Optimization and Regularization Under Arbitrary Objectives

Jared Lakhani
Supervisor: Dr Etienne Pienaar

Neural networks, along with various other machine learning architectures, constitute a class of overparameterized models - models with more parameters than training samples. As a consequence of this overparameterization, such models exhibit a propensity to achieve near-perfect fit on the training data yet require the incorporation of regularization techniques to ensure satisfactory generalization to previously unseen data. This concept of overparameterization can be formally understood through the Vapnik-Chervonenkis (VC) Generalisation Bound, which relates in-sample and out-of-sample errors. With probability $1-\delta$ we have: $E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{N}\ln\left(\frac{4(2N)^{d_{VC}}+1}{\delta}\right)}$ [1], wherein the VC dimension d_{VC} serves as a measure of model complexity. Specifically, an increase in the complexity of the hypothesis space corresponds to a higher VC dimension, which in turn, induces an expansion of the generalization gap thereby increasing the risk of overfitting as stated in Vapnik (1991).

Regularization, as formally expounded in Hastie et al. (2009), is a fundamental technique in statistical learning that introduces a penalty term to the model's objective function, thereby discouraging excessive model complexity. One widely adopted regularization approach is L2 regularization, also known as ridge regularization, wherein the squared magnitude of the model coefficients (or weights) is penalized. Unlike L1 regularization, which promotes sparsity by driving certain coefficients exactly to zero, L2 regularization instead constrains the magnitude of all parameters, thereby ensuring a controlled reduction in model complexity without entirely eliminating any particular parameter. In the context of fitting a single-ouptut neural network, if we wanted to minimize the mean square error objective of $\sum_{i=1}^{N}(y_i - a_1(i)^L)^2$ [2], we would include the constraint $\sum_{j,k,l}(w_{kj}^l)^2 \leq \tau$. [3], where τ would act as a means to constrain the parameters of the model to be closer to zero, thus lessening model complexity. Furthermore, the model fitting procedure becomes a constrained optimization problem for which we can re-write the penalized objective as:

$$\sum_{i=1}^{N}(y_i - a_1(i)^L)^2 + \nu\sum_{j,k,l}(w_{kj}^l)^2 \tag{1}$$

whereby increasing our Lagrangian multiplier ν would be equivalent to decreasing τ, thereby enforcing a stricter penalty on the model parameters.

Given that regularization introduces an additional penalty term to the objective function, the resulting optimization problem remains differentiable with respect to the model parameters, thereby necessitating the use of gradient-based optimization techniques such as gradient descent. Specifically, gradient descent iteratively updates the model parameters in the direction of the negative gradient of the objective function, ensuring convergence to an optimal solution under appropriate step-size selection and convexity conditions as described in Amari (1993). Furthermore, the choice of regularization strength, governed by the parameter ν, critically influences the trade-off between model complexity and generalization. As such, determining an optimal value for ν requires an empirical approach, typically employing a validation set to assess model performance under varying degrees of regularization. This validation set approach, entails training the model with multiple candidate values of ν and subsequently evaluating its predictive performance on held-out validation data. The value of ν that minimizes the validation error is then selected, ensuring that the model is sufficiently complex to replicate nuances of the underlying pattern in the data, but also not too complex such that the model simply 'recalls' what has been seen in the data.

In the context of reinforcement learning (RL), an agent learns to perform tasks by interacting with an environment, making sequential decisions, and receiving feedback in the form of rewards as described in Painter and Brunskill (2018). The objective function, in this case, is to maximize the cumulative reward of the agent within the environment. Unlike supervised learning, gradients cannot be computed directly with respect to labeled data, as RL models are not trained on explicit input-output pairs. This lack of direct supervision renders gradient-based optimization techniques unsuitable for RL tasks. Consequently, we introduce the concept of an arbitrary objective - an objective that need not be differentiable or directly tied to the data. It only requires that the objective function assigns high values to desirable behaviour and low values to undesirable behaviour.

Given that tasks with arbitrary objectives render traditional gradient-based optimization infeasible, alternative optimization methods which allow for the solving of complex and arbitrary objectives, must be employed. Genetic algorithms (GAs) are a prominent class of such methods. They simulate the evolutionary process by initiating a population of random candidate

[1] E_{out} and E_{in} denote the out-of-sample and in-sample errors respectively, g denotes the final chosen hypothesis function, N denotes the number of samples and δ denotes the tolerance.

[2] Here, y_i denotes the i^{th} outcome of the N observations, where $a_1(i)^L$ represents the sole node of the i^{th} observation in the output layer L.

[3] Here, w_{kj}^l denotes the kj-th weight linking the k-th node in layer l - 1 and the j-th node in layer l.

solutions and employing selection mechanisms to iteratively recombine and mutate these candidates as stated in Holland (1992). This process yields new populations with potentially enhanced fitness (objective function values) relative to previous populations. Now given sufficient iterations, this process systematically favours traits (parameter values) that should enhance performance in achieving the objective function. As a result, we obtain an evolutionary algorithm in which the population of solutions gradually evolves toward an optimal solution.

While GAs provide a biologically inspired mechanism for navigating complex search spaces, it is important to contextualize their performance against simpler baseline strategies. In particular, Random Search (RS) offers a natural point of comparison due to its algorithmic simplicity and lack of heuristic bias. RS operates by sampling candidate solutions uniformly at random from the feasible domain, evaluating their objective values, and retaining the best-performing solution observed over a fixed number of trials. Despite its simplicity and lack of adaptive guidance, RS has demonstrated competitive performance in a range of settings - particularly when the objective landscape is noisy, discontinuous, or lacks exploitable structure as detailed in Bergstra and Bengio (2012). This makes RS an appropriate benchmark for assessing whether a GA is truly needed, or if its performance gains are only marginal relative to an unguided search.

From a Bayesian perspective, regularization can be interpreted as placing prior distributions on the parameter space of an overparameterized model. Specifically, L2 regularization corresponds to assuming a normal (Gaussian) prior on the parameters.

Proof. We assume our observed response $y_i|\mathbf{w}, \mathbf{a}(i)^0 \sim \mathcal{N}(a_1(i)^L, \sigma_y^2)$ and are independent from all other N observations. That is, we assume our observations follow a Gaussian distribution with mean equal to the sole node of the i^{th} observation in the output layer L, with some variance σ_y^2. Here, the vector $\mathbf{w}$ denotes the R number of weights and biases of a single-output neural network, with vector of inputs $\mathbf{a}(i)^0$ denoting the input nodes (on the 0^{th} layer) for the i^{th} observation.

Furthermore, we assume $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \sigma_w^2 \mathbf{I}_R)$. That is, the vector of R independent weights and biases is multivariate Gaussian distributed with mean of $\mathbf{0}$ with some covariance matrix $\sigma_w^2 \mathbf{I}_R$.

Using Bayes rule we have:

$$\begin{aligned} p(\mathbf{w} \mid \mathcal{D}) &= \frac{p(\mathcal{D} \mid \mathbf{w}) p(\mathbf{w})}{p(\mathcal{D})} \\ &\propto p(\mathcal{D} \mid \mathbf{w}) p(\mathbf{w}) \\ &\propto \left[\prod_{i=1}^{N} \mathcal{N}(y_i; \mathbf{a}(i)^1, \sigma_y^2) \right] \mathcal{N}(\mathbf{w}; \mathbf{0}, \sigma_w^2 \mathbf{I}) \\ &\propto \prod_{i=1}^{N} \mathcal{N}(y_i; \mathbf{a}(i)^1, \sigma_y^2) \prod_{i=1}^{R} \mathcal{N}(w_i; 0, \sigma_w^2) \end{aligned}$$

where given a dataset $\mathcal{D}$, $p(\mathcal{D} \mid \mathbf{w})$ is the likelihood - representing how well the parameters explain the observed data, $p(\mathbf{w})$ is the prior - encoding our beliefs about the parameters before seeing the data, and $p(\mathbf{w} \mid \mathcal{D})$ is the parameter posterior - the updated distribution of the parameters after observing the data.

Now taking the negative log probability of the parameter posterior:

$$\begin{aligned} -\log[p(\mathbf{w} \mid \mathcal{D})] &\propto -\sum_{i=1}^{N} \log \left[\mathcal{N}(y_i; \mathbf{a}(i)^1, \sigma_y^2) \right] - \sum_{i=1}^{R} \log \left[\mathcal{N}(w_i; 0, \sigma_w^2) \right] \\ &\propto \frac{1}{2\sigma_y^2} \sum_{i=1}^{N} \left(y_i - a_1(i)^L \right)^2 + \frac{1}{2\sigma_w^2} \sum_{i=1}^{R} w_i^2 \\ &\propto \sum_{i=1}^{N} \left(y_i - a_1(i)^L \right)^2 + \nu \sum_{i=1}^{R} w_i^2 \end{aligned}$$

whereby ν controls the strength of the regularization as in the L2 penalized loss function in Equation 1. □

Clearly, minimizing the L2 penalized loss function is equivalent to maximizing the posterior distribution of the parameters under a Gaussian prior, which corresponds to Maximum a Posteriori (MAP) estimation.

In the framework of Bayesian Neural Networks, the parameters of the neural network are treated as random variables rather than fixed values. This means they follow a probability distribution, reflecting our uncertainty about their true values. Given a dataset $\mathcal{D}$, our goal is to infer the posterior distribution over the parameters $\boldsymbol{\theta}$, which is given by Bayes' theorem: $p(\boldsymbol{\theta} \mid \mathcal{D}) \propto p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})$. Since the parameter posterior is typically intractable due to high-dimensional parameter spaces, that is, it either has a complex or unknown form as stated in Dobson and Barnett (2018), Markov Chain Monte Carlo (MCMC) methods are employed to approximate it. MCMC generates samples from the parameter posterior by constructing a Markov chain whose stationary distribution is the true posterior - in this way, the utilization of MCMC aims to provide the entire

posterior distribution $p(\boldsymbol{\theta} \mid \mathcal{D})$ and not just the MAP estimate; corresponding to $\text{argmax}_{\boldsymbol{\theta}}\, p(\boldsymbol{\theta} \mid \mathcal{D})$ which is the mode of the posterior. According to Dobson and Barnett (2018), Markov chains simplify complex problems since the next sample in the chain depends solely on the previous sample: $p(\boldsymbol{\theta}^{(j)} = \mathbf{a}|\boldsymbol{\theta}^{(j-1)}, \boldsymbol{\theta}^{(j-2)}, \ldots, \boldsymbol{\theta}^{(0)}) = p(\boldsymbol{\theta}^{(j)} = \mathbf{a}|\boldsymbol{\theta}^{(j-1)})$.

1 Introduction

One of the earliest approaches to applying MCMC to arbitrary objective functions was achieved through simulated annealing (SA). The method reformulates a cost function $C(x)$ into a Boltzmann-like distribution, $\pi_\beta(x) \propto \exp(-\beta C(x))$, where $\beta = 1/T$ denotes the inverse temperature. At high temperature (small β), the Markov chain explores the state space broadly, whereas at low temperature (large β) it increasingly concentrates around global minima of the cost function. The algorithm proceeds by running a standard MH sampler targeting π_β and gradually increasing β according to a cooling schedule. In the limit as $T \to 0$, the chain places its mass on the global minima of $C(x)$. This approach was popularized in physics and combinatorial optimization by Kirkpatrick et al. (1983) and rigorously studied in the context of Bayesian image analysis by Geman and Geman (1984), who also established theoretical convergence guarantees under logarithmic cooling.

This study extends the discussion by examining the limitations of applying MCMC to arbitrary objectives, with particular emphasis on two-block MCMC (alternating MH and Gibbs sampling). While such methods are often presented as advantageous for allowing the training set to infer regularization, we demonstrate that this claim is sensitive to the sharpness of the likelihood functions employed. Specifically, rather than simply exponentiating an arbitrary objective as in SA, we investigate alternative likelihood formulations deliberately shaped to remain proportional to the arbitrary objective function under maximization. This shifts MCMC from a sampling paradigm towards a mode-seeking algorithm, paralleling the perspective of SA. Our analysis highlights that the sharpness of the likelihood form plays a decisive role in both determining in-sample performance and the strength of regularization inferred by the training set.

To illustrate these points, we apply two-block MCMC to reinforcement learning tasks - specifically a navigation problem and the tic-tac-toe. Beforehand, we investigate the role of regularization on out-of-sample performance for solutions obtained from a genetic algorithm (GA), and additionally compare these results to solutions obtained by random search (RS). The study concludes by illustrating the implications of increased likelihood sharpness, demonstrating this through reinforcement learning tasks associated with blackjack. Here, we simplify the two-block MCMC by replacing the first block with an iterative optimisation procedure and compare this hybrid approach to the original scheme. The resulting near-identical performance demonstrates that increasing likelihood sharpness ultimately collapses posterior mass onto the dominant mode - echoing the observation of Kirkpatrick et al. (1983) that lower temperatures (that is, higher likelihood sharpness β) increasingly concentrate samples around the global minima of the cost function.

2 Metropolis-Hastings

We detail one of the fundamental MCMC algorithms: the Metropolis-Hastings (MH) algorithm as illustrated by Hastings (1970) and Metropolis et al. (1953). Given the current state of $\boldsymbol{\theta} \in \mathbb{R}^S$, that is $\boldsymbol{\theta}^{(j)}$, the MH algorithm proposes a new value $\boldsymbol{\theta}^*$ obtained from $\boldsymbol{\theta}^* = \boldsymbol{\theta}^{(j)} + \mathbf{Q}$. Subsequently, $\boldsymbol{\theta}^*$ is accepted as the new value in the Markov chain under the following acceptance criterion:

$$\boldsymbol{\theta}^{(j+1)} = \begin{cases} \boldsymbol{\theta}^*, & \text{if } U < \alpha \\ \boldsymbol{\theta}^{(j)}, & \text{otherwise.} \end{cases} \tag{2}$$

Now the vector $\mathbf{Q}$ denotes drawn values from a proposal density (usually $\mathbf{Q} \sim \mathcal{N}(\mathbf{0}, \sigma_Q^2 \mathbf{I}_S)$), and U is a drawn value from a uniform distribution between 0 and 1, that is $U \sim \mathcal{U}(0, 1)$. Furthermore, α is the acceptance probability given by:

$$\begin{aligned} \alpha &= \min\left(\frac{p(\boldsymbol{\theta}^* \mid \mathcal{D})}{p(\boldsymbol{\theta}^{(j)} \mid \mathcal{D})} \cdot \frac{Q(\boldsymbol{\theta}^{(j)} \mid \boldsymbol{\theta}^*)}{Q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{(j)})}, 1\right) \\ &= \min\left(\frac{p(\mathcal{D} \mid \boldsymbol{\theta}^*)p(\boldsymbol{\theta}^*)}{p(\mathcal{D} \mid \boldsymbol{\theta}^{(j)})p(\boldsymbol{\theta}^{(j)})} \cdot \frac{Q(\boldsymbol{\theta}^{(j)} \mid \boldsymbol{\theta}^*)}{Q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{(j)})}, 1\right). \end{aligned}$$

If the proposal density is symmetric, then α simplifies to

$$\alpha = \min\left(\frac{p(\mathcal{D} \mid \boldsymbol{\theta}^*)p(\boldsymbol{\theta}^*)}{p(\mathcal{D} \mid \boldsymbol{\theta}^{(j)})p(\boldsymbol{\theta}^{(j)})}, 1\right), \tag{3}$$

where, for our study, we assume Gaussian priors for our parameters $\boldsymbol{\theta} \in \mathbb{R}^S$, that is, $\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \sigma_\theta^2 \mathbf{I}_S)$ (as elaborated in Section 3). Here, the likelihood ratio $\frac{p(\mathcal{D}|\boldsymbol{\theta}^*)}{p(\mathcal{D}|\boldsymbol{\theta}^{(j)})}$ reflects how much more likely the proposed parameter $\boldsymbol{\theta}^*$ is, compared to the current parameter $\boldsymbol{\theta}^{(j)}$, in explaining the observed data $\mathcal{D}$. Since the likelihood function quantifies the plausibility of the data given $\boldsymbol{\theta}$, the MH algorithm accepts proposed moves with higher likelihoods more readily, while still allowing occasional transitions to

lower-likelihood regions to ensure proper exploration of the posterior $p(\boldsymbol{\theta} \mid \mathcal{D})$. Additionally, the prior ratio $\frac{p(\boldsymbol{\theta}^*)}{p(\boldsymbol{\theta}^{(j)})}$ contributes to exploration by favouring moves toward regions of the parameter space that are more consistent with prior beliefs, especially when the likelihood offers little guidance such as being relatively flat. Furthermore, one may view the the prior ratio as a means to ensure acceptance is not solely driven by proposals which increase the likelihood. In terms of terminology used later, we refer to a likelihood-driven sampler as one in which the likelihood ratio dominates the prior ratio in the acceptance probability expression of Equation 3.

