Technical Abstract

This project looks at methods for solving populational inverse problems. The problem involves estimating population-level parameters from indirect, noisy observations of multiple systems. Think of measuring properties across wind turbines in a wind farm where each turbine is slightly different. Some windmills are older and less efficient, others are newer with better bearings, but they all share common characteristics like experiencing the same wind patterns. We observe N systems and want to learn the mean and variance of their parameters from noisy measurements. This problem is quite common in engineering, as it is frequent to be dealing with batches of components or repeated experiments. For example, this work will focus on parameter estimation for a simple but ubiquitous model, the damped harmonic oscillator.

The two methods discussed in this project are Hierarchical Bayesian Models (HBMs) and a distribution-matching (DM) approach using gradient descent. While HBMs provide robust parameter estimation even with small populations, the DM method's performance improves with larger population sizes. Since HBMs become computationally more expensive as population size increases, this project aims to determine when each method is most appropriate, i.e. when the computational cost of HBMs outweighs their improved inference accuracy.

Hierarchical Bayesian Models tackle this by building a full probabilistic model. Individual system parameters come from population distributions, and we use Markov Chain Monte Carlo to explore all possible values. Specifically, we implement Hamiltonian Monte Carlo with the No-U-Turn Sampler, using non-centered parametrization to improve efficiency. The big advantage is getting complete uncertainty information along with population parameter estimates. When you only have data from a few systems, these confidence intervals are wide, which corresponds to high uncertainty. The downside is that the computation time of the MCMC grows with the number of systems.

On the other hand, the distribution-matching approach turns everything into an optimization problem. We search for parameters that minimize the Sliced-Wasserstein distance between what we observe and what the model predicts. We add regularization to keep parameters reasonably close to initial values and use gradient descent to minimise the objective. The main advantage of this method is that computation time is constant with respect to the number of systems. However, the distribution-matching approach only

achieves point estimates, which don't account for uncertainty like HBMs do.

For the damped harmonic oscillator problem, synthetic, noisy data were generated for different population sizes (between N = 1 and N = 1000 systems) for 100 independent experiments, ensuring our conclusions were robust. For small populations (i.e. under 10 systems), HBMs are more suitable. With just one system, the Bayesian approach correctly produces huge confidence intervals, indicating we can't reliably estimate population parameters from one example. The MCMC chains explore a wide range of parameter values, expected behaviour, given the lack of data. As we add systems, confidence intervals shrink appropriately while still containing the true values. Meanwhile, the distribution-matching method gives you a number but no way to know if it's trustworthy. Furthermore, the runtime of both methods is comparable for this number of systems. For larger populations (approximately N = 10 systems), both methods now find accurate parameter values. However, we find HBM's runtime scales linearly with respect to population size, whereas the runtime for the distribution-matching method stays constant.

This project provides concrete guidance for practitioners faced with hierarchically-structured problems. Although there is no specific rule that determines when each methodology is more suitable, a N = 10 threshold gives a simple initial decision boundary as found in our experiments. HBMs should be reserved for smaller numbers of systems, when uncertainty matters, whereas the use of the distribution-matching methodology is more suitable for larger populations, where speed matters more than uncertainty quantification.

The framework naturally extends beyond linear systems. Initial tests with nonlinear predator-prey dynamics (Lotka-Volterra equations) show both methods can handle more complex systems, though convergence becomes more challenging.

Contents

1 Introduction

1.1 Aims and Motivation

Machine Learning (ML) methods have demonstrated remarkable success in computational physics, particularly in solving inverse problems, i.e., inferring system parameters from observational data. Many statistical applications to physics involve multiple groups of parameters that can be regarded as related by the structure of the problem. For instance, consider the problem of inferring the parameters of several windmills on a wind farm. Although each windmill is different and possesses unique features (e.g., some windmills are older, hence less efficient than others), all windmills share inherent characteristics that are important to consider during inference (e.g., same wind speed).

Models should appropriately reflect these dependencies and require substantial prior information. While incorporating prior distributions is necessary, it alone isn't sufficient for accurate inference: these systems also need to encode this prior information within the model architecture itself. Hierarchical Bayesian Models (HBMs) offer a solution to this challenge. However, this report aims to challenge this belief that HBMs are the only way to solve such problems, notably through a distribution-matching method.

This report is concerned with learning a generative model for unobserved parameters $\{z^{(n)}\}_{n=1}^{N}$ for N systems from indirect and noisy data $\{y^{(n)}\}_{n=1}^{N}$, given by:

$$y^{(n)} = \mathcal{G}(z^{(n)}) + \epsilon^{(n)} \tag{1}$$

where the i.i.d. noise $\epsilon \sim \eta := \mathcal{N}(0, \sigma^2 I)$ represents measurement errors with variance σ^2 and $\mathcal{G}(\cdot)$ is the forward operator mapping parameters to observations. Further details on the problem will be developed in Apparatus and Experimental Techniques chapter.

1.2 Literature Review

This chapter aims to provide an overview of the current state of the literature for our aims, particularly focusing on progress from Hierarchical Bayesian Models, which itself relies very heavily on recent advances in Markov Chain Monte-Carlo (MCMC) algorithms, as

well as tools from optimal transport.

1.2.1 HBMs

Bayesian statistics is a framework for updating our beliefs about parameters by combining prior knowledge (expressed as prior distributions) with observed data (through the likelihood function) to produce posterior distributions that quantify uncertainty in our beliefs about these parameters of interest. The uncertainty quantification in Bayes' theorem is a particularly powerful tool when there is little data, which can lead to imprecise inferences. However, implementing Bayes' theorem involves difficult analytical analysis. The development of computational methods in the early 1990s called Markov Chain Monte Carlo (MCMC) methods, which simulate draws from a distribution instead of requiring analytical solutions, helps bypass this problem. Hierarchical Bayesian Models apply these Bayesian methods to a model written in hierarchical form. A hierarchical model can be defined as one that is written modularly and is made up of a sequence of linked submodels. By linking sub-models together and correctly propagating uncertainties in each sub-model from one level to the next, we can build a complete model of the data and provide a principled framework to include systematic errors and selection effects[14].

The power of Hierarchical Bayesian Models can best be understood through an example. Imagine we want to infer the parameters of several turbines on a wind farm, such as the damping or stiffness properties. At first, an intuitive approach could be to ignore the specificity of each turbine and our data's grouped structure, lumping all turbines into one sample. This model, called a completely-pooled model, is an oversimplification of our problem, as we consider that all our data comes from the same source. We completely remove the individuality of each different system. A solution to this problem involves considering each turbine separately. This approach, called the no-pooled model, ignores data from a turbine when learning about another turbine. This method is particularly problematic when looking at turbine with a lack of data points. Also, the no-pooled model cannot be generalised to turbines outside of our sample, i.e. if we have no data on a particular turbine, we cannot make any inference on the parameters of that turbine. This failure to generalise highlights the overcomplexity of this model. Both observed models possess interesting advantages but problematic and opposite inconveniences. This challenge may be resolved by the use of partial pooling models, a mix of the completely and no-pooled models. Although each turbine is different and possesses unique features,

all turbines share inherent characteristics that are important to consider during inference. HBMs also share many advantages with classical Bayesian statistics, most notably the ability to handle inferences with only small amounts of data. The hierarchical component establishes an architecture where data observations depend on population parameters, which in turn depend on hyperparameters. This structure enables population inference by treating group-specific parameters as samples from a common distribution. While this framework offers powerful modeling capabilities, its probability density structure demonstrates sensitivity to hyperparameter specifications[6].

However, HBMs also have challenges that need to be addressed. Firstly, HBMs have underlying assumptions. Most notably, Bayesian methods are based on the assumption that probability is operationalized as a degree of belief, and not a frequency as is done in classical, or frequentist, statistics. Finally, the complexity of hierarchical models makes the analytical computation of posterior distributions intractable. Hence, HBMs need to use MCMC sampling methods, which we discuss in the following section.[2]

1.2.2 MCMC algorithms

As previously mentioned, in Bayesian analysis, the posterior probability distribution characterizes the uncertainty in the model parameters estimated from a given set of measurements. Markov Chain Monte Carlo (MCMC) techniques help us explore this posterior and, consequently, characterise parameter uncertainty.[4;16;10;28] This is done by effectively generating a sequence of model realisations randomly drawn from the posterior distribution.

Most Bayesian analyses use one of two standard MCMC algorithms: Gibbs sampling or the Metropolis-Hastings algorithm, the latter being usually employed due to its simplicity. For the Metropolis-Hastings algorithm, all the parameters are varied at once. The parameter vector is perturbed from the current sequence point by adding a trial step drawn randomly from a symmetric pdf. This proposed trial position is either accepted or rejected based on the probability of the trial position relative to the current one.[18] However, the Metropolis-Hastings algorithm's optimal efficiency for Gaussian distributions is proportional to $1/n$,[15;19] where n is the number of variables, which poses considerable problems at higher dimensions.

More sophisticated techniques have been developed to effectively explore the posterior

space at reasonable computational expenses. In particular, Hamiltonian Monte Carlo utilizes techniques from Hamiltonian dynamics to generate transitions spanning the full marginal variance. Gradients from the Hamiltonian equations guide the transitions through regions of high probability and admit the efficient exploration of the entire target distribution.[5] The Hamiltonian Monte Carlo algorithm can yield much higher performance for general hierarchical models than other common MCMC implementations.[5] While HMC's performance is sensitive to both step size and the number of steps, this limitation has been largely addressed by the No-U-Turn Sampler (NUTS) algorithm,[20] which helps preserve the detailed balance of the transitions.[5] Although NUTS requires computing the derivatives of the target distribution,[5] state-of-the-art Python libraries such as NumPyro handle this issue through automatic differentiation.[27]. As such, the usage of the NUTS algorithm for our problem is most appropriate. Further details on its implementation to our problem are described in Section 3.2.

1.2.3 Optimal transport based metrics

Literature shows that simple, non-hierarchical models are usually inappropriate for hierarchical data. Using too little amount of parameters leads to underfitting and using too many parameters leads to overfitting difficulties.[14] However, more recent literature also provides examples of successful applications of non-hierarchical methods to populational problems.

The concept of minimisation of regularised loss functions over the space of probability measures is central to modern computational statistics and machine learning.[1] Given the problem we are faced with, we are interested in computationally tractable optimal transport-based metrics on the space of probability distributions. The use of optimal transport-based metrics has already been explored in ML inference tasks. Candidates include the Wasserstein and Sliced-Wasserstein distances.

The Wasserstein distance, originating from optimal transport theory, provides a geometrically meaningful way to compare probability distributions. Unlike divergences such as Kullback-Leibler that require absolute continuity between measures, the Wasserstein distance is well-defined between distributions with non-overlapping supports and preserves the underlying geometric structure of the sample space.[32]

Intuitively, optimal transport can be understood as the problem of moving piles of sand

from one configuration to another with minimal effort, where effort is measured as the amount of sand moved multiplied by the distance it travels. The Wasserstein distance quantifies this minimal transportation cost, earning it the alternative name of "Earth Mover's Distance" [23].

The Wasserstein distance is usually computationally expensive to evaluate. On the other hand, the sliced-Wasserstein is an efficient alternative regarded as a proxy of Wasserstein distances. It works by projecting high-dimensional distributions to one-dimensional spaces and leveraging the fact that optimal transport in 1D has a simple closed-form solution. Wasserstein and sliced-Wasserstein metrics have both found use in approximate Bayesian computation, where the recovered measures have been shown to converge to the Bayesian posterior.[1] However, the easier tractability of the Sliced-Wasserstein makes it a great candidate for our problem (detailed implementation presented in Section 2.2).

2 Distribution-matching (DM) methodology

2.1 Regularised divergence minimisation

Our DM methodology aims to find the optimal hyperparameter values $\boldsymbol{\alpha}^* = [\boldsymbol{\mu}, \boldsymbol{\tau}]$ using a gradient descent algorithm, in our case Adaptive Moment Estimation (ADAM), such that:

$$\boldsymbol{\alpha}^* = \arg\min_{\alpha} J(\boldsymbol{\alpha}) \tag{2}$$

where the objective function to minimise, $J(\cdot)$, is the sum of a data fit term d_1 and a regularization term d_2:

$$J(\alpha) = d_1(\nu, \eta * \mathcal{G}_{\#} P^{\alpha}(z)) + h(\alpha) \tag{3}$$

where η & $\mathcal{G}$ are as previous defined. d_1 represents a divergence between two probability distributions: the empirical measure ν of the observed data and the pushforward (#) measure obtained by transforming a parameterized probability measure P^{α} that defines the hierarchical structure of our model through the forward operator G and convolving (*)

it with the noise distribution η. To effectively and efficiently compare empirical measures, the divergence d_1 is chosen to be the sliced-Wasserstein distance[1]. h corresponds to a suitably chosen regulariser to quantify the discrepancy between the parameterized probability measure and an initial reference measure. It is worth adding that Equation 3 is a distributional representation of Equation 1.

2.2 Sliced-Wasserstein distance for data fitting

As mentioned in the literature review, despite its theoretical appeal, computing the Wasserstein distance in higher dimensions presents significant computational challenges. For discrete distributions with n points, the complexity scales as $O(n^3 \log(n))$, rendering it impractical for many large-scale machine learning applications. This computational bottleneck has motivated the development of more efficient alternatives.

The Sliced-Wasserstein (SW) distance elegantly addresses this challenge by leveraging a key insight: the one-dimensional Wasserstein distance can be computed efficiently in $O(n \log n)$ time through simple sorting operations. The SW distance works by projecting high-dimensional distributions onto multiple one-dimensional subspaces and averaging the resulting Wasserstein distances.[1] Other versions of the Sliced-Wasserstein we studied prior to this report, most notably Energy-Based Sliced Wasserstein Distances[25] and Augmented Sliced Wasserstein Distances[11].

