE  $v_{k,t+1} = v_{k,t} + \rho_t \hat{\gamma}_{\omega_t, \mathbf{\Lambda}_{t,B}}^{(\pi_k)}$ .
13     end
14 end
15 COMPUTE  $(\pi_{k,T})_{k \in [K]}$  from  $(v_{k,T})_{k \in [K]}$ .
16 return  $\theta_T$ .
```

In practice, in NVA-GM (and NVA-M), there is no guarantee that all components will converge to distinct modes. The annealing schedule plays an essential role in helping the algorithm avoid this issue, as we will discuss in the next section. A discussion on the choice of the hyperparameters can be found in Appendix O.

4.4 Importance of the annealing schedule

Since we aim to reach g_0 , one might question the benefits of using an annealing schedule as opposed to directly solving (P3) for a small, but fixed value of ω , which effectively corresponds to a constant annealing schedule. Intuitively, for a sufficiently small ω , the Gibbs measure g_ω should approximate g_0 closely.

Actually, the rationale for using an annealing schedule lies in the fact that although problem (P2) is strictly convex, problem (P3) is generally not convex and can admit multiple local solutions. As a result, the solution obtained by the natural gradient ascent algorithm may not be the global solution $\mathbf{\Lambda}^{*,\omega}$. In particular, local solutions can occur if several components of the mixture become overly close, leading to their merging and sharing responsibility for the same mode, which can result in the failure to identify the remaining modes.

We can interpret the component means as particles moving in the search space, as discussed in Appendix N.1 and the temperature ω as a parameter regulating the exploration-exploitation balance. Appendix N.3 analyzes how the value of ω influences the behavior of the system of particles formed by the component means, particularly in terms of exploration versus exploitation. A discussion on the choice of the annealing schedule is given in Appendix O.4.

5 Fitness-shaped NVA-GM

NVA-M (and NVA-GM) can be interpreted in the light of evolution strategies, where each component corresponds to a *search distribution* (Appendix N.2). Now, this section introduces *fitness shaping*, a technique stemming from evolutionary computation (Wierstra et al., 2014; Ollivier et al., 2017) aiming to make optimization algorithms more robust, to the NVA framework.

5.1 Fitness shaping in the IGO framework

In natural gradient updates, the gradients are highly sensitive to the values of the fitness function f . Extreme values of f can distort the gradients, potentially leading to premature convergence or numerical instability (Sun et al., 2009). To circumvent this issue, *fitness shaping* is used by the NES (Wierstra et al., 2014) and the IGO framework (Ollivier et al., 2017). Fitness shaping replaces the fitness function f at each iteration t with a *utility function*, which is an adaptive monotonic transformation W_{λ_t} of f , based on its quantiles relative to the current search distribution p_{λ_t} . In other words, for a given generation of samples, this transformation ranks fitness values in the current population, telling how good an observed fitness value is relative to the fitness distribution induced by the search distribution. This ensures that the gradient flow is invariant under any rank-preserving transformation of the fitness function (Ollivier et al., 2017), which provides robustness to the optimization process.

The transformation W_{λ} is defined as $W_{\lambda} : x \mapsto W_{\lambda}(x) = w(\mathbb{P}_{\xi' \sim p_{\lambda}}\{f(\xi') \leq x\})$, where $w : [0, 1] \rightarrow \mathbb{R}$ is a non-increasing function known as the *weighting scheme*. The choice of W_{λ} can be made through the choice of w . This definition guarantees that $W_{\lambda}(f(\xi))$ remains invariant under monotonic transformations of f . Consequently, W_{λ} can effectively compare the values of f and the modes of f would remain modes of $W_{\lambda}(f(\cdot))$, i.e. $\arg \max_{\xi} f(\xi) \subseteq \arg \max_{\xi} W_{\lambda_t}(f(\xi))$.

The natural gradient ascent update rule can be modified as follows:

$$\lambda_{t+1} = \lambda_t + \rho_t \tilde{\nabla}_{\lambda} \mathcal{L}_{W_{\lambda_t}}(\lambda)|_{\lambda=\lambda_t}, \text{ where } \mathcal{L}_{W_{\lambda_t}}(\lambda) = \mathbb{E}_{p_{\lambda}}[W_{\lambda_t}(f(\xi))].$$

If the search distribution is a Gaussian $p_{\lambda_t} = \mathcal{N}(\mathbf{m}, \mathbf{S}^{-1})$, then similar to equations (25) and (26), the gradients can be expressed as:

$$\begin{aligned} \nabla_{\mu} \mathcal{L}_{W_{\lambda_t}}(\lambda) &= \mathbb{E}_{\mathcal{N}(\mu, \mathbf{S}^{-1})}[\mathbf{S}(\xi - \mu) W_{\lambda_t}(f(\xi))], \\ \nabla_{\mathbf{S}^{-1}} \mathcal{L}_{W_{\lambda_t}}(\lambda) &= \frac{1}{2} \mathbb{E}_{\mathcal{N}(\mu, \mathbf{S}^{-1})}[(\mathbf{S}(\xi - \mu)(\xi - \mu)^T \mathbf{S} - \mathbf{S}) W_{\lambda_t}(f(\xi))]. \end{aligned}$$

These gradients are then used to compute the natural gradients, according to the update rules (13) and (14). In practice, for a sample $\xi_1, \dots, \xi_B \stackrel{\text{i.i.d.}}{\sim} p_{\lambda_t}$ of size B , we can estimate

$$W_{\lambda_t}(f(\xi_b)) = w(\mathbb{P}_{\xi' \sim p_{\lambda_t}}\{f(\xi') \leq f(\xi_b)\}) \approx \hat{w}_b := w\left(\frac{\text{rk}(\xi_b) + 1/2}{B}\right),$$

where $\text{rk}(\xi_b) = \text{Card}(\{j : f(\xi_j) < f(\xi_b)\})$. According to Theorem 6 of Ollivier et al. (2017), the following estimators are consistent for the gradients

$$\begin{aligned} \hat{v}_{\lambda_t, B}^{(\mu)} &:= \frac{1}{B} \mathbf{S} \sum_{b=1}^B (\xi_b - \mu) \hat{w}_b & \xrightarrow[B \rightarrow \infty]{\mathbb{P}} \nabla_{\mu} \mathcal{L}_{W_{\lambda_t}}(\lambda), \\ \hat{v}_{\lambda_t, B}^{(\mathbf{S}^{-1})} &:= \frac{1}{2B} \mathbf{S} \sum_{b=1}^B ((\xi_b - \mu)(\xi_b - \mu)^T \mathbf{S} - \mathbf{I}) \hat{w}_b & \xrightarrow[B \rightarrow \infty]{\mathbb{P}} \nabla_{\mathbf{S}^{-1}} \mathcal{L}_{W_{\lambda_t}}(\lambda). \end{aligned}$$

If B remains constant throughout the algorithm, then specifying the complete weighting scheme w is unnecessary because only the ordered *utility values* $\mathbf{u} = (u_b)_{1 \leq b \leq B} := (\widehat{w}_{(b)})_{1 \leq b \leq B} = (w\{(b-1/2)/B\})_{1 \leq b \leq B}$ matter. These utility values \mathbf{u} must satisfy $u_1 \geq \dots \geq u_B$, and can be chosen as a hyperparameter of the algorithm.

5.2 Adaptation to the NVA-M framework

In the IGO framework, fitness shaping is a crucial technique ensuring the algorithm's robustness, therefore facilitating the convergence of the search distribution. In our method, as discussed in Appendix N.2, each component of the mixture behaves similarly to a search distribution. Therefore, it is logical to apply fitness shaping independently to each component. Adapting our method, we suggest new updates of the natural parameters λ_k , while keeping the updates on the weights of the mixture $\pi = (\pi_1, \dots, \pi_K)$ unchanged.

To incorporate fitness shaping, we define a decreasing function w on $(0, 1)$. For each $k \in [K]$, the so-called *quantile rewriting* $W_{\lambda_k}^{(\omega, \Lambda)}$ for $f_{\omega}(\cdot; \Lambda)$ is defined by

$$W_{\lambda_k}^{(\omega, \Lambda)}(x) = w\left(\mathbb{P}_{\xi' \sim q_{\lambda_k}}\{f_{\omega}(\xi'; \Lambda) \leq x\}\right). \quad (15)$$

Natural gradient update with fitness shaping. Without fitness shaping, the natural gradient update rule for λ_k at each iteration is

$$\lambda_{k,t+1} = \lambda_{k,t} + \rho_t \tilde{\nabla}_{\lambda_k} \mathbb{E}_{q_{\Lambda}}[f_{\omega_t}(\xi; \Lambda)]|_{\Lambda=\Lambda_t}.$$

Proposition H.1 and the fact that λ_k is an element of Λ ensure that

$$\tilde{\nabla}_{\lambda_k} \mathbb{E}_{q_{\Lambda}}[f_{\omega_t}(\xi; \Lambda)]|_{\Lambda=\Lambda_t} = \tilde{\nabla}_{\lambda_k} \mathbb{E}_{q_{\Lambda}}[f_{\omega_t}(\xi; \Lambda_t)]|_{\Lambda=\Lambda_t} = \tilde{\nabla}_{\lambda_k} \mathbb{E}_{q_{\lambda_k}}[\pi_k f_{\omega_t}(\xi; \Lambda_t)]|_{\lambda_k=\lambda_{k,t}}.$$

This reveals that our update rule is equivalent to performing one step of natural gradient ascent to solve

$$\lambda_k^* = \arg \max_{\lambda} \mathbb{E}_{q_{\lambda}}[f_{\omega_t}(\xi; \Lambda_t)].$$

Now, given that $W_{\lambda_{k,t}}^{(\omega, \Lambda_t)}$ is rank-preserving for $f_{\omega_t}(\cdot; \Lambda_t)$, it is reasonable to replace $f_{\omega_t}(\cdot; \Lambda_t)$ by $W_{\lambda_{k,t}}^{(\omega, \Lambda_t)}(f_{\omega_t}(\cdot; \Lambda_t))$. This yields the following update rule for each λ_k , $k \in [K]$:

$$\lambda_{k,t+1} = \lambda_{k,t} + \rho_t \tilde{\nabla}_{\lambda} \mathcal{L}_{W_{\lambda_{k,t}}^{(\omega, \Lambda_t)}}(\lambda)|_{\lambda_{k,t}}, \text{ where } \mathcal{L}_{W_{\lambda_{k,t}}^{(\omega, \Lambda_t)}}(\lambda) := \mathbb{E}_{q_{\lambda}}[W_{\lambda_{k,t}}^{(\omega, \Lambda_t)}(f_{\omega_t}(\xi; \Lambda_t))].$$

The natural gradients can be estimated as before using Monte Carlo approximations, as equation (11) becomes

$$\tilde{\nabla}_{\lambda_k} \mathcal{L}_{\omega}(\Lambda)|_{\Lambda_t} = \mathbb{E}_{q_{\lambda_k}}[\mathbf{J}_{\lambda_k}(\mathbf{M}_k)^T (\mathbf{T}_k(\xi) - \nabla_{\lambda_k} A_k(\lambda_{k,t})) W_{\lambda_{k,t}}^{(\omega, \Lambda_t)}(f_{\omega_t}(\xi; \Lambda_t))]. \quad (16)$$

FS-NVA-GM algorithm. Now, for simplicity, we only give the fitness-shaped version of NVA-GM which uses Gaussian mixtures. The gradient derivations are given in Appendix P, and estimators $\hat{\nu}_{\omega, \Lambda_t, \mathbf{u}}^{(\mu_k)}$ and $\hat{\nu}_{\omega, \Lambda_t, \mathbf{u}}^{(S_k^{-1})}$ are defined in equations (38) and (39). FS-NVA-GM algorithm (fitness-shaped NVA-GM, Algorithm 3) incorporates fitness shaping via the free parameter utility values \mathbf{u} (Appendix Q provides some examples of utility values). Note that changes compared to NVA-GM (Algorithm 2) are colored in blue.

Algorithm 3: Fitness-shaped NVA-GM (FS-NVA-GM)

```
1 GIVEN a function  $\ell$ .
2 SET  $T, B, K, \theta_0, (\rho_t)_{1 \leq t \leq T}, (\omega_t)_{1 \leq t \leq T}, \mathbf{u} = (u_b)_{1 \leq b \leq B}$ .
3 COMPUTE  $(v_{k,0})_{k \in [K]} = (\log(\pi_{k,0}/\pi_{K,t}))_{k \in [K]}$ .
4 for  $t = 0:(T-1)$  do
5   for  $k = 1:K$  do
6     SAMPLE  $\xi_b^{(k)} \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(\mu_{k,t}, \mathbf{S}_{k,t}^{-1})$ , for  $b = 1:B$ .
7     SORT  $(\xi_b^{(k)})_{1 \leq b \leq B}$  in decreasing order for  $f_{\omega_t}(\cdot, \mathbf{\Lambda}_t)$ .
8     COMPUTE  $\hat{\mathbf{v}}_{\omega_t, \mathbf{\Lambda}_t, \mathbf{u}}^{(\mu_k)}, \hat{\mathbf{v}}_{\omega_t, \mathbf{\Lambda}_t, \mathbf{u}}^{(\mathbf{S}_k^{-1})}$ .
9     UPDATE  $\mathbf{S}_{k,t+1} = \mathbf{S}_{k,t} - \rho_t \hat{\mathbf{v}}_{\omega_t, \mathbf{\Lambda}_t, \mathbf{u}}^{(\mathbf{S}_k^{-1})}$  AND  $\mu_{k,t+1} = \mu_{k,t} + \rho_t \mathbf{S}_{k,t+1}^{-1} \hat{\mathbf{v}}_{\omega_t, \mathbf{\Lambda}_t, \mathbf{u}}^{(\mu_k)}$ .
10  end
11  for  $k = 1:(K-1)$  do
12    COMPUTE  $\hat{\gamma}_{\omega_t, \mathbf{\Lambda}_t, B}^{(\pi_k)}$ .
13    UPDATE  $v_{k,t+1} = v_{k,t} + \rho_t \hat{\gamma}_{\omega_t, \mathbf{\Lambda}_t, B}^{(\pi_k)}$ .
14  end
15 end
16 COMPUTE  $(\pi_{k,T})_{k \in [K]}$  from  $(v_{k,T})_{k \in [K]}$ .
17 return  $\theta_T$ .
```

6 Simulation study

In this section, we show simulation results to illustrate the ability of our algorithms to answer our initial optimization problem. First, we check the mode-finding performances of our algorithms. Then, we showcase the ability of the mixture weights to capture information on the flatness of the modes.

6.1 Mode-finding properties

We apply the Hessian-based version of NVA-GM and FS-NVA-GM (Algorithms 2 and 3) to find the modes of several test functions. For FS-NVA-GM, we use the utility values based on the CMA-ES weights described in Appendix Q. For simplicity, the same annealing schedules are used for both algorithms, even though their optimal annealing schedules might differ. Our algorithms fitting a mixture of K components are compared against running K independent instances of traditional Stochastic Gradient Ascent (parallel SGA, denoted by pSGA) and CMA-ES (parallel CMA-ES, denoted by pCMA-ES).

Computational cost. For all the algorithms considered, we set comparable numbers of evaluations for ℓ (or $\nabla \ell, \nabla^2 \ell$). For pSGA and NVA-GM, we use a mini-batch size \tilde{B} . For pCMA-ES and FS-NVA-GM, the same number of samples is used to compute the gradients, after truncation selection of level $\eta \in (0, 1]$. pSGA requires $\tilde{B}KT$ evaluations of $\nabla \ell$. NVA-GM uses $\tilde{B}KT$ evaluations of ℓ for the mixture weight updates, which can also be used to update the means and precision matrices through the black-box estimators. If one wishes to use the derivatives of ℓ to compute the gradients, $\tilde{B}KT$ further evaluations of $\nabla \ell$, and possibly $\nabla^2 \ell$ are needed. For pCMA-ES and FS-NVA-GM, we increase the mini-batch size by a factor η^{-1} compared to pSGA and NVA-GM,

Number of evaluations of:	$\ell(\boldsymbol{\xi})$	$\nabla\ell(\boldsymbol{\xi})$	$\nabla^2\ell(\boldsymbol{\xi})$
pSGA	0	$\tilde{B}KT$	0
pNewton	0	$\tilde{B}KT$	$\tilde{B}KT$
pCMA-ES	$\eta^{-1}\tilde{B}KT$	0	0
NVA-GM (Hessian)	$\tilde{B}KT$	0 (or $\tilde{B}KT$)	$\tilde{B}KT$
NVA-GM (gradient)	$\tilde{B}KT$	$\tilde{B}KT$	0
NVA-GM (black-box)	$\tilde{B}KT$	0	0
NVA-GM-IS (Hessian)	$\tilde{B}T$	0 (or $\tilde{B}T$)	$\tilde{B}T$
NVA-GM-IS (gradient)	$\tilde{B}T$	$\tilde{B}T$	0
NVA-GM-IS (black-box)	$\tilde{B}T$	0	0
FS-NVA-GM	$\eta^{-1}\tilde{B}KT$	0	0
FS-NVA-GM-IS	$\eta^{-1}\tilde{B}T$	0	0

Table 2: Number of calls to ℓ or its derivatives for pSGA, pNewton, pCMA-ES, NVA-GM and FS-NVA-GM. The *black-box* versions of NVA-GM and NVA-GM-IS mean that only evaluations of ℓ are performed while *gradient* (resp. *Hessian*) refers to the addition of evaluations of $\nabla\ell$ (resp. $\nabla^2\ell$).

ensuring that the effective number of samples used to compute the gradients after truncation selection remains the same. Therefore, pCMA-ES and FS-NVA-GM require $\eta^{-1}\tilde{B}KT$ evaluations of ℓ . Overall, the computational cost of all five algorithms is $O(\tilde{B}KT)$ evaluations of ℓ or its derivatives.

It is noteworthy that the number of evaluations needed for NVA-GM and FS-NVA-GM can be reduced by a factor of K through importance sampling. This consists of using a single sample per iteration to update all components, leading to algorithm variants that we call NVA-GM-IS and FS-NVA-GM-IS. Typically, to compute $\nabla_{\boldsymbol{\mu}}\mathcal{L}_{W_{\boldsymbol{\lambda}_k}^{(\omega, \Lambda_t)}}(\boldsymbol{\lambda})|_{\boldsymbol{\lambda}_{k,t}}$ and $\nabla_{S^{-1}}\mathcal{L}_{W_{\boldsymbol{\lambda}_k}^{(\omega, \Lambda_t)}}(\boldsymbol{\lambda})|_{\boldsymbol{\lambda}_{k,t}}$ for all $k \in [K]$, Lin et al. (2019a) used samples $\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_B$ from the mixture q_{Λ_t} to update the parameters of all components. Using this technique, the number of calls to ℓ or its derivatives for NVA-GM-IS and FS-NVA-GM-IS is reduced to $O(\tilde{B}T)$. Table 2 summarizes the number of calls to ℓ and its derivatives required by these algorithms. For comparison, we also include the number of calls used by K independent runs of Newton–Raphson’s method (parallel Newton, pNewton) is indicated, although it is important to note that Newton–Raphson’s method is generally unsuitable for non-convex optimization, as it does not distinguish between minima and maxima.

We use two metrics to assess the performances of the algorithms, the global peak ratio (GPR) and the all-peak ratio (APR), defined as follows:

$$\text{GPR} = \frac{1}{I \times H} \sum_{h=1}^H \text{GF}_h, \quad \text{and} \quad \text{APR} = \frac{1}{J \times H} \sum_{h=1}^H \text{AF}_h, \quad (17)$$

where H algorithm runs with random initializations have been performed and for each run h , GF_h (resp. AF_h) is the number of global (resp. global and local) modes found, out of the I (resp. J) admitted by the target function. We consider that a mode $\boldsymbol{\xi}^*$ has been found if there is at least one final component mean $\boldsymbol{\mu}_{k,t}$ in the rectangle $\boldsymbol{\xi}^* \pm \epsilon(1, \dots, 1)^T$, where $\epsilon = 10^{-2}$. These two metrics evaluate two desired properties of a multimodal optimization algorithm: identifying the global

modes and identifying distinct (global and local) modes. Empirically, NVA-GM-IS and FS-NVA-GM-IS tend to require more iterations to converge, and thus are excluded from the comparison.

Symmetric Gaussian mixture. We consider ℓ to be the log-density of the Gaussian mixture having three components, with means located at $\gamma_k = (\sin(\pi/2 + 2k\pi/3), \cos(\pi/2 + 2k\pi/3))$, $k \in \{1, 2, 3\}$ and covariance matrices set to $\sigma^2 \mathbf{I}$ where $\sigma^2 = 0.54$. ℓ has three global modes at $0.511\gamma_k$, $k \in \{1, 2, 3\}$ and one local mode at $(0, 0)$.

For $K \in \{2, 3, 4, 5\}$, we have uniformly drawn K locations in $[-2, 2]^2$. These locations are used as initial component means for NVA-GM and FS-NVA-GM and the initial positions for pSGA and pCMA-ES. In NVA-GM and FS-NVA-GM, we use $T = 5000$, $\pi_{1,0} = \dots = \pi_{K,0} = 1/K$, $\mathbf{S}_{1,0}^{-1} = \dots = \mathbf{S}_{K,0}^{-1} = \mathbf{I}$, $\omega_t = \omega_1/t$ with $\omega_1 = 1$, and $\rho_t = 10^{-1}(\omega_1/\omega_t)^{0.8}$. For NVA-GM, we use the gradient estimators $\hat{\gamma}_{\omega, \mathbf{A}, B}^{(\mu_k, 1)}$ and $\hat{\gamma}_{\omega, \mathbf{A}, B}^{(\mathbf{S}_k^{-1}, 2)}$, where $B = 4$. For FS-NVA-GM, we use the CMA-ES utility values given by (40) in Appendix Q, with $B = 16$ and $\eta = 1/4$, i.e. $B_0 = 4$. For pSGA, we use $T^{(\text{SGA})} = 10000$ iterations and a learning rate sequence $\rho_t^{(\text{SGA})} = t^{-1.1/2}$. For pCMA-ES, we use the same population size and number of selected individuals as FS-NVA-GM, i.e. $B = 16$ and $B_0 = 4$. The other hyperparameters are selected as advised in Hansen (2006). To detect if one mode is found, we check if the coordinates of at least one component mean (or pSGA particle) at the end of the algorithm are equal to the ones of the mode, with a tolerance of 0.1 in all dimensions. We replicated the experiment 100 times, recovering the GPR and APR metrics.

These estimates are given in Figure 1. We remark that with these parameters, the component means in both NVA-GM and FS-NVA-GM with CMA-ES utility weights almost always find different modes for $K \in \{2, 3, 4\}$. FS-NVA-GM almost always finds the three global modes, for $K \geq 3$. NVA-GM has slightly lower performance when $K = 3$, meaning that it sometimes finds two global modes and the local modes instead of all three global modes, but performs similarly as FS-NVA-GM when $K \geq 4$. Expectedly, although pCMA-ES finds global modes more often than pSGA, it does not find more modes than pSGA. For all values of K considered here, both NVA-GM and FS-NVA-GM heavily outperform pSGA and pCMA-ES with respect to both metrics.