2.1 σ_θ^2: A parameter with a hyperprior

We now assume the variance of the prior of $\boldsymbol{\theta}$, σ_θ^2, to not be fixed, but rather having its own distribution. Instead of sampling the entire parameter vector $\boldsymbol{\Lambda} = \left[\boldsymbol{\theta}', \sigma_\theta^2\right]' \in \mathbb{R}^{S+1}$ in a joint MH framework (as illustrated in Appendix A), we split the parameters into groups (or blocks) and sample each block conditionally on the others, leveraging their conditional distributions. Hence, under the two-sample MCMC framework, the algorithm effectively samples from the joint posterior $p(\boldsymbol{\theta}, \sigma_\theta^2 \mid \mathcal{D})$ by alternately drawing from the conditionals $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D})$ and $p(\sigma_\theta^2 \mid \mathcal{D}, \boldsymbol{\theta})$. To reiterate, we know that since $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D}) = \frac{p(\boldsymbol{\theta}, \sigma_\theta^2 \mid \mathcal{D})}{p(\sigma_\theta^2 \mid \mathcal{D})}$, when sampling $\boldsymbol{\theta}$ from $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D})$, σ_θ^2 is treated as fixed. This is because it is conditioned on the value sampled in the previous step of the sampler, which results in $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D}) \propto p(\boldsymbol{\theta}, \sigma_\theta^2 \mid \mathcal{D})$. The same logic may be used to conclude $p(\sigma_\theta^2 \mid \boldsymbol{\theta}, \mathcal{D}) \propto p(\boldsymbol{\theta}, \sigma_\theta^2 \mid \mathcal{D})$ - since when sampling from $p(\sigma_\theta^2 \mid \boldsymbol{\theta}, \mathcal{D})$, $\boldsymbol{\theta}$ is treated as fixed. The two-sample MCMC framework can simplify the sampling process, particularly when the joint proposal distribution is complex or high-dimensional. Additionally, it can improve mixing in cases where there is strong posterior correlation between $\boldsymbol{\theta}$ and σ_θ^2, which can hinder the efficiency of joint updates. In this context, adequate mixing refers to the sampler's ability to explore the parameter space effectively - typically indicated by reduced autocorrelation between successive samples (for example, between $\boldsymbol{\theta}^{(j)}$ and $\boldsymbol{\theta}^{(j+1)}$ for all j), thereby promoting more reliable convergence to the stationary distribution.

2.1.0.1 Block 1: Sampling $\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D}$ We assume the prior $\boldsymbol{\theta} \mid \sigma_\theta^2 \sim \mathcal{N}(\mathbf{0}, \sigma_\theta^2 \mathbf{I}_S)$ and a symmetric proposal density $\mathbf{Q}_\theta \sim \mathcal{N}(\boldsymbol{\theta}^{(j)}, \sigma_{Q_\theta}^2 \mathbf{I}_S)$. Hence, the acceptance probability for the first block is:

$$
\begin{aligned}
\alpha_\theta &= \min\left(\frac{p\left(\boldsymbol{\theta}^* \mid \left(\sigma_\theta^2\right)^{(j)}, \mathcal{D}\right)}{p\left(\boldsymbol{\theta}^{(j)} \mid \left(\sigma_\theta^2\right)^{(j)}, \mathcal{D}\right)} \cdot \frac{Q(\boldsymbol{\theta}^{(j)} \mid \boldsymbol{\theta}^*)}{Q\left(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{(j)}\right)}, 1\right) \\
&= \min\left(\frac{p\left(\mathcal{D} \mid \boldsymbol{\theta}^*, \left(\sigma_\theta^2\right)^{(j)}\right) p\left(\boldsymbol{\theta}^* \mid \left(\sigma_\theta^2\right)^{(j)}\right)}{p(\mathcal{D} \mid \boldsymbol{\theta}^{(j)}, (\sigma_\theta^2)^{(j)}) p\left(\boldsymbol{\theta}^{(j)} \mid \left(\sigma_\theta^2\right)^{(j)}\right)} \cdot \frac{Q\left(\boldsymbol{\theta}^{(j)} \mid \boldsymbol{\theta}^*\right)}{Q\left(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}^{(j)}\right)}, 1\right) && \text{Bayes Theorem} \\
&= \min\left(\frac{p\left(\mathcal{D} \mid \boldsymbol{\theta}^*\right) p\left(\boldsymbol{\theta}^* \mid \left(\sigma_\theta^2\right)^{(j)}\right)}{p\left(\mathcal{D} \mid \boldsymbol{\theta}^{(j)}\right) p\left(\boldsymbol{\theta}^{(j)} \mid \left(\sigma_\theta^2\right)^{(j)}\right)}, 1\right) && \text{Likelihood not dependent on } \sigma_\theta^2 \\
&= \min\left(\frac{p\left(\mathcal{D} \mid \boldsymbol{\theta}^*\right) \cdot \frac{1}{\sqrt{\left(2\pi\left(\sigma_\theta^2\right)^{(j)}\right)^S}} \exp\left(-\frac{1}{2\left(\sigma_\theta^2\right)^{(j)}} \|\boldsymbol{\theta}^*\|^2\right)}{p\left(\mathcal{D} \mid \boldsymbol{\theta}^{(j)}\right) \cdot \frac{1}{\sqrt{\left(2\pi\left(\sigma_\theta^2\right)^{(j)}\right)^S}} \exp\left(-\frac{1}{2\left(\sigma_\theta^2\right)^{(j)}} \|\boldsymbol{\theta}^{(j)}\|^2\right)}, 1\right).
\end{aligned}
$$

Hence, taking the log, we obtain:

$$
\log\left(\alpha_\theta\right) = \min\left(\log\left(p\left(\mathcal{D} \mid \boldsymbol{\theta}^*\right)\right) - \frac{1}{2\left(\sigma_\theta^2\right)^{(j)}} \|\boldsymbol{\theta}^*\|^2 - \log\left(p\left(\mathcal{D} \mid \boldsymbol{\theta}^{(j)}\right)\right) + \frac{1}{2\left(\sigma_\theta^2\right)^{(j)}} \|\boldsymbol{\theta}^{(j)}\|^2, 0\right), \tag{4}
$$

after which, using Equation 2 to ascertain $\boldsymbol{\theta}^{(j+1)}$.

2.1.0.2 Block 2: Sampling $\sigma_\theta^2 \mid \boldsymbol{\theta}, \mathcal{D}$ We assume the hyperprior $\sigma_\theta^2 \sim \text{Inv-Gamma}(a, b)$. Since σ_θ^2 does not appear in the likelihood, the data $\mathcal{D}$ provides no additional information about σ_θ^2 beyond what $\boldsymbol{\theta}$ already does. Hence, conditioning on $\mathcal{D}$ does not change the distribution of $\sigma_\theta^2 \mid \boldsymbol{\theta}$:

$$
\begin{aligned}
p(\sigma_\theta^2 \mid \boldsymbol{\theta}, \mathcal{D}) &\propto p(\mathcal{D} \mid \boldsymbol{\theta}, \sigma_\theta^2) \cdot p(\sigma_\theta^2 \mid \boldsymbol{\theta}) \\
&= p(\mathcal{D} \mid \boldsymbol{\theta}) \cdot p(\boldsymbol{\theta} \mid \sigma_\theta^2) \cdot p(\sigma_\theta^2) && \text{Likelihood independent of } \sigma_\theta^2 \\
&\propto p(\boldsymbol{\theta} \mid \sigma_\theta^2) \cdot p(\sigma_\theta^2) && \text{Likelihood constant for fixed } \boldsymbol{\theta} \\
&\propto \frac{1}{\sqrt{(2\pi\sigma_\theta^2)^S}} \exp\left(-\frac{\|\boldsymbol{\theta}\|^2}{2\sigma_\theta^2}\right) \cdot \frac{b^a}{\Gamma(a)} (\sigma_\theta^2)^{-(a+1)} \exp\left(-\frac{b}{\sigma_\theta^2}\right) \\
&\propto (\sigma_\theta^2)^{-(a+\frac{S}{2}+1)} \exp\left(-\frac{b + \frac{\|\boldsymbol{\theta}\|^2}{2}}{\sigma_\theta^2}\right).
\end{aligned}
$$

Hence $\left(\sigma_\theta^2\right)^{(j+1)} \mid \boldsymbol{\theta}^{(j+1)}, \mathcal{D} \sim \text{Inv-Gamma}\left(a + \frac{S}{2}, b + \frac{\|\boldsymbol{\theta}^{(j+1)}\|^2}{2}\right)$, a distribution from which we can sample directly. Therefore, the MH algorithm is not required for this block. We choose $a, b \approx 0$ in order to specify a nearly uninformative hyperprior of σ_θ^2 - meaning that the prior exerts minimal influence on the posterior $p(\sigma_\theta^2 \mid \boldsymbol{\theta})$, allowing the observed data to primarily determine the inference. This approach avoids imposing strong assumptions on σ_θ^2 and reflects prior ignorance about its scale.

To compute the MAP estimate $\hat{\boldsymbol{\theta}}^{MAP} = \text{argmax}_{\boldsymbol{\theta}}\, p(\boldsymbol{\theta} \mid \mathcal{D})$, we merely identify the mode of the posterior distribution $p(\boldsymbol{\theta} \mid \mathcal{D})$. Likewise, $\hat{\theta}_i^{MAP} = \text{argmax}_{\theta_i}\, p(\theta_i \mid \mathcal{D})$, which is merely the mode of the marginal posterior $p(\theta_i \mid \mathcal{D})$, which implicitly integrates out both the remaining components $\boldsymbol{\theta}_{-i}$ and the dispersion parameter σ_θ^2. In this context, each component, θ_i for $i = 1, \ldots, S$, has a marginal posterior distribution given by $p(\theta_i \mid \mathcal{D}) = \int p(\boldsymbol{\theta}, \sigma_\theta^2 \mid \mathcal{D}) d\boldsymbol{\theta}_{-i} d\sigma_\theta^2$.

In hierarchical Bayesian modeling, assigning a prior distribution to σ_θ^2 allows the training data to inform the appropriate degree of regularization. Section 3 elucidates $\sigma_\theta^2 \propto \frac{1}{\nu}$ therefore smaller σ_θ^2 values imply stronger shrinkage toward zero, while larger values allow greater flexibility. By endowing σ_θ^2 with a hyperprior, we integrate over uncertainty in the regularization strength rather than fixing it arbitrarily. Consequently, the resulting MAP estimates of the parameters inherently reflect an optimal degree of regularization - effectively "baking in" regularization informed by the training data.

Thus, in a hierarchical Bayesian framework, the training data plays a dual role: it informs the marginal posterior distributions of the model parameters $\boldsymbol{\theta}$, $p(\theta_i \mid \mathcal{D})$ for $i = 1, \ldots, S$, while simultaneously guiding the level of regularization through inference on the dispersion parameter σ_θ^2.

2.2 Adaptive Metropolis-Hastings

In standard MH algorithms, the proposal distribution remains fixed throughout sampling. In contrast, adaptive MCMC methods dynamically tune aspects of the proposal distribution using information gathered during earlier iterations. This adaptive strategy aims to improve mixing by better matching the geometry of the stationary distribution as expounded in Roberts et al. (1997).

We previously stated that the proposal distribution for $\boldsymbol{\theta} \in \mathbb{R}^S$ at current iteration j is $\mathbf{Q}_\theta \sim \mathcal{N}(\boldsymbol{\theta}^{(j)}, \boldsymbol{\Sigma_j})$ where $\boldsymbol{\Sigma_j} = \sigma_{Q_\theta}^2 \mathbf{I}_S$: a multivariate normal random walk centered at the current state of $\boldsymbol{\theta}$, $\boldsymbol{\theta}^{(j)}$, with isotropic covariance matrix $\sigma_{Q_\theta}^2 \mathbf{I}_S$. During the burn-in phase only, we adapt $\boldsymbol{\Sigma_j}$ based on previously accepted proposals of $\boldsymbol{\theta}$. By limiting adaptation to the burn-in period, we avoid violating the diminishing adaptation and ergodicity conditions required for convergence to the correct stationary distribution in fully adaptive MCMC algorithms, as explained by Roberts and Rosenthal (2007). Additionally, we may still utilise Equation 3 to compute acceptance probabilities, α_θ, as our covariance matrices used will all be equivalent after the burn-in phase, that is $\boldsymbol{\Sigma_j} = \boldsymbol{\Sigma_{j+1}}$ for $j >$ burn-in, giving rise to symmetric proposal densities.

Per Haario et al. (2001), the covariance matrix at iteration j is given as:

$$\boldsymbol{\Sigma_j} = \text{Cov}\left(f\left(\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(j)}\right)\right) + \epsilon \mathbf{I},$$

where $\boldsymbol{\Sigma_j}$ is the empirical covariance matrix of a function of the first j samples and $\epsilon \mathbf{I}_S$ is a small positive-definite matrix (for example, $\epsilon = 10^{-6}$) added to maintain numerical stability. The proposal then becomes $\mathbf{Q}_\theta \sim \mathcal{N}(\boldsymbol{\theta}^{(j)}, s^2 \boldsymbol{\Sigma_j})$, where $s^2 = \frac{2.38}{S}$. Roberts et al. (1997) showed that for efficient exploration of high-dimensional target distributions, the acceptance rate should be around 0.234, and the optimal step size (variance) scales as $s^2 = \frac{2.38}{S}$.