We begin by defining the weighted Wasserstein distance:

$$W_{2,\mathrm{B}}^2(\nu, \mu) = \inf_{\gamma \in \Pi(\nu,\mu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|_{\mathrm{B}}^2 \mathrm{d}\gamma(x, y), \tag{4}$$

where $\Pi(\nu, \mu)$ represents the set of all joint probability measures on $\mathbb{R}^d \times \mathbb{R}^d$ with marginals ν and μ, and B is a positive, self-adjoint operator. For the standard case, we use $W_2 := W_{2,\mathrm{I}}$ where I is the identity matrix.

The following lemma establishes an important relationship between the weighted Wasserstein distance and pushforward measures[1]:

For $P_{\mathrm{B}}(\cdot) = \mathrm{B}^{-1/2}\cdot$, it follows that:

$$W_{2,\mathrm{B}}^2(\nu, \mu) = W_2^2(P_{\mathrm{B}}\#\nu, P_{\mathrm{B}}\#\mu), \tag{5}$$

where $P_{\mathrm{B}}\#\nu$ denotes the pushforward of measure ν under the mapping P_{B}.

Building on this result, we define the weighted Sliced-Wasserstein distance:

$$SW_{2,\mathrm{B}}^2(\nu,\mu) = \int_{\mathbb{S}^{d-1}} W_2^2(P_{\mathrm{B}}^{\theta}\#\nu, P_{\mathrm{B}}^{\theta}\#\mu)\mathrm{d}\theta, \tag{6}$$

where $P_{\mathrm{B}}^{\theta}(\cdot) = \langle \mathrm{B}^{-\frac{1}{2}}\cdot, \theta\rangle$ and $\mathbb{S}^{d-1}$ represents the unit sphere in $\mathbb{R}^d$. The integration over all possible directions θ effectively captures the geometric structure of the distributions while maintaining computational efficiency.[1]

In practice, we approximate this integral using Monte Carlo sampling:

$$SW_{2,\mathrm{B}}^2(\nu,\mu) \approx \frac{1}{N}\sum_{i=1}^{N} W_2^2(P_{\mathrm{B}}^{\theta_i}\#\nu, P_{\mathrm{B}}^{\theta_i}\#\mu), \tag{7}$$

where $\{\theta_i\}_{i=1}^N$ are random samples drawn uniformly from the unit sphere $\mathbb{S}^{d-1}$.[1]

The Sliced-Wasserstein distance's effectiveness in Bayesian inference problems, particularly for prior calibration from indirect data, has already been demonstrated.[1] Their approach is especially valuable for working with empirical measures, as the SW distance remains well-defined under empirical approximation and avoids the support issues that plague other divergences. The Sliced-Wasserstein can be calculated through the following 4-step algorithm*:[7]

1. Generate N random projections from the unit sphere
2. Project both input measures onto one-dimensional spaces using these random directions
3. Compute the 1D Wasserstein distance between each pair of projected distributions
4. Average these distances to obtain the final Sliced-Wasserstein estimate

Using JAX for efficient computation and automatic differentiation allows the SW distance to be used within gradient-based optimization frameworks. Their implementation provides a configurable Monte Carlo approximation with parameters for the number of projections and the order of the metric.

*Detailed implementation can be found on https://github.com/yanndivet/iib_project_ssh

2.3 Tikhonov regularisation

In many inverse-problem and distribution-matching settings, especially when optimising with stochastic gradient methods like ADAM[21], adding a quadratic regularization term around an initial guess is quite common. Perhaps the most widely referenced regularization method is the Tikhonov method.[8] This regulariser directly incorporates prior information through the addition of the additional term described in Equation 8 to the objective function. From the regularised objective function (Equation 3), we define the regulariser function h as:

$$h(\alpha) = \lambda||\alpha - \alpha_0||^2, \tag{8}$$

where $\lambda > 0$ is the regularisation parameter and α_0 represents our initial parameter estimate, typically derived from prior knowledge or preliminary analyses.

This quadratic penalty serves multiple purposes in our optimisation framework. First, it stabilises the gradient descent trajectory by providing a convex contribution to the loss landscape,[3] particularly important in regions where the Sliced-Wasserstein distance may exhibit local irregularities.[26] Second, it prevents the optimisation from diverging too far from physically plausible parameter values, effectively encoding soft constraints based on domain knowledge. Third, for ill-posed inverse problems where multiple parameter configurations may yield similar observations, Tikhonov regularisation helps select the solution closest to our prior belief α_0.[9]

The choice of regularisation parameter λ involves balancing data fidelity against prior confidence. In our implementation, we adopt an adaptive approach where λ is initially set to 10^{-2} and decreased logarithmically during training, allowing the optimisation to initially stay close to the prior before increasingly trusting the data. This annealing schedule proved more effective than fixed regularisation.

While Kullback-Leibler (KL) divergence was experimented with as an alternative regularisation term (as documented in the Technical Milestone Report), it provided inferior results. The KL divergence $d_2(P_\alpha(z), P_0(z))$ requires careful handling of support mismatch and can produce infinite values when distributions have non-overlapping supports[12]. In contrast, the quadratic Tikhonov term remains well-defined and differentiable throughout

the parameter space, making it more suitable for gradient-based optimisation. Furthermore, the computational overhead of evaluating KL divergence between hierarchical distributions significantly exceeded that of the simple quadratic penalty, without corresponding improvements in parameter recovery.

3 Hierarchical Bayes Models (HBMs) methodology

3.1 Constructing a Hierarchical Bayesian Model

The parameters $\{\boldsymbol{z}^{(n)}\}_{n=1}^{N}$ are sampled as illustrated in Equation 9 and Figure 1

$$z_i^{(n)} \sim \mathcal{N}(\mu_i, \tau_i^2) \tag{9}$$

$$\mu_i \sim \mathcal{N}(\mu_i^{\phi}, \sigma_i^{\phi 2}) \tag{10}$$

$$\tau_i \sim \text{Inv-Gamma}(a_i^{\phi}, b_i^{\phi}) \tag{11}$$

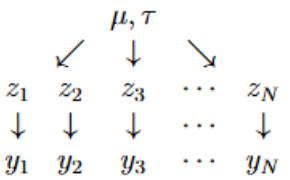

Figure 1: Structure of a one-layer hierarchical model

We use a one-way normal hierarchical model to fit the model described previously and recover the distributions of hyperparameters $\boldsymbol{\mu}$, $\boldsymbol{\tau}$, and population parameters $\boldsymbol{z}^{(1:N)}$.

Using Bayes' theorem we are able to get the following expression for the Bayesian posterior distribution:

$$P(\boldsymbol{z}^{(1:N)}, \boldsymbol{\mu}, \boldsymbol{\tau}|\boldsymbol{y}^{(1:N)}) = \frac{P(\boldsymbol{\mu}, \boldsymbol{\tau}) \prod_{n=1}^{N} P(\boldsymbol{z}^{(n)}|\boldsymbol{\mu}, \boldsymbol{\tau}) P(\boldsymbol{y}^{(n)}|\boldsymbol{z}^{(n)})}{P(\boldsymbol{y}^{(1:N)})} \tag{12}$$

Using Equation 12, the joint log-posterior density is found to be proportional to:

$$\log p(\boldsymbol{z}^{(1:N)}, \boldsymbol{\mu}, \boldsymbol{\tau}|\boldsymbol{y}^{(1:N)}) \propto \log p(\boldsymbol{\mu}, \boldsymbol{\tau}) + \sum_{n=1}^{N} \left(\log p(\boldsymbol{z}^{(n)}|\boldsymbol{\mu}, \boldsymbol{\tau}) + \log p(\boldsymbol{y}^{(n)}|\boldsymbol{z}^{(n)}) \right) \tag{13}$$

where using the log scale avoids numerical overflow.[30] Using Equations 1 & 9 and $\log p(\boldsymbol{x}) \propto \sum_i \left(\frac{x-\mu_i}{\sigma_i} \right)^2$ for $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma^2} I)$, we can reduce Equation 13 into Equation 14:

$$\begin{aligned} \therefore \log p(\boldsymbol{z}^{(1:N)}, \boldsymbol{\mu}, \boldsymbol{\tau}|\boldsymbol{y}^{(1:N)}) \propto & \sum_i \left(\frac{\mu_i - \mu_i^\phi}{\sigma_i^\phi} \right)^2 + \sum_i \log \text{Inv-Gamma}(\tau_i; a_i^\phi, b_i^\phi) \\ & + \sum_{n=1}^{N} \left(\sum_i \left(\frac{z_i^{(n)} - \mu_i}{\tau_i} \right)^2 + \left(\frac{\boldsymbol{y}^{(n)} - \mathcal{G}(\boldsymbol{z}^{(n)})}{\sigma} \right)^2 \right) \end{aligned} \tag{14}$$

3.2 Hamiltonian Monte Carlo and the No-U-Turn Sampler

Hamiltonian Monte Carlo (HMC) fundamentally transforms MCMC sampling by exploiting the geometric structure of probability distributions through Hamiltonian dynamics[24]. The method introduces auxiliary momentum variables $\boldsymbol{r}$ for each parameter $\boldsymbol{\theta}$, constructing an extended phase space with joint distribution:

$$p(\boldsymbol{\theta}, \boldsymbol{r}) \propto \exp(-H(\boldsymbol{\theta}, \boldsymbol{r})) = \exp(-U(\boldsymbol{\theta}) - K(\boldsymbol{r})) \tag{15}$$

where $U(\boldsymbol{\theta}) = -\log p(\boldsymbol{\theta})$ represents potential energy and $K(\boldsymbol{r}) = \frac{1}{2}\boldsymbol{r}^T \boldsymbol{M}^{-1} \boldsymbol{r}$ represents kinetic energy[24].

HMC relies on simulating Hamiltonian dynamics to generate proposals. These dynamics possess three crucial properties: reversibility, volume preservation (Liouville's theorem),

and conservation of the Hamiltonian.[24] These properties ensure that proposals can be distant from the current state while maintaining high acceptance probabilities, effectively suppressing the random-walk behavior found in traditional Metropolis methods.

Hamilton's equations govern the system evolution:

$$\frac{d\boldsymbol{\theta}}{dt} = \frac{\partial H}{\partial \boldsymbol{r}} = \boldsymbol{M}^{-1}\boldsymbol{r} \tag{16}$$

$$\frac{d\boldsymbol{r}}{dt} = -\frac{\partial H}{\partial \boldsymbol{\theta}} = \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) \tag{17}$$

Practical implementation requires discretization via symplectic integrators, with the leapfrog method being most common due to its simplicity and preservation of phase space volume[24]. The algorithm alternates between momentum resampling from the marginal distribution $p(\boldsymbol{r}) \propto \exp(-K(\boldsymbol{r}))$ and deterministic trajectory simulation followed by a Metropolis accept/reject step.

HMC also possesses superior scaling properties.[18] While random-walk Metropolis efficiency decreases inversely with dimension, HMC maintains nearly constant efficiency even for high-dimensional problems. Crucially, HMC handles correlated distributions effectively without requiring careful adaptation of proposal distributions[18].

However, HMCs need to manually specify the trajectory length L. This limitation proves catastrophic for hierarchical models.[5] Hierarchical structures induce "funnel" geometries in the posterior, where probability mass concentrates in regions with vastly different scales. The pathology arises because changes in hyperparameters induce correlated changes across all lower-level parameters, creating extreme variations in posterior density. No single trajectory length can efficiently explore both the narrow neck and wide mouth of such funnels[5].

Furthermore, even with position-dependent metrics such as the Riemannian HMC, the extreme density variations in hierarchical models require trajectory lengths that vary by orders of magnitude across parameter space.[5] Instead, we must look into algorithms that can "adapt the simulation length to the local structure of the target distribution."

The No-U-Turn Sampler (NUTS)[20] provides precisely such adaptation. NUTS constructs trajectories through recursive doubling, building balanced binary trees where each doubling randomly extends the trajectory forward or backward in time. The fundamental innovation

is the no-U-turn criterion:

$$(\boldsymbol{\theta}^+ - \boldsymbol{\theta}^-) \cdot \boldsymbol{r}^- < 0 \quad \text{or} \quad (\boldsymbol{\theta}^+ - \boldsymbol{\theta}^-) \cdot \boldsymbol{r}^+ < 0 \tag{18}$$

This criterion, derived from the time derivative of squared distance in phase space, detects when continued simulation would bring the trajectory closer to its starting point[20].

To maintain detailed balance with adaptive trajectories, NUTS employs a sophisticated slice sampling scheme. The algorithm introduces an auxiliary variable $u \sim \text{Uniform}(0, \exp\{L(\boldsymbol{\theta}) - K(\boldsymbol{r})\})$ and builds the set of candidate states satisfying the slice constraint. As Hoffman and Gelman[20] prove, this construction ensures the correct stationary distribution despite variable trajectory lengths.

Beyond trajectory adaptation, NUTS incorporates dual averaging for step size tuning:

$$\log \epsilon_{t+1} = \mu - \frac{\sqrt{t}}{\gamma} \frac{1}{t + t_0} \sum_{i=1}^{t} (\delta - \alpha_i) \tag{19}$$

This scheme, adapted from Nesterov's optimization methods, automatically finds step sizes achieving target acceptance rates during warmup[20].

Hoffman and Gelman[20] also address practical implementation concerns. They present memory-efficient algorithms using $O(\log n)$ rather than $O(n)$ storage by exploiting the binary tree structure. Their "efficient NUTS" variant samples incrementally from candidate sets without storing complete trajectories.

The empirical advantages prove substantial. On multivariate normal distributions, NUTS achieves effective sample sizes comparable to optimally-tuned HMC without manual intervention[20]. For hierarchical models—including logistic regression and stochastic volatility models—NUTS demonstrates robust performance where fixed-length HMC would require problem-specific tuning across orders of magnitude[20].

Betancourt and Girolami's[5] analysis explains why NUTS succeeds where standard HMC fails for hierarchical models. In funnel geometries, NUTS automatically takes short trajectories in high-curvature regions (preventing divergence) while extending trajectories in low-curvature regions (avoiding random walks). This geometric adaptivity proves

essential for the complex posterior geometries arising in modern Bayesian applications.