Styblinski–Tang’s function. Styblinski–Tang’s function is defined by $\ell(\xi) = -\sum_{i=1}^d (\xi_i^4 - 16\xi_i^2 + 5\xi_i)/2$. ℓ has $2^d - 1$ modes, including one global mode. The locations of the modes are $\{(x_{i_1}, \dots, x_{i_d}) : (i_1, \dots, i_d) \in \{1, 2\}^d\}$, where $x_1 \approx -2.90$ and $x_2 \approx 2.75$, and the global mode is located at (x_1, \dots, x_1) .

We pick $d = 4$ so there are 16 modes. For $K \in \{2, 4, 6, 8, 10, 12, 14, 16, 18, 20\}$, we have uniformly drawn K locations in $[-4, 4]^d$. These locations are used as initial component means for NVA-GM and FS-NVA-GM and the initial particle positions for pSGA and pCMA-ES. In NVA-GM and FS-NVA-GM, we use $T = 200$, $\pi_{1,0} = \dots = \pi_{K,0} = 1/K$, $\mathbf{S}_{1,0}^{-1} = \dots = \mathbf{S}_{K,0}^{-1} = \mathbf{I}$, $\omega_t = \omega_1/t^2$ with $\omega_1 = 40000$, and $\rho_t = 10^{-4}\sqrt{\omega_1/\omega_t}$. For NVA-GM, we use the gradient estimators $\hat{\gamma}_{\omega, \mathbf{A}, B}^{(\mu_k, 1)}$ and $\hat{\gamma}_{\omega, \mathbf{A}, B}^{(\mathbf{S}_k^{-1}, 2)}$, where $B = 4$. For FS-NVA-GM, we use the CMA-ES utility values given by (40) with $B = 16$ and $\eta = 1/4$, i.e. $B_0 = 4$. For pSGA, we use $T^{(\text{SGA})} = 10000$ iterations and a learning rate sequence $\rho_t^{(\text{SGA})} = t^{-1.1/2}$. For pCMA-ES, we use the same population size and number of selected individuals as FS-NVA-GM, i.e. $B = 16$ and $B_0 = 4$. The other hyperparameters are selected as advised in Hansen (2006). To detect if one mode is found, we check if the coordinates of at least one component mean (or pSGA particle) at the end of the algorithm are equal to the ones of the mode, with a tolerance of 0.1 in all dimensions. We replicated the experiment 100 times, recovering the GPR and APR metrics.

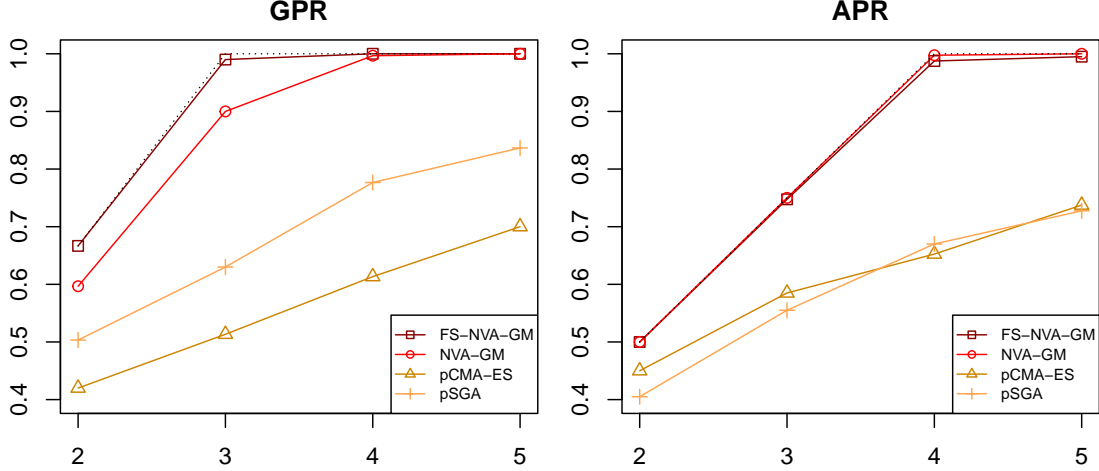


Figure 1: The first graph shows that FS-NVA-GM with CMA-ES utility values almost always locates distinct global modes of the symmetric mixture log-density. For $K \geq 3$, it almost always finds all three of them. The second graph indicates that the identified modes are distinct as long as there are at most as many components as (global and local) modes. NVA-GM performs similarly, except when $K \leq 3$, where it sometimes misses a global mode, returning the local mode instead. Parallel CMA-ES and parallel SGA struggle more to find distinct modes. The dashed line represents the maximum performance achievable for given values of K .

These estimates are given in Figure 2. The unique mode of Styblinski–Tang’s function is almost always found by NVA-GM and FS-NVA-GM for all K , and by pCMA-ES for $K \geq 6$. pSGA cannot reliably find the mode even for $K = 20$. Both NVA-GM and FS-NVA-GM show good capability to capture distinct modes. FS-NVA-GM performs slightly worse than NVA-GM, as for $K = 16$, it finds in average $16 \times 0.84 \approx 13.4$ modes out of 16, against $16 \times 0.92 \approx 14.7$ out of 16 for NVA-GM. As K is larger, the gap between NVA-GM and FS-NVA-GM on one side and pSGA and pCMA-ES on the other side is larger, with pCMA-ES performing particularly worse than pSGA. This is not surprising, since the CMA-ES has been designed for global optimization, it is more likely to find the global mode rather than a local mode when compared to SGA.

Other benchmark functions. Additionally, we have tested FS-NVA-GM on several benchmark functions from the CEC2013 suite for black-box multimodal global optimization (Li et al., 2013). A detailed description and simulation results are provided in Appendix S. For simpler functions, FS-NVA-GM achieves good performance. However, for more complex or high-dimensional functions, FS-NVA-GM is limited by its computational cost. In such cases, FS-NVA-GM would also require a significantly larger function evaluation budget to match the performance of current state-of-the-art black-box algorithms (Ahrari et al., 2017; Maree et al., 2018; de Nobel et al., 2024). It is important to note that FS-NVA-GM is only a straightforward implementation of the NVA framework using Gaussian mixtures with fitness shaping and that the NVA framework itself is not primarily designed for black-box problems. Achieving state-of-the-art performance by black-box optimization criteria would certainly require incorporating additional techniques.

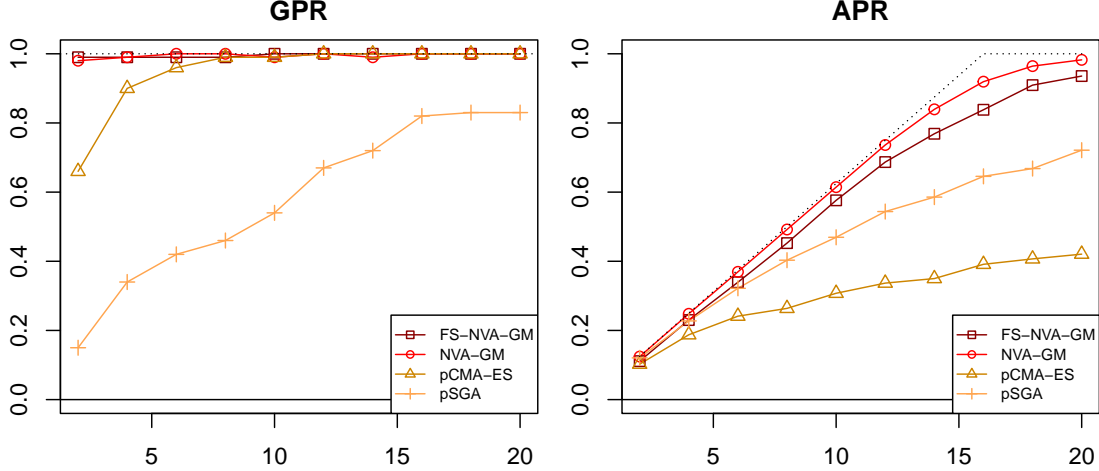


Figure 2: The first graph shows that both NVA-GM and FS-NVA-GM with CMA-ES utility values almost always locates the unique global mode of the 4-dimensional Styblinski–Tang’s function for all values of $K \geq 2$. The second graph indicates that the identified modes are distinct most of the time, except when K is close to the total number of modes (16). From this point of view, NVA-GM performs slightly better than FS-NVA-GM due to this latter’s greater tendency to favor convergence to the global mode over the 15 other local modes. Parallel CMA-ES and parallel SGA struggle more to find the global mode, as well as diverse local modes, although parallel CMA-ES matches NVA-GM and FS-NVA-GM’s performance when $K \geq 8$ in finding the global mode. The dashed line represents the maximum performance achievable for given values of K . for different values of K .

6.2 Limit of mixture weights

Here, we showcase the properties of the limit mixture weights. Both NVA-GM and FS-NVA-GM have the same weight updates, hence the same properties. We apply NVA-GM (Algorithm 1) to three objective functions with different mode flatness. With these experiments, we validate the expression of the coefficients specified in Theorem 2.2. Appendix R gives an example in a degenerate case, which is not described by Theorem 2.2, but discussed in Appendix E.

Symmetric Gaussian mixture. Here, ℓ is the log-density of the symmetric Gaussian mixture described above in Section 6.1. We apply NVA-GM with $K = 5$, using the same hyperparameters as above, except $T = 10000$ to let the weights converge further. We recover the positions of the component means and the value of the weights at each iteration.

Figure 3 illustrates the trajectories of the component means and the evolution of the mixture weights during a run where each mode is successfully identified by a distinct mean. We see that component means 1, 3, and 4 converge to the three global modes, whereas component mean 2 converges to the local mode. According to Theorem 2.2, the weights of the search mixture should be shared between components 1, 3, and 4. The target being symmetric, the three global modes have the same Hessian determinant, so all three weights should converge to $1/3$, which is the case in this run.

Asymmetric Gaussian mixture. Now, we consider the log-density of a Gaussian mixture with three components with means $\gamma_1 = (-1, 0)$, $\gamma_2 = (0, 0)$ and $\gamma_3 = (1, 0)$, covariance matrices

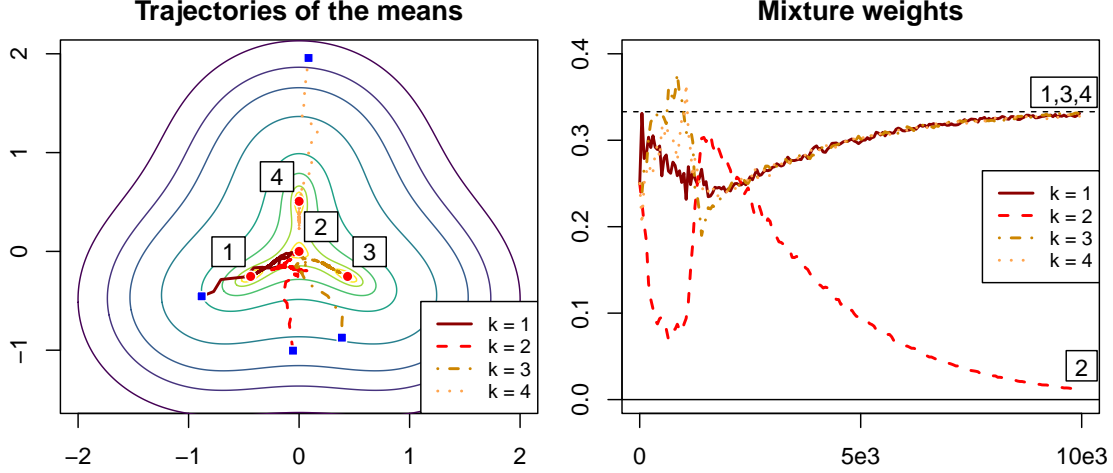


Figure 3: Left: The trajectories of the means in a run of NVA-GM on the log-density of the symmetric mixture with $K = 4$ show that the four components track different modes. The contour plot represents non-equally spaced levels of ℓ . Right: As expected, the weights of components tracking global modes (1, 3, 4) converge to $1/3$, whereas the weight of the component tracking the local mode (2) vanishes.

$\Sigma_1 = \text{diag}(0.3, 0.6)$, $\Sigma_2 = \text{diag}(0.005, 0.9)$, $\Sigma_3 = \text{diag}(0.03, 0.3)$, and weights $\pi_2 = 0.1$, $\pi_1 = (1 - \pi_2) \det(\Sigma_3)^{-1/2} / (\det(\Sigma_2)^{-1/2} + \det(\Sigma_3)^{-1/2}) \approx 0.527$ and $\pi_3 = 1 - \pi_1 - \pi_2 \approx 0.373$. This mixture is designed so that its components are well-separated and it has two global modes at γ_1 and γ_3 and one local mode in-between at γ_2 . However, the curvatures at the modes differ.

We apply NVA-GM with $K = 3$, using $T = 1000$, $\pi_{1,0} = \pi_{2,0} = \pi_{3,0} = 1/3$, $\mathbf{S}_{1,0} = \mathbf{S}_{2,0} = \mathbf{S}_{3,0} = \mathbf{I}$, $\omega_t = \omega_1/t$ with $\omega_1 = 100$, and $\rho_t = 10^{-3}(\omega_1/\omega_t)^{0.8}$. For the gradients, we use the estimators $\hat{\gamma}_{\omega, \Lambda, B}^{(\mu_k, 1)}$ and $\hat{\gamma}_{\omega, \Lambda, B}^{(S_k^{-1}, 2)}$, where $B = 4$. We recover the positions of the component means and the value of the weights at each iteration.

Figure 4 illustrates the trajectories of the component means and the evolution of the mixture weights during a run where each mode is successfully identified by a distinct mean. We notice that component mean 2 converges to the local mode near γ_2 , hence its weight vanishes. The mixture weight is shared between components 1 and 3, converging to the two global modes, near γ_1 and γ_3 respectively. According to Theorem 2.2, the weight of component 1 should converge to $\tilde{c}_1 \approx 0.586$. Analogously, the weight of component 3 should converge to $\tilde{c}_1 \approx 0.414$. These claims are supported by the behavior of the weights across this run.

7 Remote sensing inverse problem in planetary science

In general, the modes of a Gaussian mixture cannot be found analytically (Carreira-Perpiñán, 2000; Ray and Lindsay, 2005). To illustrate the practical interest of finding the modes of a Gaussian mixture in applications, we consider a remote sensing task, which consists of characterizing, with a few meaningful parameters, sites at the surface of a planet. The composition of the surface materials is generally established, with remote sensing techniques, from images produced by hyperspectral cameras, from different angles during a site flyover. An example for the planet Mars is described by Murchie et al. (2009); Fernando et al. (2016). These images carry information on the surface parameters which are extracted by inverting a model of radiative transfer. The Hapke model is

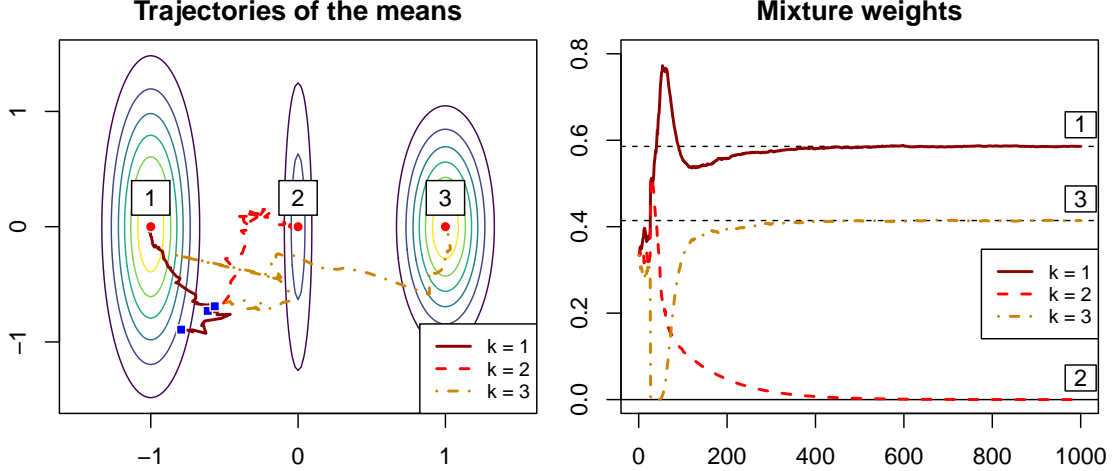


Figure 4: Left: The trajectories of the means in a run of NVA-GM on the log-density of the asymmetric mixture with $K = 3$ show that the three components track different modes. The contour plot represents non-equally spaced levels of ℓ . Right: As expected, the weights of components tracking global modes (1, 3) converge to their respective limits $\tilde{c}_1 \approx 0.586$ and $\tilde{c}_3 \approx 0.414$, whereas the weight of the component tracking the local mode (2) vanishes.

such a photometric model, that relates physically meaningful parameters to the reflectivity of a granular material for a given geometry of illumination and viewing. Formally, in our illustration, it links a set of parameters $\boldsymbol{\theta} \in \mathbb{R}^4$ to a *theoretical* surface Bidirectional Reflectance Distribution Factor (BRDF) denoted by $\mathbf{y} = F_{\text{Hapke}}(\boldsymbol{\theta}) \in \mathbb{R}^{10}$ corresponding to 10 geometries. This number of geometries is typical of real observations for which the number of possible measurements during a planet flyover is limited. The parameters are $\boldsymbol{\theta} = (\omega, \bar{\theta}, b, c)$, representing respectively the single scattering albedo, macroscopic roughness, asymmetry parameter and backscattering fraction. More details on these quantities and their photometric meanings may be found in [Schmidt and Fernando \(2015\)](#); [Labarre \(2017\)](#). Although available, the expression of F_{Hapke} is very complex and tedious to handle analytically, with a number of approximations required ([Labarre, 2017](#)). In practice, it is therefore mainly used via a numerical code, allowing simulations from the model.

Previous studies ([Kugler et al. 2022](#); [Schmidt and Fernando 2015](#); [Forbes et al. 2022](#)) have shown evidence for the existence of multiple solutions or for the possibility to obtain very similar observations from different sets of parameters, which makes this setting appropriate for testing the ability of our procedures to recover multiple solutions. To do so, a training set of 10^5 pairs $(\boldsymbol{\theta}, \mathbf{y})$ of parameters and observations are available and correspond to the application of the Hapke simulator to each $\boldsymbol{\theta}$ in the training set. We use the same experimental setting and preprocessing described in [Kugler et al. \(2022\)](#); [Forbes et al. \(2022\)](#). The approach of [Deleforge et al. \(2015\)](#) is used to learn a parametric approximation of the posterior distributions. More specifically, we use the so-called GLLiM model to produce for each possible observation \mathbf{y} an approximate posterior distribution on $\boldsymbol{\theta}$, $p(\boldsymbol{\theta}|\mathbf{y})$, which is a Gaussian mixture model with $K = 40$ components in dimension 4.

In this illustration, we focus on the characterisation of one specific observation \mathbf{y}_o coming from a mineral called Nontronite which has been chosen as likely to exhibit multiple $\boldsymbol{\theta}$ solutions (see [Kugler et al. 2022](#) for a description). The hope is then to recover these likely solutions as the modes of the 40-component Gaussian mixture posterior distribution, which approximate the desired posterior $p(\boldsymbol{\theta}|\mathbf{y}_o)$, as provided by GLLiM. We thus apply the NVA-GM algorithm to find these mixture modes. As the number of modes is a priori unknown ([Carreira-Perpiñán and Williams, 2003a,b](#);

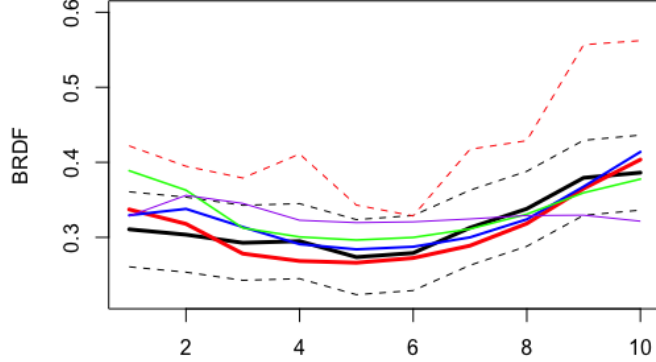


Figure 5: Remote sensing illustration: BRDF signal reconstructions. The Nontronite observed signal \mathbf{y}_o (black) is compared to the reconstructed signals from the 4 modes found by NVA-GM, θ_1 (red), θ_2 (blue), θ_3 (green) and θ_4 (mauve). The black dashed lines show a band of ± 0.05 around \mathbf{y}_o . The red dashed line shows for comparison the signal in the training set with the highest correlation to \mathbf{y}_o .

Aprausheva et al., 2006; Améndola et al., 2020), we ran NVA-GM several times with $K = 10$ and found consistently 4 candidate modes at most, the 4th one being much smaller than the others and more difficult to find. The 4 solutions found are $\theta_1 = (0.584, 0.296, 0.201, 0.143)$, $\theta_2 = (0.676, 0.317, 0.627, 0.030)$, $\theta_3 = (0.583, 0.309, 0.017, 0.614)$ and $\theta_4 = (0.729, 0.232, 0.992, 0.159)$. Their respective log-density values are 6.16, 5.17, 4.35 and 1.91, reflecting the mode order.

In the absence of ground truth, it is difficult to fully validate these estimations. However a simple inspection consists of checking the reconstructed signals obtained by applying the Hapke model to the 4 sets of estimated parameters and to compare them with the inverted signal \mathbf{y}_o . Figure 5 shows the inverted signal compared to the 4 reconstructed signals. The proximity of the reconstructions confirms the existence of multiple solutions and thus the relevance of a multimodal posterior. One solution can be selected by choosing the parameters that provide the best reconstruction assessed for instance by the smallest mean square error (MSE). For the 4 solutions, we get respectively $\text{MSE}_1 = 0.00338$, $\text{MSE}_2 = 0.00343$, $\text{MSE}_3 = 0.0115$ and $\text{MSE}_4 = 0.0173$. Figure 5 is visually consistent with the MSE values, with the θ_1 and θ_2 solutions being close to the observed signal and within a ± 0.05 band corresponding to the standard deviation of the noise added to the training set. The two other solutions are less satisfying in terms of shape although not too far from \mathbf{y}_o either. Figure 5 also shows in the red dashed line the signal in the training set that is the most correlated to \mathbf{y}_o . This signal is much further away from \mathbf{y}_o showing the gain provided by considering posterior modes instead.

8 Conclusion

In this paper, we proposed a novel, principled framework for multimodal optimization, contrasting with state-of-the-art approaches using niching, particle swarms or restart strategies. This framework uses principles from variational inference to reformulate the initial optimization task as a variational problem, an optimization task focused on the parameters of a Gaussian mixture search distribution. Geometry-informed methods, such as natural gradient ascent or the improved

Bayesian Learning Rule, are used to solve this variational problem. An entropy penalty is introduced to the variational objective, promoting convergence of the means of the mixture components to different optima.