To reduce the correlation between the proposals, we employ:

$$f\left(\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(j)}\right) = \begin{cases} \left(\boldsymbol{\theta}^{(j-\delta\Delta)}, \boldsymbol{\theta}^{(j-\delta\Delta+\delta)}, \ldots, \boldsymbol{\theta}^{(j-\delta\Delta+(\Delta-1)\delta)}, \boldsymbol{\theta}^{(j)}\right), & \text{if } j > \delta\Delta \\ \left(\boldsymbol{\theta}^{(j-\lfloor \frac{j}{\delta} \rfloor \delta)}, \boldsymbol{\theta}^{(j-\lfloor \frac{j}{\delta} \rfloor \delta+\delta)}, \ldots, \boldsymbol{\theta}^{(j-\lfloor \frac{j}{\delta} \rfloor \delta+(\lfloor \frac{j}{\delta} \rfloor -1)\delta)}, \boldsymbol{\theta}^{(j)}\right), & \text{if } \delta < j \leq \delta\Delta \\ \left(\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(j)}\right), & \text{if } j \leq \delta, \end{cases} \tag{5}$$

with $\boldsymbol{\theta}^{(1)}$ randomly drawn from the multivariate normal distribution $\mathcal{N}\left(\mathbf{0}_{S\times 1}, \sigma_{\text{Init}}^2 \mathbf{I}_S\right)$. Here, δ serves as a stride parameter that ensures non-consecutive $\boldsymbol{\theta}$ values are used in the empirical covariance calculation, thereby mitigating autocorrelation between proposals. The window size parameter Δ is purely curated to ease computation of the empirical covariances at each iteration (its inclusion is optional in which case one would omit the first case of Equation 5).

2.3 Adaptive scaling

Previously, it was noted that according to the optimal scaling theory of Roberts et al. (1997), the optimal proposal scale for high-dimensional target distributions is given by $s^2 = \frac{2.38}{S}$. However, in order to directly control the average acceptance rate α toward its theoretical optimum of approximately 0.234, we adopt an alternative strategy for scaling the covariance matrix $\boldsymbol{\Sigma_j}$ during sampling. Specifically, during the burn-in phase, the scaling factor s^2 is adaptively updated at each iteration j according to the observed acceptance rate α_j:

$$(s^2)^{(j+1)} = (s^2)^{(j)} \times \exp\left(\gamma_j \cdot (\alpha_j - 0.234)\right),$$

where γ_j is a sequence of diminishing adaptation rates. In accordance with the theoretical guarantees for ergodicity of adaptive MCMC methods established by Roberts and Rosenthal (2007), we choose $\gamma_j = \frac{1}{j^\kappa}$ for some $0.5 < \kappa < 1$. This choice satisfies the standard conditions $\sum_j \gamma_j = \infty$ and $\sum_j \gamma_j^2 < \infty$ as per Roberts and Rosenthal (2007) ensuring that the adaptation is both diminishing and stable. While these conditions are necessary for ensuring ergodicity in fully adaptive MCMC, in our implementation, adaptation is restricted to the burn-in phase. This pragmatic restriction alleviates these theoretical concerns.

3 The Objective & Regularization

This section demonstrates the necessity for the likelihood function, $p(\mathcal{D} \mid \boldsymbol{\theta})$, to be proportional to (or a monotonically increasing function of) the objective function being maximized. Consider the arbitrary objective, for a given parameter configuration $\boldsymbol{\theta} \in \mathbb{R}^S$, denoted as $\text{argmax}_{\boldsymbol{\theta}} \text{Obj}(\boldsymbol{\theta})$. By including L2 regularization, the L2 penalized objective becomes:

$$\underset{\boldsymbol{\theta}}{\text{argmax}} \left(\text{Obj}(\boldsymbol{\theta}) - \nu \sum_{i=1}^{S} \theta_i^2 \right). \tag{6}$$

Under the assumption that the likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$ is monotonic increasing with respect to $\text{Obj}(\boldsymbol{\theta})$, and assuming Gaussian priors for our parameters $\boldsymbol{\theta} \in \mathbb{R}^S$, that is $\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \sigma_\theta^2 \mathbf{I}_S)$, using Bayes rule we have:

$$\begin{aligned} p(\boldsymbol{\theta} \mid \mathcal{D}) &= \frac{p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{D})} \\ &\propto p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) \\ &\propto p(\mathcal{D} \mid \boldsymbol{\theta}) \mathcal{N}(\boldsymbol{\theta}; \mathbf{0}, \sigma_\theta^2 \mathbf{I}) \\ &\propto p(\mathcal{D} \mid \boldsymbol{\theta}) \prod_{i=1}^{S} \mathcal{N}(\theta_i; 0, \sigma_\theta^2). \end{aligned}$$

Now taking the log probability of the parameter posterior:

$$\begin{aligned} \log[p(\boldsymbol{\theta} \mid \mathcal{D})] &\propto \log\left[p(\mathcal{D} \mid \boldsymbol{\theta})\right] + \sum_{i=1}^{S} \log\left[\mathcal{N}(\theta_i; 0, \sigma_\theta^2)\right] \\ &\propto \log\left[p(\mathcal{D} \mid \boldsymbol{\theta})\right] - \frac{1}{2\sigma_\theta^2} \sum_{i=1}^{S} \theta_i^2. \end{aligned}$$

Now since we assumed the likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$ to be proportional to $\text{Obj}(\boldsymbol{\theta})$, that is, $p(\mathcal{D} \mid \boldsymbol{\theta}) \propto \text{Obj}(\boldsymbol{\theta})$, we have:

$$\log[p(\boldsymbol{\theta} \mid \mathcal{D})] \propto \text{Obj}(\boldsymbol{\theta}) - \nu \sum_{i=1}^{S} \theta_i^2,$$

whereby $\nu \propto \frac{1}{\sigma_\theta^2}$ controls the strength of the regularization as in the L2 penalized objective in Equation 6. Clearly, $\text{argmax}_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \mathcal{D})$, is equivalent to the L2 penalized objective in Equation 6, that is, $\hat{\boldsymbol{\theta}}^{MAP} = \text{argmax}_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \mathcal{D}) = \text{argmax}_{\boldsymbol{\theta}} \left(\text{Obj}(\boldsymbol{\theta}) - \frac{1}{2\sigma_\theta^2} \sum_{i=1}^{S} \theta_i^2 \right)$. We may say that the L2 penalization constraint depends on fixed σ_θ^2, the number of coefficients, S, and their magnitudes θ_i, written as function $f(\sigma_\theta^2, S, \theta_i)$.

However, within the context of the two-sampler MCMC scheme, since σ_θ^2 is sampled at each iteration, it is no longer the case that the MAP estimate $\hat{\boldsymbol{\theta}}^{MAP}$ is equivalent to $\text{argmax}_{\boldsymbol{\theta}} \left(\text{Obj}(\boldsymbol{\theta}) - \nu \sum_{i=1}^{S} \theta_i^2 \right)$ with $\nu \propto \frac{1}{\sigma_\theta^2}$, as this equivalence only holds when σ_θ^2 is fixed across all iterations. Rather, at each iteration j, we induce a shrinkage factor $\nu^{(j)} \propto \frac{1}{(\sigma_\theta^2)^{(j)}}$ on the proposal $\boldsymbol{\theta}^*$, as shown in Equation 4. Nevertheless, we may still characterize the amount of regularization associated with the MAP estimates by examining the marginal posterior $p(\sigma_\theta^2 \mid \mathcal{D})$. This distribution reflects the posterior uncertainty about the degree of shrinkage, that is, how much regularization the data supports. For example, we may utilise the mean of $p(\sigma_\theta^2 \mid \mathcal{D})$ to reflect an 'effective ridge penalty' or have the posterior intervals to show uncertainty in the amount of regularization inferred.

This two-sampler MCMC framework also results in our marginal prior for $\boldsymbol{\theta}$ to not be Gaussian as before. Since we assume priors: $\boldsymbol{\theta} \mid \sigma_\theta^2 \sim \mathcal{N}(\mathbf{0}, \sigma_\theta^2 \mathbf{I}_S)$ and $\sigma_\theta^2 \sim \text{Inv-Gamma}(a, b)$, we have:

$$\begin{aligned} p(\boldsymbol{\theta}) &= \int_0^\infty p(\boldsymbol{\theta}, \sigma_\theta^2)\, d\sigma_\theta^2 \\ &= \int_0^\infty p(\boldsymbol{\theta} \mid \sigma_\theta^2) p(\sigma_\theta^2)\, d\sigma_\theta^2 \\ &= \int_0^\infty (2\pi\sigma_\theta^2)^{(-\frac{S}{2})} \exp\left(-\frac{1}{2\sigma_\theta^2} \|\boldsymbol{\theta}\|^2 \right) \frac{b^a}{\Gamma(a)} (\sigma_\theta^2)^{-(a+1)} \exp\left(-\frac{b}{\sigma_\theta^2} \right) d\sigma_\theta^2 \\ &= \frac{b^a}{\Gamma(a)} (2\pi)^{-(\frac{S}{2})} \int_0^\infty (\sigma_\theta^2)^{-(a+\frac{S}{2}+1)} \exp\left(-\frac{1}{2\sigma_\theta^2} \left(\|\boldsymbol{\theta}\|^2 + 2b \right) \right) d\sigma_\theta^2. \end{aligned}$$

Using $\int_0^\infty x^{-(\alpha+1)} \exp(-\frac{\beta}{x})\, dx = \beta^{-\alpha}\Gamma(\alpha)$, we have the following:

$$\begin{aligned} p(\boldsymbol{\theta}) &= \frac{\Gamma(a+\frac{S}{2})}{\Gamma(a)}(2\pi)^{-(\frac{S}{2})} b^a \left(b + \frac{1}{2}\|\boldsymbol{\theta}\|^2\right)^{-(a+\frac{S}{2})} \\ &= \frac{\Gamma\left(\frac{2a+S}{2}\right)}{\Gamma\left(\frac{2a}{2}\right)(2a)^{\frac{S}{2}}\pi^{\frac{S}{2}}\left(\frac{b}{a}\right)^{\frac{S}{2}}} \left(1 + \frac{1}{2a}\boldsymbol{\theta}'\left(\frac{b}{a}\mathbf{I}_S\right)^{-1}\boldsymbol{\theta}\right)^{-\left(\frac{2a+S}{2}\right)}. \end{aligned}$$

Hence, $\boldsymbol{\theta} \sim t_S\left(\mathbf{0}, \frac{b}{a}\mathbf{I}_S, 2a\right)$ - that is, the marginal prior on $\boldsymbol{\theta}$ is multivariate Student-t with $2a$ degrees of freedom, location vector $\mathbf{0}$, and scale matrix $\frac{b}{a}\mathbf{I}_S$, rather than multivariate Gaussian. Accordingly:

$$\log[p(\boldsymbol{\theta} \mid \mathcal{D})] \propto \text{Obj}(\boldsymbol{\theta}) - \left(a + \frac{S}{2}\right)\log\left(b + \frac{1}{2}\sum_{i=1}^{S}\theta_i^2\right).$$

Hence $\hat{\boldsymbol{\theta}}^{MAP} = \text{argmax}_{\boldsymbol{\theta}}\left(\text{Obj}(\boldsymbol{\theta}) - \left(a+\frac{S}{2}\right)\log\left(b + \frac{1}{2}\sum_{i=1}^{S}\theta_i^2\right)\right)$. Constraining $\text{Obj}(\boldsymbol{\theta})$ with a penalty of the form $\log\left(f(\sum_{i=1}^{S}\theta_i^2)\right)$, rather than the quadratic penalty $\sum_{i=1}^{S}\theta_i^2$ that arises under a Gaussian marginal prior, highlights the difference between the two. Both behave similarly for small coefficients θ_i, but the Student-t prior imposes much weaker shrinkage in the tails, applying almost no penalization to large coefficients relative to the Gaussian case. Previously, under a fixed Gaussian prior, the penalization constraint for L2 regularization was a function $f(\sigma_\theta^2, S, \theta_i)$. In the two-block MCMC framework, since σ_θ^2 itself is treated hierarchically via the hyperparameters (a, b), the penalization constraint is now a function $f(a, b, S, \theta_i)$.

4 The Likelihood

We emphasize that, because our objectives are arbitrary, no concrete assumptions about the data-generating process can be made. In other words, there is no well-defined probabilistic model for the data. Conventionally, the likelihood is derived from the joint density of the data samples $\{x_i\}_{i=1}^n$ given the parameters $\boldsymbol{\theta}$: $p(\mathcal{D} \mid \boldsymbol{\theta}) = f(x_1, \ldots, x_n \mid \boldsymbol{\theta}) = \prod_{i=1}^n f(x_i \mid \boldsymbol{\theta})$ (for x_i i.i.d.). In contrast, in our setting, we only have an objective $\text{Obj}(\boldsymbol{\theta})$ that we aim to maximize. Hence, the study utilises likelihoods - more accurately described as pseudo-likelihoods, although we use the terms interchangeably throughout the study - which are purposely fabricated to be proportional to $\text{Obj}(\boldsymbol{\theta})$, as prescribed in Section 3, to ensure $\hat{\boldsymbol{\theta}}^{MAP} = \text{argmax}_{\boldsymbol{\theta}}\left(\text{Obj}(\boldsymbol{\theta}) - f\left(\sum_{i=1}^{S}\theta_i^2\right)\right)$. Additionally, the likelihoods are tempered - that is, constructed to allow control over their sharpness with respect to $\text{Obj}(\boldsymbol{\theta})$. This tempering increases the sensitivity of the likelihood to changes in the objective function values, since $p(\mathcal{D} \mid \boldsymbol{\theta}) \propto \text{Obj}(\boldsymbol{\theta})$. This sharpness may also be interpreted as a greater concentration of mass around the modal regions of their densities. Consequently, it translates into a more pronounced influence of the likelihood on the conditional posterior distribution $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D})$, since $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D}) \propto p(\mathcal{D} \mid \boldsymbol{\theta}) \cdot p(\boldsymbol{\theta} \mid \sigma_\theta^2)$. Within the context of the two-sample MCMC framework described in Section 2.1, increased likelihood sharpness implies that the sampler becomes more likelihood-driven, thereby diminishing the influence of the prior ratio in exploring the conditional posterior.

In this light, the pseudo-likelihoods employed in the study are not intended to represent rigorous data-generating models. Instead, they are used within the MCMC framework, not to sample from a full posterior, but to facilitate optimisation over the parameter space. The MCMC algorithm is thus repurposed as a mode-seeking procedure, targeting the mode of the conditional posterior $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D})$ - that is to say, to concentrate samples around a dominant mode of $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D})$. It suffices that the proposal mechanism is guided by likelihood functions that monotonically increase with the objective of interest, biasing the sampling process toward regions of high posterior density - these regions necessarily correspond to high-likelihood (thus high-valued objective) areas since $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D}) \propto p(\mathcal{D} \mid \boldsymbol{\theta}) \cdot p(\boldsymbol{\theta} \mid \sigma_\theta^2)$. As such, the requirement for a fully specified likelihood linking the data to the model becomes less critical.

We emphasise now, for the purposes of convergence in MCMC, it is often more desirable for the sampler to be likelihood-driven rather than prior-driven. Equation 4 further illustrates this concept. In the limiting case of a negligible likelihood, it simplifies to:

$$\log(\alpha_\theta) = \min\left(\frac{1}{2\left(\sigma_\theta^2\right)^{(j)}}\left(\|\boldsymbol{\theta}^{(j)}\|^2 - \|\boldsymbol{\theta}^*\|^2\right), 0\right),$$

indicating that when the likelihood is approximately flat, that is, $\log\left(p\left(\mathcal{D} \mid \boldsymbol{\theta}^*\right)\right) \approx \log\left(p\left(\mathcal{D} \mid \boldsymbol{\theta}^{(j)}\right)\right)$, proposals that reduce $\|\boldsymbol{\theta}^*\|^2$ are more likely to be accepted, leading to an indefinite contraction of $\boldsymbol{\theta}$ towards zero. Although a balanced contribution between prior and likelihood is theoretically desirable, the practical impossibility of pre-specifying this balance motivates us to ensure that the likelihood is sufficiently dominant in shaping the conditional posterior $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D})$.