The theoretical and practical advantages of NUTS have established it as the default sampling algorithm in probabilistic programming systems. As Hoffman and Gelman[20] conclude, NUTS "allows researchers and data analysts to spend more time developing and testing models and less time worrying about how to fit those models to data." For our implementation, we employ NumPyro's NUTS implementation, leveraging its automatic differentiation and just-in-time compilation capabilities for efficient gradient computation—addressing another practical concern raised by Hanson[18] regarding the computational cost of gradient evaluations in complex models.

3.3 Non-centered parametrisation

The convergence properties of MCMC algorithms for hierarchical models are greatly influenced by the choice of parametrisation. As discussed in Section 3.2, achieving computationally efficient posterior evaluation requires careful consideration of the parameter space structure. Hierarchical models are particularly sensitive to parametrisation choices due to their multi-level structure and the potentially strong dependencies between parameters at different levels of the hierarchy.

The model previously defined in Equation 14 in represents what is commonly known as centered parametrisation. In this formulation, we directly model the parameters $z^{(n)}$ as being drawn from a distribution with mean μ and standard deviation τ:

$$z_i^{(n)} \sim \mathcal{N}(\mu_i, \tau_i^2) \tag{20}$$

While conceptually straightforward, this parametrisation often creates problematic posterior dependencies between $z^{(n)}$ and the hyperparameters μ and τ. Betancourt and Girolami (2013) demonstrate that these dependencies can create challenging geometries for MCMC algorithms in the form of "funnel" structures in the posterior density. These funnels occur because as τ approaches zero, the conditional variance of $z^{(n)}$ given μ and τ also approaches zero, creating regions of high curvature that standard samplers struggle to navigate efficiently.

To address this challenge, we implement a non-centered parametrisation, which reformulates

the model through a change of variables that reduces posterior correlations. Rather than directly sampling $z^{(n)}$, we introduce standardised auxiliary parameters $z_{\text{raw}}^{(n)}$ that are independent of the hyperparameters:

$$z_{\text{raw}}^{(n)} \sim \mathcal{N}(0, 1) \tag{21}$$

$$z^{(n)} = \mu + \tau \cdot z_{\text{raw}}^{(n)} \tag{22}$$

This transformation alters the dependency structure in our model. The parameters $z_{\text{raw}}^{(n)}$ are completely independent of μ and τ in the prior, which allows more efficient exploration of the posterior distribution. This parametrisation can dramatically accelerate convergence by reducing coupling in parameter updates.[29]

The efficacy of non-centered versus centered parametrisation depends on the relative magnitudes of the likelihood and prior variances. Neither approach is universally optimal.[34] When data are highly informative (small measurement noise relative to prior variance), the centered parametrisation often performs better; conversely, when data are weakly informative (large measurement noise), the non-centered parametrisation typically excels.

The mathematical foundation for this improvement can be understood through the lens of posterior correlation structures. In the centered parametrisation, strong correlations between $z^{(n)}$ and the hyperparameters create a challenging geometry for MCMC samplers. The non-centered parametrisation effectively eliminates these correlations in the prior, leaving only the likelihood-induced correlations, which are often less problematic for sampling.

This reparameterisation strategy complements the Hamiltonian Monte Carlo methodology that will be discussed in Section 3.2.2. The NUTS algorithm benefits substantially from the improved geometry offered by non-centered parametrisation, as the challenging funnel geometries that would otherwise require adaptive trajectory lengths are significantly ameliorated. The elimination of strong parameter correlations allows for more efficient momentum trajectories during the Hamiltonian simulation phase, leading to more effective exploration of the posterior distribution.

By implementing non-centered parametrisation, we significantly enhance the efficiency of

our posterior sampling approach without altering the underlying model specification or its statistical properties. This improvement is crucial for reliable inference in our hierarchical framework, particularly for problems with complex posterior geometries and multiple layers of parameters.

4 Apparatus and Experimental Techniques

In order to compare and contrast the distribution-matching and HBM techniques, we designed an experiment protocol for two different physical systems. The first one is a linear system, the damped harmonic oscillator, while our second experiment is a non-linear system called the Lotka-Volterra system of ODEs. This chapter is split into 3 sections, firstly talking about the general methodology applicable to both experiments, before then diving into the specificity of the damped harmonic oscillator then the Lotka-Volterra systems.

4.1 General Experiment Methodology

In this section, we outline the experimental framework used to evaluate both inference methods. The methodology is designed to ensure fair comparison between the distribution-matching (DM) and Hierarchical Bayesian Model (HBM) approaches while remaining applicable to different physical systems.

4.1.1 Hyperprior Specification and Initialization

The hierarchical structure requires careful specification of prior distributions, which serve different purposes for each method:

For HBM: The hyperpriors form an integral part of the model structure, encoding our prior beliefs about the population distribution:

For a system with d-dimensional parameter space, we specify:

- Location parameters: $\mu_i \sim \mathcal{N}(\mu_{\phi_i}, \sigma^2_{\phi_i})$ for $i = 1, \ldots, d$

- Scale parameters: $\tau_i \sim \text{Inv-Gamma}(a_{\phi_i}, b_{\phi_i})$ for $i = 1, \ldots, d$

where $\boldsymbol{\mu}_\phi, \boldsymbol{\sigma}_\phi, \boldsymbol{a}_\phi, \boldsymbol{b}_\phi \in \mathbb{R}^d$ correspond to appropriate prior values chosen based on domain knowledge to be weakly informative, i.e. providing reasonable bounds without overly constraining the inference.

For DM: These same hyperprior specifications provide the initialization point for the optimization algorithm. Specifically, we initialize:

- $\mu_i = \mu_{\phi_i}$ (the mean of the normal hyperprior) for $i = 1, \ldots, d$
- $\tau_i = b_{\phi_i}/(a_{\phi_i}+1)$ (the mode estimate of the inverse-gamma hyperprior) for $i = 1, \ldots, d$

This approach ensures fair comparison between methods—both have access to the same prior information, but utilize it differently. The HBM incorporates the full prior distribution into its inference, while the DM method uses only the mode as a starting point for optimization. A small amount of noise ($\epsilon = 10^{-2}$) is added to the DM initialization to avoid potential local minima at the exact prior mode.

Population Distribution: Given the hyperparameters (either sampled for HBM or optimized for DM), individual system parameters follow:

- $z_i^{(n)} \sim \text{LogNormal}(\mu_i, \tau_i^2)$ for systems $n = 1, \ldots, N$ and dimensions $i = 1, \ldots, d$

The log-normal distribution ensures parameter positivity, which is essential for the physical parameters we will be learning. This choice also captures the multiplicative nature of many physical processes.

4.1.2 Data Generation Process

To ensure reproducible and fair comparison between methods, we implement a systematic data generation pipeline:

Step 1: Initialize Random Seeds Each experiment is assigned a unique seed based on the experiment number and population size to ensure reproducibility while maintaining independence across experiments.

Step 2: Sample True Hyperparameters We draw the "ground truth" hyperparameters from the specified distributions:

- $\mu_{\text{true}} \sim \mathcal{N}(\mu_{\text{target}}, \tau_{\text{target}})$
- τ_{true} is set to predetermined values representing realistic population variability

Step 3: Generate Population Parameters For each of N systems, we sample parameters from the population distribution:

- $z^{(n)} \sim \text{LogNormal}(\mu_{\text{true}}, \tau^2_{\text{true}})$

where the log-normal distribution is chosen as it ensures the positivity of the parameters. These represent the true physical parameters for each system in our population.

Step 4: Simulate System Observations For each system, we:

1. Apply the forward operator $G(z^{(n)})$ to generate noise-free observations
2. Add Gaussian measurement noise: $y^{(n)} = G(z^{(n)}) + \varepsilon^{(n)}$, where $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$
3. Record observations at T equally-spaced time points

The forward operator G encapsulates the system dynamics and is the only component that changes between different physical systems.

Step 5: Data Persistence All generated observations are saved and organized by experiment number and population size. This ensures that both DM and HBM methods operate on identical data, eliminating any variation due to data generation from our comparative analysis.

The noise level $\sigma = 0.01$ is chosen to represent realistic measurement uncertainty while maintaining sufficient signal-to-noise ratio for meaningful inference. The number of observation time points $T = 50$ provides adequate temporal resolution without excessive computational burden.

This standardized approach allows us to systematically investigate how both methods perform across different population sizes $N \in \{1, 10, 50, 100, 500, 1000\}$ and multiple experimental realizations, providing robust statistical evidence for our comparative analysis.

4.2 Experiment 1: Damped Harmonic Oscillator

To validate our methodology, we implement and evaluate it on a linear physical system: the damped harmonic oscillator, although the methodology generalizes to other similar physical systems. We generate synthetic noisy observations using numerical integration techniques, specifically employing the leapfrog integrator to simulate system dynamics given system parameters. To ensure physical consistency, we employ log-transformation of the μ parameters, thereby enforcing positivity constraints on the physical parameters during the learning process.

4.2.1 Governing Equation

The damped harmonic oscillator problem is governed by a second-order differential equation:[13]

$$\frac{d^2x}{dt^2} + 2\zeta\omega_0\frac{dx}{dt} + \omega_0^2 x = 0 \tag{23}$$

where:

- x is the position of the mass;
- $\omega_0 = \sqrt{\frac{k}{m}}$ is the undamped angular frequency of the oscillator;
- $\zeta = \frac{c}{2\sqrt{mk}}$ is the damping ratio.

$$\frac{d^2x}{dt^2} + 2\beta\frac{dx}{dt} + \omega_0^2 x = 0 \tag{24}$$

$$\beta = \zeta\omega_0 \tag{25}$$

Equation 23 can be rewritten as Equation 24 and henceforth will be the form used in our study.[31]

The value of the damped ratio ζ (and by extension β) determines the behaviour of the system.[22] As such, the damped harmonic oscillator can be one of three states:

- Underdamped ($\zeta < 1$)
- Critically damped ($\zeta = 1$)
- Overdamped ($\zeta > 1$)

Equation 26 is the analytical solution of the underdamped state.[17]

$$x(t) = Ae^{-\beta t}\cos(\omega t - \delta). \tag{26}$$

Equation 27 is the analytical solution of the critically damped state.[17]

$$x(t) = C_1 e^{-\beta t} + C_2 t e^{-\beta t}. \tag{27}$$

Equation 28 is the analytical solution of the overdamped state.[17]

$$x(t) = C_1 e^{-(\beta - \sqrt{\beta^2 - \omega_0^2})t} + C_2 e^{-(\beta + \sqrt{\beta^2 - \omega_0^2})t}. \tag{28}$$

The analytical solution for all three of these states is plotted in Figure 2. Figure 2 illustrates the drastic differences in behaviour between those three states.

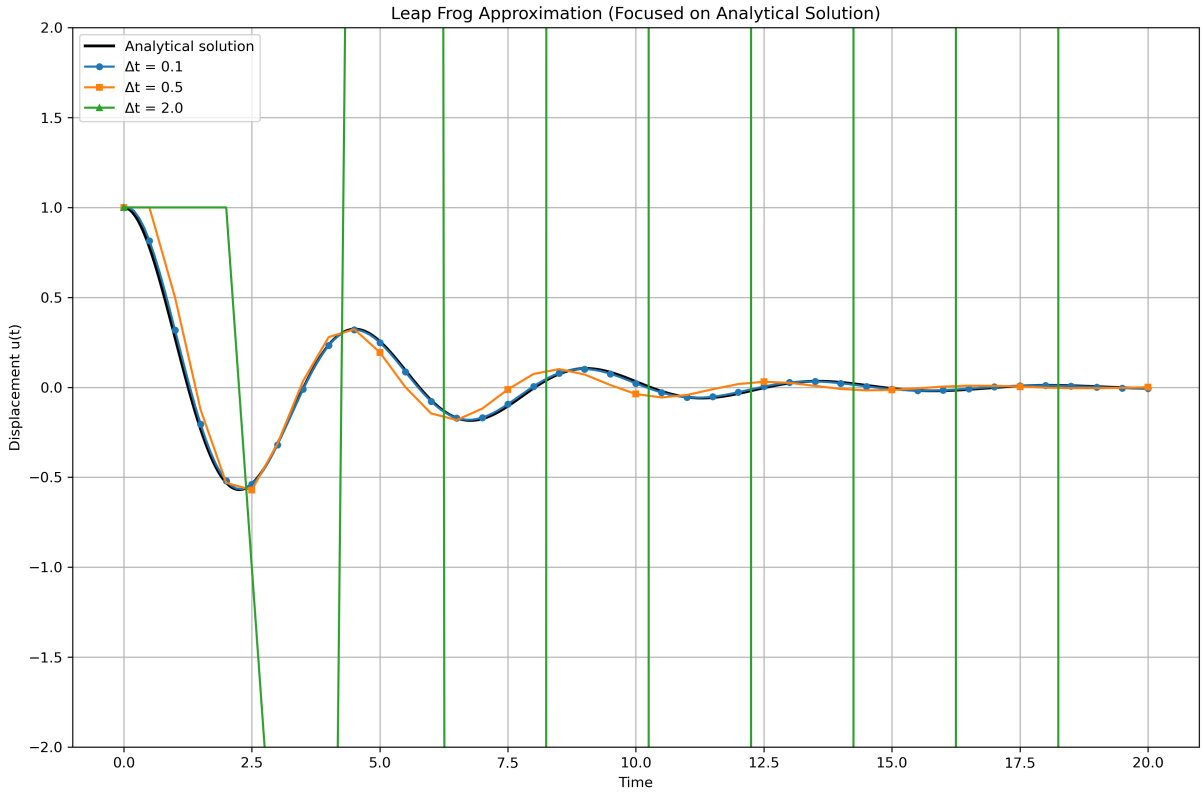

Figure 2: Leapfrog approximations and analytical solution against time for different Δt values. The timestop $\Delta t = 2$ demonstrates the importance of choosing suitably small time steps in numerical integration

However, to make this experiment as applicable to a wide array of problems, we will use a numerical integrator to make observations at equally spaced time steps.

4.2.2 Choice of numerical integrator

Using a numerical integrator when an analytical solution exists might sound counter-productive. However, the damped harmonic oscillator is an example used to show such methodology can be used for any type of linear problem. As such, we need to use a methodology that can generalise to problems that may not have analytical solutions.