We derived a baseline algorithm for this framework, the NVA-GM algorithm, and we proposed several extensions, including Hessian-free and black-box (derivative-free) versions. We also incorporated fitness shaping into the black-box version, resulting in the FS-NVA-GM algorithm. In our experience, the Hessian-based NVA-GM and FS-NVA-GM versions perform the best, although these algorithms face challenges when the target function number of modes K is large, as the number of objective function evaluations scales linearly with K . Importance sampling can be used to mitigate this dependency on K , although it would require a larger mini-batch size. Finding the optimal proposal distribution at each iteration is an open problem. These versions of NVA-GM and FS-NVA-GM also suffer in high-dimensional settings (d is large) due to the large covariance matrices of the Gaussian mixture, inducing high computational cost. However, constraining the variational family, for example, to Gaussian mixtures with diagonal or isotropic Gaussian covariance matrices, can address this issue.

In summary, the NVA-M framework introduces a novel approach to multimodal optimization with variational optimization as a central principle, providing a foundation for developing more sophisticated problem-specific algorithms. For example, as variational optimization gains traction in deep learning (Osawa et al., 2019; Shen et al., 2024), it would be interesting to adapt a multimodal optimization algorithm suitable to handle similarly challenging problems. Future work then includes developing solutions to reduce the computational cost and extend the framework scalability. Mode-finding performance can also be improved by incorporating new techniques or heuristics inspired from existing methods. Finally, further research may focus on the development of adaptive learning rates and annealing schedules, which are currently tuned by trial-and-error due to their problem-dependent nature.

Acknowledgements

The authors would like to thank Jacopo Iollo, Yuesong Shen and Asuka Takatsu for fruitful discussions at different stages of this project. This work is supported by ANR-21-JSTM-0001 grant.

Appendices

A Strict convexity of the Kullback–Leibler divergence

In this section, we prove the strict convexity of the Kullback–Leibler divergence, stated by the following proposition.

Proposition A.1. *If $\text{Supp}(q) \subset \text{Supp}(p)$, then $\text{KL}(q \parallel p)$ is strictly convex with respect to q .*

Proof. Let q_0 and q_1 be two distinct probability distributions such that $\text{Supp}(q_0) \subset \text{Supp}(p)$ and $\text{Supp}(q_1) \subset \text{Supp}(p)$. For $t \in (0, 1)$, define

$$q_t = (1 - t)q_0 + tq_1.$$

Our aim is to show that for all $t \in (0, 1)$,

$$\Delta_t := (1 - t)\text{KL}(q_0 \parallel p) + t\text{KL}(q_1 \parallel p) - \text{KL}(q_t \parallel p) > 0.$$

Let $g : q \mapsto q \log q$. Since q_0 , q_1 and q_t are absolutely continuous with respect to p , then we have

$$\Delta_t = \int [(1 - t)g(q_0) + tg(q_1) - g(q_t)](\xi) d\xi.$$

To conclude that $\Delta_t > 0$, we write the first order Taylor expansions of $g(q_0)$ and $g(q_1)$.

For $i \in \{0, 1\}$, define $\varphi_i : u \mapsto (1 - u)q_t + uq_i$, then

$$\begin{aligned} g(q_i) &= g(\varphi_i(1)) \\ &= g(\varphi_i(0)) + (\varphi_i(1) - \varphi_i(0))g'(\varphi_i(0)) - \int_0^1 (\varphi_i(u) - \varphi_i(0))g''(\varphi_i(u))d\varphi_i(u) \\ &= g(q_t) + (q_i - q_t)g'(q_t) + (q_i - q_t)^2 \int_0^1 \frac{1}{(1 - u)q_t + uq_i} (1 - u)du. \end{aligned}$$

Therefore,

$$(1 - t)g(q_0) + tg(q_1) - g(q_t) = t(1 - t)(q_1 - q_0)^2 \int_0^1 \left(\frac{t}{(1 - u)q_t + uq_0} + \frac{1 - t}{(1 - u)q_t + uq_1} \right) (1 - u)du.$$

Since q_0 and q_1 are distinct, it can be deduced that $\Delta_t > 0$. \square

B Link between the Gibbs measure and the variational problem

In this section, we prove Proposition 2.1, which is stated in Section 2.1.

Proof of Proposition 2.1. This proof is similar to the one of Alquier (2024). Let q be a probability distribution. Consider the Kullback–Leibler divergence between q and g_ω :

$$\begin{aligned} 0 &\geq -\text{KL}(q \parallel g_\omega) \\ &= - \int q(\xi) \log \frac{q(\xi)}{g_\omega(\xi)} d\xi \\ &= \int q(\xi) \frac{\ell(\xi)}{\omega} d\xi - \int q(\xi) \log q(\xi) d\xi - \log Z(\omega) \\ &= \frac{1}{\omega} (\mathbb{E}_q[\ell(\xi)] + \omega \mathcal{H}(q)) - \log Z(\omega). \end{aligned}$$

Since $\text{KL}(q \parallel g_\omega) = 0$ if and only if $q = g_\omega$, it follows that the quantity $\mathbb{E}_q[\ell(\xi)] + \omega \mathcal{H}(q)$ is maximized if and only if $q = g_\omega$. \square

C Continuity of Gibbs measures with respect to the temperature

In this section, we prove that the mapping $\omega \mapsto g_\omega$ is continuous with respect to the total variation distance.

Proposition C.1. *For some ω_0 in $(0, +\infty)$, we have*

$$\|g_\omega - g_{\omega_0}\|_{\text{TV}} \xrightarrow{\omega \rightarrow \omega_0} 0,$$

where $\|p - q\|_{\text{TV}}$ denotes the distance in total variation between p and q , i.e. $\|p - q\|_{\text{TV}} = \sup_A |p(A) - q(A)|$.

Before giving a proof for this proposition, we recall Scheffé's theorem (e.g. [Tsybakov, 2008](#), p. 84).

Lemma C.2 (Scheffé's theorem).

$$\|p - q\|_{\text{TV}} = \frac{1}{2} \int |p(\xi) - q(\xi)| d\xi.$$

Proof of Proposition C.1. By Scheffé's theorem, it is enough to prove that $g_\omega \xrightarrow{\omega \rightarrow \omega_0} g_{\omega_0}$ almost everywhere on \mathbb{R}^d . Let $\xi \in \mathbb{R}^d$. Denote $\varphi_\xi : \omega \mapsto g_\omega(\xi)$.

The analytical expression of the derivative $\varphi'_\xi(\omega)$ can be derived on $(0, +\infty)$. Because $Z(\omega) = \int \exp(\ell(\xi)/\omega) d\xi$, we have $Z'(\omega) = -\omega^{-2} Z(\omega) \mathbb{E}_{g_\omega}[\ell(\xi)]$. Therefore,

$$|\varphi'_\xi(\omega)| = \left| \frac{d}{d\omega} \left(\frac{e^{\frac{\ell(\xi)}{\omega}}}{Z(\omega)} \right) \right| = \left| \frac{\ell(\xi) e^{\frac{\ell(\xi)}{\omega}}}{\omega^2 Z(\omega)} - \frac{e^{\frac{\ell(\xi)}{\omega}} Z'(\omega)}{Z(\omega)^2} \right| = \frac{g_\omega(\xi)}{\omega^2} |\ell(\xi) - \mathbb{E}_{g_\omega}[\ell(\xi)]| < \infty.$$

This shows that φ_ξ is a continuous function on $(0, +\infty)$, for all $\xi \in \mathbb{R}^d$. This ensures that $\varphi_\xi(\omega) \xrightarrow{\omega \rightarrow \omega_0} \varphi_\xi(\omega_0)$ almost everywhere on \mathbb{R}^d , which concludes the proof of this proposition. \square

D Laplace's method

In this section, we will prove Theorem 2.2. Before that, we state a multivariate version of the well-known Laplace's theorem (e.g. Section 6.4 of [Bender and Orszag, 1999](#)), which seems to be due to [Hsu \(1948\)](#) (see also IX.5 of [Wong, 2001](#)).

Theorem D.1 (Laplace's theorem). *Let $\xi^* \in \mathbb{R}^d$, $\delta > 0$ and $\mathcal{B}_{\xi^*}(\delta)$ be the ball centered around ξ^* and with radius δ . Let $\phi : \mathcal{B}_{\xi^*}(\delta) \rightarrow \mathbb{R}$ be a twice continuously differentiable function such that*

- $\phi(\xi^*) = 0$,
- $\phi(\xi) < 0$ for all $\xi \in \mathcal{B}_{\xi^*}(\delta)$,
- $\det(-\nabla^2 \phi(\xi^*)) > 0$.

Then, for all bounded function f , for all $0 < \epsilon \leq \delta$, we have

$$\int_{\mathcal{B}_{\xi^*}(\epsilon)} f(\xi) \exp(\phi(\xi)/\omega) d\xi \underset{\omega \rightarrow 0}{\sim} (2\pi\omega)^{d/2} f(\xi^*) \det(-\nabla^2 \phi(\xi^*))^{-1/2}.$$

Now, to prove Theorem 2.2, we recall that it is assumed that ℓ is twice continuously differentiable and all the global modes of ℓ are non-degenerate, i.e. the Hessian matrix of ℓ is negative definite at these modes. Without loss of generality, we can also consider that ℓ is non-positive. Therefore, the following assumptions is made throughout this section.

Assumption D.2. *The function ℓ satisfies the following conditions:*

- the set $\{\xi_1^*, \dots, \xi_I^*\} = \arg \max_{\xi} \ell(\xi)$ is finite and $\max_{\xi} \ell(\xi) = 0$,
- $\int_{\mathbb{R}^d} \exp(\ell(\xi)/\omega) d\xi < \infty$ for all $\omega > 0$,
- ℓ is twice differentiable,
- for all the global modes $\{\xi_1^*, \dots, \xi_I^*\}$ of ℓ , we have

$$\det(-\nabla^2 \ell(\xi_i^*)) > 0.$$

Now, we need to define the following notations and prove some lemmas.

Notations. Here, some useful notations are defined for the proof.

- In the metric space $(\mathbb{R}^d, \|\cdot\|)$, for some $\epsilon > 0$, $\mathcal{B}_\epsilon(\xi)$ denotes the ball of \mathbb{R}^d centered on ξ with radius ϵ .
- Let ϵ be such that $0 < \epsilon < \min_{i,j} \|\xi_j^* - \xi_i^*\|$. Such an ϵ exists because there are a finite number of global modes, so they are all isolated. In this case,

$$\begin{aligned} - K_\epsilon &:= \bigcup_{i=1}^I \mathcal{B}_\epsilon(\xi_i^*), \\ - M_\epsilon &:= \frac{1}{2} \max_{\xi \in \mathbb{R}^d \setminus K_\epsilon} \ell(\xi), \\ - K_\epsilon^{(M_\epsilon)} &:= K_\epsilon \cap \{\xi \in \mathbb{R}^d : \ell(\xi) \geq M_\epsilon\}. \end{aligned}$$

- Let f be a continuous and bounded function. For $\omega > 0$, we define

$$\begin{aligned} - \text{for all } i \in [I], C_i^f(\omega) &:= \int_{\mathcal{B}_\epsilon(\xi_i^*)} f(\xi) e^{\ell(\xi)/\omega} d\xi, \\ - C_0^f(\omega) &:= \int_{\mathbb{R}^d \setminus K_\epsilon} f(\xi) e^{\ell(\xi)/\omega} d\xi = Z(\omega) - \sum_{i=1}^I C_i^f(\omega). \end{aligned}$$

- We write $C_i^1(\omega) := C_i^{f_0}(\omega)$ and $C_0^1(\omega) := C_0^{f_0}(\omega)$ for $f_0 \equiv 1$.

Lemma D.3. *For a continuous and bounded function f on \mathbb{R}^d , we have*

$$\frac{C_0^f(\omega)}{Z(\omega)} \xrightarrow{\omega \rightarrow 0} 0.$$

Proof. For all $\xi \in K_\epsilon^{(M_\epsilon)}$, we have $M_\epsilon \leq \ell(\xi) \leq 0$. This means that

$$\begin{aligned} Z(\omega) &\geq \int_{K_\epsilon^{(M_\epsilon)}} e^{\ell(\xi)/\omega} d\xi \\ &\geq e^{M_\epsilon/\omega} \mathbb{P}(K_\epsilon^{(M_\epsilon)}). \end{aligned}$$

Therefore, for a continuous and bounded function f , we have

$$\begin{aligned} \left| \frac{C_0^f(\omega)}{Z(\omega)} \right| &\leq \frac{1}{\mathbb{P}(K_\epsilon^{(M_\epsilon)})} \int_{\mathbb{R}^d \setminus K_\epsilon} |f(\xi)| e^{\frac{1}{\omega}(\ell(\xi) - M_\epsilon)} d\xi \\ &\leq \frac{\max_{\xi \in \mathbb{R}^d \setminus K_\epsilon} |f(\xi)|}{\mathbb{P}(K_\epsilon^{(M_\epsilon)})} \int_{\mathbb{R}^d \setminus K_\epsilon} e^{\frac{1}{\omega}(\ell(\xi) - M_\epsilon)} d\xi. \end{aligned}$$

Any sequence $(\omega_t)_{t \geq 1}$ such that $\omega_t \xrightarrow[t \rightarrow \infty]{} 0$ is bounded, so there is a value ω_M such that $\omega_t \leq \omega_M$ for all $t \geq 1$. Therefore, on $\mathbb{R}^d \setminus K_\epsilon$, we have $e^{\frac{1}{\omega_t}(\ell(\xi) - M_\epsilon)} \leq e^{\ell(\xi)/\omega_M}$ for all $t \geq 1$, since $\ell(\xi) - M_\epsilon < 0$. Thus, we can apply the dominated convergence theorem and

$$\begin{aligned} \lim_{\omega \rightarrow 0} \left| \frac{C_0^f(\omega)}{Z(\omega)} \right| &\leq \lim_{\omega \rightarrow 0} \frac{\max_{\xi \in \mathbb{R}^d \setminus K_\epsilon} |f(\xi)|}{\mathbb{P}(K_\epsilon^{(M_\epsilon)})} \int_{\mathbb{R}^d \setminus K_\epsilon} e^{\frac{1}{\omega}(\ell(\xi) - M_\epsilon)} d\xi \\ &= \frac{\max_{\xi \in \mathbb{R}^d \setminus K_\epsilon} |f(\xi)|}{\mathbb{P}(K_\epsilon^{(M_\epsilon)})} \int_{\mathbb{R}^d \setminus K_\epsilon} \lim_{\omega \rightarrow 0} e^{\frac{1}{\omega}(\ell(\xi) - M_\epsilon)} d\xi \\ &= 0. \end{aligned}$$

□

Lemma D.4. For a continuous and bounded function f on \mathbb{R}^d , we have for all $i \in [I]$

$$(2\pi)^{-d/2} \omega^{-d/2} C_i^f(\omega) \xrightarrow[\omega \rightarrow 0]{} f(\xi_i^*) \det(L_i)^{-1},$$

where L_i is the matrix such that $L_i^2 = -\nabla^2 \ell(\xi_i^*)$.

Proof. This is a direct consequence of Laplace's theorem (Thm. D.1). □

Lemma D.5. For a continuous and bounded function f on \mathbb{R}^d , we have for all $i \in [I]$

$$\frac{C_i^f(\omega)}{Z(\omega)} \xrightarrow[\omega \rightarrow 0]{} \tilde{c}_i f(\xi_i^*),$$

where $\tilde{c}_i = \frac{\det(L_i)^{-1}}{\sum_{i'=1}^I \det(L_{i'})^{-1}} f(\xi_i^*)$ and L_i is the matrix such that $L_i^2 = -\nabla^2 \ell(\xi_i^*)$.

Proof. Remark that

$$\begin{aligned} Z(\omega) &= \int_{\mathbb{R}^d} e^{\ell(\xi)/\omega} d\xi \\ &= \int_{\mathbb{R}^d \setminus K_\epsilon} e^{\ell(\xi)/\omega} d\xi + \sum_{i'=1}^I \int_{B_\epsilon(\xi_{i'}^*)} e^{\ell(\xi)/\omega} d\xi \\ &= C_0^1(\omega) + \sum_{i'=1}^I C_{i'}^1(\omega). \end{aligned}$$

Therefore, applying Lemma D.3 and then Lemma D.4, this ensures that

$$\begin{aligned} \lim_{\omega \rightarrow 0} \frac{C_i^f(\omega)}{Z(\omega)} &= \lim_{\omega \rightarrow 0} \frac{C_i^f(\omega)}{C_0^1(\omega) + \sum_{i'=1}^I C_{i'}^1(\omega)} \\ &= \lim_{\omega \rightarrow 0} \frac{C_i^f(\omega)}{\sum_{i'=1}^I C_{i'}^1(\omega)} \\ &= \frac{\det(L_i)^{-1}}{\sum_{i'=1}^I \det(L_{i'})^{-1}} f(\xi_i^*). \end{aligned}$$

□

Finally, thanks to Lemmas D.3 and D.5, we can concisely prove Theorem 2.2.

Proof of Theorem 2.2. Using Lemmas D.3 and D.5, we have for any continuous and bounded function f ,

$$\begin{aligned}
\lim_{\omega \rightarrow 0} \int_{\mathbb{R}^d} f(\boldsymbol{\xi}) g_\omega(d\boldsymbol{\xi}) &= \lim_{\omega \rightarrow 0} \frac{\int_{\mathbb{R}^d \setminus K_\epsilon} f(\boldsymbol{\xi}) e^{\ell(\boldsymbol{\xi})/\omega} d\boldsymbol{\xi} + \sum_{i'=1}^I \int_{B_\epsilon(\boldsymbol{\xi}_{i'}^*)} f(\boldsymbol{\xi}) e^{\ell(\boldsymbol{\xi})/\omega} d\boldsymbol{\xi}}{Z(\omega)} \\
&= \lim_{\omega \rightarrow 0} \left\{ \frac{C_0^f(\omega)}{Z(\omega)} + \sum_{i=1}^I \frac{C_i^f(\omega)}{Z(\omega)} \right\} \\
&= 0 + \sum_{i=1}^I \tilde{c}_i f(\boldsymbol{\xi}_i^*) \\
&= \int_{\mathbb{R}^d} f(\boldsymbol{\xi}) \left(\sum_{i=1}^I \tilde{c}_i \delta_{\boldsymbol{\xi}_i^*}(d\boldsymbol{\xi}) \right).
\end{aligned}$$

This proves that g_ω converges weakly to $\sum_{i=1}^I \tilde{c}_i \delta_{\boldsymbol{\xi}_i^*}$. □

E Interpretation of the weights of the annealed Gibbs measure

We claim that the weights $(\tilde{c}_i)_{i \in [I]}$ of the annealed Dirac mixture g_0 of Theorem 2.2 provide valuable insight on the curvature of the function ℓ at its modes. Specifically, these weights offer a means to assess the “flatness” of the modes. To explore this, we introduce a measure of flatness and discuss the implications of the non-degeneracy assumption.

A measure of flatness based on curvature. The Hessian matrix $\nabla^2 \ell(\boldsymbol{\xi})$ characterizes the local convexity (or concavity) of ℓ at a location $\boldsymbol{\xi}$. To quantify the flatness of a mode, we consider the eigenvalues of $\nabla^2 \ell(\boldsymbol{\xi})$, which indicate the curvature of ℓ along the directions specified by the eigenvectors. At a maximum $\boldsymbol{\xi}_i^*$, the eigenvalues $\{\lambda_{i,1}, \dots, \lambda_{i,d}\}$ of $\nabla^2 \ell(\boldsymbol{\xi}_i^*)$ are non-positive, and the sharpness of the mode is reflected by the magnitude of $|\lambda_{i,1}|, \dots, |\lambda_{i,d}|$. To characterize this sharpness, one can compute the geometric mean $|\prod_{j=1}^d \lambda_{i,j}|^{1/d}$. Conversely, the flatness of the mode can be measured by

$$\zeta(\boldsymbol{\xi}_i^*) := \left| \prod_{j=1}^d \lambda_{i,j} \right|^{-1/d}.$$

To illustrate the utility of ζ as a measure of flatness, consider a multivariate normal distribution $\ell(\cdot) := \mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$, with mode located at $\boldsymbol{\mu}$. The Hessian at the mode is given by $\nabla_{\boldsymbol{\xi}}^2 \mathcal{N}(\boldsymbol{\mu}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\det(\boldsymbol{\Sigma})^{-1/2} \boldsymbol{\Sigma}^{-1}$, leading to

$$\zeta(\boldsymbol{\mu}) = \det(-\nabla_{\boldsymbol{\xi}}^2 \mathcal{N}(\boldsymbol{\mu}; \boldsymbol{\mu}, \boldsymbol{\Sigma}))^{-1/d} = (\det \boldsymbol{\Sigma})^{\frac{d+2}{2d}}.$$

Thus, $\zeta(\boldsymbol{\mu})$ increases with the flatness of the Gaussian distribution, as larger values of $\det(\boldsymbol{\Sigma})$ correspond to flatter distributions.

Link with the annealed mixture weights. Now, for a general function ℓ with modes $\{\xi_1^*, \dots, \xi_I^*\}$, Theorem 2.2 establishes that the weight \tilde{c}_i assigned to the mode ξ_i^* by the annealed Dirac mixture is proportional to $\zeta(\xi_i^*)$. Indeed,

$$\tilde{c}_i \propto \det(-\nabla^2 \ell(\xi_i^*))^{-1/2} = \left(\prod_{j=1}^d (-\lambda_j) \right)^{-1/2} = \zeta(\xi_i^*)^{d/2}.$$

Therefore, under the non-degeneracy assumption, the annealed Dirac mixture assigns greater weight to flatter maxima, as measured by ζ .

Degenerate modes. The Hessian matrix $\nabla^2 \ell$ is negative semi-definite at maxima, meaning that the curvature of ℓ is non-positive in all directions. A zero eigenvalue indicates complete flatness along the corresponding eigenvector direction. Therefore, the number of zero eigenvalues determines the dimension of the subspace on which the mode is completely flat. This can be interpreted as the “degree of flatness” of the mode, with modes having more zero eigenvalues being considered flatter. The concept of degree of flatness helps clarify the interpretation of ζ . In particular, ζ measures flatness only for non-degenerate modes (where the “degree of flatness” is 0). For degenerate modes, where $\zeta(\xi_i^*) = \infty$, ζ views them as “infinitely” flatter than non-degenerate ones.