4.1 Likelihood dominance & implications

In the context of the two-sample MCMC scheme as in Section 2.1, where sampling is performed from the conditional posterior $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D}) \propto p(\mathcal{D} \mid \boldsymbol{\theta}) \cdot p(\boldsymbol{\theta} \mid \sigma_\theta^2)$, it is important to note that the mode of conditional $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D})$ need not coincide exactly

with the mode of the marginal $p(\boldsymbol{\theta} \mid \mathcal{D})$. This is because the former is directly influenced by the specific value of σ_θ^2: that is, σ_θ^2 directly determines the shape of the prior $p(\boldsymbol{\theta} \mid \sigma_\theta^2)$. We note that:

$$\begin{aligned} p(\boldsymbol{\theta} \mid \mathcal{D}) &= \int p(\boldsymbol{\theta}, \sigma_\theta^2 \mid \mathcal{D})\, d\sigma_\theta^2 \\ &= \int p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D}) \cdot p(\sigma_\theta^2 \mid \mathcal{D})\, d\sigma_\theta^2. \end{aligned} \tag{7}$$

This expression implies that the marginal posterior $p(\boldsymbol{\theta} \mid \mathcal{D})$ is a weighted average of the conditionals $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D})$, where the weights come from the marginal $p(\sigma_\theta^2 \mid \mathcal{D})$. In other words, to obtain $p(\boldsymbol{\theta} \mid \mathcal{D})$, one averages over the uncertainty in σ_θ^2.

There are two main scenarios where the mode of the marginal posterior $p(\boldsymbol{\theta} \mid \mathcal{D})$ will approximately coincide with the mode of the conditional posterior $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D})$. First, if the marginal $p(\sigma_\theta^2 \mid \mathcal{D})$ has low variance - that is, it is sharply peaked around a single value - then the integral in Equation 7 is dominated by a narrow range of σ_θ^2, effectively treating σ_θ^2 as nearly constant. Second, if the likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$ is highly informative (that is, dominates the conditional $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D})$), then it largely determines the shape of the conditional, making it sharply peaked in roughly the same region of $\boldsymbol{\theta}$ regardless of the specific value of σ_θ^2. In this case, all the conditional posteriors in the integral of Equation 7 are peaked in the same region, and so the resulting marginal $p(\boldsymbol{\theta} \mid \mathcal{D})$ will also be peaked there.

Furthermore, given the assumption that the likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$ is sufficiently dominant, we have $\operatorname{argmax}_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D}) \approx \hat{\boldsymbol{\theta}}^{MAP} = \operatorname{argmax}_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \mathcal{D})$. Additionally, it may be argued that through likelihood dominance, since $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D}) \propto p(\mathcal{D} \mid \boldsymbol{\theta}) \cdot p(\boldsymbol{\theta} \mid \sigma_\theta^2)$, we may have $\operatorname{argmax}_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D}) \approx \operatorname{argmax}_{\boldsymbol{\theta}} p(\mathcal{D} \mid \boldsymbol{\theta}) = \hat{\boldsymbol{\theta}}^{MLE}$. Furthermore, due to proportionality $p(\mathcal{D} \mid \boldsymbol{\theta}) \propto \text{Obj}(\boldsymbol{\theta})$, $\hat{\boldsymbol{\theta}}^{MLE} = \operatorname{argmax}_{\boldsymbol{\theta}} \text{Obj}(\boldsymbol{\theta})$. Hence, altogether, we may argue that, given sufficient likelihood dominance, $\operatorname{argmax}_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D}) \approx \hat{\boldsymbol{\theta}}^{MAP} = \operatorname{argmax}_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \mathcal{D}) \approx \hat{\boldsymbol{\theta}}^{MLE} = \operatorname{argmax}_{\boldsymbol{\theta}} p(\mathcal{D} \mid \boldsymbol{\theta}) = \operatorname{argmax}_{\boldsymbol{\theta}} \text{Obj}(\boldsymbol{\theta})$. That is, the mode of the conditional $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D})$ will necessarily lie in a region where $\text{Obj}(\boldsymbol{\theta})$ attains high values. This imples that, when the prior is severely undermined, two-block MCMC merely serves as an optimisation technique to seek out $\operatorname{argmax}_{\boldsymbol{\theta}} \text{Obj}(\boldsymbol{\theta})$ - where negligible regularization is inferred to the MAP estimates.

5 The Navigation Problem

Consider T drones navigating within a two-dimensional circular arena. The arena is defined as the annular region bounded by an inner radius R_{inner} and an outer radius R_{outer}. Each drone aims to escape this arena within K steps, where each step has a fixed length δ. A drone is considered to have crashed if it comes within a distance R_{crash} of any of the J orbiting obstacles. Upon crashing, the drone ceases further navigation.

5.1 Encoding

We represent the annular arena by the region $[R_{inner}\cos(\theta), R_{outer}\cos(\theta)] \times [R_{inner}\sin(\theta), R_{outer}\sin(\theta)] \in \mathbb{R}^2$ for $\theta \in [0, 2\pi)$. The J obstacles orbit the arena with unique angular frequencies, ω_j, and orbital radii, r_j where the j^{th} obstacle has a coordinate at the k^{th} iteration: $\mathbf{o}_j^{(k)} = (o_{j1}^{(k)}, o_{j2}^{(k)}) : j = 1, 2, \ldots, J$ where $o_{j1}^{(k)} = r_j\cos(\omega_j \cdot k + \phi_j)$ and $o_{j2}^{(k)} = r_j\sin(\omega_j \cdot k + \phi_j)$ for angular frequencies, $\omega_j = 2\pi U(\frac{1}{P_{upper}}, \frac{1}{P_{lower}})$, phase shifts, $\phi_j = 2\pi U(0, 1)$ and orbital radii, $r_j = U(R_{inner} + R_{crash}, R_{outer})$. We draw the initial T drones at coordinates $\mathbf{x}_t^{(0)} = (x_{t1}^{(0)}, x_{t2}^{(0)}) : t = 1, 2, \ldots, T$ where $x_{t1}^{(0)} = r_t\cos(\theta_t)$ and $x_{t2}^{(0)} = r_t\sin(\theta_t)$ for $\theta_t = 2\pi U(0, 1)$ and $r_t = U(0, R_{inner})$.

5.2 The game state

After each movement made by an obstacle, the game state must be evaluated to determine whether a collision with a drone has occurred. Similarly, following each drone movement, the game state must again be assessed to establish whether the drone has either succeeded or failed. A failure is defined as the drone either (i) coming within a distance R_{crash} of an obstacle, signifying a crash, or (ii) failing to reach the boundary of the circular arena within K permitted steps. A success is defined as the drone reaching the edge of the arena. Consequently, the game state is evaluated twice during each time step k - once after the obstacle's move and once after the drone's. If neither a success nor failure condition is met, the drone proceeds to the next step. We encode this as:

$$s_t^k = \begin{cases} +1, & \text{if } \sqrt{r_{t1}^2 + r_{t2}^2} \geq R_{outer}, \\ -1, & \text{if } \exists j \in \{1, 2, \ldots, J\} \text{ such that } \|\mathbf{x}_t - \mathbf{o}_j\| \leq R_{crash}, \\ 0, & \text{otherwise.} \end{cases}$$

where s_t^k denotes the games status for drone t at turn k while $k \leq K$. Now if $s_t^k \in \{-1, 1\}$ for $k \leq K$ then the navigation for that particlar t^{th} drone ends. Otherwise, the game continues until $k > K$ and the status is recorded as -1 for that particlar t^{th} drone.

5.3 Control: Drone movement

During each step, we must move each drone by changing $(x_{t1}^{(k)}, x_{t2}^{(k)})$ by at most δ for at most K steps. Hence, the game-updating equation follows: $\mathbf{x}_t^{(k+1)} = \mathbf{x}_t^{(k)} + \mathbf{ct}(\mathbf{x}_t^{(k)}, \boldsymbol{\theta})\delta_t$ where $\mathbf{ct}(\mathbf{x}_t^{(k)}, \boldsymbol{\theta}) \in [-1,1]^2$ is some control vector for some parameter configuration $\boldsymbol{\theta} \in \mathbb{R}^S$. Now the interface between a model and the navigation is undergone through this control vector for which $\mathbf{ct} : (\mathbf{x}_t^{(k)}, \mathbf{o}_j^{(k)}, \boldsymbol{\theta}) \rightarrow \mathbf{model}\left(\boldsymbol{\Omega}\left(\mathbf{x}_t^{(k)}, \mathbf{o}_j^{(k)}, \boldsymbol{\phi}\left(\boldsymbol{\theta}_1\right)\right), \boldsymbol{\theta}_2\right) \xrightarrow{\sigma_L(.)} \left(\tilde{x}_{t1}^{(k)}, \tilde{x}_{t2}^{(k)}\right) \in [-1,1]^2$ where $\boldsymbol{\theta}_1 \subset \boldsymbol{\theta}, \boldsymbol{\theta}_2 \subseteq \boldsymbol{\theta}$ and $\boldsymbol{\theta}$ is fixed throughout the navigation period and all $t = 1, 2, \ldots, T$ drones move according to this $\boldsymbol{\theta}$. Additionally, $\left(\tilde{x}_{t1}^{(k)}, \tilde{x}_{t2}^{(k)}\right)$ represents the additions to the x and y coordinate of the t^{th} drone for the k^{th} step such that $\mathbf{x}_t^{(k+1)} = \mathbf{x}_t^{(k)} + \tilde{\mathbf{x}}_t^{(k)}\delta_t$. Furthermore, when the t^{th} drone is deemed terminal, we set $\delta_t = 0$, resulting in no positional update, that is, $\mathbf{x}_t^{(k+1)} = \mathbf{x}_t^{(k)}$ for the k^{th} iteration in which terminality is detected. Otherwise, $\delta_t = \delta$. In other words, if the t^{th} drone has reached a terminal state, its coordinates $\mathbf{x}_t^{(k)}$ are still passed through to the control vector $\mathbf{ct}$. Now in the framework of using a neural network as our model, we define $\boldsymbol{\Omega} : \left(\mathbf{x}_t^{(k)}, \mathbf{o}_j^{(k)}, \boldsymbol{\phi}\left(\boldsymbol{\theta}_1\right)\right) \rightarrow \mathbf{a}(t)^0 \in \mathbb{R}^{d_0}$ which signifies the vector of input nodes for the t^{th} drone, where $\boldsymbol{\phi}$ is a user-defined mapping. Additionally, $\boldsymbol{\theta}_2$ are the weights and biases of the neural network, $\mathbf{w} \in \mathbb{R}^R$, where $R \leq S$, and $\sigma_L(.)$ represents the hyperbolic tangent activation function applied at the output layer - allowing movement of the t^{th} drone in all directions.

5.3.1 Feature engineering

For effective manoeuvring of a drone around 'clumps' of obstacles, we propose to set the neural net input, $\boldsymbol{\Omega}\left(\mathbf{x}_t^{(k)}, \mathbf{o}_j^{(k)}, \boldsymbol{\phi}\left(\boldsymbol{\theta}_1\right)\right)$, for the t^{th} drone to merely the sum of all reciprocal Euclidean distances between said drone and all obstacles within a specified radius $R_{detection}$ of the drone. This approach has two key benefits: first, the input value increases when there are many obstacles within the drone's vicinity. Second, the input value becomes larger the closer a obstacle is to the drone. In this way, the input can quantify the level of caution the t^{th} drone should exercise.

Hence we set $\boldsymbol{\phi}\left(\boldsymbol{\theta}_1\right) = R_{detection}$ and $\boldsymbol{\Omega} : \left(\mathbf{x}_t^{(k)}, \mathbf{o}_j^{(k)}, R_{detection}\right) \rightarrow a(t)^0 \in \mathbb{R}^{d_0=1}$, that is, a single input node is created for each t^{th} drone with:

$$\boldsymbol{\Omega}(\mathbf{x}_t^{(k)}, \mathbf{o}_j^{(k)}, R_{detection}) = \begin{cases} \sum_{j^*=1}^{J^*} \frac{1}{\|\mathbf{x}_t^{(k)} - \mathbf{o}_{j^*}\| - R_{crash}} & \text{if } J^* > 0, \\ \frac{1}{\min_{1 \leq j \leq J} \left\|\mathbf{x}_t^{(k)} - \mathbf{o}_j\right\| - R_{crash}} & \text{if } J^* = 0. \end{cases} \tag{8}$$

where we subtract R_{crash} here as $\|\mathbf{x}_t^{(k)} - \mathbf{o}_{j^*}\| - R_{crash}$ represents the true distance the t^{th} drone may come to an obstacle without crashing. Now, $\|\mathbf{x}_t^{(k)} - \mathbf{o}_{j^*}\|$ denotes the euclidean distance between the t^{th} drone and the $j^{*^{th}}$ obstacle within radius $R_{detection}$ of the drone and there exists $J^* \leq J$ obstacles within the t^{th} drone's 'detection' circle. Furthermore, if $J^* = 0$ - that is, if there are no trees within radius $R_{detection}$ from the t^{th} drone - we return the reciprocal of the minimum euclidean distances between the t^{th} drone and all the J obstacles. We observe that, during the optimisation process, if it is estimated that $\hat{R}_{detection} < R_{crash}$, the first case in Equation 8 becomes futile in controlling the movement of non-terminal drones. For example, if $J^* = 1$ - that is, if there is a single obstacle within radius $R_{detection}$ of the t^{th} drone - then the drone would have already collided with that obstacle, rendering it terminal. Consequently, in such scenarios, the only viable control feedback for non-terminal drones is obtained from the second case of Equation 8, which governs behavior when no obstacles are detected within $R_{detection}$ of the t^{th} drone. We refer to this configuration, with the entirety of Equation 8 dictating the input node, as $\overset{(\text{I})}{\mathbf{model}}$ with input node $\overset{(\text{I})}{a}(t)^0$ and weights $\boldsymbol{\theta}_2^{(\text{I})}$ embedded in $\boldsymbol{\theta}^{(\text{I})}$. Now to ensure a meaningful $R_{detection}$ is estimated during the optimisation process, we introduce a second model, namely $\overset{(\text{II})}{\mathbf{model}}$, which replaces the second case in Equation 8 with 0 if $J^* = 0$, to create input node $\overset{(\text{II})}{a}(t)^0$ with weights $\boldsymbol{\theta}_2^{(\text{II})}$ as a subset of $\boldsymbol{\theta}^{(\text{II})}$.

Lastly, since we assume Gaussian priors for our parameters $\boldsymbol{\theta} \in \mathbb{R}^S$, that is $\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \sigma_\theta^2 \mathbf{I}_S)$, where for our case we have $\boldsymbol{\theta} = [\boldsymbol{\theta_1}, \boldsymbol{\theta_2}]'$ for which $\boldsymbol{\theta_1} = \theta_1 \in \mathbb{R}$ and $\boldsymbol{\theta_2} = \mathbf{w} \in \mathbb{R}^R$, we must ensure $\boldsymbol{\phi}\left(\boldsymbol{\theta}_1 = \theta_1\right) = R_{detection} \geq 0$. Accordingly, we define $\boldsymbol{\phi} = \phi$ as a logistic function with a scaling factor sf to ensure that the radius remains positive while being constrained by an upper bound equal to sf. More formally, define $\boldsymbol{\phi} = \phi$ as a mapping $\phi : \theta_1 \xrightarrow{\frac{sf}{1+\exp(-\theta_1)}} R_{detection}$, where the upper bound sf is a user-specified parameter determined by the geometry of the problem.