Two different numerical integrators were studied for this problem: the Forward Euler and Leapfrog Integrator. The main conclusions are that the Leapfrog Integrator significantly outperforms the Forward Euler for the damped harmonic oscillator problem.

The Leap Frog method (sometimes called Stormer-Verlet method) is a symplectic integrator particularly well-suited for Hamiltonian systems. The method is similar to the Forward Euler method, but we replace u_n by u_{n+1} in the second equation, leading to a better preservation of the system's energy:

$$\begin{cases} u_{n+1} = u_n + \Delta t\, v_n, \\ v_{n+1} = v_n + \Delta t \left(\frac{f - c v_n - k u_{n+1}}{m} \right). \end{cases} \tag{29}$$

With this method, we will "leapfrog" between the x & v values, hence the name of the algorithm.

The Leapfrog method also has superior stability properties compared to Forward Euler. Indeed, the stability condition for the Leapfrog method, found in Equation 30, is twice as high as Forward Euler's condition

$$\Delta t \leq 2\sqrt{\frac{m}{k}}. \tag{30}$$

4.2.3 Hyperprior Specification and Initialization for Harmonic Oscillator

The role played by the hyperprior is defined previously in this chapter. This section is only concerned with the numerical values used for the hyperprior and initial values for the damped harmonic oscillator problem.

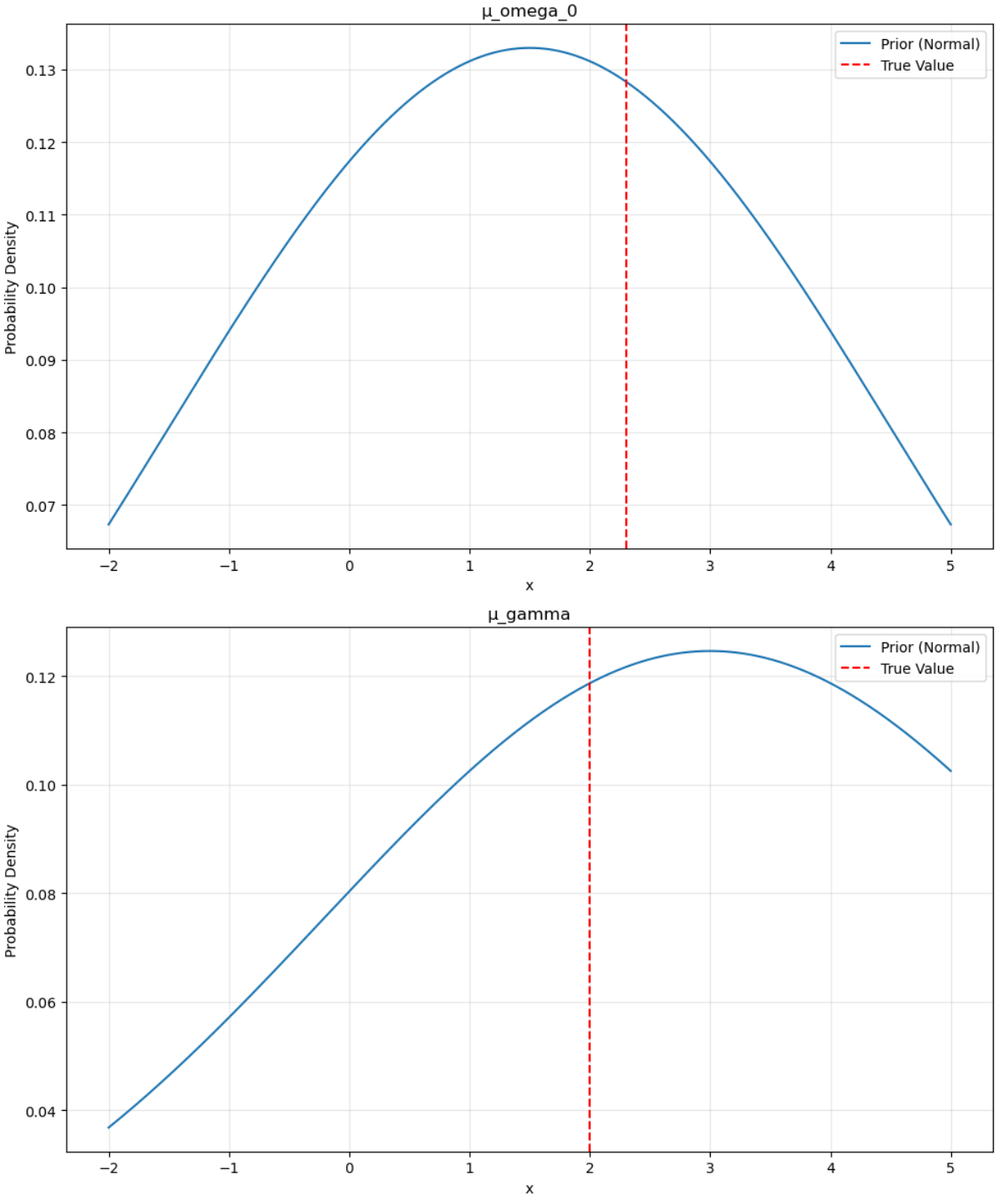

Figure 3: Hyperpriors of μ (mean) terms

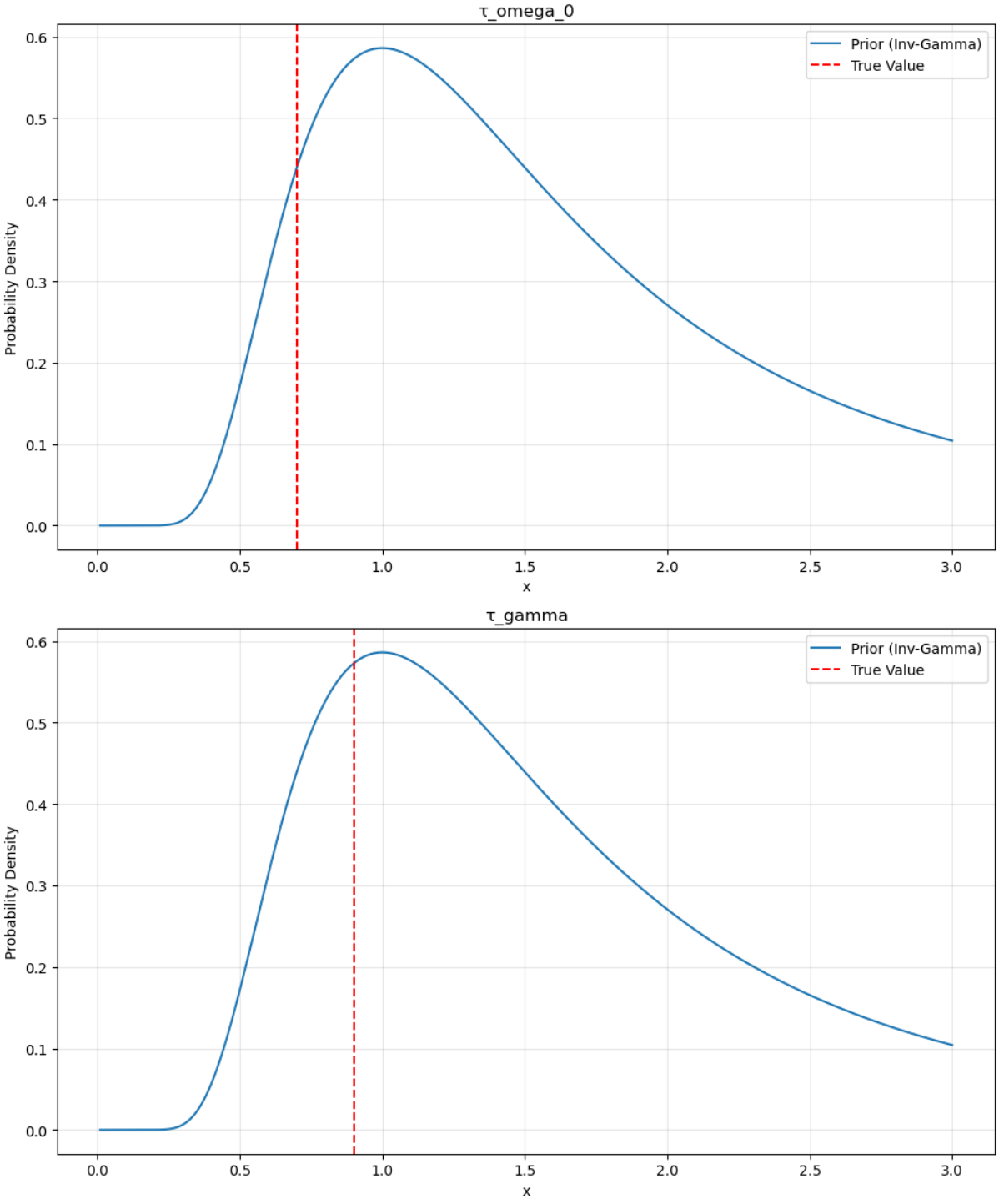

Figure 4: Hyperpriors of τ (standard deviation) terms

These distributions are relevant for the HBM methodology where we look at distributions. We set the initial values to be the same for both the DM and HBM methodologies. These

values can be found in the table below.

Parameter	μ_ω	τ_ω	μ_γ	τ_γ
Initial Value for DM methodology	1.9	0.9	2.8	1.0
True Value	2.3	0.7	2.0	0.9

Table 1: True and Initial Values used for Damped Harmonic Oscillator problem

The main takeaway from these hyperpriors and initialisation values is that they are reasonable starting points, without being too informative, but at the same time, provide the model with enough information to work with. Quite logically, if we gave more informative priors, the model would be able to learn the parameters faster, and vice versa with less informative priors.

Another important graph to visualise the hyperpriors is by looking at the pushforward (Figure 5). The pushforward is meant to show the impact of the hyperprior on the populational level. This can be done by sampling a μ and τ value from the hyperprior distribution, and then "pushing forward" through the log-normal distribution to have a sample population value. The push-forward measure is mathematically defined as follows

Let $(\mathcal{Z}, \mathcal{F}_\mathcal{Z})$ and $(\mathcal{Y}, \mathcal{F}_\mathcal{Y})$ be measurable spaces, let

$$G \,:\, \mathcal{Z} \longrightarrow \mathcal{Y}$$

be a measurable map, and let P be a probability measure on $(\mathcal{Z}, \mathcal{F}_\mathcal{Z})$. The *push-forward* of P through G, denoted $G_\# P$, is the measure on $(\mathcal{Y}, \mathcal{F}_\mathcal{Y})$ defined by

$$\big(G_\# P\big)\big(A\big) \;=\; P\big(G^{-1}(A)\big) \qquad \text{for every } A \in \mathcal{F}_\mathcal{Y}.$$

We then simulate 10,000 realisations to empirically estimate the population distribution.

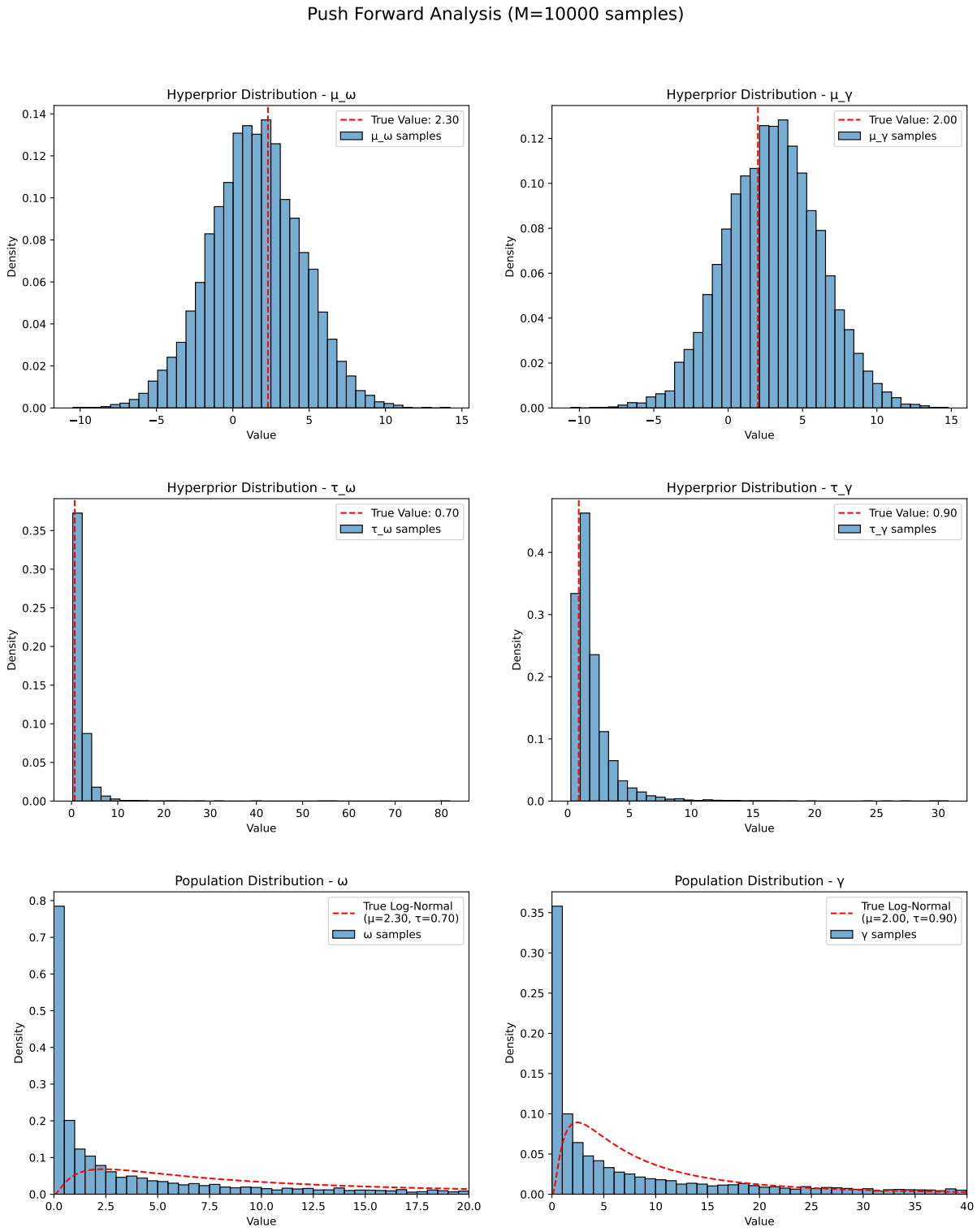

Figure 5: Hyperprior distributions for the mean (μ) and standard deviation (τ) of ω and γ are shown with vertical lines marking true values. The pushforward distributions depict how hyperparameter uncertainty induces population-level variability in ω and γ.