Beyond the non-degeneracy assumption. While Theorem 2.2 excludes degenerate cases, we can provide an intuitive explanation of the annealing process when degenerate modes are present. Since the weights of the Dirac mixture are related to the flatness index ζ , non-degenerate modes will have comparable curvatures. In contrast, when degenerate modes are present, the weights of the Dirac mixture are predominantly assigned to the modes with the highest degree of flatness, as these are “infinitely” flatter than modes with a lower degree of flatness. The exact weights depend on the higher-order derivatives of ℓ . Deriving these explicitly is beyond the scope of this paper, which focuses on the non-degenerate case that is applicable to most scenarios.

F Link with Bayesian inference

The main concepts of NVA-M are the variational formulation of the initial optimization problem giving rise to Gibbs measures, their approximation in the space of mixtures of exponential family distributions using natural gradients, and annealing of the temperature parameter. In this section, we show that they admit a Bayesian interpretation. Then, we conclude that our optimization framework can be seen as a generalized Bayesian problem where the prior is flat.

F.1 Variational formulation and generalized posterior

Bayesian approaches to statistical inference consist in updating prior beliefs about an event A in light of an observable event B . Given a prior probability $\mathbb{P}(A)$, Bayesian inference seeks to determine the posterior probability $\mathbb{P}(A | B)$, which is governed by Bayes’ rule:

$$\mathbb{P}(A | B) = \frac{\mathbb{P}(B | A)\mathbb{P}(A)}{\mathbb{P}(B)}.$$

In many cases, prior probabilities are represented by a distribution $p(\xi)$ over a parameter of interest $\xi \in E$. For a set of observables X_1, \dots, X_n , the posterior distribution of ξ is

$$q_p^*(\xi) = \frac{L(\xi; X_1, \dots, X_n)p(\xi)}{\int L(\xi; X_1, \dots, X_n)p(\xi)d\xi}, \quad (18)$$

where $L(\boldsymbol{\xi}; X_1, \dots, X_n)$ denotes the likelihood, defined as the joint distribution of observations X_1, \dots, X_n given the parameter $\boldsymbol{\xi}$. It is well-established (Zellner, 1988) that the posterior q_p^* also satisfies

$$q_p^* = \arg \min_{q \in \mathcal{P}(E)} -\mathbb{E}_q[\log L(\boldsymbol{\xi}; X_1, \dots, X_n)] + \text{KL}(q \parallel p),$$

where $\mathcal{P}(E)$ represents the space of probability distributions on E , and $\text{KL}(q \parallel p)$ is the Kullback–Leibler divergence between q and p , defined as:

$$\text{KL}(q \parallel p) = \int q(\boldsymbol{\xi}) \log \left(\frac{q(\boldsymbol{\xi})}{p(\boldsymbol{\xi})} \right) d\boldsymbol{\xi}.$$

This formulation casts posterior inference as an optimization problem, where the objective is to balance the trade-off between two competing terms. The first expectation term represents the expected negative log-likelihood, measuring the discrepancy between the distribution q and the observed data. The second term $\text{KL}(q \parallel p)$ quantifies the information loss incurred by approximating the prior p with the distribution q , therefore favoring distributions close to the prior.

Remarkably, this optimization-based view of posterior inference can be generalized by replacing the negative log-likelihood with a loss function \mathfrak{L} , often referred to as a quasi-likelihood. Reformulating the minimization problem as a maximization problem yields:

$$q_p^* = \arg \max_{q \in \mathcal{P}(E)} \mathbb{E}_q[\ell(\boldsymbol{\xi})] - \text{KL}(q \parallel p), \quad (19)$$

where $\ell(\boldsymbol{\xi}) = -\mathfrak{L}(\boldsymbol{\xi}; X_1, \dots, X_n)$. Recent studies have increasingly adopted this generalized approach to updating prior beliefs, as it offers greater flexibility compared to the traditional Bayesian framework (Bissiri et al., 2016; Fong and Holmes, 2020; Rigon et al., 2023; Agnoletto et al., 2023). In this context, the resulting q_p^* is referred to as a *generalized posterior*.

F.2 Gibbs posteriors and annealing

Many Bayesian methods use a *Gibbs posterior* defined by

$$q_p^{*,\omega}(\boldsymbol{\xi}) = \frac{\exp(\ell(\boldsymbol{\xi})/\omega)p(\boldsymbol{\xi})}{\int \exp(\ell(\boldsymbol{\xi})/\omega)p(\boldsymbol{\xi})d\boldsymbol{\xi}}, \quad (20)$$

where $\omega > 0$. Gibbs posteriors are also commonly referred to as the tempered posterior (Andersen et al., 2002; Girolami, 2008), fractional posterior (O’Hagan, 1995; Gilks, 1995) or power posterior (Friel and Pettitt, 2008) and have been utilized in a wide range of statistical methodologies. For instance, they have been used to incorporate historical data in Bayesian data analyses (Ibrahim and Chen, 2000), offering a robust, but flexible, mechanism for incorporating prior knowledge. Furthermore, in scenarios where the model is misspecified, i.e. the true data-generating process lies outside of the support of the prior, it has been shown that Gibbs posteriors can outperform the traditional likelihood-based approach in terms of risk minimization (Jiang and Tanner, 2008).

The role of ω is clear when comparing (20) with the generalized Bayesian update, derived from (18). Specifically, ω modulates the influence of the information derived from the data relative to the prior distribution, effectively controlling how fast the model “learns” from the data. This relationship becomes even more apparent in the variational formulation of the Gibbs posterior, given by:

$$q_p^{*,\omega} = \arg \max_{q \in \mathcal{P}(E)} \mathbb{E}_q[\ell(\boldsymbol{\xi})] - \omega \text{KL}(q \parallel p). \quad (21)$$

Here, if $\omega = 1$, (20) reduces to the generalized Bayesian update (19). If $\omega > 1$, the loss of information with respect to the prior is given greater importance than the data. Conversely, when $0 < \omega < 1$, the data is given a stronger influence than in the usual Bayesian framework.

The process of powering up the quasi-likelihood in (20) draws a parallel with thermodynamic systems in statistical physics, where the probability of the state is given by a Gibbs measure. For this reason, the parameter ω is also commonly referred to as the temperature in the Bayesian literature, and the process of adjusting the temperature is also called annealing (Geman and Geman, 1984). This analogy underscores the foundational role of ω in modulating the influence of prior information versus new data, akin to controlling the temperature in a physical system.

F.3 Posterior approximation by variational inference

In classical Bayesian inference, the prior distribution is often selected from a family of distributions that is conjugate to the likelihood function. In this case, the posterior distribution remains in the same family as the prior, enabling analytical computation of the posterior parameters. However, in modern Bayesian problems, this assumption of conjugacy is not always feasible. In many scenarios, the posterior distribution is intractable. To address this challenge, one common approach is to approximate the posterior by a distribution using a member of a parametric family of distributions \mathcal{Q} . For example, Laplace’s approximation can be used to approximate the posterior with a Gaussian distribution, providing tractable solutions (Tierney et al., 1989).

An alternative approach uses the variational formulation presented by (21). Consider a Gibbs posterior $q_p^{*,\omega}$ solving (21) for some value of $\omega > 0$. Since the optimization problem defined by (21) is infinite-dimensional, deriving an algorithm to find $q_p^{*,\omega}$ is not straightforward. To overcome this, one may approximate $q_p^{*,\omega}$ with a distribution $\tilde{q}_p^{*,\omega}$ from the parametric family of probability distributions $\mathcal{Q} = (q_\Lambda)_{\Lambda \in \Theta}$, solving the constrained optimization problem

$$\tilde{q}_p^{*,\omega} = \arg \min_{q \in \mathcal{Q}} \text{KL}(q, q_p^{*,\omega}).$$

It is well-established that the solution $\tilde{q}_p^{*,\omega}$ is also the solution of the variational problem defined by (21), but constrained to the parametric family \mathcal{Q} :

$$\tilde{q}_p^{*,\omega} = \arg \max_{q \in \mathcal{Q}} \mathbb{E}_q[\ell(\xi)] - \omega \text{KL}(q \parallel p).$$

Usually, the parametric family \mathcal{Q} is chosen to simplify the optimization problem by reducing it from an infinite-dimensional problem to a finite-dimensional one involving the parameters of the family. Consequently, an algorithm can be derived to find the optimal parameters

$$\Lambda^{*,\omega} = \arg \max_{\Lambda \in \Theta} \mathbb{E}_{q_\Lambda}[\ell(\xi)] - \omega \text{KL}(q_\Lambda \parallel p), \quad (22)$$

where the resulting approximate posterior distribution is given by $\tilde{q}_p^{*,\omega} = q_{\Lambda^{*,\omega}}$. This methodology defines variational inference and the chosen parametric family \mathcal{Q} is referred to as the variational family.

F.4 Link with the initial optimization problem

Our optimization problem (P1) is not a Bayesian problem in nature. However, our approach to solving it is inspired by the Bayesian principles discussed above. Now, we provide an intuitive exploration of the behavior of a Gibbs posterior as the temperature parameter ω approaches zero.

From the expression of the Gibbs posterior $q_p^{*,\omega}$ given by (20), it can be observed that $q_p^{*,\omega}$ increasingly concentrates around the global modes of ℓ when the temperature ω decreases. Therefore, when ω tends to 0, the Gibbs posterior becomes a very sharp distribution, resembling a mixture of Dirac measures centered on the global modes of ℓ , provided all these modes lie within the support of the prior distribution p . Therefore, the modes of ℓ can be identified by locating the regions where $q_p^{*,\omega}$ remains non-negligible.

Our objective is to exploit this phenomenon by approximating $q_p^{*,\omega}$ with a Gaussian mixture model. Provided the number of mixture components is at least equal to the number of global modes of ℓ , then each component should fit under one of the modes. Thus, the modes of ℓ can be effectively recovered as the means of the mixture components. To help the convergence of the variational parameters during the optimization process, we propose the use of an annealing schedule for the temperature ω . In particular, at each iteration t , we sequentially fit q_p^{*,ω_t} , where the temperature ω_t gradually decreases to 0.

Given the absence of a compelling reason to select a specific prior, we opt for a uniform prior by default. Under a uniform prior, the Kullback–Leibler divergence term simplifies to $\text{KL}(q_{\Lambda} \parallel p) = -\mathcal{H}(q_{\Lambda})$, where $\mathcal{H}(q_{\Lambda})$ denotes the entropy of q_{Λ} . It is also noteworthy that with a uniform prior, the modes of $q_p^{*,\omega}$ remain invariant with respect to the value of $\omega > 0$, which is a property that may not hold for other prior distributions.

G Natural gradient optimization and information geometry

We recall some properties of the natural gradient and its relation with information geometry, motivating its use. More details can be found in [Khan and Rue \(2023\)](#).

Limitation of the traditional gradient approach. Let us first examine the (vanilla) gradient update rule to solve problem (P3b):

$$\Lambda_{t+1} = \Lambda_t + \rho_t \nabla_{\Lambda} \mathcal{L}_{\omega}(\Lambda)|_{\Lambda_t},$$

which is the update obtained when solving:

$$\Lambda_{t+1} = \arg \min_{\Lambda} \langle \nabla_{\Lambda} \mathcal{L}_{\omega}(\Lambda)|_{\Lambda_t}, \Lambda \rangle + \frac{1}{2\rho_t} \|\Lambda - \Lambda_t\|_2^2.$$

This makes clear that the update is driven by an Euclidean penalty in the parameter space. However, since the goal is to update Λ such that q_{Λ} converges to g_{ω} , it is more appropriate to use a penalty based on a distributional metric rather than a Euclidean metric in the parameter space. The Euclidean distance, which depends on the parameterization, is not an ideal metric for evaluating the proximity between distributions.

Natural gradients and link to information geometry. As discussed in [Khan and Rue \(2023\)](#), using a KL divergence penalty leads to the update rule

$$\Lambda_{t+1} = \arg \min_{\Lambda} \langle \nabla_{\Lambda} \mathcal{L}_{\omega}(\Lambda)|_{\Lambda_t}, \Lambda \rangle + \frac{1}{2\rho_t} \text{KL}(q_{\Lambda} \parallel q_{\Lambda_t}),$$

which results in (8). Typically, the natural gradient amounts to scaling the vanilla gradient by the inverse of the Fisher Information Matrix (FIM). However, for mixtures of exponential family distributions, the FIM associated with the density function $q_{\Lambda}(\xi)$ may be singular. Therefore,

it is more convenient to define the natural gradient using the joint distribution $q_{\Lambda}(\xi, Z)$ of the MCEF (5). The natural gradient is then given by (6).

The natural gradient is narrowly linked to information geometry (Bonnabel, 2013). The core idea behind information geometry is to use the shape of the manifold induced by the set of mixtures in the space of all probability distributions, which is independent of their parameterization (Amari and Nagaoka, 2000; Ollivier et al., 2017; Nielsen, 2020). The natural gradient points in the direction of steepest ascent on the Riemannian manifold with respect to the metric induced by the FIM, offering a faster convergence in variational inference (Sato, 2001; Honkela et al., 2008; Hoffman et al., 2013; Wu and Gardner, 2024). This motivates the choice of natural gradient ascent for solving our optimization problem.

Inexact Riemannian gradient ascent. Natural gradient ascent can be viewed as a form of inexact Riemannian gradient ascent, as it only uses the first-order approximation of the geodesic defined by the FIM, i.e. the direction of steepest ascent (Bonnabel, 2013). Indeed, at each iteration t , the update Λ_{t+1} is achieved by taking a step ρ_t forward from Λ_t , following a straight line rather than the true curved geodesic. However, this approximation is only valid within a neighborhood with a small radius. Typically, if the step-size ρ_t is too large, the positive-definite constraint of the precision matrices can be violated. Lin et al. (2020) suggested an improved learning rule (*improved Bayesian Learning Rule*, iBLR) using a second-order approximation of the geodesic, which can be easily obtained in the case of exponential family distributions and their mixtures through a retraction map. The iBLR update ensures that the precision matrices remain positive-definite. In this paper, we have also derived the iBLR update rule to solve our optimization problem.

H Identities for gradient derivation

This section proves some identities used in Sections 3 and 4 and Appendix J. They mostly come from Lin et al. (2019a) and Carreira-Perpiñán (2000).

Proposition H.1. *Let $q_{\theta}(\xi)$ be a probability density function parameterized by $\theta \in \Theta$. We have*

$$\nabla_{\theta} \mathbb{E}_{q_{\theta}}[\log q_{\theta}(\xi)]|_{\theta=\theta_0} = \int \log q_{\theta_0}(\xi) \nabla_{\theta} q_{\theta}(\xi)|_{\theta=\theta_0} d\xi = \nabla_{\theta} \mathbb{E}_{q_{\theta}}[\log q_{\theta_0}(\xi)]|_{\theta=\theta_0}.$$

Proof. The result is obtained when developing

$$\begin{aligned} \nabla_{\theta} \mathbb{E}_{q_{\theta}}[\log q_{\theta}(\xi)]|_{\theta=\theta_0} &= \int \nabla_{\theta} (q_{\theta}(\xi) \log q_{\theta}(\xi))|_{\theta=\theta_0} d\xi \\ &= \int (\nabla_{\theta} q_{\theta}(\xi)|_{\theta=\theta_0}) \log q_{\theta_0}(\xi) d\xi + \mathbb{E}_{q_{\theta_0}}[\nabla_{\theta} \log q_{\theta}(\xi)|_{\theta=\theta_0}] \\ &= \int \log q_{\theta_0}(\xi) \nabla_{\theta} q_{\theta}(\xi)|_{\theta=\theta_0} d\xi, \end{aligned}$$

where we have used the fact that $\mathbb{E}_{q_{\theta_0}}[\nabla_{\theta} \log q_{\theta}(\xi)|_{\theta=\theta_0}] = \int \nabla_{\theta} q_{\theta}(\xi)|_{\theta=\theta_0} d\xi = 0$. \square

Proposition H.2. *Let $\xi \in \mathbb{R}^d$. We have*

$$\begin{aligned} \nabla_{\mu} \mathcal{N}(\xi; \mu, \Sigma) &= \Sigma^{-1}(\xi - \mu) \mathcal{N}(\xi; \mu, \Sigma), \\ \nabla_{\Sigma} \mathcal{N}(\xi; \mu, \Sigma) &= \frac{1}{2} \{ \Sigma^{-1}(\xi - \mu)(\xi - \mu)^T \Sigma^{-1} - \Sigma^{-1} \} \mathcal{N}(\xi; \mu, \Sigma). \end{aligned}$$

Proof. We have

$$\mathcal{N}(\boldsymbol{\xi}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2} \det(\boldsymbol{\Sigma})^{1/2}} \exp \left\{ -\frac{1}{2} \|\boldsymbol{\xi} - \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}}^2 \right\}.$$

The first equality is obtained after remarking that $\nabla_{\boldsymbol{\mu}} \|\boldsymbol{\xi} - \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}}^2 = 2\boldsymbol{\Sigma}^{-1}(\boldsymbol{\xi} - \boldsymbol{\mu})$.

For the second equality, we have to use the fact that $\nabla_{\boldsymbol{\Sigma}} \det(\boldsymbol{\Sigma}) = \det(\boldsymbol{\Sigma})\boldsymbol{\Sigma}^{-1}$ and that $\nabla_{\boldsymbol{\Sigma}} \|\boldsymbol{\xi} - \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}}^2 = -\boldsymbol{\Sigma}^{-1}(\boldsymbol{\xi} - \boldsymbol{\mu})(\boldsymbol{\xi} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}$, so that

$$\begin{aligned} \nabla_{\boldsymbol{\Sigma}} \mathcal{N}(\boldsymbol{\xi}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \frac{1}{(2\pi)^{d/2}} \nabla_{\boldsymbol{\Sigma}} \det(\boldsymbol{\Sigma})^{-1/2} \exp \left\{ -\frac{1}{2} \|\boldsymbol{\xi} - \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}}^2 \right\} \\ &\quad + \frac{1}{(2\pi)^{d/2} \det(\boldsymbol{\Sigma})^{1/2}} \nabla_{\boldsymbol{\Sigma}} \exp \left\{ -\frac{1}{2} \|\boldsymbol{\xi} - \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}}^2 \right\} \\ &= -\frac{\det(\boldsymbol{\Sigma})\boldsymbol{\Sigma}^{-1}}{(2\pi)^{d/2} \times 2 \det(\boldsymbol{\Sigma})^{3/2}} \exp \left\{ -\frac{1}{2} \|\boldsymbol{\xi} - \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}}^2 \right\} \\ &\quad + \frac{\boldsymbol{\Sigma}^{-1}(\boldsymbol{\xi} - \boldsymbol{\mu})(\boldsymbol{\xi} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}}{(2\pi)^{d/2} \det(\boldsymbol{\Sigma})^{1/2} \times 2} \exp \left\{ -\frac{1}{2} \|\boldsymbol{\xi} - \boldsymbol{\mu}\|_{\boldsymbol{\Sigma}}^2 \right\} \\ &= \frac{1}{2} \{ \boldsymbol{\Sigma}^{-1}(\boldsymbol{\xi} - \boldsymbol{\mu})(\boldsymbol{\xi} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1} \} \mathcal{N}(\boldsymbol{\xi}; \boldsymbol{\mu}, \boldsymbol{\Sigma}). \end{aligned}$$

□

Proposition H.3. Let $q_{\boldsymbol{\Lambda}}(\boldsymbol{\xi}) = \sum_{k=1}^K q_{\boldsymbol{\lambda}_k}(\boldsymbol{\xi})$ where for all $k \in [K]$, $q_{\boldsymbol{\lambda}_k}(\boldsymbol{\xi}) := \mathcal{N}(\boldsymbol{\xi}; \boldsymbol{\mu}_k, \boldsymbol{S}_k^{-1})$. For all $k \in [K]$, let $r_k(\boldsymbol{\xi}) := \pi_k q_{\boldsymbol{\lambda}_k}(\boldsymbol{\xi}) / q_{\boldsymbol{\Lambda}}(\boldsymbol{\xi})$ and, in addition, for all $\ell \in [K]$, $A_{\ell k}(\boldsymbol{\xi}) := \boldsymbol{S}_{\ell}(\boldsymbol{\mu}_{\ell} - \boldsymbol{\xi})(\boldsymbol{\mu}_k - \boldsymbol{\xi})^T \boldsymbol{S}_k$. Then, we have

$$\begin{aligned} \nabla_{\boldsymbol{\xi}} \log q_{\boldsymbol{\Lambda}}(\boldsymbol{\xi}) &= \sum_{k=1}^K r_k(\boldsymbol{\xi}) \boldsymbol{S}_k(\boldsymbol{\mu}_k - \boldsymbol{\xi}), \\ \nabla_{\boldsymbol{\xi}}^2 \log q_{\boldsymbol{\Lambda}}(\boldsymbol{\xi}) &= \sum_{k=1}^K r_k(\boldsymbol{\xi}) \left[A_{kk}(\boldsymbol{\xi}) - \boldsymbol{S}_k - \sum_{\ell=1}^K r_{\ell}(\boldsymbol{\xi}) A_{\ell k}(\boldsymbol{\xi}) \right] \\ &= -(\nabla_{\boldsymbol{\xi}} \log q_{\boldsymbol{\Lambda}}(\boldsymbol{\xi})) (\nabla_{\boldsymbol{\xi}} \log q_{\boldsymbol{\Lambda}}(\boldsymbol{\xi}))^T + \sum_{k=1}^K r_k(\boldsymbol{\xi}) [A_{kk}(\boldsymbol{\xi}) - \boldsymbol{S}_k]. \end{aligned}$$

Proof. Let us first write that, by Proposition H.2,

$$\nabla_{\boldsymbol{\xi}} q_{\boldsymbol{\lambda}_k}(\boldsymbol{\xi}) = \nabla_{\boldsymbol{\xi}} \mathcal{N}(\boldsymbol{\xi}; \boldsymbol{\mu}_k, \boldsymbol{S}_k^{-1}) = \nabla_{\boldsymbol{\xi}} \mathcal{N}(\boldsymbol{\mu}_k; \boldsymbol{\xi}, \boldsymbol{S}_k^{-1}) = \boldsymbol{S}_k(\boldsymbol{\mu}_k - \boldsymbol{\xi}) q_{\boldsymbol{\lambda}_k}(\boldsymbol{\xi}). \quad (23)$$

The first identity comes from (23), as we have

$$\nabla_{\boldsymbol{\xi}} \log q_{\boldsymbol{\Lambda}}(\boldsymbol{\xi}) = \sum_{k=1}^K \pi_k \frac{\nabla_{\boldsymbol{\xi}} q_{\boldsymbol{\lambda}_k}(\boldsymbol{\xi})}{q_{\boldsymbol{\Lambda}}(\boldsymbol{\xi})} = \sum_{k=1}^K \frac{q_{\boldsymbol{\lambda}_k}(\boldsymbol{\xi})}{q_{\boldsymbol{\Lambda}}(\boldsymbol{\xi})} \boldsymbol{S}_k(\boldsymbol{\mu}_k - \boldsymbol{\xi}).$$

In order to obtain the second identity, we can differentiate the first identity and use (23) again,

so that

$$\begin{aligned}
& \nabla_{\xi}^2 \log q_{\Lambda}(\xi) \\
&= \sum_{k=1}^K \pi_k \left\{ \frac{\nabla_{\xi}(S_k(\mu_k - \xi)q_{\lambda_k}(\xi))}{q_{\Lambda}(\xi)} - \frac{\nabla_{\xi}q_{\Lambda}(\xi)q_{\lambda_k}(\xi)(\mu_k - \xi)^T S_k}{q_{\Lambda}(\xi)^2} \right\} \\
&= \sum_{k=1}^K \pi_k \frac{q_{\lambda_k}(\xi)}{q_{\Lambda}(\xi)} \left\{ \frac{\nabla_{\xi}q_{\lambda_k}(\xi)(\mu_k - \xi)^T S_k}{q_{\lambda_k}(\xi)} - S_k - \sum_{\ell=1}^K \pi_{\ell} \frac{\nabla_{\xi}q_{\lambda_{\ell}}(\xi)(\mu_k - \xi)^T S_k}{q_{\Lambda}(\xi)} \right\} \\
&= \sum_{k=1}^K \pi_k \frac{q_{\lambda_k}(\xi)}{q_{\Lambda}(\xi)} \left\{ S_k(\mu_k - \xi)(\mu_k - \xi)^T S_k - S_k - \sum_{\ell=1}^K \pi_{\ell} \frac{q_{\lambda_{\ell}}(\xi)}{q_{\Lambda}(\xi)} S_{\ell}(\mu_{\ell} - \xi)(\mu_k - \xi)^T S_k \right\}.