5.4 The arbitrary objective

Consider an arbitrary objective, where - for a given parameter configuration $\boldsymbol{\theta} \in \mathbb{R}^S$ and there exists T number of drones such $t = 1, 2, \ldots, T$ - we record the relative frequency of successes. Hence, the arbitrary objective is:

$$\begin{aligned}\underset{\boldsymbol{\theta}}{\operatorname{argmax}} \operatorname{Obj}(\boldsymbol{\theta}) &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \frac{1}{T} \sum_{t=1}^{T} \mathbb{I}(s_t^K(\boldsymbol{\theta}) = +1) \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \frac{1}{T} k(\boldsymbol{\theta}).\end{aligned}$$

where $s_t^K(\boldsymbol{\theta})$ represents the success status of the t^{th} drone after K steps, dependent on our parameter vector $\boldsymbol{\theta}$. By including L2 regularization, the L2 penalized objective becomes:

$$\underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left(\frac{1}{T} k(\boldsymbol{\theta}) - \nu \sum_{i=1}^{S} \theta_i^2 \right). \tag{9}$$

5.5 Likelihoods

The subsequent section introduces three distinct likelihood formulations - more accurately described as pseudo-likelihoods as previously explained in Section 4. The former two are loosely motivated by the fact that for count-based objectives - that is, objectives quantifying the number of successes - it is natural to construct a likelihood based on the binomial distribution, which defines a probability mass function for a fixed success probability and number of trials. However, as discussed in Section 3, it is necessary that these likelihoods be proportional to the objective function, that is, $p(\mathcal{D} \mid \boldsymbol{\theta}) \propto \operatorname{Obj}(\boldsymbol{\theta})$. Accordingly, we restructure these likelihoods to ensure they align with this notion. Additionally, Section 4 indicates that our likelihoods are tempered; hence, we incorporate a likelihood sharpness parameter $\beta \in \mathbb{R}^+$ to enable control over this.

More broadly, the pseudo-likelihood formulations in this section inherently exhibit differing sharpness (without explicitly inducing sharpness through the parameter β). Therefore, this section aims to facilitate a discussion on how different likelihoods - with inherently different sharpnesses - can induce distinct behaviours within the MCMC framework.

5.5.1 Binomial-based likelihood

We model the likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$ as the the probability of observing $k(\boldsymbol{\theta}) = \sum_{t=1}^{T} \mathbb{I}(s_t^K(\boldsymbol{\theta}) = +1) \leq T$ successes for T total drones given a success probability of $p_{\boldsymbol{\theta}} \approx \frac{k(\boldsymbol{\theta})}{T}$, that is $k(\boldsymbol{\theta}) \sim BIN(T, p_{\boldsymbol{\theta}})$ for $k(\boldsymbol{\theta}) \in \{0, 1, \ldots, T\}$. We note that the number of successes $k(\boldsymbol{\theta})$ is dependent on the specific parameter $\boldsymbol{\theta}$ used, and $p_{\boldsymbol{\theta}}$ is obtained empirically, hence, rendering our likelihood a simulation-based likelihood as follows:

$$\begin{aligned}p(\mathcal{D} \mid \boldsymbol{\theta}) &= \binom{T}{k(\boldsymbol{\theta})} p_{\boldsymbol{\theta}}^{k(\boldsymbol{\theta})} (1 - p_{\boldsymbol{\theta}})^{T - k(\boldsymbol{\theta})} \\ &= \binom{T}{k(\boldsymbol{\theta})} \left(\frac{k(\boldsymbol{\theta})}{T} \right)^{k(\boldsymbol{\theta})} \left(1 - \left(\frac{k(\boldsymbol{\theta})}{T} \right) \right)^{T - k(\boldsymbol{\theta})}\end{aligned} \tag{10}$$

We emphasize that this is not a classical likelihood function in the strictest sense, as both the observed outcome $k(\boldsymbol{\theta})$ and the estimated success probability $p_{\boldsymbol{\theta}}$ are derived from the same data. This creates a circularity in which the probability of the outcome, $\boldsymbol{\theta}$, is conditioned on a parameter, $p_{\boldsymbol{\theta}}$, that itself depends on the outcome. We view this pseudo-likelihood as a proxy for how well $\boldsymbol{\theta}$ explains the observed outcomes by quantifying the plausibility of $k(\boldsymbol{\theta})$ under a binomial model with $p_{\boldsymbol{\theta}} \approx \frac{k(\boldsymbol{\theta})}{T}$.

Now, one can not assume that $p(\mathcal{D} \mid \boldsymbol{\theta})$ in Equation 10 is monotonic increasing with respect to the number of successes $k(\boldsymbol{\theta})$. In fact, $p(\mathcal{D} \mid \boldsymbol{\theta})$ in Equation 10 exhibits a sole minimum when $k(\boldsymbol{\theta}) = \frac{T}{2}$, Theorem 1 in conjunction with Lemma 1 clarifies this.

Lemma 1. *$\frac{1}{n} > \psi^{(1)}(n+1)$ for $n > 0$.*

Proof. By Guo et al. (2015), we know $\psi^{(1)}(x) < \frac{1}{x + \frac{1}{2}} + \frac{1}{x^2}$ for $x > 0$. Hence $\psi^{(1)}(n+1) < \frac{1}{(n+1) + \frac{1}{2}} + \frac{1}{(n+1)^2} < \frac{1}{n+1} + \frac{1}{(n+1)^2}$ for $n > 0$. We show $\frac{1}{n} > \psi^{(1)}(n+1)$ by contradiction. Consider:

$$\begin{aligned}\frac{1}{n} &\leq \frac{1}{n+1} + \frac{1}{(n+1)^2} \\ \therefore 1 &\leq \frac{n^2 + 2n}{n^2 + 2n + 1} \\ \therefore 1 &\leq 0\end{aligned}$$

Which is a contradiction, implying $\frac{1}{n} > \frac{1}{n+1} + \frac{1}{(n+1)^2} > \psi^{(1)}(n+1)$ for $n > 0$. □

Theorem 1. *$f(x) = \binom{T}{x} (\frac{x}{T})^x (1 - \frac{x}{T})^{T-x}$ exhibits a sole minimum at $x = \frac{T}{2}$ for $x \in [0, T]$ with $T \in (0, \infty)$, with $f(x)$ being symmetric about $x = \frac{T}{2}$.*

Proof. Since $\lim_{x\to 0^+} f(x) = 1$ and $\lim_{x\to T^-} f(x) = 1$, we know $f(x)$ is continuous on the closed interval $[0, T]$. Consider:

$$l(x) = \log[f(x)] = \log(T!) - \log(x!) - \log[(T-x)!] + x\log\left(\frac{x}{T}\right) + (T-x)\log\left(\frac{T-x}{T}\right).$$

Using $\log(n!) = \log[\Gamma(n+1)]$ we have:

$$l(x) = \log[\Gamma(T+1)] - \log[\Gamma(x+1)] - \log[\Gamma(T-x+1)] + x\log\left(\frac{x}{T}\right) + (T-x)\log\left(\frac{T-x}{T}\right).$$

Taking the first derivative with respect to x and using the digamma function $\psi(n) = \frac{d}{dn}\log[\Gamma(n)]$:

$$l'(x) = -\psi(x+1) + \psi(T-x+1) + \log\left(\frac{x}{T-x}\right).$$

Since $l'(\frac{T}{2}) = -\psi(\frac{T}{2}+1) + \psi(\frac{T}{2}+1) + \log(1) = 0$, we know $f(x)$ has a critical point at $x = \frac{T}{2}$. We now show $f''(x) > 0$ on the interval $(0, T)$ implying $f(x)$ is convex and has most one minimum on this interval. We use the trigamma function $\frac{d\psi(n)}{dn} = \psi^{(1)}(n)$ and consider:

$$l''(x) = -\psi^{(1)}(x+1) - \psi^{(1)}(T-x+1) + \frac{1}{x} + \frac{1}{T-x}.$$

By Lemma 1, since $\frac{1}{x} > \psi^{(1)}(x+1)$ and $\frac{1}{T-x} > \psi^{(1)}(T-x+1)$, we have $l''(x) > 0$ on $x \in (0, T)$. Additionally, since:

$$\begin{aligned} f(T-x) &= \binom{T}{T-x}\left(\frac{T-x}{T}\right)^{T-x}\left(1-\frac{T-x}{T}\right)^{T-(T-x)} \\ &= \binom{T}{x}\left(\frac{x}{T}\right)^{x}\left(1-\frac{x}{T}\right)^{T-x} \\ &= f(x). \end{aligned}$$

we have shown that $f(x)$ is symmetric about $x = \frac{T}{2}$. □

To ensure a monotonic increasing likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$ on interval $k(\boldsymbol{\theta}) \in \{0, 1, \ldots, T\}$, we propose artificially changing the structure of the likelihood given in Equation 10. Since Theorem 1 implies that Equation 10 is monotonic decreasing on $k(\boldsymbol{\theta}) \in \{0, 1, \ldots, \frac{T}{2}\}$, after which being monotonic increasing on $k(\boldsymbol{\theta}) \in \{\frac{T}{2}, \frac{T}{2}+1, \ldots, T\}$, to enforce monotonicity over the entire interval, we define a piecewise function $h(x)$ as follows:

$$h(x) = \begin{cases} g(x), & \text{if } x < \frac{T}{2}, \\ f(x), & \text{if } x \geq \frac{T}{2}. \end{cases}$$

where $f(x) = \binom{T}{x}(\frac{x}{T})^x(1-\frac{x}{T})^{T-x}$ for $x \in \{0, 1, \ldots, T\}$ represents the original binomial-based likelihood with $g(x) = ax$ merely being a linear function where scaling factor $a = \frac{2}{T}f\left(\frac{T}{2}\right)$ has a dual function of ensuring continuity at $x = \frac{T}{2}$ as well as ensuring $g(x) \geq 0$ for $x \in \{0, 1, \ldots, \frac{T}{2}\}$. Additionally, one could make the argument that any monotonic increasing function $g(x)$ could be used, satisfying $g(x) \geq 0$ for $x \in \{0, 1, \ldots, \frac{T}{2}\}$ as well as the continuity constraint of $g(\frac{T}{2}) = f(\frac{T}{2})$. This notion is further elaborated on in Section 5.5.1.1.

To allow for sharper likelihoods - specifically, to increase the rate of change of $h(x)$ over the domain - we introduce a sharpness parameter $\beta \in \mathbb{R}^+$. This parameter modulates the steepness of the likelihood function by exponentiating it directly. That is, we define $\left[h(x)\right]^\beta$, where larger values of β yield a more pronounced increase in likelihood as a function of $k(\boldsymbol{\theta})$, effectively sharpening the likelihood surface, further elucidated in Section 5.5.1.2. We exponentiate $h(x)$ by β - rather than scale it via $\beta \cdot h(x)$ - because our focus is on amplifying the steepness of the log-likelihood, which plays a central role in acceptance probability computations (as seen in Equation 4). Specifically, taking the logarithm of $[h(x)]^\beta$ yields $\beta \cdot \log[h(x)]$, thereby linearly scaling the log-likelihood. Alternatively, the parameter β can be interpreted as a means of amplifying the likelihood ratio in Equation 3, yielding the modified expression $\left(\frac{p(\mathcal{D}|\boldsymbol{\theta}^*)}{p(\mathcal{D}|\boldsymbol{\theta}^{(j)})}\right)^\beta$. Increasing β makes the Markov chain more inclined to accept proposed solutions $\boldsymbol{\theta}^*$ that yield higher objective values, given the proportionality $p(\mathcal{D} \mid \boldsymbol{\theta}) \propto \text{Obj}(\boldsymbol{\theta})$ - in effect, making the MCMC sampler more likelihood-driven. As a result, the influence of the prior ratio during the acceptance step is diminished.

Furthermore, for $\boldsymbol{\theta} \in \mathbb{R}^S$, $k(\boldsymbol{\theta}) \in \{0, 1, \ldots, T\}$ and sharpness $\beta \in \mathbb{R}^+$, we have our new likelihood as:

$$p(\mathcal{D} \mid \boldsymbol{\theta}) = \left[h\left(k(\boldsymbol{\theta})\right)\right]^\beta. \tag{11}$$

Clearly, $\text{argmax}_{\boldsymbol{\theta}}\, p(\boldsymbol{\theta}|\mathcal{D})$, would now be equivalent to maximizing the L2 penalized objective in Equation 9 due to the monotonic increasing nature of $h(x)$ in Equation 11, as displayed in Figure 1 and compared to the original likelihood of Equation 10, in Figure 2. Additionally, Figure 3 illustrates the scaled log-likelihood, $\beta \cdot \log[h(x)]$, highlighting how increasing the sharpness

parameter β amplifies the curvature of the log-likelihood. This results in a steeper surface, thereby enhancing the sensitivity of the likelihood to changes in $x = k(\boldsymbol{\theta})$.

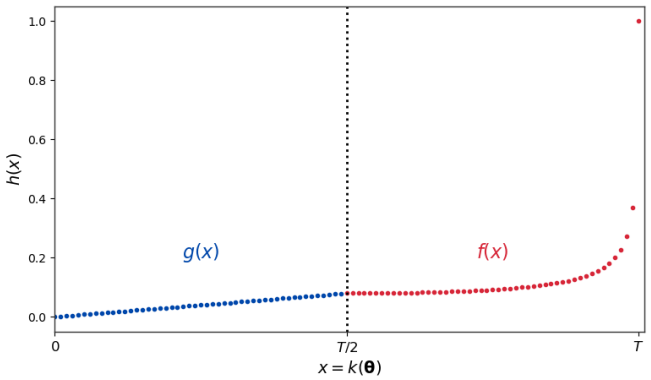


Figure 1: $h(x)$ on interval $[0, 1]$ for $x = k(\boldsymbol{\theta}) \in \{0, 1, \ldots, T\}$.

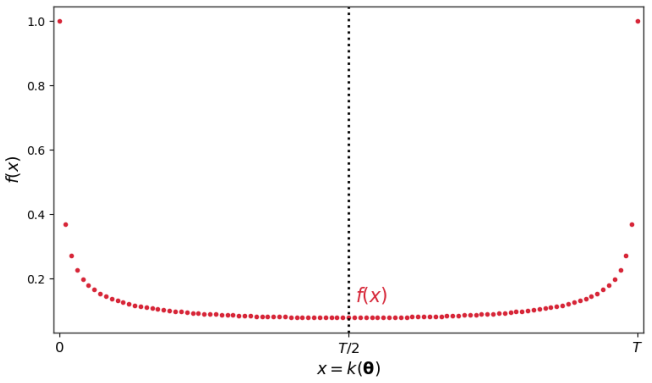


Figure 2: $f(x)$ on interval $[0, 1]$ for $x = k(\boldsymbol{\theta}) \in \{0, 1, \ldots, T\}$

.

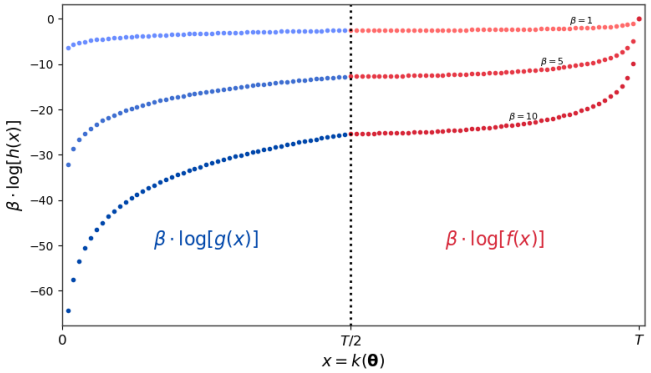


Figure 3: $\beta \cdot \log[h(x)]$ for $x = k(\boldsymbol{\theta}) \in \{0, 1, \ldots, T\}$ for various β.