4.2.4 Generalisation to other physical systems

The power of using the damped harmonic oscillator for our problem lies in its ability to generalise to other linear problems. More specifically, several other engineering and physical problems have the same model form as the damped harmonic oscillator in mechanics and electricity.

Translational mechanical	**Rotational mechanical**	**Series RLC circuit**	**Parallel RLC circuit**
Position x	Angle θ	Charge q	Flux linkage φ
Velocity $\frac{dx}{dt}$	Angular velocity $\frac{d\theta}{dt}$	Current $\frac{dq}{dt}$	Voltage $\frac{d\varphi}{dt}$
Mass m	Moment of inertia I	Inductance L	Capacitance C
Momentum p	Angular momentum L	Flux linkage φ	Charge q
Spring constant k	Torsion constant μ	Elastance $1/C$	Magnetic reluctance $1/L$
Damping c	Rotational friction Γ	Resistance R	Conductance $G = 1/R$
Drive force $F(t)$	Drive torque $\tau(t)$	Voltage v	Current i
Undamped resonant frequency f_n:			
$\frac{1}{2\pi}\sqrt{\frac{k}{m}}$	$\frac{1}{2\pi}\sqrt{\frac{\mu}{I}}$	$\frac{1}{2\pi}\sqrt{\frac{1}{LC}}$	$\frac{1}{2\pi}\sqrt{\frac{1}{LC}}$
Damping ratio ζ:			
$\frac{c}{2}\sqrt{\frac{1}{km}}$	$\frac{\Gamma}{2}\sqrt{\frac{1}{I\mu}}$	$\frac{R}{2}\sqrt{\frac{C}{L}}$	$\frac{G}{2}\sqrt{\frac{L}{C}}$
Differential equation:			
$m\ddot{x} + c\dot{x} + kx = F$	$I\ddot{\theta} + \Gamma\dot{\theta} + \mu\theta = \tau$	$L\ddot{q} + R\dot{q} + q/C = v$	$C\ddot{\varphi} + G\dot{\varphi} + \varphi/L = i$

Table 2: Analogies between mechanical and electrical systems

We could also generalise problems modeled using non-homogeneous second-order differential equations (we only look at homogeneous in this first experiment).

4.3 Experiment 2: Lotka-Volterra

To verify the robustness of our methodology, we have also designed a very similar experiment for non-linear equations. In the interest of space, we will not go into as much detail as previously, given the close similarity in methodologies.

The governing equations of the Lotka-Volterra model are:[33]

$$\begin{aligned} \frac{dx}{dt} &= \alpha x - \beta xy \\ \frac{dy}{dt} &= \delta xy - \gamma y \end{aligned} \tag{31}$$

This time, we will use the Runge-Kutta 4th order as our numerical operator. An example of what the prey-predator looks like visually is given below:

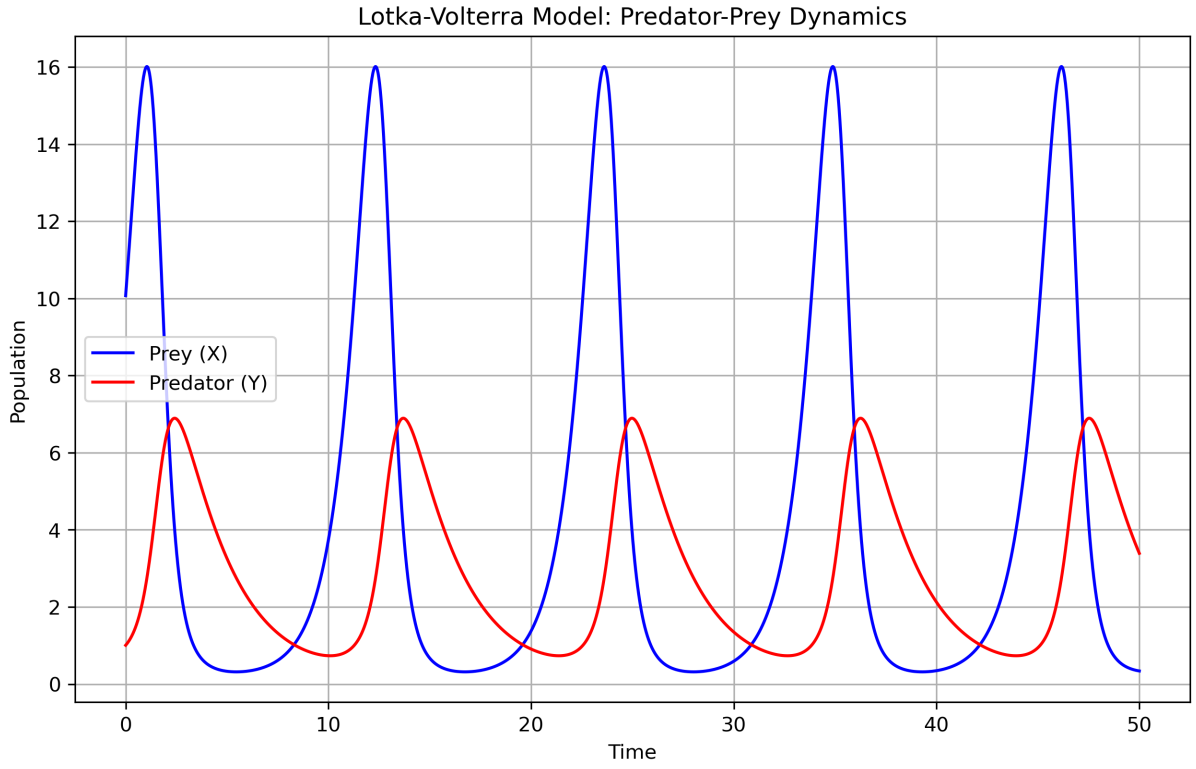

Figure 6: Example Predator-Prey model

5 Results & Discussion

To ensure statistical robustness and quantify estimation variability, we conduct 100 independent experimental realizations following the methodology outlined in Section 4.1. Each methodology (Hierarchical Bayesian Models and distribution-matching) is evaluated separately to characterize their convergence properties, parameter recovery accuracy, and

computational scaling behavior. Subsequently, we present a comprehensive comparative analysis examining both inferential performance and runtime efficiency across varying population sizes $N \in \{1, 10, 50, 100, 500, 1000\}$.

5.1 Accuracy Performance

5.1.1 HBMs

This section examines the MCMC sampling performance and resulting posterior distributions obtained from the HBM methodology. We analyze the posterior summary statistics across population sizes ranging from $N = 1$ to $N = 1000$, characterizing the evolution from diffuse to concentrated posterior distributions as sample size increases.

Each Markov chain was initialized with a 500-iteration warmup phase to ensure convergence to regions of high posterior probability, followed by 1000 sampling iterations. The warmup phase enables the sampler to reach the typical set of the posterior distribution, thereby ensuring that subsequent samples provide valid Monte Carlo estimates of posterior expectations.

For the single-system case ($N = 1$), Figure 7 reveals substantial challenges in posterior exploration. The trace plots exhibit poor mixing characteristics, with sporadic excursions to extreme values exceeding 10 for parameters μ_γ, τ_ω, and τ_γ. Only the μ_ω parameter demonstrates reasonable convergence behavior. This pattern indicates that limited data from a single system provides insufficient information for reliable hyperparameter inference, resulting in a diffuse posterior with significant uncertainty.

Table 3: Posterior summary statistics for $N = 1$ system

Parameter	5th percentile	Mean	95th percentile
μ_ω	-0.36	2.46	4.42
μ_γ	0.24	2.15	4.74
τ_ω	0.48	1.87	3.43
τ_γ	0.30	1.63	2.94

The posterior credible intervals presented in Table 3 corroborate this assessment, with 90% credible intervals spanning several units for all parameters. The wide posterior distributions reflect the fundamental limitation of inferring population-level parameters

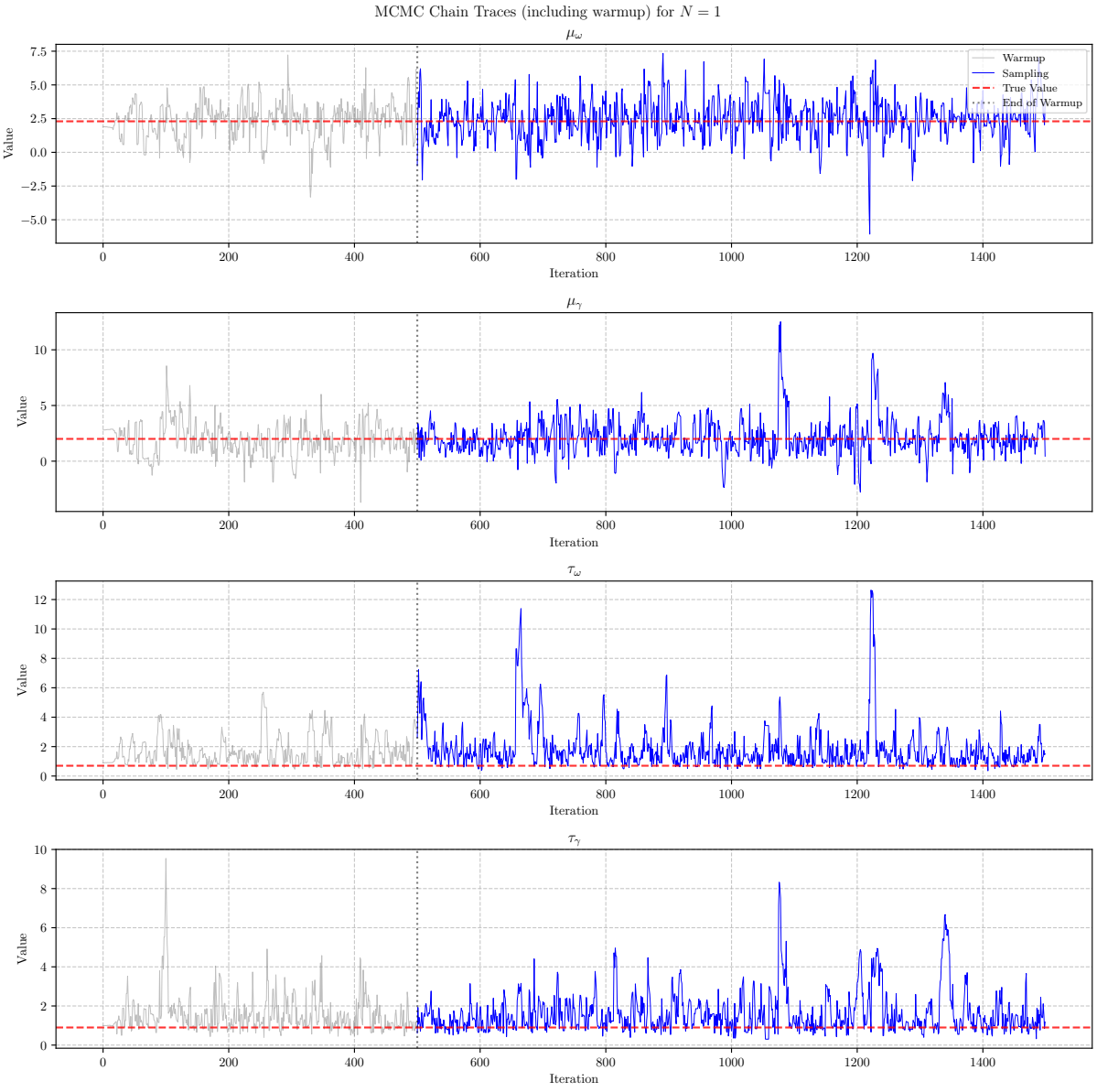

Figure 7: MCMC trace plots for $N = 1$ system showing hyperparameter evolution during warmup and sampling phases.

from minimal data.