\end{aligned}$$

This also implies the alternative form of the second identity, by remarking that

$$(\nabla_{\xi} \log q_{\Lambda}(\xi)) (\nabla_{\xi} \log q_{\Lambda}(\xi))^T = \sum_{k=1}^K \sum_{\ell=1}^K \pi_k \pi_{\ell} \frac{q_{\lambda_k}(\xi)}{q_{\Lambda}(\xi)} \frac{q_{\lambda_{\ell}}(\xi)}{q_{\Lambda}(\xi)} S_{\ell}(\mu_{\ell} - \xi)(\mu_k - \xi)^T S_k.$$

□

I Bonnet's and Price's theorems

In this section, we recall Bonnet's and Price's theorems (Bonnet, 1964; Price, 1958; Opper and Archambeau, 2009; Rezende et al., 2014; Lin et al., 2019a), formulated as consequences of Stein's lemma (Stein, 1981). Trading generality for clarity, the following statements require stronger, but simpler assumptions than in Lin et al. (2019b).

Theorem I.1 (Bonnet's theorem). *Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be a continuously differentiable function and $q(\xi) := \mathcal{N}(\xi; \mu, \Sigma)$ a multivariate Gaussian distribution. We have*

$$\nabla_{\mu} \mathbb{E}_q[f(\xi)] = \mathbb{E}_q[\nabla_{\xi} f(\xi)].$$

Theorem I.2 (Price's theorem). *Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be a twice continuously differentiable function and $q(\xi) := \mathcal{N}(\xi; \mu, \Sigma)$ a multivariate Gaussian distribution. We have*

$$\nabla_{\Sigma} \mathbb{E}_q[f(\xi)] = \mathbb{E}_q[\nabla_{\xi}^2 f(\xi)].$$

J Computation of natural gradients

In this section, we derive the natural gradient update rules and the gradient estimators for the variational mixtures, including when they are Gaussian mixtures (Sect. 3 and 4).

J.1 Natural gradient update rules for the mixture parameters

In the case of Gaussian mixtures, we have the natural parameters $\lambda_k = (\lambda_k^{(1)}, \lambda_k^{(2)}) := (S_k \mu_k, -S_k/2)$ and the expectation parameters $M_k := (m_k^{(1)}, m_k^{(2)}) = (\pi_k \mu_k, \pi_k(S_k^{-1} + \mu_k \mu_k^T))$ (see Lin et al., 2019a for details). Using the chain rule, it comes

$$\begin{aligned}
\nabla_{m_k^{(1)}} \mathcal{L}_{\omega}(\Lambda) &= \frac{1}{\pi_k} \left(\nabla_{\mu_k} \mathcal{L}_{\omega}(\Lambda) - 2(\nabla_{S_k^{-1}} \mathcal{L}_{\omega}(\Lambda)) \mu_k \right), \\
\nabla_{m_k^{(2)}} \mathcal{L}_{\omega}(\Lambda) &= \frac{1}{\pi_k} \nabla_{S_k^{-1}} \mathcal{L}_{\omega}(\Lambda).
\end{aligned}$$

Using (7) and substituting these gradients into the natural gradient update rule (8), we derive update equations for the parameters $\boldsymbol{\mu}_k$, \mathbf{S}_k and π_k :

$$\begin{aligned}\mathbf{S}_{k,t+1} &= \mathbf{S}_{k,t} - \frac{2\rho_t}{\pi_{k,t}} \nabla_{\mathbf{S}_k^{-1}} \mathcal{L}_\omega(\boldsymbol{\Lambda})|_{\boldsymbol{\Lambda}_t}, \\ \boldsymbol{\mu}_{k,t+1} &= \boldsymbol{\mu}_{k,t} + \frac{\rho_t}{\pi_{k,t}} \mathbf{S}_{k,t+1}^{-1} \nabla_{\boldsymbol{\mu}_k} \mathcal{L}_\omega(\boldsymbol{\Lambda})|_{\boldsymbol{\Lambda}_t}, \\ v_{k,t+1} &= v_{k,t} + \rho_t \nabla_{\pi_k} \mathcal{L}_\omega(\boldsymbol{\Lambda})|_{\boldsymbol{\Lambda}_t},\end{aligned}$$

where $v_{k,t} := \log(\pi_{k,t}/\pi_{K,t})$.

J.2 Expression for $\nabla_{\pi_k} \mathcal{L}_\omega(\boldsymbol{\Lambda})$

Proposition H.1 (Appendix H) provides the identity

$$\nabla_{\pi_k} \mathcal{L}_\omega(\boldsymbol{\Lambda}) = \int (\nabla_{\pi_k} q_{\boldsymbol{\Lambda}}(\boldsymbol{\xi})) f_\omega(\boldsymbol{\xi}; \boldsymbol{\Lambda}) d\boldsymbol{\xi}.$$

The gradient with respect to π_k depends only on $\nabla_{\pi_k} q_{\boldsymbol{\Lambda}}(\boldsymbol{\xi})$. Since $\pi_K = 1 - \sum_{k=1}^{K-1} \pi_k$, we have

$$\nabla_{\pi_k} q_{\boldsymbol{\Lambda}}(\boldsymbol{\xi}) = q_{\boldsymbol{\lambda}_k}(\boldsymbol{\xi}) - q_{\boldsymbol{\lambda}_K}(\boldsymbol{\xi}),$$

and thus,

$$\begin{aligned}\nabla_{\pi_k} \mathcal{L}_\omega(\boldsymbol{\Lambda}) &= \mathbb{E}_{q_{\boldsymbol{\lambda}_k}}[f_\omega(\boldsymbol{\xi}; \boldsymbol{\Lambda})] - \mathbb{E}_{q_{\boldsymbol{\lambda}_K}}[f_\omega(\boldsymbol{\xi}; \boldsymbol{\Lambda})] \\ &= \mathbb{E}_{\mathcal{N}(\boldsymbol{\mu}_k, \mathbf{S}_k^{-1})}[f_\omega(\boldsymbol{\xi}; \boldsymbol{\Lambda})] - \mathbb{E}_{\mathcal{N}(\boldsymbol{\mu}_K, \mathbf{S}_K^{-1})}[f_\omega(\boldsymbol{\xi}; \boldsymbol{\Lambda})].\end{aligned}\tag{24}$$

J.3 Expressions for $\nabla_{\mathbf{S}_k^{-1}} \mathcal{L}_\omega(\boldsymbol{\Lambda})$ and $\nabla_{\boldsymbol{\mu}_k} \mathcal{L}_\omega(\boldsymbol{\Lambda})$

To obtain the gradients with respect to the means and the precision matrices, we use Proposition H.1 again:

$$\begin{aligned}\nabla_{\mathbf{S}_k^{-1}} \mathcal{L}_\omega(\boldsymbol{\Lambda}) &= \int \left(\nabla_{\mathbf{S}_k^{-1}} q_{\boldsymbol{\Lambda}}(\boldsymbol{\xi}) \right) f_\omega(\boldsymbol{\xi}; \boldsymbol{\Lambda}) d\boldsymbol{\xi}, \\ \nabla_{\boldsymbol{\mu}_k} \mathcal{L}_\omega(\boldsymbol{\Lambda}) &= \int (\nabla_{\boldsymbol{\mu}_k} q_{\boldsymbol{\Lambda}}(\boldsymbol{\xi})) f_\omega(\boldsymbol{\xi}; \boldsymbol{\Lambda}) d\boldsymbol{\xi}.\end{aligned}$$

These expressions lead to a few equivalent identities, from which different estimators can be suggested.

Black-box method. Proposition H.2 provides expressions for the gradients derived from direct differentiation of $q_{\boldsymbol{\Lambda}}(\boldsymbol{\xi})$ under the integral with respect to \mathbf{S}_k^{-1} and $\boldsymbol{\mu}_k$. The following expressions are obtained:

$$\nabla_{\mathbf{S}_k^{-1}} \mathcal{L}_\omega(\boldsymbol{\Lambda}) = \frac{\pi_k}{2} \mathbb{E}_{\mathcal{N}(\boldsymbol{\mu}_k, \mathbf{S}_k^{-1})}[(\mathbf{S}_k(\boldsymbol{\xi} - \boldsymbol{\mu}_k)(\boldsymbol{\xi} - \boldsymbol{\mu}_k)^T \mathbf{S}_k - \mathbf{S}_k) f_\omega(\boldsymbol{\xi}; \boldsymbol{\Lambda})],\tag{25}$$

$$\nabla_{\boldsymbol{\mu}_k} \mathcal{L}_\omega(\boldsymbol{\Lambda}) = \pi_k \mathbb{E}_{\mathcal{N}(\boldsymbol{\mu}_k, \mathbf{S}_k^{-1})}[\mathbf{S}_k(\boldsymbol{\xi} - \boldsymbol{\mu}_k) f_\omega(\boldsymbol{\xi}; \boldsymbol{\Lambda})].\tag{26}$$

These expressions are consistent with those given in Section 2 of Wierstra et al. (2014) for the natural gradient with respect to a Gaussian distribution.

Bonnet’s and Price’s theorems. Instead of differentiating $q_{\Lambda}(\xi)$, Bonnet’s and Price’s theorems (Price, 1958; Bonnet, 1964; Lin et al., 2019b) can be used to find alternative expressions for these gradients. We recall these results in Appendix I. In fact, this amounts to a reparameterization trick, using the properties of the Gaussian distribution to transform the gradients with respect to the parameters into gradients with respect to ξ , and then performing integration by parts:

$$\nabla_{S_k^{-1}} \mathcal{L}_{\omega}(\Lambda) = \frac{\pi_k}{2} \mathbb{E}_{\mathcal{N}(\mu_k, S_k^{-1})} [S_k(\xi - \mu_k) \nabla_{\xi} f_{\omega}(\xi; \Lambda)], \quad (27)$$

$$\nabla_{\mu_k} \mathcal{L}_{\omega}(\Lambda) = \pi_k \mathbb{E}_{\mathcal{N}(\mu_k, S_k^{-1})} [\nabla_{\xi} f_{\omega}(\xi; \Lambda)]. \quad (28)$$

An alternative expression for $\nabla_{S_k^{-1}} \mathcal{L}_{\omega}(\Lambda)$ can be obtained by integrating by parts once more:

$$\nabla_{S_k^{-1}} \mathcal{L}_{\omega}(\Lambda) = \frac{\pi_k}{2} \mathbb{E}_{\mathcal{N}(\mu_k, S_k^{-1})} [\nabla_{\xi}^2 f_{\omega}(\xi; \Lambda)]. \quad (29)$$

A more detailed derivation for these expressions is given in Lin et al. (2019a), Appendix B.2.

J.4 Estimation of the gradients

The gradients expressed in the previous section can be estimated using Monte Carlo approximations from simulated samples. For some $t \geq 1$, $B \geq 1$, for all $k \in [K]$, let $\xi_1^{(k)}, \dots, \xi_B^{(k)} \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(\mu_{k,t}, S_{k,t}^{-1})$.

Estimating $\nabla_{\pi_k} \mathcal{L}_{\omega}(\Lambda)|_{\Lambda_t}$. We propose the following estimator for $\nabla_{\pi_k} \mathcal{L}_{\omega}(\Lambda)|_{\Lambda_t}$ defined by

$$\widehat{\nabla_{\pi_k} \mathcal{L}_{\omega}(\Lambda)}|_{\Lambda_t} = \widehat{\gamma}_{\omega, \Lambda_t, B}^{(\pi_k)} := \frac{1}{B} \sum_{b=1}^B \left(f_{\omega}(\xi_b^{(k)}; \Lambda_t) - f_{\omega}(\xi_b^{(K)}; \Lambda_t) \right), \quad (30)$$

which is unbiased and consistent from (24).

Estimating $\nabla_{\mu_k} \mathcal{L}_{\omega}(\Lambda)|_{\Lambda_t}$. We propose two estimators for $\nabla_{\mu_k} \mathcal{L}_{\omega}(\Lambda)|_{\Lambda_t}$ of the form $\widehat{\nabla_{\mu_k} \mathcal{L}_{\omega}(\Lambda)}|_{\Lambda_t} = \pi_{k,t} \widehat{\gamma}_{\omega, \Lambda_t, B}^{(\mu_k, i)} := \pi_{k,t} \widehat{\gamma}_{\omega, \Lambda_t, B}^{(\mu_k, i)}$, $i \in \{0, 1\}$, where

$$\widehat{\gamma}_{\omega, \Lambda_t, B}^{(\mu_k, 0)} := \frac{1}{B} S_{k,t} \sum_{b=1}^B (\xi_b^{(k)} - \mu_{k,t}) f_{\omega}(\xi_b^{(k)}; \Lambda_t), \quad (31)$$

and

$$\widehat{\gamma}_{\omega, \Lambda_t, B}^{(\mu_k, 1)} := \frac{1}{B} \sum_{b=1}^B \nabla_{\xi} f_{\omega}(\xi_b^{(k)}; \Lambda_t), \quad (32)$$

lead to unbiased and consistent estimators from (26) and (28) respectively.

Estimating $\nabla_{S_k^{-1}} \mathcal{L}_{\omega}(\Lambda)|_{\Lambda_t}$. We propose three estimators for $\nabla_{S_k^{-1}} \mathcal{L}_{\omega}(\Lambda)|_{\Lambda_t}$ of the form $\widehat{\nabla_{S_k^{-1}} \mathcal{L}_{\omega}(\Lambda)}|_{\Lambda_t} = \pi_{k,t} \widehat{\gamma}_{\omega, \Lambda_t, B}^{(S_k^{-1})} / 2 := \pi_{k,t} \widehat{\gamma}_{\omega, \Lambda_t, B}^{(S_k^{-1}, i)} / 2$, $i \in \{0, 1, 2\}$, where

$$\widehat{\gamma}_{\omega, \Lambda_t, B}^{(S_k^{-1}, 0)} := \frac{1}{B} S_{k,t} \sum_{b=1}^B \left((\xi_b^{(k)} - \mu_{k,t})(\xi_b^{(k)} - \mu_{k,t})^T S_{k,t} - I \right) f_{\omega}(\xi_b^{(k)}; \Lambda_t), \quad (33)$$

$$\hat{\gamma}_{\omega, \Lambda_t, B}^{(S_k^{-1}, 1)} := \frac{1}{B} \mathbf{S}_{k,t} \sum_{b=1}^B (\boldsymbol{\xi}_b^{(k)} - \boldsymbol{\mu}_{k,t}) \nabla_{\boldsymbol{\xi}} f_{\omega}(\boldsymbol{\xi}_b^{(k)}; \Lambda_t), \quad (34)$$

and

$$\hat{\gamma}_{\omega, \Lambda_t, B}^{(S_k^{-1}, 2)} := \frac{1}{B} \sum_{b=1}^B \nabla_{\boldsymbol{\xi}}^2 f_{\omega}(\boldsymbol{\xi}_b^{(k)}; \Lambda_t), \quad (35)$$

lead to unbiased and consistent estimators from (25), (27) and (29) respectively.

Computability of $\nabla_{\boldsymbol{\xi}} f_{\omega}$ and $\nabla_{\boldsymbol{\xi}}^2 f_{\omega}$. From Proposition H.3, the quantities $\log q_{\Lambda}(\boldsymbol{\xi})$, $\nabla_{\boldsymbol{\xi}} \log q_{\Lambda}(\boldsymbol{\xi})$ and $\nabla_{\boldsymbol{\xi}}^2 \log q_{\Lambda}(\boldsymbol{\xi})$ can be exactly computed. Therefore, the computability of the derivative of f_{ω} only depend on the computability of the corresponding derivatives of ℓ .

K Stochastic natural gradient ascent algorithm

This section gives the stochastic natural gradient ascent algorithm in the general mixture case (Algorithm A).

Algorithm A: Stochastic natural gradient ascent for mixture of exponential family variational approximation (SNGA-M)

```

1 GIVEN a function  $\ell$  and a value for  $\omega$ .
2 SET  $T, B, K, \Lambda_0, (\rho_t)_{1 \leq t \leq T}$ .
3 COMPUTE  $(v_{k,0}) = (\log(\pi_{k,0}/\pi_{K,t}))_{k \in [K]}$ .
4 for  $t = 0:(T-1)$  do
5   for  $k = 1:K$  do
6     SAMPLE  $\boldsymbol{\xi}_b^{(k)} \stackrel{\text{i.i.d.}}{\sim} q_{\boldsymbol{\lambda}_{k,t}}, \quad \text{for } b = 1:B.$ 
7     COMPUTE  $\widehat{\nabla_{M_k} \mathcal{L}_{\omega}}(\Lambda_t)$ .
8     UPDATE  $\boldsymbol{\lambda}_{k,t+1} = \boldsymbol{\lambda}_{k,t} + \rho_t \widehat{\nabla_{M_k} \mathcal{L}_{\omega}}(\Lambda_t)$ .
9   end
10  for  $k = 1:(K-1)$  do
11    COMPUTE  $\hat{\gamma}_{\omega, \Lambda_t, B}^{(\pi_k)}$ .
12    UPDATE  $v_{k,t+1} = v_{k,t} + \rho_t \hat{\gamma}_{\omega, \Lambda_t, B}^{(\pi_k)}$ .
13  end
14 end
15 COMPUTE  $(\pi_{k,T})_{k \in [K]}$  from  $(v_{k,T})_{k \in [K]}$ .
16 return  $\Lambda_T$ .
```

L Behavior of a single Gaussian parameters when annealing

In this section, we provide the proof of Proposition 4.1, stated in Section 4.1.

Proof of Proposition 4.1. For some t , the parameters of $q_{\theta_1^*, \omega_t} = \mathcal{N}(\boldsymbol{\mu}_1^{*, \omega_t}, (\mathbf{S}_1^{*, \omega_t})^{-1})$ minimize

$\text{KL}(q_{\theta_1^*, \omega_t} \parallel g_{\omega_t})$. They can be found by solving the equation system

$$\begin{cases} \nabla_{\mu_1} \text{KL}(q_{\theta_1} \parallel g_{\omega}) = 0 \\ \nabla_{S_1^{-1}} \text{KL}(q_{\theta_1} \parallel g_{\omega}) = 0 \end{cases}.$$

We also recall that

$$\text{KL}(q_{\theta_1} \parallel g_{\omega}) = -\mathcal{H}(q_{\theta_1}) - \mathbb{E}_{q_{\theta_1}} \left[\frac{\ell(\xi)}{\omega} - \log Z(\omega) \right],$$

where $\mathcal{H}(q_{\theta_1}) = d(1 + \log(2\pi))/2 - \log(\det(\mathbf{S})) / 2$.

First, we prove that solutions $(q_{\theta_1^*, \omega_t})_{t \rightarrow \infty}$ converge to a Dirac measure centered on ξ^* as $t \rightarrow \infty$. Using Price's theorem (Appendix I), we have

$$\nabla_{S_1^{-1}} \text{KL}(q_{\theta_1} \parallel g_{\omega}) = -\frac{1}{2} \mathbf{S}_1 - \nabla_{S_1^{-1}} \mathbb{E}_{q_{\theta_1}} \left[\frac{\ell(\xi)}{\omega} \right] = -\frac{1}{2} \mathbf{S}_1 - \frac{1}{2} \mathbb{E}_{q_{\theta_1}} \left[\frac{\nabla_{\xi}^2 \ell(\xi)}{\omega} \right].$$

So for all $t \geq 1$, $\mathbf{S}_1^{*, \omega_t}$ satisfies

$$\mathbf{S}_1^{*, \omega_t} = -\frac{1}{\omega_t} \mathbb{E}_{\mathcal{N}(\mu_1^{*, \omega_t}, (S_1^{*, \omega_t})^{-1})} [\nabla_{\xi}^2 \ell(\xi)].$$

Because ℓ is strictly concave, the eigenvalues of $-\nabla_{\xi}^2 \ell(\xi)$ are strictly positive for all ξ . This implies that the eigenvalues of any $\mathbf{S}_1^{*, \omega_t}$ cannot be bounded over t because $\omega_t \xrightarrow[t \rightarrow \infty]{} 0$, so we deduce that $(\mathbf{S}_1^{*, \omega_t})^{-1} \xrightarrow[t \rightarrow \infty]{} 0$.

Furthermore, using Bonnet's theorem (Appendix I), we have

$$\nabla_{\mu_1} \text{KL}(q_{\theta_1} \parallel g_{\omega}) = \nabla_{\mu} \mathbb{E}_{q_{\theta_1}} \left[\frac{\ell(\xi)}{\omega} \right] = \mathbb{E}_{q_{\theta_1}} \left[\frac{\nabla_{\xi} \ell(\xi)}{\omega} \right].$$

Therefore, μ_1^{*, ω_t} must satisfy

$$\mathbb{E}_{\mathcal{N}(\mu_1^{*, \omega_t}, (S_1^{*, \omega_t})^{-1})} [\nabla_{\xi} \ell(\xi)] = 0.$$

Since ξ^* is the only critical point of ℓ and $(\mathbf{S}_1^{*, \omega_t})^{-1} \xrightarrow[t \rightarrow \infty]{} 0$, then we have $\mu_1^{*, \omega_t} \xrightarrow[t \rightarrow \infty]{} \xi^*$. We also deduce that the sequence $(q_{\theta_1^*, \omega_t})_{t \geq 1}$ converges weakly to the Dirac measure centered on ξ^* as $t \rightarrow \infty$.