5.5.1.1 Rebuttals against different $g(x)$ forms It is important to re-emphasize that the modified likelihood consisting of $h(k(\boldsymbol{\theta}))$ is constructed primarily to facilitate efficient sampling in an MCMC context, where the goal is to identify the mode of the conditional posterior, $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D})$. Since the mode necessarily lies in the high-likelihood (high-valued objective) region (due to $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D}) \propto p(\mathcal{D} \mid \boldsymbol{\theta}) \cdot p(\boldsymbol{\theta} \mid \sigma_\theta^2)$) - specifically in the upper half of the domain $k(\boldsymbol{\theta}) \in \{\frac{T}{2}, \frac{T}{2} + 1, \ldots, T\}$, where the original likelihood function $f(k(\boldsymbol{\theta}))$ is already monotonic increasing - our primary interest lies in accurately sampling from this latter half. The lower half $k(\boldsymbol{\theta}) \in \{0, 1, \ldots, \frac{T}{2}\}$, where the original likelihood is decreasing, serves primarily as a transitional region that we wish to exit efficiently during sampling. As such, the choice of $g(k(\boldsymbol{\theta}))$ on this interval can be quite flexible: it need only be monotonic increasing and continuous at $k(\boldsymbol{\theta}) = \frac{T}{2}$. For this reason, we adopt a simple linear form $g(k(\boldsymbol{\theta})) = a \cdot k(\boldsymbol{\theta})$, which accelerates the sampler's movement through low-likelihood regions and thus enhances convergence toward the high-likelihood regions that contribute meaningfully to the conditional posterior mode. This design choice is well justified given that our objective is not full posterior sampling, but rather efficient localization of the posterior mode (achieved by deliberately concentrating samples around it).

5.5.1.2 Sharpness $\beta \in \mathbb{R}^+$ In the context of the two-sample MCMC framework, wherein a pseudo-likelihood is modulated by a sharpness parameter β, this parameter plays a critical role in shaping the behaviour of the Markov chain by governing the peakedness of the likelihood. Since β exponentiates $p(\boldsymbol{\theta} \mid \mathcal{D})$, not only is the likelihood's rate of change across the domain increased, but the density itself becomes sharper in the sense that mass is more strongly concentrated around its mode. This is because exponentiating the likelihood amplifies higher values and suppresses lower ones, thereby steepening the posterior landscape. Specifically, a high value of β accentuates the pseudo-likelihood, yielding a sharply peaked conditional posterior $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D})$ - given that $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D}) \propto p(\mathcal{D} \mid \boldsymbol{\theta}) \cdot p(\boldsymbol{\theta} \mid \sigma_\theta^2)$ - and thus concentrates samples around the mode. This enhances exploitation by focusing the chain on high-likelihood regions. However, in the presence of multimodality, a large β may cause the chain to become trapped in a single dominant mode, hindering exploration of other dominant modes, as proposals that move away from the current region can receive extremely low acceptance probabilities α_θ. Conversely, a low β reduces the sharpness of the likelihood, flattening the conditional posterior and enabling the chain to move more freely across the parameter space, thereby promoting exploration of multiple modes at the expense of slower convergence to high-likelihood regions. The choice of β therefore embodies a trade-off between exploration and exploitation in the MCMC process.

An alternative perspective on the role of the sharpness parameter β is that it controls the number and prominence of modes in the conditional posterior. For example, a low β flattens the likelihood, allowing multiple regions of the parameter

space to "compete" for conditional posterior mass and thereby inducing multimodality. In contrast, a high β sharpens the conditional posterior, collapsing it onto dominant modes and potentially suppressing minor alternatives.

5.5.2 Beta-based likelihood

Invoking the identity $n! = \Gamma(n+1)$ for $n \in \mathbb{W}$, Equation 10 may be written as such:

$$p(\mathcal{D} \mid \boldsymbol{\theta}) = \frac{\Gamma(T+1)}{\Gamma(k(\boldsymbol{\theta})+1) \cdot \Gamma(T-k(\boldsymbol{\theta})+1)} \left(\frac{k(\boldsymbol{\theta})}{T}\right)^{k(\boldsymbol{\theta})} \left(1 - \left(\frac{k(\boldsymbol{\theta})}{T}\right)\right)^{T-k(\boldsymbol{\theta})}.$$

for $k(\boldsymbol{\theta}) \in \{0, 1, \ldots, T\}$. Yet, since we would like to model the proportion of successes $\frac{1}{T}k(\boldsymbol{\theta}) \in [0,1]$, yet retain the constraint of the original likelihood $p(\mathcal{D} \mid \boldsymbol{\theta}) \in [0,1]$ (better seen as $f(x)$ in Figure 2), we model the likelihood as such:

$$p(\mathcal{D} \mid \boldsymbol{\theta}) = \frac{\Gamma(2)}{\Gamma\left(\frac{1}{T}k(\boldsymbol{\theta})+1\right) \cdot \Gamma\left(2-\frac{1}{T}k(\boldsymbol{\theta})\right)} \left(\frac{k(\boldsymbol{\theta})}{T}\right)^{\frac{1}{T}k(\boldsymbol{\theta})} \left(1 - \left(\frac{k(\boldsymbol{\theta})}{T}\right)\right)^{1-\frac{1}{T}k(\boldsymbol{\theta})} \tag{12}$$

for $\frac{1}{T}k(\boldsymbol{\theta}) \in \{0, \frac{1}{T}, \frac{2}{T}, \ldots, 1\}$. Rescaling Equation 10, using the proportion of successes, $\frac{1}{T}k(\boldsymbol{\theta}) \in [0,1]$, instead of the number of successes, $k(\boldsymbol{\theta})$, resembles a binomial distribution for a single trial, $T = 1$. Now, as before, since Theorem 1 implies that Equation 12 is monotonic decreasing on $\frac{1}{T}k(\boldsymbol{\theta}) \in [0, \frac{1}{2})$, after which being monotonic increasing on $\frac{1}{T}k(\boldsymbol{\theta}) \in [\frac{1}{2}, 1]$, to enforce monotonicity over the entire interval, we define a piecewise function $h(x)$ as follows:

$$h(x) = \begin{cases} g(x), & \text{if } x < \frac{1}{2}, \\ f(x), & \text{if } x \geq \frac{1}{2}. \end{cases}$$

where $f(x) = \frac{\Gamma(2)}{\Gamma(x+1)\Gamma(2-x)}(x)^x(1-x)^{1-x}$ for $x \in [0,1]$ represents the original likelihood in Equation 12. Furthermore, we note that $f(x)$ is proportional to the density function of a Beta$(\alpha, 3-\alpha)$ distribution scaled by $\frac{1}{2}$ where $\alpha = x+1$. Now $g(x) = ax$ is a linear function with scaling factor $a = 2f\left(\frac{1}{2}\right)$. Furthermore, for $\boldsymbol{\theta} \in \mathbb{R}^S$, $\frac{1}{T}k(\boldsymbol{\theta}) \in [0,1]$ and sharpness $\beta \in \mathbb{R}^+$, we have our new likelihood as:

$$p(\mathcal{D} \mid \boldsymbol{\theta}) = \left[h\left(\frac{1}{T}k(\boldsymbol{\theta})\right)\right]^{\beta} \tag{13}$$

Figure 4 illustrates the re-scaled likelihood function (beta-based) from Equation 13, while Figure 5 displays the original likelihood from Equation 12. Additionally, Figure 6 illustrates the scaled log-likelihood, $\beta \cdot \log[h(x)]$ for various β.

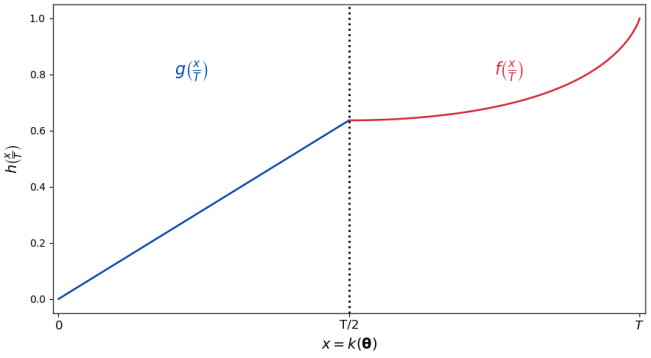


Figure 4: $\beta \cdot h\left(\frac{x}{T}\right)$ on interval $[0, \beta]$ for $x = k(\boldsymbol{\theta}) \in \{0, 1, \ldots, T\}$ for various β.

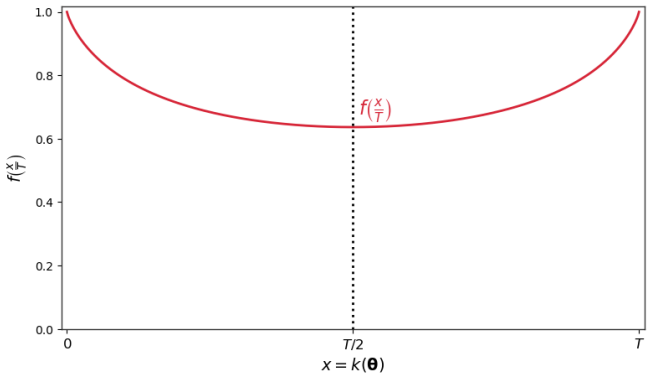


Figure 5: $f\left(\frac{x}{T}\right)$ on interval $[0,1]$ for $x = k(\boldsymbol{\theta}) \in \{0, 1, \ldots, T\}$

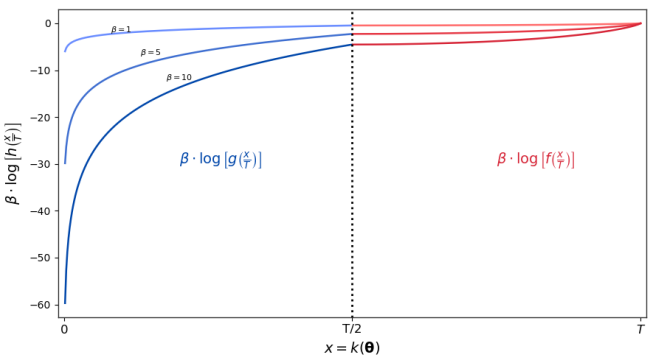


Figure 6: $\beta \cdot \log[h(x)]$ for $x = k(\boldsymbol{\theta}) \in \{0, 1, \ldots, T\}$ for various β.

A key distinction between the beta-based likelihood in Equation 13 and the binomial-based likelihood in Equation 11 (see Section 5.5.1) lies in the differing rates of change across the support $k(\boldsymbol{\theta}) \in \{0, 1 \ldots, T\}$. Specifically, the binomial-based log-likelihood exhibits a relatively flat profile across much of the support, followed by a pronounced increase in steepness at higher values of $k(\boldsymbol{\theta})$. This behavior is visually evident in Figure 8, where the gradient of the binomial-based log-likelihood clearly exceeds that of the beta-based log-likelihood for large $k(\boldsymbol{\theta})$. In the context of the MCMC algorithm, such a steep ascent corresponds to a more sharply peaked posterior $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D})$ - hence acceptance into higher likelihood regions, $k(\boldsymbol{\theta}) \in \{\frac{T}{2}+1, \frac{T}{2}+2, \ldots, T\}$, are likely to be accelerated when using the binomial-based likelihood. It is also worthwhile to note that the derivative of both binomial and beta-based log-likelihoods for domain $k(\boldsymbol{\theta}) \in \{0, 1, \ldots, \frac{T}{2}\}$ are equal, that is, $\beta \cdot \log\left[\frac{d}{dx} g^{\text{Binomial}}(x)\right] = \beta \cdot \log\left[\frac{d}{dx} g^{\text{Beta}}(\frac{x}{T})\right] = \beta \cdot \frac{1}{x}$ as shown in Figure 7. This similarity implies that, from an MCMC perspective, transitions out of lower-likelihood regions, $k(\boldsymbol{\theta}) \in \{0, 1, \ldots, \frac{T}{2}\}$, into the higher likelihood regions, $k(\boldsymbol{\theta}) \in \{\frac{T}{2}+1, \frac{T}{2}+2, \ldots, T\}$, are likely to proceed with similar efficiency under both likelihoods. It is important to note, however, that the derivatives depicted reflect infinitesimal changes in the log-likelihood. In contrast, the MH algorithm typically evaluates differences over finite, and often substantial, differences in log-likelihood values. As such, our derivative plots provide interpretive value primarily in scenarios where the proposed $\boldsymbol{\theta}$ yield objective function values of comparable magnitude.

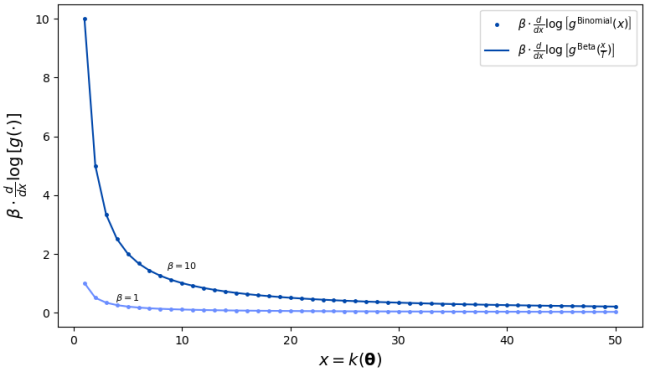


Figure 7: $\beta \cdot \frac{d}{dx} \log[g(\cdot)]$ on interval $[0, \beta]$ for $x = k(\boldsymbol{\theta}) \in \{0, 1, \ldots, T\}$ for various β.

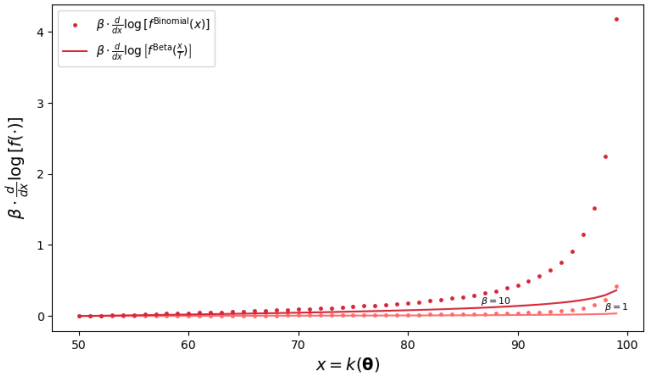


Figure 8: $\beta \cdot \frac{d}{dx} \log[f(\cdot)]$ for $x = k(\boldsymbol{\theta}) \in \{0, 1, \ldots, T\}$

5.5.3 Exponential-based likelihood

Now to reiterate, in the context of this work, our primary objective is not to perform full Bayesian inference via MCMC, but rather to identify the mode of the conditional $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D})$. In such cases, the MCMC algorithm is not employed for its traditional role in posterior sampling, but rather as a stochastic optimisation tool that facilitates a random search over the parameter space. From this perspective, strict adherence to the exact posterior structure is unnecessary. It suffices that the proposal mechanism is guided by a function that monotonically increases with the objective of interest, thereby biasing the random walk toward high-likelihood (or high-valued objective) regions. In treating MCMC as a mode-seeking algorithm, the requirement for an explicit, well-defined likelihood linking $\boldsymbol{\theta}$ to the data becomes less critical.