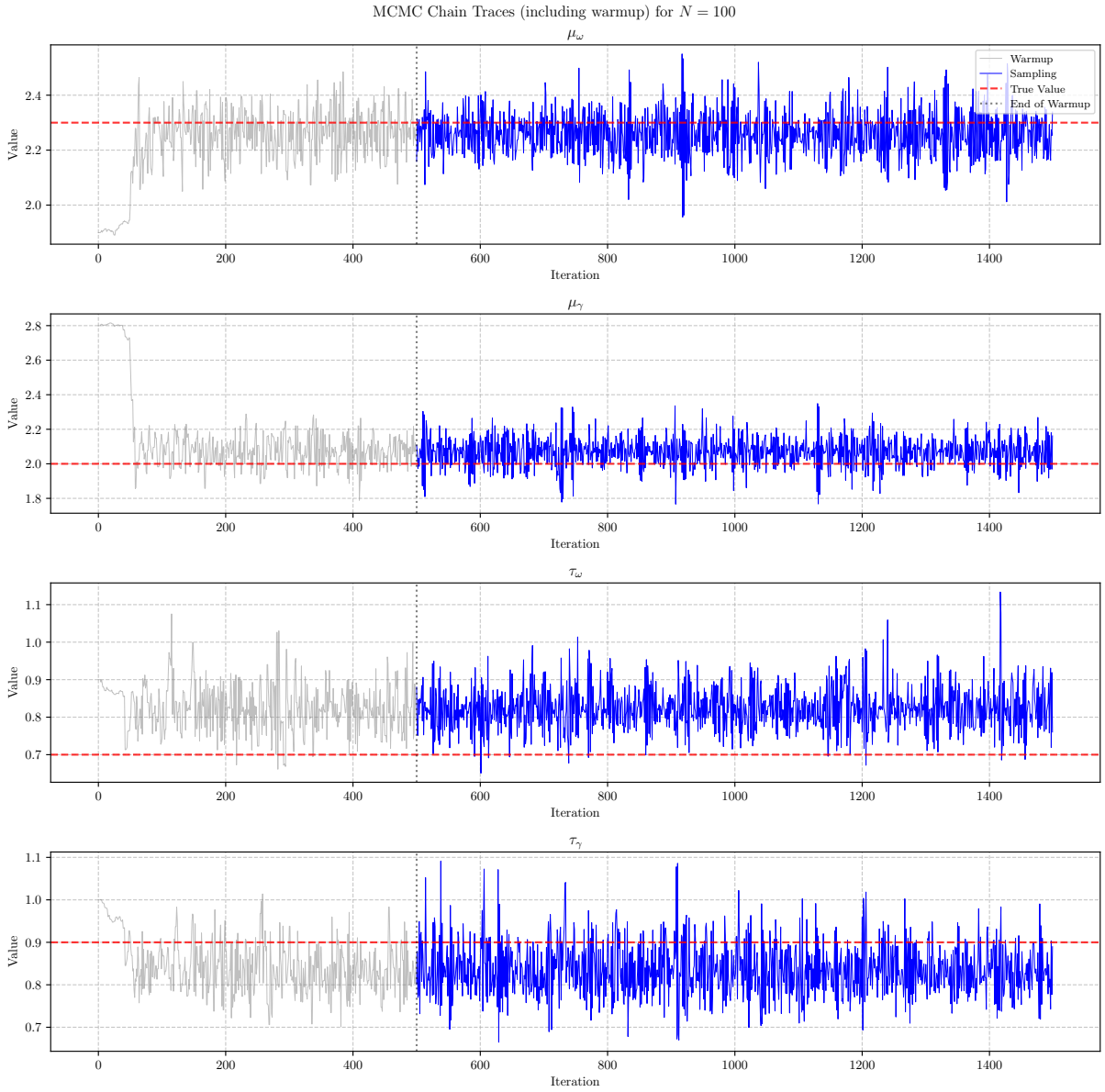

Figure 8: MCMC trace plots for $N = 100$ systems demonstrating improved but suboptimal convergence.

Increasing the population size to $N = 100$ yields marked improvements in sampler behavior, as evidenced in Figure 8. The trace plots exhibit reduced variance and more stable exploration of the posterior space. The location parameters μ_ω and μ_γ demonstrate good convergence to their true values (indicated by horizontal reference lines). However, the scale parameters τ_ω and τ_γ display systematic bias, with τ_ω particularly underestimating the true value. This asymmetric performance between location and scale parameters

suggests inherent challenges in estimating population variability from finite samples.

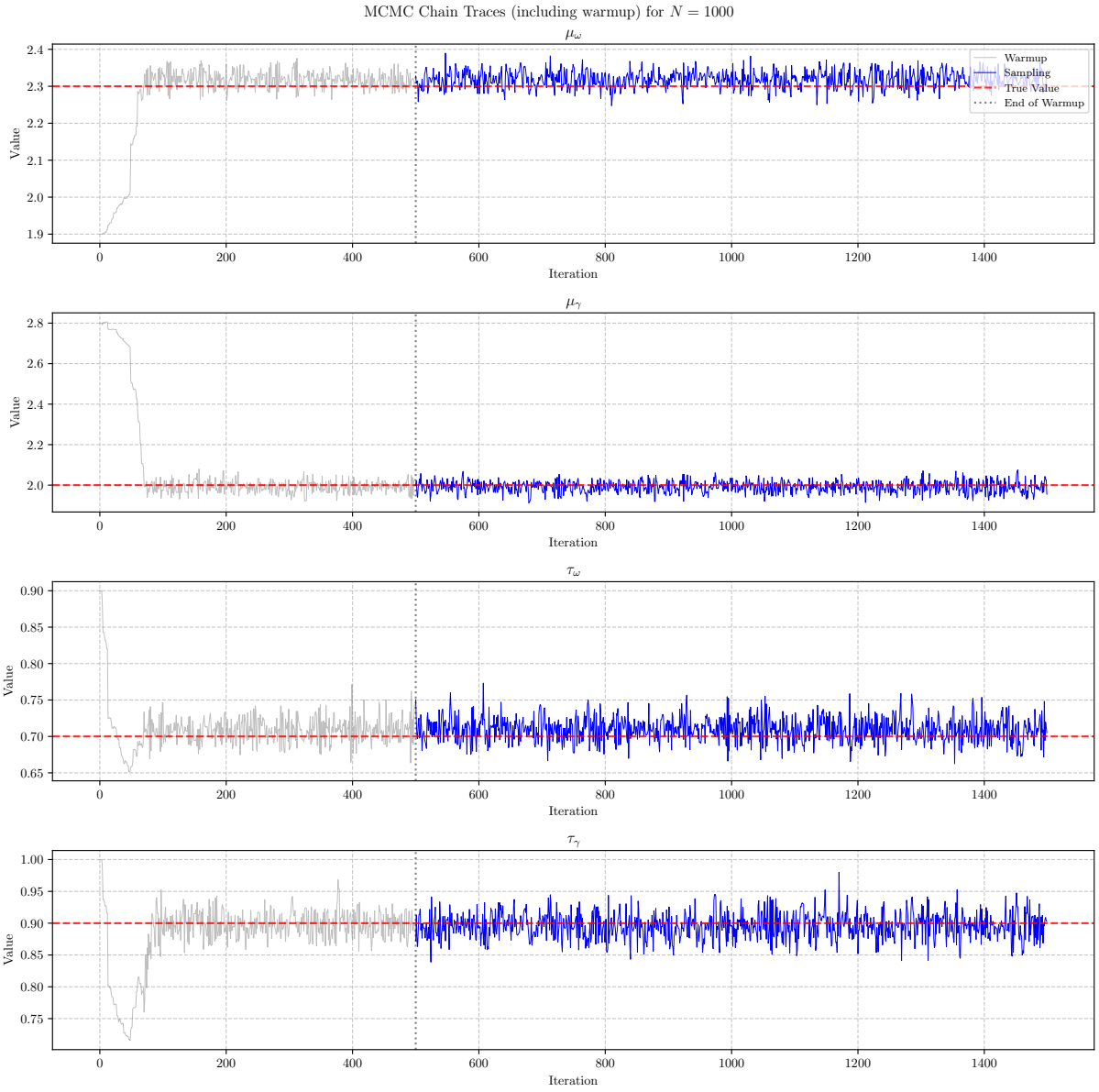

Figure 9: MCMC trace plots for $N = 1000$ systems showing excellent convergence properties.

The large population regime ($N = 1000$) demonstrates the asymptotic efficiency of the HBM approach. Figure 9 reveals tight, well-mixed chains with minimal variance around the posterior mean. The sampler efficiently explores the concentrated posterior distribution, indicating that sufficient data effectively constrains the hyperparameter space.

The posterior summary statistics in Table 4 confirm the precision of hyperparameter

Table 4: Posterior summary statistics for $N = 1000$ systems

Parameter	5th percentile	Mean	95th percentile
μ_ω	2.28	2.32	2.35
μ_γ	1.95	1.99	2.04
τ_ω	0.68	0.71	0.74
τ_γ	0.86	0.90	0.93

estimation at large N. The 90% credible intervals span less than 0.1 units for all parameters, representing an order-of-magnitude reduction in uncertainty compared to the single-system case. These results demonstrate that HBM performance scales favorably with population size, transitioning from highly uncertain estimates at small N to precise inference as N increases.

5.1.2 Distribution-matching

The distribution-matching methodology demonstrates robust convergence behavior across varying population sizes, as evidenced by the optimization trajectories presented in Figures 10 and 11. The loss function evolution exhibits characteristic features of successful gradient-based optimization, with rapid initial descent followed by asymptotic convergence to stable minima.

For the single-system case ($N = 1$), Figure 10 reveals an initial loss value of approximately 17, which undergoes steep reduction within the first 250 iterations. The optimization trajectory follows an exponential decay pattern, with the gradient magnitude decreasing as the algorithm approaches the optimal hyperparameter configuration. After 2000 iterations, the loss stabilizes at approximately 8, indicating convergence to a local minimum of the Sliced-Wasserstein distance combined with the Tikhonov regularization term.

The large population regime ($N = 1000$) exhibits similar convergence characteristics, albeit from a higher initial loss value of 22, as shown in Figure 11. This elevated starting point reflects the increased complexity of matching distributions when aggregating observations from multiple systems. Despite this initial challenge, the optimizer successfully navigates the loss landscape, achieving a final loss value of approximately 8.3. The marginally higher terminal loss compared to the $N = 1$ case is consistent with the increased data complexity inherent in larger population sizes.

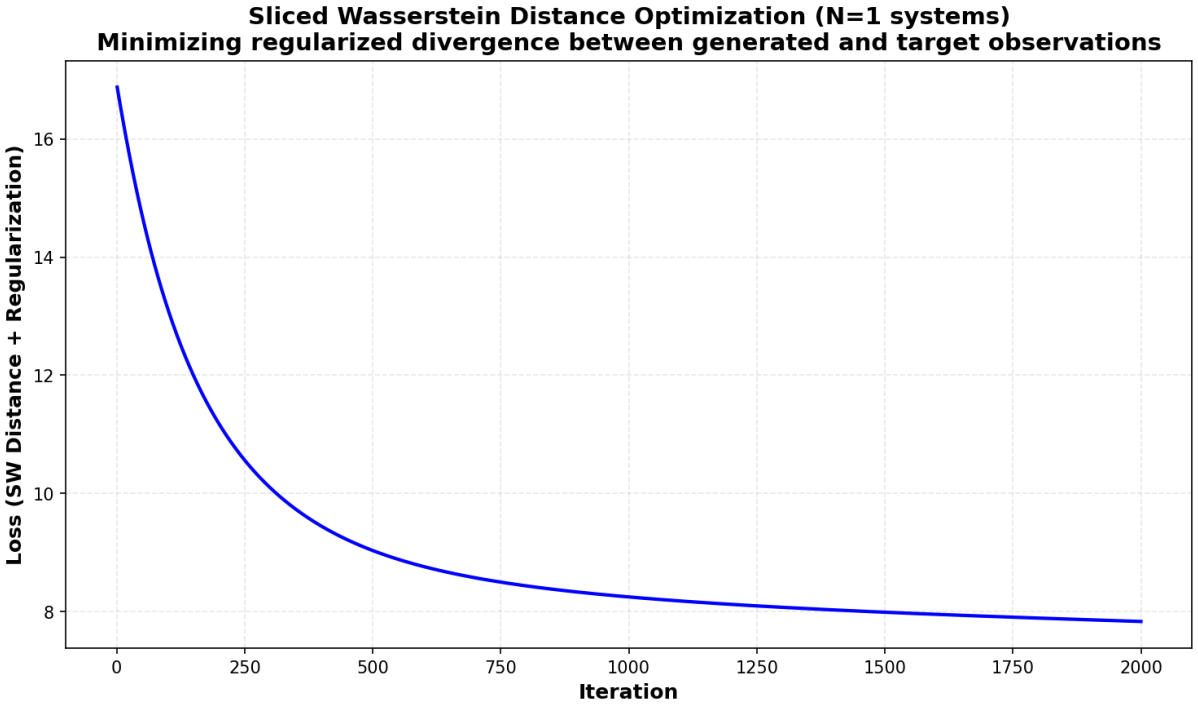

Figure 10: Loss function with respect to number of iterations for N = 1

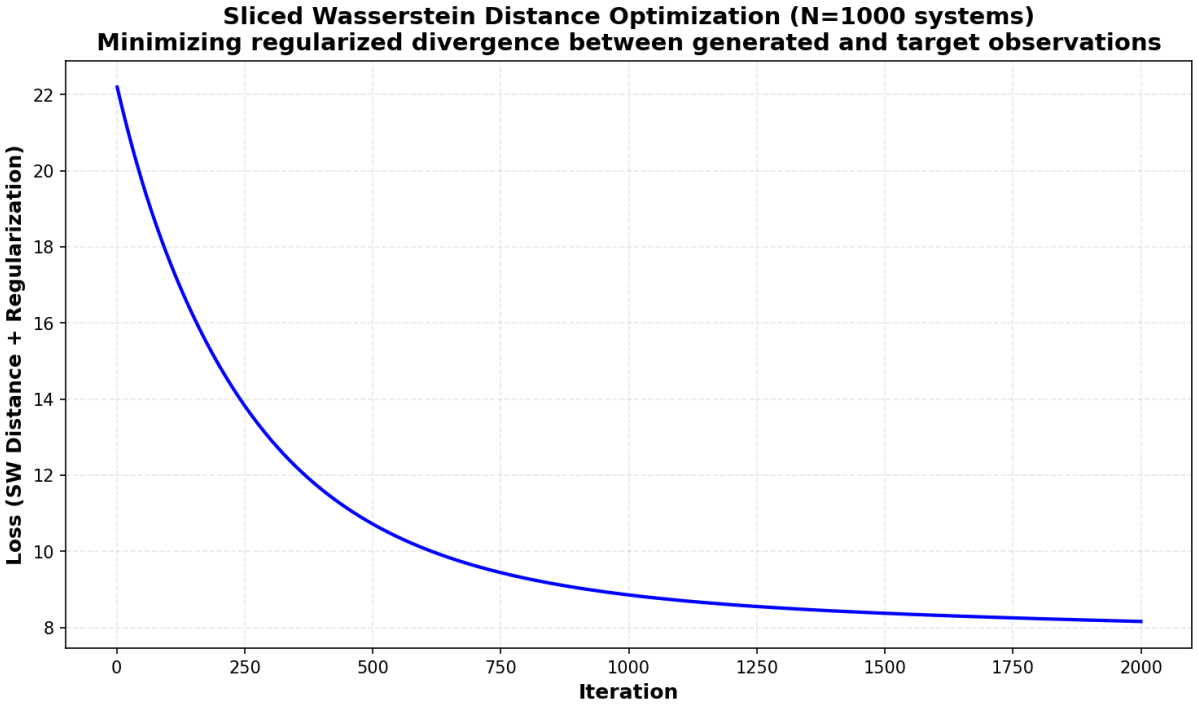

Figure 11: Loss function with respect to number of iterations for N = 1000

Both optimization trajectories demonstrate the effectiveness of the ADAM optimizer in minimizing the regularized Sliced-Wasserstein distance. The smooth, monotonic decrease in loss values confirms the absence of significant local minima or optimization instabilities. The convergence rate, characterized by rapid initial progress followed by diminishing returns, aligns with theoretical expectations for first-order optimization methods applied to well-conditioned objective functions. These results validate the distribution-matching approach as a computationally tractable alternative for hierarchical parameter inference, particularly in scenarios where uncertainty quantification is not the primary concern.