Finally, we use this weak convergence result and the following expression, already shown above

$$\mathbf{S}_1^{*, \omega_t} = -\frac{1}{\omega_t} \mathbb{E}_{q_{\theta_1^*, \omega_t}} [\nabla_{\xi}^2 \ell(\xi)],$$

to conclude that

$$\omega_t \mathbf{S}_1^{*, \omega_t} \xrightarrow[t \rightarrow \infty]{} -\nabla_{\xi}^2 \ell(\xi^*).$$

□

M Behavior of Gaussian covariance matrices when annealing

In this section, we will see how $\mathbf{S}_k^{*, \omega} \underset{\omega \rightarrow 0}{\sim} -\omega^{-1} \nabla_{\xi}^2 \ell(\xi_k^*)$ for all $k \in [K]$. We use the notations of Section 4.1. First, we prove a preliminary result.

M.1 Component separation and entropy approximation lemma

The following lemma show that if the components of the mixture are separated, then the entropy of the mixture can be approximated by a sum of entropies associated to the components. In order to prove the validity of the approximation, we will pick a Gaussian mixture where components have fixed means and vanishing covariance matrices. We assume that the eigenvalues of a covariance matrix are decreasing to 0 uniformly within itself but also with respect to all the other covariance matrices.

Assumption M.1. *We assume that there exists*

- a positive function $\phi : h \mapsto \phi(h)$ with $\phi(h) \xrightarrow{h \rightarrow \infty} +\infty$,
- for $k \in [K]$, a matrix \mathbf{E}_k ,

such that

$$\phi(h)\Sigma_k(h) \xrightarrow{h \rightarrow \infty} \mathbf{E}_k.$$

Let $\alpha_{k,1}(h) \leq \dots \leq \alpha_{k,d}(h)$ denote the eigenvalues of $\Sigma_k(h)^{1/2}$ ordered in increasing order and analogously, $\beta_{k,1} \leq \dots \leq \beta_{k,d}$ the ordered eigenvalues of $\mathbf{E}_k^{1/2}$. A consequence is that for all $1 \leq i \leq d$, we have

$$\phi(h)\alpha_{k,i}(h) \xrightarrow{h \rightarrow \infty} \beta_{k,i}.$$

Lemma M.2. *Suppose that the mixture parameters Λ are constant except for the covariance matrices $\Sigma_k(h)$, for $k \in [K]$, converging accordingly to Assumption M.1. Then*

$$\mathcal{H}(q_\Lambda) \xrightarrow{h \rightarrow \infty} \tilde{\mathcal{H}}(q_\Lambda) := - \sum_{k=1}^K \pi_k \log \pi_k + \sum_{k=1}^K \pi_k \mathcal{H}(q_{\Lambda_k}).$$

Proof. Let $u_{k,k'} := \|\mu_k - \mu_{k'}\|_{\Sigma_k} / (1 + \|\Sigma_k^{-1/2} \Sigma_{k'}^{1/2}\|_{\text{op}})$ where $\|\Sigma\|_{\text{op}}$ is the operator norm of Σ , given by its largest eigenvalue.

Theorem 4.2 of [Furuya et al. \(2024\)](#) states that for $s \in (0, 1)$, we have

$$|\mathcal{H}(q_\Lambda) - \tilde{\mathcal{H}}(q_\Lambda)| \leq \min \left\{ \frac{K}{2}, \frac{2}{(1-s)^{d/4}} \sum_{k=1}^K \sum_{k' \neq k} \sqrt{\pi_k \pi_{k'}} \exp \left(\frac{-s u_{k,k'}}{4} \right) \right\}.$$

But we have $\|\Sigma_k^{-1/2} \Sigma_{k'}^{1/2}\|_{\text{op}} \leq \alpha_{k',d}(h) \alpha_{k,1}(h)$ and $\|\mu_k - \mu_{k'}\|_{\Sigma_k} \geq \|\mu_k - \mu_{k'}\|_2 \alpha_{k,d}(h)^{-1}$, so that

$$u_{k,k'} \geq \frac{\|\mu_k - \mu_{k'}\|_2}{\alpha_{k,1}(h) + \alpha_{k',d}(h)} \sim_{h \rightarrow \infty} \phi(h) \frac{\|\mu_k - \mu_{k'}\|_2}{\beta_{k,1} + \beta_{k',d}} \xrightarrow{h \rightarrow \infty} +\infty.$$

Therefore, we can conclude that $|\mathcal{H}(q_\Lambda) - \tilde{\mathcal{H}}(q_\Lambda)| \xrightarrow{h \rightarrow \infty} 0$. □

M.2 Asymptotic behavior of Gaussian covariance matrices

Now, we can explain how $\mathbf{S}_k^{*,\omega} \underset{\omega \rightarrow 0}{\sim} -\omega^{-1} \nabla_{\boldsymbol{\xi}}^2 \ell(\boldsymbol{\xi}_k^*)$ for all $k \in [K]$. If $\boldsymbol{\theta}^{*,\omega} \rightarrow \boldsymbol{\theta}^{*,0}$ as $\omega \rightarrow 0$, then for a sufficiently small ω , $\boldsymbol{\theta}^{*,\omega}$ is a mixture of very sharp Gaussian components, as their covariance matrices vanish when $\omega \rightarrow 0$. Following Lemma M.2, this implies that below a certain threshold of ω , the entropy of $q_{\boldsymbol{\theta}^{*,\omega}}$ can be approximately decomposed as

$$\mathcal{H}(q_{\boldsymbol{\theta}^{*,\omega}}) \approx - \sum_{k=1}^K \pi_k^{*,\omega} \log \pi_k^{*,\omega} + \sum_{k=1}^K \pi_k^{*,\omega} \mathcal{H}(\boldsymbol{\theta}_k^{*,\omega}).$$

Therefore,

$$\mathbb{E}_{q_{\boldsymbol{\theta}^{*,\omega}}}[\ell(\boldsymbol{\xi})] + \omega \mathcal{H}(q_{\boldsymbol{\theta}^{*,\omega}}) \approx \sum_{k=1}^K \pi_k^{*,\omega} \left(\mathbb{E}_{q_{\boldsymbol{\theta}_k^{*,\omega}}}[\ell(\boldsymbol{\xi})] - \omega \mathcal{H}(q_{\boldsymbol{\theta}_k^{*,\omega}}) \right).$$

This indicates that, for sufficiently small ω , problem (P4) can be nearly decomposed into K separated problems

$$\boldsymbol{\theta}_k^{*,\omega} \in \arg \max_{\boldsymbol{\theta}_k} \mathbb{E}_{q_{\boldsymbol{\theta}_k}}[\ell(\boldsymbol{\xi})] + \omega \mathcal{H}(q_{\boldsymbol{\theta}_k}).$$

Given that $\boldsymbol{\theta}^{*,\omega} \rightarrow \boldsymbol{\theta}^{*,0}$ as $\omega \rightarrow 0$, this suggests that in the mixture approximation problem (P4), if the components are well-separated and close to the modes of ℓ , then the behavior of each individual component will resemble that in the single-mode case as described by Proposition 4.1, hence, for all $k \in [K]$,

$$\mathbf{S}_k^{*,\omega} \underset{\omega \rightarrow 0}{\sim} -\omega^{-1} \nabla_{\boldsymbol{\xi}}^2 \ell(\boldsymbol{\xi}_k^*).$$

While this conclusion is based on theoretical considerations, its practical utility will be demonstrated for the choice of algorithm hyperparameters, more specifically the learning rate (see Appendix O.5).

N Interpretations of the NVA-GM algorithm

NVA-GM implements a natural gradient method to solve a variational problem, with an annealed objective function. The core idea presented was that in fitting a Gaussian mixture, each Gaussian component is likely to adjust to a location where the objective function is high in order to favor a high expectation in (P3). If there are as many modes of ℓ as components in the Gaussian mixture ($K = I$), then we hope that each Gaussian component takes responsibility for one mode of ℓ . Alternatively, we can look at this algorithm from other perspectives.

N.1 Particle interpretation

Khan and Rue (2023) proved that the *Bayesian Learning Rule*, consisting in solving the approximate Bayesian problem (22) with $\omega = 1$, using a variational family from the exponential family and natural gradients, can lead to classic optimization algorithms such as stochastic gradient algorithms and Newton–Raphson’s method. These algorithms iteratively update the position of a particle until convergence to an optimum. For example, in the case of Stochastic Gradient Ascent (SGA), the updates are of the form

$$\boldsymbol{\xi}_{t+1} = \boldsymbol{\xi}_t + \rho_t \nabla \ell(\boldsymbol{\xi}),$$

where $\ell(\boldsymbol{\xi})$ is the function to be maximized, and $\boldsymbol{\xi}_t$ is expected to converge to a local or global maximum of ℓ . Conceptually, the particle is subject to the potential ℓ , generating a driving force $\nabla \ell$, directed towards the sinks of the potential, i.e. the maxima of ℓ .

When the variational family in (22) is restricted to Gaussian distributions with fixed covariance, the optimization is only performed on the mean $\boldsymbol{\mu}$ of the Gaussian. In [Khan and Rue \(2023\)](#), it is shown that a delta-method approximation of the natural gradient leads to the update rule

$$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t + \rho_t \nabla \ell(\boldsymbol{\mu}_t).$$

Thus, the trajectory of the mean $(\boldsymbol{\mu}_t)_{t \geq 1}$ of the Gaussian distribution is completely equivalent to the trajectory of the particle $(\boldsymbol{\xi}_t)_{t \geq 1}$ in the SGA algorithm.

This analogy can be extended to Gaussian mixtures by considering the component means as individual particles, each moving according to forces generated by potentials. From this perspective, a Gaussian mixture with K components can be viewed as a system of K particles with locations $(\boldsymbol{\mu}_{k,t})_{k \in [K]}$ at time t , with the goal that they converge to the sinks of the potential ℓ . In our situation, the function to optimize at each iteration t is $\mathcal{L}_{\omega_t}(\boldsymbol{\Lambda}) = \mathbb{E}_{q_{\boldsymbol{\Lambda}}}[f_{\omega_t}(\boldsymbol{\xi}; \boldsymbol{\Lambda})]$, which can be interpreted as an average of a general potential $f_{\omega_t}(\boldsymbol{\xi}; \boldsymbol{\Lambda})$ (where $\boldsymbol{\xi} \sim q_{\boldsymbol{\Lambda}}$). This general potential can be decomposed into two components:

$$f_{\omega_t}(\boldsymbol{\xi}; \boldsymbol{\Lambda}) = \underbrace{\ell(\boldsymbol{\xi})}_{\text{driving potential}} + \omega_t \times \underbrace{(-\log q_{\boldsymbol{\Lambda}}(\boldsymbol{\xi}))}_{\text{repulsive potential}}. \quad (36)$$

To clarify the role of these two potentials, recall that the gradient of $\mathcal{L}_{\omega_t}(\boldsymbol{\Lambda})$ with respect to $\boldsymbol{\mu}_k$ is given by equation (26), which shows that the gradient is the weighted average of vectors $(\boldsymbol{\xi} - \boldsymbol{\mu}_k)$ where $\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{\mu}_k, \mathbf{S}_k)$ is sampled symmetrically around $\boldsymbol{\mu}_k$, with weights determined by the potential $f_{\omega_t}(\boldsymbol{\xi}; \boldsymbol{\Lambda})$. The decomposition of the potential makes it clear that $\ell(\boldsymbol{\xi})$ contributes positively to the weight, whereas $\log q_{\boldsymbol{\Lambda}}(\boldsymbol{\xi})$ contributes negatively. Consequently, the driving potential directs the gradient towards regions where ℓ is high, whereas the repulsive potential pushes it away from regions where $q_{\boldsymbol{\Lambda}}$ is high, i.e. away from the other component means of the mixture. Therefore, the covariance \mathbf{S}_k^{-1} of a component determines the strength of this repulsive force. Whereas the individual covariance affects the repulsion force of the particles individually, ω_t can be seen as a parameter setting the strength of the repulsive potential of all particles. This is analogous to many setups in statistical physics, where temperature influences the interparticle forces ([Israelachvili, 1992](#)).

If the aim of the algorithm were only to let the particles converge to the sinks of ℓ , the driving potential alone would achieve this. However, the inclusion of the repulsive potential, induced by the entropy term, forces the particles to spread out, which is crucial when we want them to converge to different sinks. A discussion on how one should balance the potentials over time through the specification of an annealing schedule is given in [Section 4.4](#).

N.2 Evolutionary interpretation

Evolutionary algorithms represent a class of black-box optimization algorithms. To optimize a function, called fitness function, with respect to certain parameters, these algorithms generate a population of individuals in the search space. Each individual is assigned a fitness value, typically the function's value at the parameters represented by that individual. At each iteration, a new generation of individuals is produced through a process that simulates mechanisms of natural evolution, specifically mutation, and selection. The mutation step involves randomly perturbing the current population to explore the parameter space, while the selection step uses the fitness values of the current population to generate new individuals with higher fitness, therefore focusing on already explored regions of the parameter space.

A subset of evolutionary algorithms, those within the Information-Geometric Optimization framework (IGO, [Ollivier et al., 2017](#)), maintains a connection with gradient-based optimization

methods. Notably, the Natural Evolution Strategies (NES, [Wierstra et al., 2014](#)) use a parameterized search distribution to generate populations. The parameters of this search distribution are updated based on an estimate of the search natural gradient, i.e. the natural gradient of the fitness function with respect to the search distribution. At each iteration, a new population is sampled under the current search distribution (mutation step), and these samples are used to estimate the search natural gradient, directing the search towards regions of the parameter space with higher expected fitness (selection step).

Our natural gradient algorithms exhibit analogous principles. At each iteration, it samples locations from each Gaussian component $\mathcal{N}(\boldsymbol{\mu}_k, \mathbf{S}_k^{-1})$ of the mixture. The covariance matrix \mathbf{S}_k^{-1} can be interpreted as the mutation strength for the corresponding search distribution, defining the dispersion of the samples. These samples are used to update the components from which they were drawn, by computing natural gradients. Thus, each component acts as a search distribution. Thus, the mixture induces K search distributions, enabling the identification of up to K distinct solutions. However, these search distributions are not updated independently, but they are interdependent due to the entropy term in our objective function, which involves the whole mixture. As seen earlier, the entropy term promotes spatial diversity between the components. Consequently, rather than acting completely independently, the different search distributions collaborate to find distinct solutions, with the temperature ω_t setting the strength of the collaboration. In the limiting case where $\omega_t = 0$, the search distributions are updated independently.

N.3 Interpretation of the annealing schedule

In this section, we analyze how the value of ω influences the behavior of the system of particles formed by the component means, particularly in terms of exploration versus exploitation. We identify and discuss two distinct regimes: when ω is very small and when ω is very large. We will then provide guidance on how to navigate between these two regimes to ensure that the particles behave well to solve our optimization problem.

Full-drive regime. We argue that when ω is too small in SNGA (Algorithm A), several particles are likely to converge to the same mode. To understand this, consider a scenario where $\omega \approx 0$ is sufficiently small so that the objective function can be approximated by this decomposition:

$$\mathcal{L}_\omega(\boldsymbol{\Lambda}) \approx \sum_{k=1}^K \pi_k \mathbb{E}_{q_{\lambda_k}}[\ell(\boldsymbol{\xi})].$$

In problem (P3), the entropy penalty, by introducing a repulsive force between the particles, prevents the optimization problem to be separable into K independent optimization sub-problems for each particle. However, when the entropy term is negligible, we can separate it: for $k \in [K]$,

$$\lambda_k = \arg \max_{\lambda} \mathbb{E}_{q_{\lambda}}[\ell(\boldsymbol{\xi})].$$

Thus, setting a very small value for ω amounts to running K parallel optimization algorithms. The result depends heavily on the initial parameters, like in most gradient algorithms, and the mixture components converge independently. In particular, the update for the component means (the positions of the particles) is identical for all $k \in [K]$, i.e.

$$\boldsymbol{\mu}_{k,t+1} = \boldsymbol{\mu}_{k,t} + \rho_t \tilde{\nabla}_{\boldsymbol{\mu}} \mathbb{E}_{\mathcal{N}(\boldsymbol{\mu}, \mathbf{S}^{-1})}[\ell(\boldsymbol{\xi})]_{(\boldsymbol{\mu}, \mathbf{S})=(\boldsymbol{\mu}_{k,t}, \mathbf{S}_{k,t})}.$$

Because all the particles follow the same dynamic, they are likely to converge to the same mode if their initial positions are close. Similar to classic gradient ascent, the landscape of ℓ can be

partitioned into basins of attractions associated with the potential’s sinks (the modes), where a particle initialized within a basin is driven to the corresponding sink. To ensure that particles converge to different modes, they must be initialized in different basins of attraction. However, even if we knew the number of modes in ℓ and their approximate locations, the exact boundaries of the basins are unknown, especially when there is no prior information on ℓ available. Setting the initial positions in well-spread locations of space might mitigate the problem in low-dimensional spaces, where there is a higher chance that they fall in different basins, but this becomes almost impossible in high-dimensional settings (for example $d \geq 3$), where the shape of these basins are more complex. This shows why using Algorithm A with a small ω is not suitable to find distinct solutions.

Equilibrium regime. When ω is non-negligible, the entropy term in (P3) plays a critical role in preserving component separation. Entropy is maximized when a distribution is uniform. For a Gaussian mixture, this implies that its entropy is higher when the mixture components are spread out, meaning the means are distant from each other and the variances are large. Recall from equation (36) that the potential governing the dynamics of the particles can be decomposed into two parts: a driving potential and a repulsive potential.

The gradient pulling the component means towards the modes is fully determined by the driving potential. Because the repulsive potential between the particles counteracts this driving force, a non-negligible ω prevents them from converging to the sinks. Instead, they converge to an equilibrium state where the driving force is compensated by the repulsive force, i.e. a set of parameters $\mathbf{\Lambda}$ where $f_\omega(\xi; \mathbf{\Lambda})$ is constant. Therefore, with a large ω , convergence of the particles is inconsistent, as the final locations of the means are not at the modes of ℓ , but rather at an equilibrium between the two potentials. Therefore, using Algorithm A with a large ω does not allow us to find the modes of ℓ . However, this repulsive potential can be used to force spatial separation between the particles, allowing them to escape one basin of attraction that is already occupied by one and move to another.

Exploration versus exploitation. Ultimately, we want the component means to converge to the modes of ℓ , and only canceling the repulsive force ($\omega \approx 0$) would achieve this. However, this also makes it likely that multiple means converge to the same mode. On the other hand, the repulsive force keeps particles spatially separated and encourages them to explore other regions of the search space, but it also prevents them from reaching the modes. The idea behind the annealing schedule in NVA-GM is to initially force the particles to spread out and explore the search space by setting a strong repulsive force (large ω), then gradually transition into a regime where the particles are fully subject to the driving force by decreasing ω over time.

To some extent, we can say that the annealing schedule is similar to an exploration-exploitation strategy, a classic analogy in applications of annealing that has already been used in optimization (Huang et al., 2018; D’Angelo and Fortuin, 2021b). Ideally, during the exploration phase, one should prioritize spatial separation of the components and exploration of the landscape of ℓ . Then, ω decreases, and the driving potential should progressively dominate the overall dynamics, until reaching a pure exploitation phase, where the particles are only guided by the landscape of ℓ . The transition phase allows the particles to be attracted to regions with high values of ℓ while still preserving some spatial separation. If the transition is slow enough, the particles are more likely to discover different basins of attraction, leading to convergence to distinct modes.

Setting an annealing schedule. Originally, ω is used to set the target g_ω of the optimization problem, like in Section 3. To implement annealing in Algorithm 1, a piecewise-constant decreasing schedule can be defined. First, set ω to a large value ω_1 and let the algorithm run until Λ^{*,ω_1} is reached. Then, reduce ω to a slightly smaller value $\omega_2 \leq \omega_1$, find Λ^{*,ω_2} and repeat until $\Lambda^{*,\omega_S} \approx \Lambda^{*,\omega_{S-1}}$. This approach essentially restarts the algorithm after convergence with a fixed ω , using the previous result as the initial condition for the next run.

However, this approach has several drawbacks. First, a criterion for convergence must be established before restarting with a new ω . There is no general guideline for choosing a relevant criterion. More importantly, because it is preferable to decrease ω slowly to ensure convergence to distinct modes, the algorithm needs to be restarted many times (S is large), which is computationally costly.

Given that setting ω can be interpreted as balancing exploration and exploitation behaviors of a mode-finding algorithm, then from this perspective, restarts are not necessary: there is no reason to wait for convergence to adjust the value of ω . Thus, it makes sense to consider any sequence $(\omega_t)_{t \geq 1}$ such that $\omega_t \rightarrow 0$ as $t \rightarrow \infty$ as a valid annealing schedule. The challenge lies in finding the optimal annealing schedule. To prevent component means from converging to the same modes, the annealing schedule should begin with large values of ω and not decrease too quickly to 0. In practice, we aim to use the smallest starting value possible and decrease it as quickly as possible, all that while ensuring convergence to distinct modes, therefore saving computation time. An example of annealing schedule is given in Appendix O.4.

O Hyperparameter tuning for NVA-GM

The NVA-GM algorithm (Algorithm 3) is governed by several hyperparameters, including the maximum number of iterations T , the mini-batch size B , the number of components of the mixture K , the initial parameters of the mixture θ_0 , the annealing schedule $(\omega_t)_{t \geq 1}$ and the learning rate $(\rho_t)_{t \geq 1}$.

The hyperparameter T can be set similarly to other classic optimization problems: increasing T allows the algorithm to perform more iterations and move closer to the solution but at a computational cost $O(T)$. Additionally, one may incorporate a stopping criterion to reduce this cost, such as stopping the algorithm when $\sum_{k=1}^K \|\mu_{k,t} - \mu_{k,t-1}\|_2 \leq \epsilon$ for some value of ϵ , which introduces an additional hyperparameter. The focus of this section is to discuss the choice and the effect of the remaining hyperparameters, with relevant insights and examples provided.

O.1 Number of components

The number of components K represents the maximum number of solutions (modes) that can be identified by our algorithms. If K is less than the number of (global) modes ($K < I$), then a single run cannot capture all the global modes. Conversely, if $K > I$ and all the global modes are identified with K components, then the remaining components will either converge to local modes or to some of the global modes already found. Indeed, although the ideal solution g_0 is a Dirac mixture concentrated on the global modes of ℓ , the smoothed version g_ω retains both global and local modes of ℓ , even though the local modes are less pronounced and therefore harder to detect.

If all the global modes are already captured by a component, then the remaining components can potentially converge to local modes due to the entropy term, which promotes separation between components. However, if no local modes are available, these remaining components will converge to already occupied (global or local) modes. However, more importantly, the weights $(\tilde{c}_i)_{i \in [I]}$ assigned to the modes, specified in g_0 , remain unchanged: the combined weight of components converging to

a global mode ξ_i^* converges to \tilde{c}_i , while components converging to local modes will see their weights vanish.