Being such, we utilize an exponential function as said monotonic increasing function as a substitute for a well-defined likelihood where we define $h(x) = \exp(x)$ for $x > 0$. Hence for $\boldsymbol{\theta} \in \mathbb{R}^S$, $\frac{1}{T} k(\boldsymbol{\theta}) \in [0, 1]$ and sharpness $\beta \in \mathbb{R}^+$, we have our new likelihood as:

$$\begin{aligned} p(\mathcal{D} \mid \boldsymbol{\theta}) &= \left[h\left(\frac{1}{T} k(\boldsymbol{\theta})\right)\right]^\beta \\ &= \exp\left(\beta \cdot \frac{1}{T} k(\boldsymbol{\theta})\right) \end{aligned} \tag{14}$$

We observe in Figure 9 a comparison between the exponential likelihood defined in Equation 14 and the alternative likelihood formulations given in Equations 11 and 13. Given that the MH algorithm bases proposal acceptance on the ratio of likelihoods - more precisely, the difference in log-likelihoods - the absolute magnitude of the likelihood function is of limited relevance. Instead, the relative rate of change, as illustrated in Figure 10, offers more informative insight into how the Markov chain is guided through the parameter space.

As illustrated in Figure 10, the derivative of the exponential-based log-likelihood, given by $\beta \cdot \frac{1}{T}$, exceeds the gradients of the alternative log-likelihoods over the domain $k(\boldsymbol{\theta}) \in \left\{\frac{T}{2}, \frac{T}{2}+1, \ldots, T\right\}$. However, this dominance diminishes as $k(\boldsymbol{\theta})$ increases. In contrast, the derivative of the binomial-based log-likelihood surpasses those of the alternatives at higher values of $k(\boldsymbol{\theta})$. In this regard, the binomial-based formulation may be considered advantageous, as its steeper gradient in the upper region of the domain can facilitate more dynamic transitions toward higher-valued objective areas, potentially reducing the risk of the Markov chain becoming trapped in local optima.

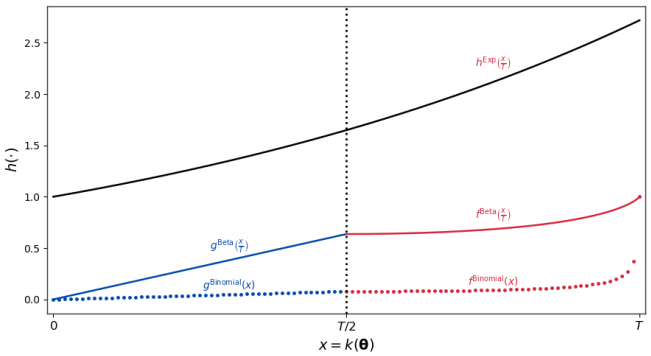


Figure 9: $h^{\text{Binomial}}(x), h^{\text{Beta}}\left(\frac{x}{T}\right)$ and $h^{\text{Exp}}\left(\frac{x}{T}\right)$ for $x = k(\boldsymbol{\theta}) \in \{0, 1, \ldots, T\}$.

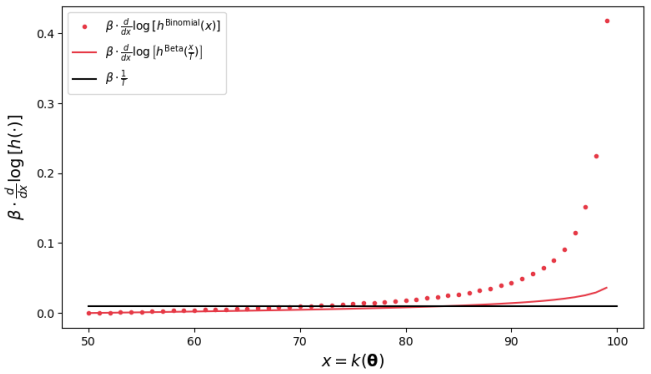


Figure 10: $\beta \cdot \frac{d}{dx} \log[h(\cdot)]$ for $x = k(\boldsymbol{\theta}) \in \{\frac{T}{2}, \frac{T}{2} + 1, \ldots, T\}$ for $\beta = 1$.

5.6 Effects of regularization

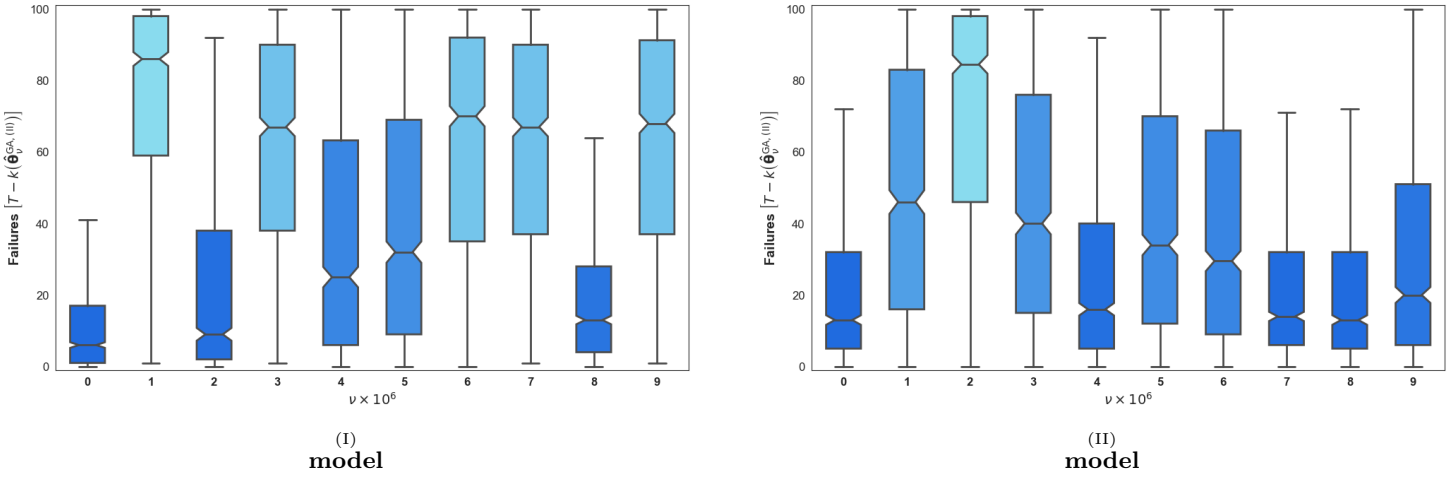
To preliminarily evaluate the effect of regularization on the performance of $\boldsymbol{\theta}$, we assess generalization under a given regularization strength ν. Specifically, we apply the estimator $\hat{\boldsymbol{\theta}}_{\nu}^{\text{GA}}$ - obtained via a GA with L_2 regularization - to a newly initialized environment. - defined by a distinct set of J obstacle coordinates at the k^{th} iteration, $\mathbf{o}_j^{(k)}$ for $j = 1, 2, \ldots, J$. These coordinates are governed by a new set of angular frequencies ω_j, phase shifts ϕ_j, and orbital radii r_j. Additionally, a new initialization is also characterized by T new initial drone coordinates, $\mathbf{x}_t^{(0)}$ for $t = 1, 2, \ldots, T$. We dictate our training initialization by using seed value ω_0^{Train}, with 1000 test initializations governed by seed values $\{\omega_j^{\text{Test}}\}_{j=1}^{1000}$. More formally, our game-updating equation becomes $\mathbf{x}_t^{(k+1)} = \mathbf{x}_t^{(k)} + \mathbf{ct}(\mathbf{x}_t^{(k)}, \hat{\boldsymbol{\theta}}_{\nu}^{\text{GA}})\delta$ for which $\forall j, k : \mathbf{o}_j^{(k)} \cap \mathbf{o}_j^{(k)^{\text{Train}}} = \emptyset$ and $\forall t : \mathbf{x}_t^{(0)} \cap \mathbf{x}_t^{(0)^{\text{Train}}} = \emptyset$. Figure 11 illustrates the distributions of failures, where the number of failures are defined as $T - k\left(\hat{\boldsymbol{\theta}}_{\nu}^{\text{GA}}\right)$, for 1000 of these initializations against varying regularization strengths ν for both $\overset{\text{(I)}}{\textbf{model}}$ and $\overset{\text{(II)}}{\textbf{model}}$. Table 1 displays the median and mean of the success distributions, where the number of successes are defined as $k\left(\hat{\boldsymbol{\theta}}_{\nu}^{\text{GA}}\right)$, as well as displaying $\hat{R}_{detection}$ which is subsumed in $\hat{\boldsymbol{\theta}}_{\nu}^{\text{GA}}$.

With respect to $\overset{\text{(I)}}{\textbf{model}}$, Table 1 indicates that improved out-of-sample performance is achieved when the optimization procedure estimates $\hat{R}_{detection} < R_{crash} = 0.05$ (see Appendix C for additional specifications). This suggests that only the second case of Equation 8 is necessary to govern the behavior of non-terminal drones in order to enhance out-of-sample performance. In contrast, for $\overset{\text{(II)}}{\textbf{model}}$, Table 1 suggests that improved out-of-sample performance is attained when the estimated detection radius satisfies $\hat{R}_{detection} \approx 0.07$. These notions are further supported by the results in Table 2.

Furthermore, no consistent pattern in drone success rates on the out-of-sample set is observed as ν varies, as illustrated in Figure 11. Instead, performance appears to depend primarily on whether $\hat{R}_{detection}$ is estimated to be greater or less than the crash radius $R_{crash} = 0.05$ under $\overset{\text{(I)}}{\textbf{model}}$, or whether $\hat{R}_{detection} \approx 0.07$ under $\overset{\text{(II)}}{\textbf{model}}$. The results suggest that there are many plausible solutions that yield high objective values on the in-sample set, although only a subset of these generalize well to the out-of-sample set.

	$\overset{(\mathrm{I})}{\mathbf{model}}\left(\overset{(\mathrm{I})}{a}(t)^0, \hat{\boldsymbol{\theta}}_{2,\nu}^{\mathrm{GA},(\mathrm{I})}\right)$				$\overset{(\mathrm{II})}{\mathbf{model}}\left(\overset{(\mathrm{II})}{a}(t)^0, \hat{\boldsymbol{\theta}}_{2,\nu}^{\mathrm{GA},(\mathrm{II})}\right)$			
	In-Sample	Out-of-Sample			In-Sample	Out-of-Sample		
$\nu \times 10^6$	$k\left(\hat{\boldsymbol{\theta}}_{\nu}^{\mathrm{GA},(\mathrm{I})}\right)$	$\tilde{k}\left(\hat{\boldsymbol{\theta}}_{\nu}^{\mathrm{GA},(\mathrm{I})}\right)$	$\overline{k}\left(\hat{\boldsymbol{\theta}}_{\nu}^{\mathrm{GA},(\mathrm{I})}\right)$	$\hat{R}_{detection}$	$k\left(\hat{\boldsymbol{\theta}}_{\nu}^{\mathrm{GA},(\mathrm{II})}\right)$	$\tilde{k}\left(\hat{\boldsymbol{\theta}}_{\nu}^{\mathrm{GA},(\mathrm{II})}\right)$	$\overline{k}\left(\hat{\boldsymbol{\theta}}_{\nu}^{\mathrm{GA},(\mathrm{II})}\right)$	$\hat{R}_{detection}$
0	100	**94.00**	**87.58**	**0.0350**	100	**87.00**	**78.67**	**0.0726**
1	99	14.00	24.86	0.1993	98	54.00	50.61	0.1056
2	99	**91.00**	**75.69**	**0.0094**	100	15.50	29.77	0.1206
3	99	33.00	37.06	0.1238	98	60.00	54.6	0.0983
4	99	75.00	63.61	0.0048	100	84.00	73.90	0.1075
5	100	68.00	59.55	0.0063	99	66.00	58.56	0.0962
6	98	30.00	36.64	0.1481	98	70.50	61.09	0.1016
7	98	33.00	37.29	0.1309	100	**86.00**	**78.47**	**0.0721**
8	100	**87.00**	**79.97**	**0.0136**	100	**87.00**	**79.34**	**0.0724**
9	98	32.00	36.51	0.1249	100	80.00	69.20	0.1161

Table 1: Number of successes $k\left(\hat{\boldsymbol{\theta}}_{\nu}^{\mathrm{GA}}\right)$ for the in-sample initialization, median ($\tilde{k}$) and mean ($\overline{k}$) for the distributions of successes on the 1000 test initializations and estimated $\hat{R}_{detection}$, against varying ν for $T = 100$ and $R_{crash} = 0.05$ using both $\overset{(\mathrm{I})}{\mathbf{model}}$ and $\overset{(\mathrm{II})}{\mathbf{model}}$.



$\overset{(\mathrm{I})}{\mathbf{model}}$ $\overset{(\mathrm{II})}{\mathbf{model}}$

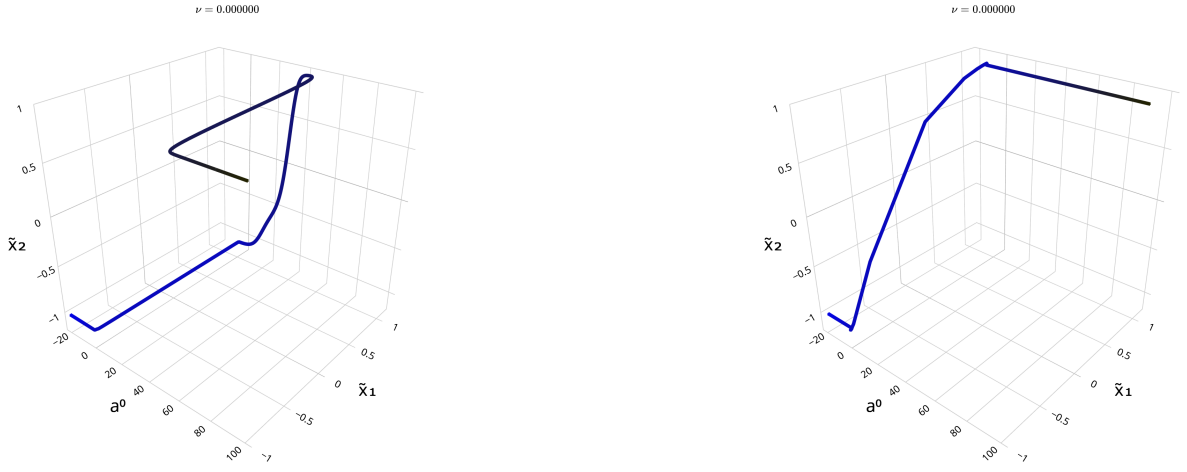
Figure 11: Boxplots illustrating distributions of failures, $T - k\left(\hat{\boldsymbol{\theta}}_{\nu}^{\mathrm{GA}}\right)$, against varying ν obtained through 1000 test initializations for $T = 100$ for $\overset{(\mathrm{I})}{\mathbf{model}}$ (left) and $\overset{(\mathrm{II})}{\mathbf{model}}$ (right).

As shown in Table 2, there is a clear trend of decreasing in-sample performance as the regularization strength increases across both models. This behaviour is consistent with underfitting resulting from excessive regularization - that is, the models become overly constrained. Furthermore, the results suggest that the use of the GA may not be essential for achieving improved performance on the in-sample set, as the RS (used only for $\overset{(\mathrm{II})}{\mathbf{model}}$) performs comparably to the GA at low values of ν. This observation implies that the fine-tuning capabilities of the GA offer limited benefit when the model is only weakly regularized - under such conditions, the GA's exploitation mechanisms appear to be inconsequential, with its role in refining existing parent solutions rendered largely unnecessary. Interestingly, at moderate regularization levels, the RS appears to estimate $\hat{R}_{detection} \approx 0.07$ more effectively than the GA. However, this outcome reinforces the notion that multiple plausible solutions exist which yield high objective values on the in-sample set, and that the RS may have coincidentally identified ones that generalize well. Nevertheless, Table 1 demonstrates that the GA is indeed capable of consistently discovering this advantageous value of $\hat{R}_{detection}$.