5.1.3 Comparing performances between methodologies

In this section, we will aim to compare the performances of the HBM methodology and the distribution-matching methodology. In both cases, we observed that increasing the number of systems helped reach better performances. But how do they compare against one another? It is worth reiterating that although HBMs output a distribution for these hyperparameter values, the distribution-matching methodology will return only the optimal values it found. This distinction is crucial when comparing results.

The following figures summarise the results over all 100 experiments of both methodologies and are undoubtedly the most important figures of this report.

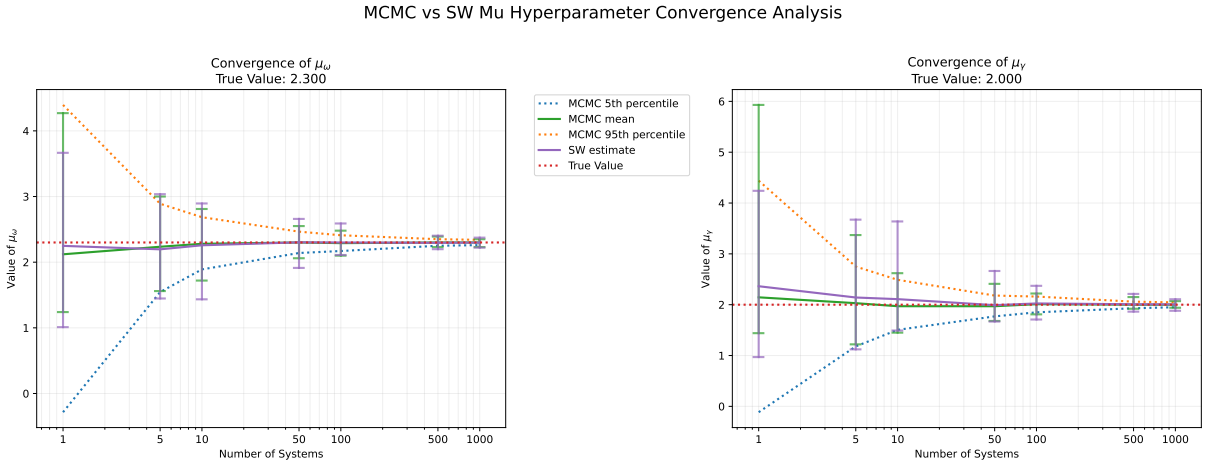

Figure 12: Estimated μ values for different N values

The convergence analysis compares Hierarchical Bayesian Models (HBM) and Sliced-Wasserstein (SW) distribution-matching methods across population sizes N. The plots

display MCMC posterior means (blue lines) with 5th and 95th percentile credible intervals (orange bands), SW point estimates (green lines), and empirical min-max bounds from 100 experimental realizations (purple brackets). True hyperparameter values are indicated by red dashed lines.

For small population sizes ($N < 10$), both methods exhibit substantial estimation uncertainty. The SW min-max brackets span wide ranges, indicating high variability across experimental realizations. The MCMC credible intervals are similarly broad, but crucially, they quantify this uncertainty within a single experiment. As N increases, the SW brackets tighten dramatically, showing improved consistency across experiments. Simultaneously, the MCMC credible intervals narrow, with the posterior distribution becoming increasingly concentrated around the true values.

At large population sizes ($N \geq 100$), both methods achieve comparable point estimation accuracy. The SW estimates converge to values near the MCMC posterior means, and both approaches successfully recover the true parameter values. The min-max brackets become narrow, indicating consistent performance across experimental realizations. The MCMC credible intervals also tighten substantially, reflecting high posterior certainty in the parameter estimates.

The fundamental distinction between methods lies in uncertainty quantification. The MCMC approach provides calibrated credible intervals that consistently contain the true parameter values across all population sizes. These intervals explicitly communicate estimation reliability—wide intervals at small N warn practitioners of high uncertainty, while narrow intervals at large N indicate reliable estimates. Notably, the SW min-max brackets typically fall within these credible intervals, suggesting the MCMC posterior captures the empirical variability observed across multiple experiments. In contrast, SW methods provide only point estimates without uncertainty measures. While aggregate performance across 100 experiments may appear satisfactory, practitioners conducting single experiments cannot assess whether their specific realization yielded reliable estimates.

Parameter-specific performance differs markedly between μ and τ. The μ parameters demonstrate robust estimation with well-calibrated uncertainty quantification across all population sizes. However, τ parameters exhibit inferior performance, with wider credible intervals and slower convergence toward true values. This disparity suggests inherent challenges in estimating scale parameters within hierarchical structures, affecting both methodologies but particularly evident in the persistent uncertainty of τ estimates even at

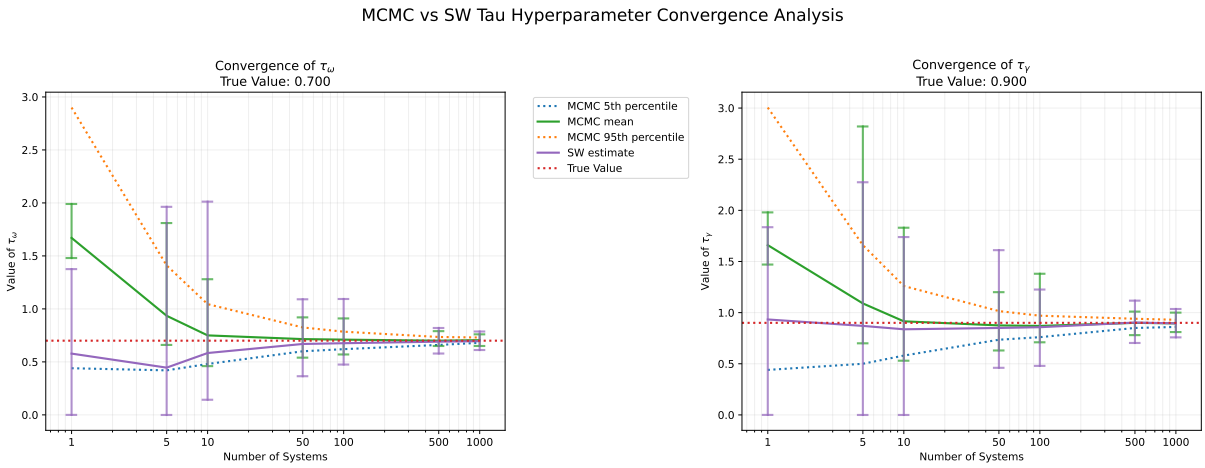

Figure 13: Estimated τ values for different N values

larger population sizes.

5.2 Computational Performance

Having established the comparative accuracy performance of both methodologies, we now examine their computational efficiency. It is worth noting that the MCMC was executed on a CPU architecture. While GPU implementations were tested, they demonstrated slower performance than CPU-based execution for our specific algorithmic implementations. On the other hand, the code used for the distribution-matching method uses a GPU, as it uses code written to optimise performance on GPUs from Girolami and Vadeboncoeur's work.[1]

The runtime analysis on Figure 14 reveals distinct computational scaling patterns between methodologies. The computational cost of MCMC increases with the number of systems in the hierarchical model. The theoretical reasoning for this observation was already developed in Chapter 3.2. This scaling reflects the expanding posterior sampling complexity as additional systems are incorporated. The Sliced-Wasserstein distribution-matching approach maintains constant computational performance regardless of population size. This efficiency stems from the gradient-based optimization framework operating on fixed-dimensional hyperparameters independent of N. The optimization complexity remains unchanged as the number of systems increases, providing superior scalability for large population inference problems.

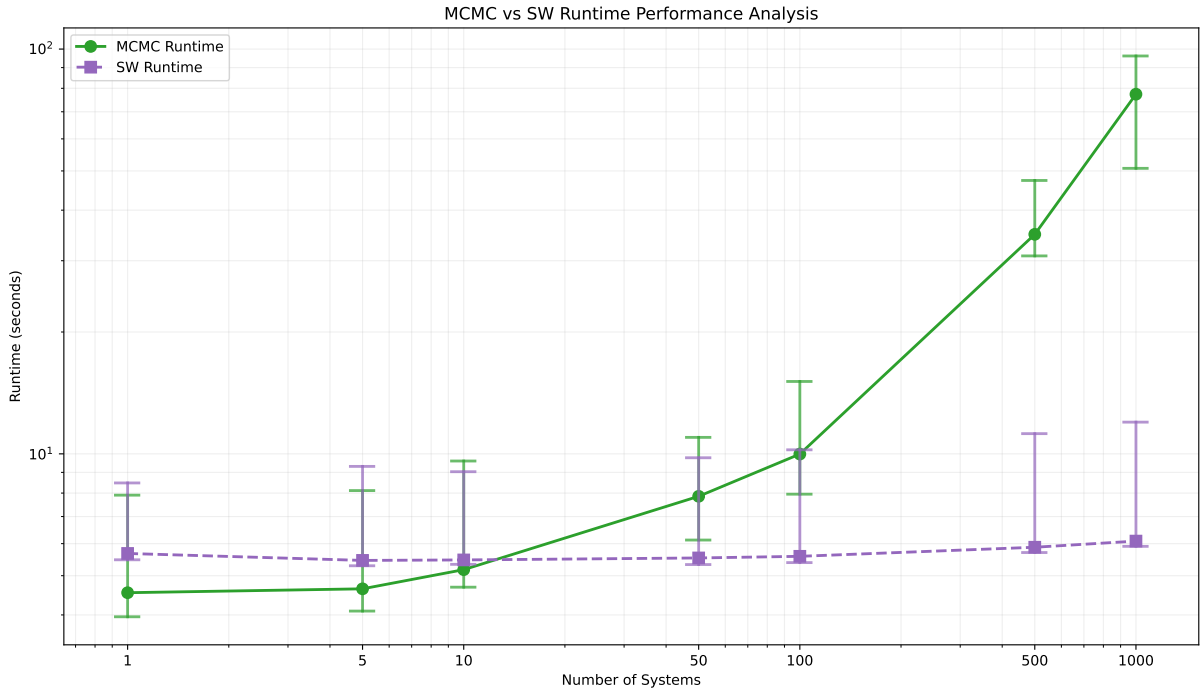

Figure 14: Runtime comparison for different N values across 100 experiments

From Figure 14, we notice that MCMC generally exhibits a smaller runtime than the distribution-matching approach for small population sizes ($N < 10$). However, for larger populations ($N > 10$), the distribution-matching methodology has a shorter runtime on average, as the MCMC's runtime scales approximately linearly. This means that, for equal performance, it would be more suitable to run the distribution-matching methodology when $N > 10$ and the HBM methodology when $N < 10$ in this particular experiment setup.

5.3 Results for a non-linear system

The same methodology, applied to the Lotka-Volterra equations, yields some similar conclusions as mentioned before. Firstly, we notice both models are able to learn very well the mean of parameters α and β in Equation 31, as seen on Figure 15. Also, the runtime (Figure 16) has a shape very similar to Figure 14. On the other hand, the learning of parameters γ and δ is much more complex for both methodologies. This complex learning is likely the result of identifiability issues and requires further investigation.