In some applications, such as Bayesian inverse problems (Stuart, 2010) or optimal experimental design (Long, 2022), finding local modes in addition to the global modes may be of particular interest. Nevertheless, global and local modes can be immediately distinguished by examining the component weights or evaluating ℓ . Thus, even if local modes are not the primary focus, identifying them is not problematic since they can be easily recognized and disregarded. The main downside of choosing a larger K is the additional computational cost associated with processing the extra components, which are not useful when the goal is only to find the I global modes.

In general, without prior knowledge of the exact number of modes I , it is generally advisable to slightly overestimate K to ensure that $K \geq I$, provided that this remains computationally reasonable. Although the number of parameters in the mixture scales linearly with K , the complexity of the algorithm is not strictly linear in K . Actually, the computation cost would be exactly $O(K)$ if $\omega = 0$, as this case allows for independent parameter updates across components. However, when $\omega > 0$, the evaluation of the gradient depends on the entropy term which involves the parameters of all components, therefore increasing the cost. Thus, the complexity of the algorithm becomes $O(K\varphi(K))$, where $\varphi(K)$ reflects the gradient evaluation cost, depending on the estimators used.

Specifically, for the NVA-GM algorithm (Algorithm 1),

- in $\hat{\gamma}_{\omega, \Lambda, B}^{(\pi_k, 0)}$, $\hat{\gamma}_{\omega, \Lambda, B}^{(\mu_k, 0)}$ and $\hat{\gamma}_{\omega, \Lambda, B}^{(S_k^{-1}, 0)}$, $\varphi(K)$ is the cost of computing $\log q_{\Lambda}(\xi)$ in $f_{\omega}(\xi; \Lambda)$, which is $O(K)$,
- in $\hat{\gamma}_{\omega, \Lambda, B}^{(\mu_k, 1)}$ and $\hat{\gamma}_{\omega, \Lambda, B}^{(S_k^{-1}, 1)}$, $\varphi(K)$ is the cost of computing $\nabla_{\xi} \log q_{\Lambda}(\xi)$ in $\nabla_{\xi} f_{\omega}(\xi; \Lambda)$, which is also $O(K)$,
- in $\hat{\gamma}_{\omega, \Lambda, B}^{(S_k^{-1}, 2)}$, $\varphi(K)$ is the cost of computing $\nabla_{\xi}^2 \log q_{\Lambda}(\xi)$ in $\nabla_{\xi}^2 f_{\omega}(\xi; \Lambda)$, which is $O(K^2)$.

O.2 Mini-batch size

The mini-batch size B is the number of samples used to compute the gradients. Larger values of B lead to more accurate gradient estimates, though at the cost of increased computation time. Indeed, the variance of the gradient estimators decreases asymptotically with B^{-1} , while the computational complexity scales as $O(B)$. Alternatively, from an evolutionary perspective, B can be viewed as the size of the population sampled by each search distribution.

In Algorithm 3, the samples must also be sorted before computing the gradients with utility values, which results in a complexity of $O(B \log B)$ (assuming a merge-sort algorithm). However, in practice, the effective time required for sorting is generally negligible compared to gradient computation, unless B is very large.

It may be relevant to use a dynamic mini-batch size, varying over time according to a schedule similar to those used for ω and ρ . For instance, a smaller mini-batch at the beginning of the run sizes can improve exploration thanks to the inherent stochasticity in the Monte Carlo gradient approximations. Although our algorithms can be easily modified to accommodate a dynamic mini-batch size, for the sake of clarity, the rest of the paper will assume a constant mini-batch size.

O.3 Initial mixture parameters

The initial parameters θ_0 consist of the initial mixture weights $(\pi_{1,0}, \dots, \pi_{K,0})$, the initial means $(\mu_{1,0}, \dots, \mu_{K,0})$, and the initial covariance matrices $(S_{1,0}^{-1}, \dots, S_{K,0}^{-1})$.

Initial mixture weights. The weights of the search mixture do not influence the parameter updates. As shown in equations (9) and (24), the weight increments are independent of $\pi_{k,t}$. Similarly, although $\pi_{k,t}$ appears in the update equations for the mean locations (14) and the covariance matrices (13), it cancels out with the gradients as shown in equations (25) to (29). Therefore, the behavior of the algorithms is not fundamentally affected by the choice of the initial weights.

Any positive weight vector summing to 1 should suffice. It takes fewer iterations for the weights to converge if their initial value is close to their limit. However, in general, we cannot predict which component will converge to which mode and we have no prior knowledge about the limit of the weights. Thus, it is reasonable, although not critical, to initialize the weight uniformly, i.e. $\pi_{k,0} = 1/K$ for all $k \in [K]$.

Initial covariance matrices. In contrast, the covariance matrices play an important role in two ways. First, they determine the dispersion of the samples used to compute the gradients. Samples give information about the landscape of the objective function. If the eigenvalues of the covariance matrices are too small, then the algorithm will sample points only near the means, limiting exploration. Second, the update equation for component means in the natural gradient approach (14) implies that

$$\|\boldsymbol{\mu}_{k,t+1} - \boldsymbol{\mu}_{k,t}\|_2 = \frac{\rho_t}{\pi_{k,t}} \|\mathbf{S}_{k,t+1}^{-1} \nabla_{\boldsymbol{\mu}_k} \mathcal{L}_\omega(\boldsymbol{\Lambda})|_{\boldsymbol{\Lambda}_t}\|_2. \quad (37)$$

One can remark that the size of the jump between $\boldsymbol{\mu}_{k,t}$ and $\boldsymbol{\mu}_{k,t+1}$ is regulated by the eigenvalues of $\mathbf{S}_{k,t+1}^{-1}$. Since we have explained in Section 4.1 how they are expected to vanish, this means that the algorithm can converge prematurely not because the means have reached stationary points, but because the eigenvalues have become too small. Picking larger initial eigenvalues for the covariance matrix helps avoid this issue.

Without prior information on ℓ , there is no reason not to pick isotropic Gaussian components. Therefore, we can reasonably set $\mathbf{S}_{k,0}^{-1} = \sigma_0^2 \mathbf{I}$ for all $k \in [K]$, where σ_0 , which should be large enough, can be picked by trial and error.

Initial means. In classic gradient algorithms, such as gradient ascent or Newton–Raphson’s method, the initial position of the particle is crucial because it may determine the (global or local) mode to which it will converge. Our algorithms also rely on gradients, but the entropy penalty in the objective function mitigates poor initialization by forcing separation between the components. Therefore, as long as there are enough components and with a well-chosen annealing schedule (Appendix O.4) and learning rate sequence (Appendix O.5), the choice of the initial means does not carry as much importance.

O.4 Annealing schedule

The annealing schedule $(\omega_t)_{1 \leq t \leq T}$ should be designed such that:

- ω_1 is large enough,
- ω_T is small enough,
- $(\omega_t)_{1 \leq t \leq T}$ is decreasing slowly enough.

Indeed, as discussed in Appendix N.3, the temperature ω governs the overall behavior of the interacting particle system. The annealing schedule must establish a smooth transition between

two regimes, each suitable to different phases of the algorithm. At the beginning of the run, a large ω is needed to encourage exploration of the landscape of ℓ , while towards the end, ω should be close to 0 to ensure convergence to the correct solutions. The transition process should be gradual and continuous.

Determining what constitutes sufficiently “large” or “small” values for ω , or when a sequence is decreasing “slowly enough”, is challenging because these factors depend heavily on ℓ and other hyperparameters such as the learning rate $(\rho_t)_{1 \leq t \leq T}$. Our recommendation is to set ω_t as a smooth monotonically decreasing function of t , parameterized by ω_1 and a parameter α controlling the rate of decrease. While ω_T could also be treated as a separate parameter, here it will be determined by ω_1 , α , and T . A typical example of annealing schedule is $\omega_t = \omega_1 t^{-\alpha}$, where $\alpha > 0$.

By trial-and-error, suitable values for ω_1 and α can be picked to find the desired solutions. For instance, if multiple components converge to the same mode, it suggests that ω_1 is not large enough or α is too large. Inversely, if all the components converge to different potential solutions, it is important to check if those are the correct solutions by assessing their convergence. This can be done by resuming the algorithm (effectively increasing T) or by restarting with a smaller ω_1 or a larger α to verify if the identified solutions remain stable. If they do, this indicates that ω_T is sufficiently small and that the identified solutions are indeed the correct ones.

O.5 Learning rate

The learning rate plays a special role in our algorithm because it is narrowly linked with ω through the precision matrices of the mixture components. Indeed, this can be seen in equation (37). The dependence of the distance between $\mu_{k,t}$ and $\mu_{k,t+1}$ on ρ_t and $\mathcal{L}_\omega(\Lambda)|_{\Lambda_t}$ is classic for gradient algorithms. The learning rate sets the proportionality constant ρ_t , while the gradient is necessary for convergence to the correct stationary point. However, here, we notice that the distance also depends on $S_{k,t+1}$.

In Appendix M, we have conjectured that $S_k^{*,\omega_t} \underset{t \rightarrow \infty}{\sim} \omega_t^{-1} (-\nabla_\xi^2 \ell(\xi_k^*))$ where ξ_k^* is the mode of ℓ such that $\mu_k^{*,\omega_t} \xrightarrow{t \rightarrow \infty} \xi_k^*$. But, near convergence of Algorithm 2, we have $S_{k,t} \approx S_k^{*,\omega_t}$, which means that

$$\|\mu_{k,t+1} - \mu_{k,t}\|_2 \approx \frac{\rho_t \omega_t}{\pi_{k,t}} \|(-\nabla_\xi^2 \ell(\xi_k^*))^{-1} \nabla_{\mu_k} \mathcal{L}_\omega(\Lambda)|_{\Lambda_t}\|_2.$$

Since $\omega_t \xrightarrow{t \rightarrow \infty} 0$, it slows the trajectory of $\mu_{k,t}$ down. As a result, our experiments have shown that with a constant learning rate, the decay in ω_t can occasionally lead to premature convergence, therefore preventing $\mu_{k,t}$ from reaching the correct value.

One potential solution is to allow ρ_t to increase to offset the decrease in ω_t . To select ρ_t adaptively, we propose defining $\rho_t = \omega_t^{-\beta}$, where $0 < \beta < 1$. In classical gradient ascent (GA), it is typically recommended to use a decreasing learning rate to facilitate convergence, as a constant learning rate does not guarantee convergence. In our algorithm, setting $\beta = 1$ effectively corresponds to a constant learning rate in GA, since ρ_t fully compensates for the effect of ω_t in the updates of μ_k . Therefore, we suggest selecting a lower value for β , such as $\beta = 0.8$ so that $\rho_t = \omega_t^{-0.8}$, to only partially offset the impact of ω_t .

P Gradient computation and estimation in FS-NVA-GM

In this section, we derive the natural gradient estimators needed in FS-NVA-GM, the fitness-shaped version of NVA-GM.

Following the derivation approach in Appendix J.1, we obtain the following update rules for μ_k and S_k

$$\begin{aligned} S_{k,t+1} &= S_{k,t} - \frac{2\rho_t}{\pi_{k,t}} \nabla_{S^{-1}} \mathcal{L}_{W_{\lambda_{k,t}}^{(\omega, \Lambda_t)}}(\lambda)|_{\lambda_{k,t}}, \\ \mu_{k,t+1} &= \mu_{k,t} + \frac{\rho_t}{\pi_{k,t}} S_{k,t+1}^{-1} \nabla_{\mu} \mathcal{L}_{W_{\lambda_{k,t}}^{(\omega, \Lambda_t)}}(\lambda)|_{\lambda_{k,t}}. \end{aligned}$$

For these updates, like in Appendix J.3, Gaussian identities yield the following gradient expressions

$$\begin{aligned} \nabla_{S^{-1}} \mathcal{L}_{W_{\lambda_{k,t}}^{(\omega, \Lambda_t)}}(\lambda) &= \frac{\pi_{k,t}}{2} \mathbb{E}_{\mathcal{N}(\mu, S^{-1})} \left[(S(\xi - \mu)(\xi - \mu)^T S - S) W_{\lambda_{k,t}}^{(\omega, \Lambda_t)}(f_{\omega_t}(\xi; \Lambda_t)) \right], \\ \nabla_{\mu} \mathcal{L}_{W_{\lambda_{k,t}}^{(\omega, \Lambda_t)}}(\lambda) &= \pi_{k,t} \mathbb{E}_{\mathcal{N}(\mu, S^{-1})} \left[S(\xi - \mu) W_{\lambda_{k,t}}^{(\omega, \Lambda_t)}(f_{\omega_t}(\xi; \Lambda_t)) \right]. \end{aligned}$$

Finally, for an arbitrary mini-batch size B , let $\mathbf{u} = (u_b)_{1 \leq b \leq B}$ be the utility values associated to w of (15). The gradients can be estimated by sampling B locations $\xi_1, \dots, \xi_B \stackrel{\text{i.i.d.}}{\sim} q_{\lambda_{k,t}} = \mathcal{N}(\mu_{k,t}, S_{k,t}^{-1})$ and sorting them as $\xi_{(1)}, \dots, \xi_{(B)}$ such that $f_{\omega}(\xi_{(1)}, \Lambda_t) \geq \dots \geq f_{\omega}(\xi_{(B)}, \Lambda_t)$. According to Theorem 6 of Ollivier et al. (2017), consistent estimators for the gradients are $\widehat{\nabla_{\mu} \mathcal{L}_{W_{\lambda_k}^{(\omega, \Lambda_t)}}(\lambda)}|_{\lambda_{k,t}} = \pi_{k,t} \widehat{\mathbf{v}}_{\omega, \Lambda_t, \mathbf{u}}^{(\mu_k)}$ and $\widehat{\nabla_{S^{-1}} \mathcal{L}_{W_{\lambda_k}^{(\omega, \Lambda_t)}}(\lambda)}|_{\lambda_{k,t}} = \pi_{k,t} \widehat{\mathbf{v}}_{\omega, \Lambda_t, \mathbf{u}}^{(S_k^{-1})}/2$, where:

$$\widehat{\mathbf{v}}_{\omega, \Lambda_t, \mathbf{u}}^{(\mu_k)} := \frac{1}{B} S_{k,t} \sum_{b=1}^B (\xi_{(b)}^{(k)} - \mu_{k,t}) u_b, \quad (38)$$

$$\widehat{\mathbf{v}}_{\omega, \Lambda_t, \mathbf{u}}^{(S_k^{-1})} := \frac{1}{B} S_{k,t} \sum_{b=1}^B \left((\xi_{(b)}^{(k)} - \mu_{k,t})(\xi_{(b)}^{(k)} - \mu_{k,t})^T S_{k,t} - \mathbf{I} \right) u_b. \quad (39)$$

Q Choice of utility values

In this section, we discuss the choice of utility values \mathbf{u} , or alternatively, of a weighting scheme w , in the FS-NVA-M (or FS-NVA-GM) algorithm, in relation with the literature in evolutionary algorithms.

In most implementations of the NES or the IGO framework, the choice of the utility values is typically arbitrary. While Wierstra et al. (2014) empirically observed that this choice does not significantly impact algorithm performance, Beyer (2014) demonstrated that it could strongly influence the behavior of the optimization process by analyzing the ordinary differential equation governing the dynamics of these algorithms. However, this impact is dependent on the specific fitness function, making it impossible to generalize. In the absence of prior information, defining an optimal \mathbf{u} or w is not feasible. Below, we present several common choices for w , or alternatively \mathbf{u} .

A natural selection rule in evolutionary algorithms is truncation selection, which involves selecting a fixed fraction of the current population, representing the samples with the highest fitness, to compute the next generation. Typically, this corresponds to the weighting scheme $w(x) = \mathbf{1}_{\{x \leq \eta\}}$, where $\eta \in (0, 1)$ is the fraction of the population kept. If the population size B is fixed, then the number of samples selected $B_0 = \lfloor B\eta + 1/2 \rfloor$ is constant and the equivalent utility values are

$$u_b = \begin{cases} B/B_0 & \text{for } 1 \leq b \leq B_0, \\ 0 & \text{for } B_0 + 1 \leq b \leq B. \end{cases}$$

Consequently, the choice of η determines the number of samples effectively used in the gradient computations. If η is too large, the algorithms may lead to premature convergence. [Ollivier et al. \(2017\)](#) suggest that η should generally be less than $1/2$, confirming the theoretical findings of [Beyer \(2001\)](#), [Jebalia and Auger \(2010\)](#) and [Beyer \(2014\)](#), who identified optimal values for specific examples of fitness functions. From a computational perspective, a lower η implies that fewer samples are used (in fact, η must be at least $0.5/N$ to select to ensure that at least one sample is selected), even though these samples have higher fitness. Thus, selecting a low η can reduce the computational cost but at the expense of increased variance in the gradient computations.

While truncation selection assigns equal weight to each retained sample, a variation of this selection scheme assigns rank-based weights, placing greater importance on the samples with the highest fitness. Notably, the weights used in the widely used CMA-ES algorithm ([Hansen and Ostermeier, 2001](#)) can be employed as utility values ([Wierstra et al., 2014](#)). They are defined by

$$u_b = \begin{cases} B \frac{\log(B_0+1) - \log(b)}{\sum_{c=1}^{B_0} (\log(B_0+1) - \log(c))} & \text{for } 1 \leq b \leq B_0, \\ 0 & \text{for } B_0 + 1 \leq b \leq B. \end{cases} \quad (40)$$

Departing from truncation selection, it is important to note that utility values can also be negative. For instance, [Arnold \(2006\)](#) proved that the optimal utility values for the spherical fitness function $f(\boldsymbol{\xi}) = -\|\boldsymbol{\xi}^* - \boldsymbol{\xi}\|_2^2$ are given by

$$u_b = \mathbb{E}[Z_{(b)}],$$

where $Z_{(1)} \geq \dots \geq Z_{(B)}$ are the order statistics of the vector (Z_1, \dots, Z_B) with $Z_b \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$, for all $1 \leq b \leq B$.

Finally, [Ollivier et al. \(2017\)](#) discussed the fact that any weighting scheme w , or vector of utility values \mathbf{u} , can be shifted by an additive constant c . While this does not change the expression of the gradient, it affects its estimation by introducing an additional term with a null expectation. Therefore, although c does not influence the expectation of the gradient estimator, it can be optimized, for example, to minimize variance.

R Simulations with degenerate modes

In this section, we present simulation results showcasing the behavior of the mixture weights in NVA-GM when ℓ has a degenerate mode and a non-degenerate mode. We set $d = 2$ and $\ell(\boldsymbol{\xi}) = \psi(\xi_1)(\xi_2^2 + 1)$ where

$$\psi(\xi) = \begin{cases} -(\xi + 3)^4 - 1 & \text{for } \xi < -2, \\ -\frac{1}{8}\xi^3 + \frac{3}{4}\xi^2 + \frac{1}{2}\xi - 5 & \text{for } -2 \leq \xi \leq 2, \\ -(\xi - 3)^2 - 1 & \text{for } 2 < \xi. \end{cases}$$

Here, ℓ has two modes, which are both global, at $\boldsymbol{\xi}_1^* = (-3, 0)$ and $\boldsymbol{\xi}_2^* = (3, 0)$. However, we have $\det(-\nabla^2 \ell(\boldsymbol{\xi}_1^*)) = 0$ and $\det(-\nabla^2 \ell(\boldsymbol{\xi}_2^*)) > 0$. Therefore, the mode at $\boldsymbol{\xi}_1^*$ is degenerate whereas the one at $\boldsymbol{\xi}_2^*$ is not.

We apply NVA-GM with $K = 2$, using $T = 50$, $\pi_{1,0} = \pi_{2,0} = 1/2$, $\mathbf{S}_{1,0}^{-1} = \mathbf{S}_{2,0}^{-1} = \mathbf{I}$, $\omega_t = \omega_1/t^2$ with $\omega_1 = 0.1$, and $\rho_t = 10^{-1}(\omega_1/\omega_t)^{0.8}$. For the gradients, we use the estimators $\hat{\gamma}_{\omega, \boldsymbol{\Lambda}, B}^{(\mu_k, 1)}$ and $\hat{\gamma}_{\omega, \boldsymbol{\Lambda}, B}^{(\mathbf{S}_k^{-1}, 2)}$, where $B = 4$. We recover the positions of the component means and the value of the weights at each iteration.

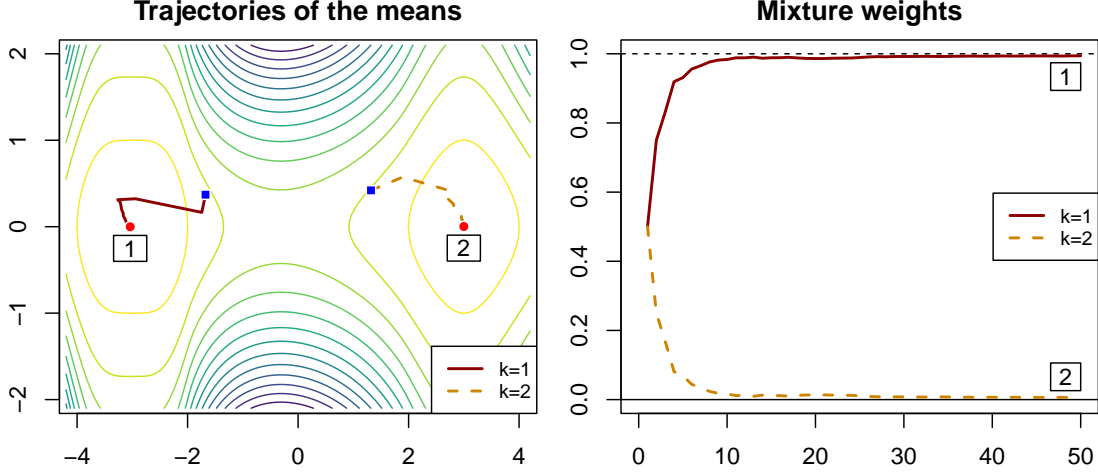


Figure 6: Left: The trajectories of the means in a run of NVA-GM on the function with a degenerate mode and a non-degenerate mode with $K = 2$ show that the two components track different modes. The contour plot represents non-equally spaced levels of ℓ . Right: As expected, the weights of the component tracking the degenerate modes (1) converge to 1, whereas the weight of the component tracking the non-degenerate mode (2) vanishes.

Figure 6 illustrates the trajectories of the component means and the evolution of the mixture weights during a run where each mode is successfully identified by a distinct mean. Although Theorem 2.2 is not directly applicable to functions with degenerate modes, in the subsequent discussion in Appendix E, we have provided an interpretation that degenerate modes are “infinitely” flatter than non-degenerate ones. Consequently, in this run of NVA-GM, it is unsurprising that all the weight is captured by component 1, which is the one converging to the degenerate mode.

S Black-box optimization benchmark

To our knowledge, the only established benchmarks for multimodal optimization are designed for black-box optimization methods, using only the objective function evaluation. For completeness, we thus illustrate the performance of FS-NVA-GM, the black-box setting of which does not resort to gradients or Hessians. The FS-NVA-GM algorithm is applied to several CEC2013 benchmark functions for multimodal optimization (Li et al., 2013), which have been used in both CEC and GECCO niching competitions. Although our framework is designed to locate all modes, the benchmark suite targets black-box multimodal global optimization. The benchmark assesses the ability to consistently find all the global modes of a set of test functions, using only function evaluations.

We use the first six functions of the benchmark suite. Many of these functions exhibit local optima. The function domains of the benchmark suite are constrained to rectangles of the form $\mathcal{D}_{a,b} = \prod_{i=1}^d [a_i, b_i]$, where d is the input variable dimension. As this is not our current illustrative purpose, to bypass the need for constrained optimization, we extend all functions to the entire space \mathbb{R}^d as follows:

$$\tilde{f}(\xi) = f(a + r(\xi)) - |q(\xi)|_1 A(f),$$

where

- $r(\xi) = (r_i(\xi_i))_{1 \leq i \leq d}$ and $q(\xi) = (q_i(\xi_i))_{1 \leq i \leq d}$ and $(q_i(\xi_i), r_i(\xi_i))$ are the quotient and the remainder of the (generalized) Euclidean division of ξ_i by $(b_i - a_i)$, i.e. $\xi_i = q_i(\xi_i)(b_i - a_i) + r_i(\xi_i)$

with $0 \leq r_i(\xi_i) < (b_i - a_i)$ and $q_i(\xi_i) \in \mathbb{Z}$, for all $1 \leq i \leq d$,

- $|\mathbf{q}(\boldsymbol{\xi})|_1 = \sum_{i=1}^d |q_i(\boldsymbol{\xi})|$,
- $A(f) = \max_{\boldsymbol{\xi} \in \mathcal{D}} f(\boldsymbol{\xi}) - \min_{\boldsymbol{\xi} \in \mathcal{D}} f(\boldsymbol{\xi})$.

The modified function \tilde{f} is a “pyramidal” extension of f on the whole space \mathbb{R}^d . This construction partitions \mathbb{R}^d into a grid of rectangular regions, each with the same dimensions as the original domain $\mathcal{D}_{\mathbf{a},\mathbf{b}}$. In each of these regions, the function f is copied but shifted down by a constant, which increases with the distance of that region from the original domain $\mathcal{D}_{\mathbf{a},\mathbf{b}}$. This construction preserves the global modes as f in \mathcal{D} , but introduces an infinite number of new local modes due to the repeated, offset copies of f across the extended domain.

We fix an accuracy level $\epsilon = 0.1$, a threshold of the target function below which a mode is considered found. To evaluate our algorithms, we use the global peak ratio (GPR) and global success rate (GSR) as advised in Li et al. (2013). The GPR has been defined earlier by (17). The GSR is given by

$$\text{GSR} = \frac{1}{H} \sum_{h=1}^H \text{AGF}_h,$$

where H is the number of runs, and for each run h , $\text{AGF}_h = 1$ if all target global modes are found, and 0 otherwise.

We use FS-NVA-GM with CMA-ES utility values, as given by (40). We made slight modifications to FS-NVA-GM. First, to prevent numerical instability, we enforce a lower bound τ on the eigenvalues of the covariance matrices in the search mixture. This is done by adding the following step after each precision matrix update in FS-NVA-GM:

$$\mathbf{S}_{k,t+1} \leftarrow (\mathbf{S}_{k,t+1}^{-1} + \tau \mathbf{I})^{-1}.$$

The hyperparameter τ can be seen as a damping factor and can be fixed to a small value, $\tau = 10^{-10}$ in our experiments. Second, we observed that skipping updates of the precision matrices for a few initial iterations can improve performance. We introduce an additional hyperparameter κ , which defines a burn-in period, i.e. the number of initial iterations before precision matrix updates begin. Mathematically, these first κ iterations correspond to solving the variational problem in the family of Gaussian mixtures with fixed covariance matrices. After κ iterations, covariances are allowed to be updated. For each run, the initial means are initialized uniformly at random in a rectangle for each function, containing all global modes, specified in Li et al. (2013), of the form $\mathcal{D}_{\mathbf{a},\mathbf{b}} = \prod_{i=1}^d [a_i, b_i]$. The initial covariance matrices are set to $\sigma_0^2 \mathbf{I}$, where $\sigma_0^2 = \|\mathcal{D}_{\mathbf{a},\mathbf{b}}/2\|_\infty^2 = \max_{i \in \{1, \dots, d\}} (b_i - a_i)^2/4$ and the initial mixture weights are set to $1/K$. The annealing schedules are defined as $\omega_t = \omega_1 t^{-\alpha}$, and the learning rates are given by $\rho_t = \rho_1 (\omega_1/\omega_t)^\beta$, where $\omega_1, \rho_1, \alpha, \beta$ are hyperparameters specified in Table 3.

In the CEC2013 competition guidelines, a maximum budget in the number of function evaluations is allocated for each target function, which is 50 000 for functions F_1 to F_5 , and 200 000 for F_6 . For FS-NVA-GM, the number of evaluations performed during a run is $f\text{-eval} = BKT$. K is set to the number of global modes to find. Increasing K improves the performance metrics but raises $f\text{-eval}$. We use B and T to control performance within a reasonable value for $f\text{-eval}$, slightly exceeding the budget specified in the CEC2013 competition. Under this setting, the benchmark results (Table 4) show that FS-NVA-GM performs well for simple functions (F_1 to F_5), locating all global modes in almost every run. For F_6 (Shubert 2D function), which is highly multimodal with 18 global and many local modes, FS-NVA-GM finds in average $0.749 \times 18 \approx 13.5$ modes per run with our chosen hyperparameters.

Function	I	T	K	B	ω_1	α	ρ_1	β	κ
F_1	2	500	2	16	100000	2	10^{-3}	0.8	0
F_2	5	2000	5	32	20	1	10^{-3}	0.9	0
F_3	1	2000	1	32	20	1	10^{-3}	0.9	0
F_4	4	2000	4	16	2000000	1.8	10^{-4}	0.7	50
F_5	2	2000	2	16	10000	2	10^{-5}	0.8	0
F_6	18	2000	18	16	1000000	1.8	10^{-5}	0.8	50

Table 3: Number of global modes I and hyperparameters used for the first six functions of the CEC2013 benchmark (F_1 to F_6).

Function	f -eval	GPR	GSR
F_1	16000	1.000	1.000
F_2	320000	0.996	0.980
F_3	64000	1.000	1.000
F_4	130000	0.990	0.960
F_5	64000	1.000	1.000
F_6	580000	0.749	0.000

Table 4: Number of evaluations of the target function and evaluation metrics GPR and GSR (the closer to 1 the better) for $H = 50$ runs of FS-NVA-GM with the hyperparameters specified in Table 3.

The results are competitive with some black-box methods. However, we expect FS-NVA-GM not to be at the same level for more complex benchmark functions, as the NVA-M framework has been primarily designed to exploit first and second order information. The black-box algorithm FS-NVA-GM may then be too simple in its current tested implementation, and in particular limited by the allowed number of function calls.

References

- Agnoletto, D., Rigon, T., and Dunson, D. B. (2023). Bayesian inference for generalized linear models via quasi-posteriors. *arXiv preprint arXiv:2311.00820*.
- Ahrari, A., Deb, K., and Preuss, M. (2017). Multimodal optimization by covariance matrix self-adaptation evolution strategy with repelling subpopulations. *Evolutionary Computation*, 25(3):439–471.
- Alquier, P. (2024). User-friendly introduction to PAC-Bayes bounds. *Foundations and Trends® in Machine Learning*, 17(2):174–303.
- Amari, S.-I. (1998). Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276.
- Amari, S.-I. and Nagaoka, H. (2000). *Methods of information geometry*, volume 191. American Mathematical Soc.
- Améndola, C., Engström, A., and Haase, C. (2020). Maximum number of modes of Gaussian mixtures. *Information and Inference: A Journal of the IMA*, 9(3):587–600.
- Andersen, H.-E., Reutebuch, S. E., and Schreuder, G. F. (2002). Bayesian object recognition for the analysis of complex forest scenes in airborne laser scanner data. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/A):35–41.
- Aprausheva, N. N., Mollaverdi, N., and Sorokin, S. V. (2006). Bounds for the number of modes of the simplest Gaussian mixture. *Pattern Recognition and Image Analysis*, 16:677–681.
- Arnold, D. V. (2006). Weighted multirecombination evolution strategies. *Theoretical Computer Science*, 361(1):18–37.
- Bäck, T., Fogel, D. B., and Michalewicz, Z. (1997). *Handbook of Evolutionary Computation*. IOP Publishing Ltd., 1st edition.
- Bender, C. M. and Orszag, S. A. (1999). *Advanced Mathematical Methods for Scientists and Engineers I: Asymptotic Methods and Perturbation Theory*, volume 1. Springer.
- Beyer, H.-G. (2001). *Theory of Evolution Strategies*. Springer.
- Beyer, H.-G. (2014). Convergence analysis of evolutionary algorithms that are based on the paradigm of information geometry. *Evolutionary Computation*, 22(4):679–709.
- Bissiri, P. G., Holmes, C. C., and Walker, S. G. (2016). A general framework for updating belief distributions. *Journal of the Royal Statistical Society: Series B*, 78(5):1103–1130.
- Bonnabel, S. (2013). Stochastic gradient descent on Riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229.
- Bonnet, G. (1964). Transformations des signaux aléatoires à travers les systèmes non linéaires sans mémoire. *Annales des Télécommunications*, 19:203–220.
- Brits, R., Engelbrecht, A. P., and van den Bergh, F. (2007). Locating multiple optima using particle swarm optimization. *Applied Mathematics and Computation*, 189(2):1859–1883.

- Carreira-Perpiñán, M. Á. (2000). Mode-finding for mixtures of Gaussian distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1318–1323.
- Carreira-Perpiñán, M. Á. and Williams, C. K. I. (2003a). An isotropic Gaussian mixture can have more modes than components. Technical Report EDI-INF-RR-0185, School of Informatics, University of Edinburgh, UK.
- Carreira-Perpiñán, M. Á. and Williams, C. K. I. (2003b). On the number of modes of a Gaussian mixture. In *International Conference on Scale-Space Theories in Computer Vision*, pages 625–640. Springer.
- Custódio, A. L. and Madeira, J. A. (2015). GLODS: global and local optimization using direct search. *Journal of Global Optimization*, 62:1–28.
- D’Angelo, F. and Fortuin, V. (2021a). Annealed Stein variational gradient descent. *AABI*.
- D’Angelo, F. and Fortuin, V. (2021b). Annealed Stein variational gradient descent. In *Advances in Approximate Bayesian Inference*.
- de Nobel, J., Vermetten, D., Kononova, A. V., Shir, O. M., and Bäck, T. (2024). Avoiding redundant restarts in multimodal global optimization. In *Parallel Problem Solving from Nature*.
- Deleforge, A., Forbes, F., and Horaud, R. (2015). High-dimensional regression with Gaussian mixtures and partially-latent response variables. *Statistics and Computing*, 25(5):893–911.
- Donsker, M. D. and Varadhan, S. R. S. (1976). Asymptotic evaluation of certain Markov process expectations for large time—III. *Communications on Pure and Applied Mathematics*, 29(4):389–461.
- Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7).
- Fernando, J., Schmidt, F., and Douté, S. (2016). Martian surface microtexture from orbital CRISM multi-angular observations: A new perspective for the characterization of the geological processes. *Planetary and Space Science*, 128:30–51.
- Fong, E. and Holmes, C. C. (2020). On the marginal likelihood and cross-validation. *Biometrika*, 107(2):489–496.
- Forbes, F., Nguyen, H. D., Nguyen, T., and Arbel, J. (2022). Summary statistics and discrepancy measures for approximate Bayesian computation via surrogate posteriors. *Statistics and Computing*, 32(5):85.
- Friel, N. and Pettitt, A. N. (2008). Marginal likelihood estimation via power posteriors. *Journal of the Royal Statistical Society: Series B*, 70(3):589–607.
- Furuya, T., Kusumoto, H., Taniguchi, K., Kanno, N., and Suetake, K. (2024). Theoretical error analysis of entropy approximation for Gaussian mixture. *arXiv preprint arXiv:2202.13059*.
- Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741.

- Gilks, W. R. (1995). Fractional Bayes factors for model comparison: Discussion. *Journal of the Royal Statistical Society: Series B*, 57(1):118–120.
- Girolami, M. (2008). Bayesian inference for differential equations. *Theoretical Computer Science*, 408(1):4–16.
- Hansen, N. (2006). The CMA evolution strategy: A comparing review. In Lozano, J. A., Larrañaga, P., Inza, I., and Bengoetxea, E., editors, *Towards a New Evolutionary Computation: Advances in the Estimation of Distribution Algorithms*, volume 192, pages 75–102. Springer.
- Hansen, N. and Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195.
- Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347.
- Honkela, A., Tornio, M., Raiko, T., and Karhunen, J. (2008). Natural conjugate gradient in variational inference. In *Neural Information Processing*.
- Hsu, L. C. (1948). A theorem on the asymptotic behavior of a multiple integral. *Duke Mathematical Journal*, 15(3):623.
- Huang, C.-W., Tan, S., Lacoste, A., and Courville, A. C. (2018). Improving explorability in variational inference with annealed variational objectives. *Advances in Neural Information Processing Systems*.
- Hwang, C.-R. (1980). Laplace’s method revisited: weak convergence of probability measures. *The Annals of Probability*, 8(6):1177–1182.
- Ibrahim, J. G. and Chen, M.-H. (2000). Power prior distributions for regression models. *Statistical Science*, 15(1):46–60.
- Israelachvili, J. N. (1992). *Intermolecular and surface forces*. Academic Press.
- Jebalia, M. and Auger, A. (2010). Log-linear convergence of the scale-invariant $(\mu/\mu_w, \lambda)$ -ES and optimal μ for intermediate recombination for large population sizes. In *Parallel Problem Solving from Nature*. Springer.
- Jiang, W. and Tanner, M. A. (2008). Gibbs posterior for variable selection in high-dimensional classification and data mining. *Annals of Statistics*, 36(5):2207–2231.
- Katahira, K., Watanabe, K., and Okada, M. (2008). Deterministic annealing variant of variational Bayes method. *Journal of Physics: Conference Series*, 95(1):012015.
- Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *International Conference on Neural Networks*.
- Khan, M. E. and Rue, H. (2023). The Bayesian learning rule. *Journal of Machine Learning Research*, 24(281):1–46.
- Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

- Kirkpatrick, S., Gelatt Jr, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.
- Knoblauch, J., Jewson, J., and Damoulas, T. (2022). An optimization-centric view on Bayes’ rule: Reviewing and generalizing variational inference. *Journal of Machine Learning Research*, 23(1).
- Kugler, B., Forbes, F., and Douté, S. (2022). Fast Bayesian inversion for high dimensional inverse problems. *Statistics and Computing*, 32(2):31.
- Kullback, S. (1959). *Information Theory and Statistics*. John Wiley & Sons.
- Labarre, S. (2017). *Caractérisation et modélisation de la rugosité multi-échelle des surfaces naturelles par télédétection dans le domaine solaire*. PhD thesis, Université Sorbonne Paris Cité.
- Larson, J. and Wild, S. M. (2018). Asynchronously parallel optimization solver for finding multiple minima. *Mathematical Programming Computation*, 10:303–332.
- Lee, D., Kim, J.-W., Lee, C.-G., and Jung, S.-Y. (2011). Variable mesh adaptive direct search algorithm applied for optimal design of electric machines based on FEA. *IEEE Transactions on Magnetics*, 47(10):3232–3235.
- Li, X. (2007). A multimodal particle swarm optimizer based on fitness Euclidean-distance ratio. In *Conference on Genetic and Evolutionary Computation*.
- Li, X., Engelbrecht, A., and Epitropakis, M. G. (2013). Benchmark functions for cec’2013 special session and competition on niching methods for multimodal function optimization. *RMIT University, Evolutionary Computation and Machine Learning Group, Australia, Tech. Rep.*
- Li, X., Epitropakis, M. G., Deb, K., and Engelbrecht, A. (2016). Seeking multiple solutions: An updated survey on niching methods and their applications. *IEEE Transactions on Evolutionary Computation*, 21(4):518–538.
- Liang, J. J., Qin, A. K., Suganthan, P. N., and Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, 10(3):281–295.
- Lin, W., Khan, M. E., and Schmidt, M. (2019a). Fast and simple natural-gradient variational inference with mixture of exponential-family approximations. In *International Conference on Machine Learning*.
- Lin, W., Khan, M. E., and Schmidt, M. (2019b). Stein’s lemma for the reparameterization trick with exponential family mixtures. *arXiv preprint arXiv:1910.13398*.
- Lin, W., Schmidt, M., and Khan, M. E. (2020). Handling the positive-definite constraint in the Bayesian learning rule. In *International Conference on Machine Learning*.
- Long, Q. (2022). Multimodal information gain in Bayesian design of experiments. *Computational Statistics*, 37(2):865–885.
- Luo, X. (2018). Minima distribution for global optimization. *arXiv preprint arXiv:1812.03457*.
- Mandt, S., McInerney, J., Abrol, F., Ranganath, R., and Blei, D. (2016). Variational tempering. In *Artificial Intelligence and Statistics*.

- Maree, S. C., Alderliesten, T., Thierens, D., and Bosman, P. A. N. (2018). Real-valued evolutionary multi-modal optimization driven by hill-valley clustering. In *Conference on Genetic and Evolutionary Computation*.
- Murchie, S. L., Seelos, F. P., Hash, C. D., Humm, D. C., Malaret, E., McGovern, J. A., Choo, T. H., Seelos, K. D., Buczkowski, D. L., Morgan, M. F., Barnouin-Jha, O. S., Nair, H., Taylor, H. W., Patterson, G. W., Harvel, C. A., Mustard, J. F., Arvidson, R. E., McGuire, P., Smith, M. D., Wolff, M. J., Titus, T. N., Bibring, J., and Poulet, F. (2009). Compact Reconnaissance Imaging Spectrometer for Mars investigation and data set from the Mars Reconnaissance Orbiter’s primary science phase. *Journal of Geophysical Research: Planets*, 114(E2):E00D07.
- Nielsen, F. (2020). An elementary introduction to information geometry. *Entropy*, 22(10):1100.
- O’Hagan, A. (1995). Fractional Bayes factors for model comparison. *Journal of the Royal Statistical Society: Series B*, 57(1):99–118.
- Ollivier, Y., Arnold, L., Auger, A., and Hansen, N. (2017). Information-geometric optimization algorithms: A unifying picture via invariance principles. *Journal of Machine Learning Research*, 18(18):1–65.
- Oppel, M. and Archambeau, C. (2009). The variational Gaussian approximation revisited. *Neural Computation*, 21(3):786–792.
- Osawa, K., Swaroop, S., Khan, M. E., Jain, A., Eschenhagen, R., Turner, R. E., and Yokota, R. (2019). Practical deep learning with Bayesian principles. *Advances in Neural Information Processing Systems*.
- Price, R. (1958). A useful theorem for nonlinear devices having Gaussian inputs. *IRE Transactions on Information Theory*, 4(2):69–72.
- Qu, B.-Y., Suganthan, P. N., and Das, S. (2012). A distance-based locally informed particle swarm model for multimodal optimization. *IEEE Transactions on Evolutionary Computation*, 17(3):387–402.
- Ray, S. and Lindsay, B. G. (2005). The topography of multivariate normal mixtures. *The Annals of Statistics*, 33(5):2042–2065.
- Regis, R. G. and Shoemaker, C. A. (2013). A quasi-multistart framework for global optimization of expensive functions using response surface models. *Journal of Global Optimization*, 56:1719–1753.
- Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*.
- Rigon, T., Herring, A. H., and Dunson, D. B. (2023). A generalized Bayes framework for probabilistic clustering. *Biometrika*, 110(3):559–578.
- Rinnooy Kan, A. H. G. and Timmer, G. T. (1987). Stochastic global optimization methods, part II: Multi level methods. *Mathematical Programming*, 39:57–78.
- Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407.
- Sanokowski, S., Berghammer, W., Hochreiter, S., and Lehner, S. (2023). Variational annealing on graphs for combinatorial optimization. *Advances in Neural Information Processing Systems*.

- Sato, M.-A. (2001). Online model selection based on the variational Bayes. *Neural Computation*, 13(7):1649–1681.
- Schmidt, F. and Fernando, J. (2015). Realistic uncertainties on Hapke model parameters from photometric measurements. *Icarus*, 260:73–93.
- Serré, G., Kalogeratos, A., and Vayatis, N. (2024). Stein Boltzmann sampling: A variational approach for global optimization. *arXiv preprint arXiv:2402.04689*.
- Shen, Y., Daheim, N., Cong, B., Nickl, P., Marconi, G. M., Bazan, C., Yokota, R., Gurevych, I., Cremers, D., Khan, M. E., and Möllenhoff, T. (2024). Variational learning is effective for large deep networks. In *International Conference on Machine Learning*.
- Staines, J. and Barber, D. (2012). Variational optimization. *arXiv preprint arXiv:1212.4507*.
- Stein, C. M. (1981). Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics*, 9(6):1135–1151.
- Stuart, A. M. (2010). Inverse problems: a Bayesian perspective. *Acta Numerica*, 19:451–559.
- Sun, Y., Wierstra, D., Schaul, T., and Schmidhuber, J. (2009). Efficient natural evolution strategies. In *Conference on Genetic and Evolutionary Computation*.
- Tao, C., Dai, S., Chen, L., Bai, K., Chen, J., Liu, C., Zhang, R., Bobashev, G., and Duke, L. C. (2019). Variational annealing of GANs: A Langevin perspective. In *International Conference on Machine Learning*.
- Tieleman, T. and Hinton, G. (2012). Lecture 6.5 - RMSprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning 4*.
- Tierney, L., Kass, R. E., and Kadane, J. B. (1989). Fully exponential Laplace approximations to expectations and variances of nonpositive functions. *Journal of the American Statistical Association*, 84(407):710–716.
- Tsybakov, A. (2008). *Introduction to Nonparametric Estimation*. Springer Series in Statistics. Springer.
- Wierstra, D., Schaul, T., Glasmachers, T., Sun, Y., Peters, J., and Schmidhuber, J. (2014). Natural evolution strategies. *Journal of Machine Learning Research*, 15:949–980.
- Wong, R. (2001). *Asymptotic approximations of integrals*. SIAM.
- Wu, K. and Gardner, J. R. (2024). Understanding stochastic natural gradient variational inference. *arXiv preprint arXiv:2406.01870*.
- Yamanaka, Y. and Yoshida, K. (2021). Simple gravitational particle swarm algorithm for multimodal optimization problems. *PLOS ONE*, 16(3):1–23.
- Yoshida, R. and West, M. (2010). Bayesian learning in sparse graphical factor models via variational mean-field annealing. *The Journal of Machine Learning Research*, 11:1771–1798.
- Zellner, A. (1988). Optimal information processing and Bayes’s theorem. *The American Statistician*, 42(4):278–280.
- Zhang, X. and Ghosh, S. (2023). ProGO: Probabilistic global optimizer. *arXiv preprint arXiv:2310.04457*.