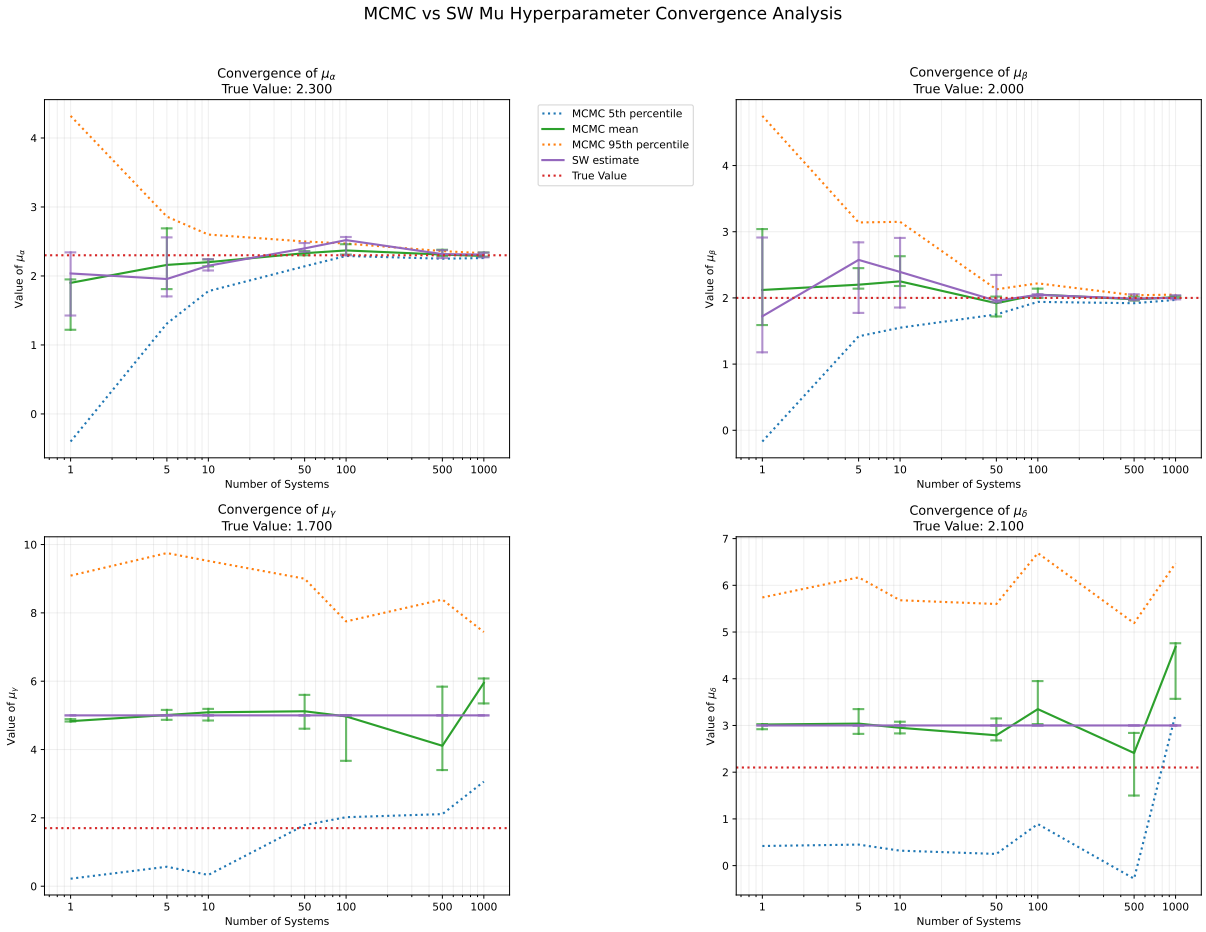

Figure 15: Estimated μ values for different N values

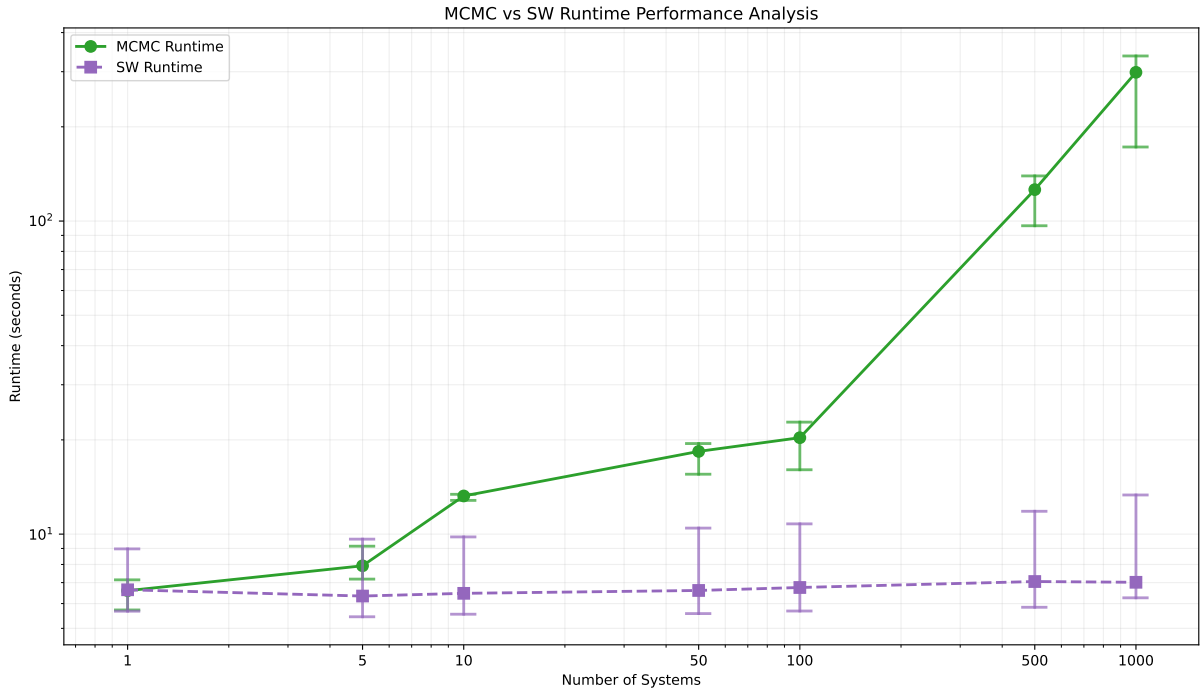

Figure 16: Runtime comparison for different N values across 100 experiments

6 Conclusion

This comparative analysis establishes fundamental performance characteristics and computational trade-offs between Hierarchical Bayesian Models and distribution-matching approaches for populational inverse problems. The investigation reveals that the optimal methodological selection depends critically on the intersection of population size, computational constraints, and uncertainty quantification requirements, with each approach demonstrating distinct advantages across different operational regimes.

The experimental framework successfully demonstrates that HBMs provide superior robustness through calibrated credible intervals and inherent uncertainty propagation, while DM methods achieve computational scalability through constant complexity $\mathcal{O}(1)$ independent of population size compared to the linear scaling $\mathcal{O}(N)$ exhibited by MCMC-based approaches. Both methodologies consistently demonstrate superior performance in location parameter estimation (μ) relative to scale parameter estimation (τ), revealing systematic challenges in population variability inference that affect hierarchical structures regardless of the underlying algorithmic framework.

A critical quantitative finding emerges at approximately $N \approx 10$ systems, where computational efficiency considerations begin to favor distribution-matching approaches while maintaining comparable statistical performance. However, this threshold exhibits sensitivity to the specific parameters of interest, with variance parameter estimation showing markedly different characteristics that significantly impact the computational trade-off analysis.

The systematic evaluation across 100 independent realizations provided robust statistical validation of these performance boundaries. The non-centered parametrization implementation significantly improved MCMC convergence properties, while the Sliced-Wasserstein distance formulation proved computationally tractable for gradient-based optimization. The damped harmonic oscillator system served as an effective controlled comparison framework, establishing baseline performance characteristics for linear systems.

Preliminary investigations extending the methodology to non-linear systems through Lotka-Volterra equations implementation yielded promising but inconclusive results. While the framework demonstrates applicability beyond linear systems, the complexity of non-linear parameter estimation requires more extensive analysis to establish definitive performance comparisons between the methodological approaches.

Several methodological limitations warrant acknowledgment. The choice of Sliced-Wasserstein distance, while computationally tractable, may not capture all relevant distributional differences compared to full Wasserstein metrics. The hyperprior specifications, though designed to be weakly informative, inevitably influence comparative performance in ways that require more systematic sensitivity analysis. Additionally, the persistent difficulties in scale parameter estimation across both methodologies suggest fundamental limitations when inferring population-level variability from finite samples.

The scale parameter estimation challenges observed across both methodologies indicate that alternative hierarchical specifications or advanced MCMC techniques specifically designed for variance parameter inference merit investigation. The asymmetric performance between location and scale parameters reflects deeper statistical challenges in hierarchical inference that extend beyond the specific algorithmic implementations examined.

This work contributes to the broader understanding of methodological selection in Physics-Informed Machine Learning by establishing quantitative performance boundaries and providing principled guidance for practitioners facing the fundamental trade-off between

statistical rigor and computational feasibility in populational inference problems. The established framework provides essential foundations for informed methodological selection across diverse application domains requiring populational parameter estimation.

6.1 Future Work

Future research should prioritize extending the framework to non-linear physical systems to validate the generalizability of our methodology. The preliminary Lotka-Volterra implementation provides a foundation for this extension, though more comprehensive analysis is required to establish definitive performance boundaries in non-linear contexts. Investigation of adaptive hybrid methodologies that dynamically switch between HBM and DM approaches based on real-time performance metrics could be promising for combining the strengths of both frameworks. However, there are still potential for further research in linear systems. Firstly, we could investigate whether advanced MCMC techniques specifically designed for hierarchical scale parameter inference could address the systematic performance limitations observed in variance parameter estimation. Also, while the present study employs weakly informative priors, practical applications frequently encounter scenarios where priors yield substantially incorrect specifications. Future work should therefore examine the degradation of both inference methodologies under adversarial prior conditions, particularly investigating whether the observed $N \approx 10$ transition point shifts when hyperpriors are miscentered by orders of magnitude or when prior variances severely underestimate true population heterogeneity.

In future work, applying our inference framework to real-world systems, such as wind farms where individual turbines exhibit slight variations in efficiency and wear, could enable more precise predictive maintenance and optimize collective performance, thereby reducing downtime, material waste, and overall carbon footprint. Ethically, ensuring that uncertainty quantification from hierarchical Bayesian models or confidence bounds in distribution matching is transparently communicated will help prevent overreliance on point estimates and avoid inappropriate decisions, for example, prematurely retiring components or overinvesting in unnecessary repairs. Moreover, by embedding fairness principles into our hierarchical models, such as treating similar turbines comparably regardless of location or ownership, we can mitigate potential biases that might disadvantage certain stakeholders. On the environmental front, extending the methodology to incorporate life-cycle assessments of materials or energy consumption during model calibration could

directly inform more sustainable design choices, such as selecting components or control strategies that minimize emissions over a turbine's lifetime. Finally, by open-sourcing our methodology and implementation after this project, we foster reproducibility and collaborative scrutiny, which not only strengthens ethical standards but also accelerates the development of greener, more accountable engineering solutions.

References

[1] O Deniz Akyildiz, Mark Girolami, Andrew M Stuart, and Arnaud Vadeboncoeur. Efficient prior calibration from indirect data. *arXiv preprint arXiv:2405.17955*, 2024.

[2] M Allenby Greg, E Rossi Peter, and E McCulloch Robert. *Hierarchical Bayes Models: A Practitioners Guide*. 2005.

[3] Hedy Attouch, Aïcha Balhag, Zaki Chbani, and Hassan Riahi. Accelerated gradient methods combining tikhonov regularization with geometric damping driven by the hessian. *Applied Mathematics & Optimization*, 88(2):29, 2023.

[4] Julian Besag, Peter Green, David Higdon, and Kerrie Mengersen. Bayesian computation and stochastic systems. *Statistical science*, pages 3–41, 1995.

[5] M Betancourt and M. Girolami. *Hamiltonian Monte Carlo for Hierarchical Models*. 2013.

[6] Michael Betancourt. A conceptual introduction to hamiltonian monte carlo. *arXiv preprint arXiv:1701.02434*, 2017.

[7] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51:22–45, 2015.

[8] Alan C Bovik. *Handbook of image and video processing*. Academic press, 2010.

[9] Daniela Calvetti and Erkki Somersalo. Distributed tikhonov regularization for ill-posed inverse problems from a bayesian perspective. *Computational Optimization and Applications*, pages 1–32, 2025.

[10] Ming-Hui Chen, Qi-Man Shao, and Joseph G Ibrahim. *Monte Carlo methods in Bayesian computation*. Springer Science & Business Media, 2012.

[11] Xiongjie Chen, Yongxin Yang, and Yunpeng Li. Augmented sliced wasserstein distances. *arXiv preprint arXiv:2006.08812*, 2020.

[12] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

[13] Grant Fowles, George Cassiday, and RW Robinett. Analytical mechanics, 2000.

[14] Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. *Bayesian Data Analysis*. 2013.

[15] Andrew Gelman, Gareth O Roberts, and Walter R Gilks. Efficient metropolis jumping rules. *Bayesian statistics 5*, 5:599–608, 1996.

[16] Walter R Gilks, Sylvia Richardson, and David Spiegelhalter. *Markov chain Monte Carlo in practice*. CRC press, 1995.

[17] Rohit Gupta, Rahul Gupta, and Sonica Rajput. Analysis of damped harmonic oscillator by matrix method. *International Journal of Research and Analytical Reviews (IJRAR)*, 2018.

[18] Kenneth M Hanson. *Markov Chain Monte Carlo posterior sampling with the Hamiltonian method*. 2001.

[19] Kenneth M Hanson and Gregory S Cunningham. Posterior sampling with improved efficiency. In *Medical Imaging 1998: Image Processing*, volume 3338, pages 371–382. SPIE, 1998.

[20] Matthew D Hoffman, Andrew Gelman, et al. *The No-U-Turn sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo.* 2014.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] Erwin Kreyszig, K Stroud, and G Stephenson. Advanced engineering mathematics. *Integration*, 9(4):1014, 2008.

[23] Alex Lipp and Pieter Vermeesch. The wasserstein distance as a dissimilarity metric for comparing detrital age spectra and other geological distributions. *Geochronology*, 5(1):263–270, 2023.

[24] Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.

[25] Khai Nguyen and Nhat Ho. Energy-based sliced wasserstein distance. *Advances in Neural Information Processing Systems*, 36:18046–18075, 2023.

[26] Sloan Nietert, Ziv Goldfeld, and Kengo Kato. Smooth p-wasserstein distance: structure, empirical approximation, and statistical applications. In *International Conference on Machine Learning*, pages 8172–8183. PMLR, 2021.

[27] Du Phan, Neeraj Pradhan, and Martin Jankowiak. *Composable effects for flexible and accelerated probabilistic programming in NumPyro*. 2019.

[28] Christian P Robert, George Casella, and George Casella. *Monte Carlo statistical methods*, volume 2. Springer, 1999.

[29] Linda SL Tan and David J Nott. Variational inference for generalized linear mixed models using partially noncentered parametrizations. 2013.

[30] Luke Tierney. *Markov Chains for Exploring Posterior Distributions*. 1994.

[31] Paul A Tipler. *Physics for Scientists and Engineers: Regular Version, Ch. 1-35 and 39*. Macmillan, 1999.

[32] Cédric Villani. *Topics in optimal transportation*, volume 58. American Mathematical Soc., 2021.

[33] Peter J Wangersky. Lotka-volterra population models. *Annual Review of Ecology and Systematics*, 9:189–218, 1978.

[34] Yaming Yu and Xiao-Li Meng. To center or not to center: That is not the question—an ancillarity–sufficiency interweaving strategy (asis) for boosting mcmc efficiency. *Journal of Computational and Graphical Statistics*, 20(3):531–570, 2011.

Appendix A: Risk Assessment Retrospective

In the Risk Assessment form completed at the beginning of this project, no hazards were found likely to be encountered in the project. In retrospect, the only potential hazards that could have occurred would have been linked with the use of GPUs from Virtual Machines, but this has not posed any complications in this project.