	$\overset{(\mathrm{I})}{\mathbf{model}}\left(\overset{(\mathrm{I})}{a}(t)^0, \hat{\boldsymbol{\theta}}_{2,\nu}^{\mathrm{GA},(\mathrm{I})}\right)$			$\overset{(\mathrm{II})}{\mathbf{model}}\left(\overset{(\mathrm{II})}{a}(t)^0, \hat{\boldsymbol{\theta}}_{2,\nu}^{\mathrm{GA},(\mathrm{II})}\right)$			$\overset{(\mathrm{II})}{\mathbf{model}}\left(\overset{(\mathrm{II})}{a}(t)^0, \hat{\boldsymbol{\theta}}_{2,\nu}^{\mathrm{RS},(\mathrm{II})}\right)$		
	In-Sample	OOS		In-Sample	OOS		In-Sample	OOS	
ν	$k\left(\hat{\boldsymbol{\theta}}_{\nu}^{\mathrm{GA},(\mathrm{I})}\right)$	$\overline{k}\left(\hat{\boldsymbol{\theta}}_{\nu}^{\mathrm{GA},(\mathrm{I})}\right)$	$\hat{R}$	$k\left(\hat{\boldsymbol{\theta}}_{\nu}^{\mathrm{GA},(\mathrm{II})}\right)$	$\overline{k}\left(\hat{\boldsymbol{\theta}}_{\nu}^{\mathrm{GA},(\mathrm{II})}\right)$	$\hat{R}$	$k\left(\hat{\boldsymbol{\theta}}_{\nu}^{\mathrm{RS},(\mathrm{II})}\right)$	$\overline{k}\left(\hat{\boldsymbol{\theta}}_{\nu}^{\mathrm{RS},(\mathrm{II})}\right)$	$\hat{R}$
0.000001	99	24.86	0.1993	98	50.61	0.1056	100	**77.51**	**0.0666**
0.00001	98	37.54	0.1362	100	68.29	0.0847	98	**79.55**	**0.0668**
0.0001	100	**75.04**	**0.0070**	100	42.63	0.2843	96	**78.36**	**0.0779**
0.001	96	31.27	0.1463	98	25.30	0.0942	70	18.68	0.0375
0.01	74	18.62	0.5863	74	18.61	0.5406	21	19.45	0.9914
0.1	62	17.63	0.4640	73	18.14	0.4550	8	18.62	0.1608
1	8	12.15	0.4403	49	13.57	0.5855	70	18.68	0.1757

Table 2: Number of successes $k\left(\hat{\boldsymbol{\theta}}_{\nu}^{\mathrm{GA}}\right)$ for the in-sample initialization, mean $(\bar{k})$ for the distributions of successes on the 1000 test initializations (OOS) and estimated $\hat{R}_{detection}$, against varying ν for $T = 100$ and $R_{crash} = 0.05$ using both $\overset{(\mathrm{I})}{\mathbf{model}}$ and $\overset{(\mathrm{II})}{\mathbf{model}}$.

We observe the response curves in Figure 12, corresponding to a given input $a^0 \in \mathbb{R}$, which yields the output $\tilde{\mathbf{x}} = (\tilde{x}_1, \tilde{x}_2)$ using $\overset{(\mathrm{I})}{\mathbf{model}}\left(\begin{matrix}\overset{(\mathrm{I})}{a}{}^0, \hat{\boldsymbol{\theta}}_{2,\nu}^{\mathrm{GA},(\mathrm{I})}\end{matrix}\right)$ and $\overset{(\mathrm{II})}{\mathbf{model}}\left(\begin{matrix}\overset{(\mathrm{II})}{a}{}^0, \hat{\boldsymbol{\theta}}_{2,\nu}^{\mathrm{GA},(\mathrm{II})}\end{matrix}\right)$. It is worth noting that values of $a^0 < 0$ are feasible, as such cases arise only when a given drone has entered a terminal state. Furthermore, no distinguishable pattern appears to emerge as ν varies - each response curve exhibits a unique shape, with no visually discernible structure that consistently correlates with improved performance as reflected in Table 1.



$\overset{(\mathrm{I})}{\mathbf{model}}$ $\overset{(\mathrm{II})}{\mathbf{model}}$

Figure 12: Response Curves of $\tilde{\mathbf{x}} = (\tilde{x}_1, \tilde{x}_2)$ given $\overset{(\mathrm{I})}{a}{}^0$ using $\overset{(\mathrm{I})}{\mathbf{model}}\left(\begin{matrix}\overset{(\mathrm{I})}{a}{}^0, \hat{\boldsymbol{\theta}}_{2,\nu}^{\mathrm{GA},(\mathrm{I})}\end{matrix}\right)$ (left) and $\tilde{\mathbf{x}} = (\tilde{x}_1, \tilde{x}_2)$ given $\overset{(\mathrm{II})}{a}{}^0$ using $\overset{(\mathrm{II})}{\mathbf{model}}\left(\begin{matrix}\overset{(\mathrm{II})}{a}{}^0, \hat{\boldsymbol{\theta}}_{2,\nu}^{\mathrm{GA},(\mathrm{II})}\end{matrix}\right)$ (right) for various ν.

5.7 MCMC

We aim to compare the behaviour of the three likelihood formulations discussed previously - the binomial-based likelihood (Equation 10), the beta-based likelihood (Equation 13), and the exponential-based likelihood (Equation 14) - in terms of their impact on the MCMC results obtained. To ensure a fair comparison, we evaluate each likelihood using the same fixed value of the sharpness parameter, specifically $\beta = 20$, selected arbitrarily. It is important to note that each pseudo-likelihood induces values of differing magnitudes when evaluated at the same $k(\boldsymbol{\theta})$. Consequently, we refrain from reporting the corresponding values of the conditional posterior $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D})$ (since $p(\boldsymbol{\theta} \mid \sigma_\theta^2, \mathcal{D}) \propto p(\mathcal{D} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta} \mid \sigma_\theta^2)$ and the scale of the likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$ varies considerably across formulations). This notion is illustrated in Figure 9. Furthermore, we employ $\overset{(\mathrm{II})}{\mathbf{model}}$ in this section.

Table 3 presents the results, from which it is evident that the binomial-based likelihood yields the best in-sample solution. However, none of the likelihood formulations appear to generalize well to the out-of-sample set of initializations. This is most likely attributable to the estimation of the parameter $R_{detection}$. As discussed in Section 5.6, the out-of-sample performance of a given solution appears to be highly sensitive to the value estimated for this parameter.

Additionally, the multivariate effective sample sizes (ESS) reported in Table 3 provide evidence of satisfactory mixing. High ESS values indicate that the sampler is generating effectively independent samples - that is, with low autocorrelation - which are representative of the target posterior distribution. All reported ESS values exceed the commonly accepted threshold of 100 (as recommended by **?**, Section 4), which supports the claim of efficient exploration. This performance is not coincidental: as discussed in Section 2.2, the sampler was deliberately tuned using an adaptive MH scheme to encourage good mixing. Furthermore, Table 3 suggests that the exponential-based likelihood exhibits the best mixing properties, as indicated by the ESS. That said, there is no clear evidence of poor mixing in the other likelihood formulations.

Figure 13 illustrates the poor convergence behaviour of the beta- and exponential-based likelihoods, as evidenced by the plots of $\|\boldsymbol{\theta}^{(j)}\|^2$ across MCMC iterations. The top panel displays all 100,000 MCMC iterations. Their trajectories exhibit an initial phase of high variability and gradual decline, followed by a transition into a region of relative stability. This behavior suggests that the sampler undergoes a prolonged adaptation phase before reaching a region of the parameter space where $\|\boldsymbol{\theta}^{(j)}\|^2$ fluctuates around a stable value. The prolonged descent and subsequent stabilisation imply delayed convergence for the beta- and exponential-based likelihood types, necessitating a longer burn-in period to ensure samples are drawn from the stationary distribution. We note, however, that the binomial-based likelihood exhibits relatively good convergence. For the beta- and exponential-based likelihoods, a burn-in of 60,000 iterations was applied - motivated by the point at which $\|\boldsymbol{\theta}^{(j)}\|^2$ appears to stabilise - whereas a shorter burn-in of 20,000 iterations was used for the binomial-based likelihood. The corresponding marginal distributions, $p(\sigma_\theta^2 \mid \mathcal{D})$, are shown in the bottom panel of Figure 13.

Despite the different convergence dynamics, the marginal distributions, $p(\sigma_\theta^2 \mid \mathcal{D})$, across all likelihood types appear to be consistent with an inverse-gamma form. Given that the conditional posterior $\sigma_\theta^2 \mid \boldsymbol{\theta}, \mathcal{D} \sim \text{Inv-Gamma}\left(a + \frac{S}{2},\ b + \frac{\|\boldsymbol{\theta}\|^2}{2}\right)$ with $a, b \approx 0$ (known from Section 2.1), we expect that if $\|\boldsymbol{\theta}^{(j)}\|^2$ fluctuates around a constant value c, then the marginal distribution $\sigma_\theta^2 \mid \mathcal{D} \sim \text{Inv-Gamma}\left(a + \frac{S}{2},\ b + \frac{c}{2}\right)$ - that is, it should also retain the inverse-gamma form with approximately constant shape and scale parameters. Regarding the level of regularization inferred from the training set - where regularization is represented via the dispersion parameter $\sigma_\theta^2 \propto \frac{1}{\nu}$ - we may draw meaningful conclusions from the marginals of $\sigma_\theta^2 \mid \mathcal{D}$ shown in Figure 13. Specifically, the variation in these distributions across different likelihood types suggests that each likelihood inherently induces a different degree of regularization, which is subsequently reflected in the MAP estimates of the parameters.

Furthermore, we may attribute the superior in-sample performance of the binomial-based likelihood to its sharper increase at high values of $k(\boldsymbol{\theta})$, as we previously alluded to in Figure 10. We propose that the likelihood function can be interpreted as a mechanism for modulating the structural complexity of the conditional posterior. Specifically, in the case of flatter likelihoods, the influence of the likelihood on the posterior is weak, resulting in a diffuse posterior landscape that may support multiple regions with comparable acceptance probabilities α_θ. This, in turn, can induce a multimodal posterior structure. In such a regime, the sampler is encouraged to explore broadly, often traversing disconnected or competing modes across the parameter space. However, this extensive exploration may lead the Markov chain to eventually converge to a mode that is not aligned with a high-value region of the objective function, given the presence of multiple modes. In contrast, a more sharply peaked likelihood suppresses minor modes and concentrates posterior mass around a dominant region, thereby sharpening the posterior and promoting unimodality. We posit that this behaviour underlies the relatively better performance observed with the binomial-based likelihood on the in-sample set, in contrast to the beta- and exponential-based alternatives, which likely converge to sampling around one of these minor modes.

Now, to potentially improve the in-sample performance using the beta- and exponential-based likelihoods, one could increase their likelihood sharpness parameter β, thereby inducing a more concentrated conditional posterior. We do not pursue this adjustment in the present analysis however.

	In-Sample	Out-of-Sample			
Likelihood Type	$k\left(\hat{\boldsymbol{\theta}}_\nu^{\text{GA,(II)}}\right)$	$\tilde{k}\left(\hat{\boldsymbol{\theta}}_\nu^{\text{GA,(II)}}\right)$	$\overline{k}\left(\hat{\boldsymbol{\theta}}_\nu^{\text{GA,(II)}}\right)$	$\hat{R}_{detection}$	ESS
Binomial-based	100	13.00	24.45	0.3912	179.8550
Beta-based	50	8.00	12.72	0.5175	170.0205
Exponential-based	52	12.00	17.25	0.5601	317.2764

Table 3: Number of successes $k\left(\hat{\boldsymbol{\theta}}_\nu^{\text{GA}}\right)$ for the in-sample initialization, summary statistics for the distributions of successes on the 1000 test initializations and estimated $\hat{R}_{detection}$, for the three likelihood types.

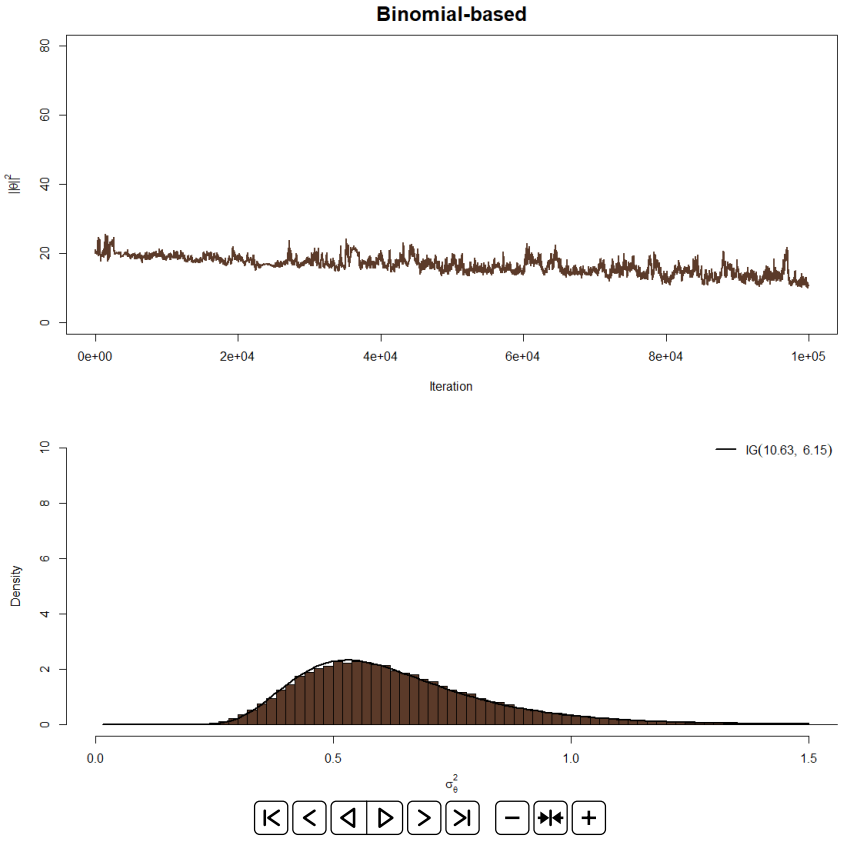


Figure 13: $\|\boldsymbol{\theta}^{(j)}\|^2$ for $j = 1, \ldots 100,000$ (no burn-in) in the top panel, with distribution of marginal $\sigma^2_\theta \mid \mathcal{D}$ (using $20,000$ burn-in for binomial-based and $60,000$ burn-in for beta- and exponential-based) for the three likelihood types in the bottom panel (all plots use the same scale).

Appendix A Joint Metropolis-Hastings

We define $\boldsymbol{\Lambda} = \left[\boldsymbol{\theta}', \sigma^2_\theta\right]' \in \mathbb{R}^{S+1}$ and re-derive our MH algorithm to include σ^2_θ. Hence, given the current state of $\boldsymbol{\Lambda}$, that is $\boldsymbol{\Lambda}^{(j)}$, the MH algorithm proposes a new value $\boldsymbol{\Lambda}^*$ obtained from $\boldsymbol{\Lambda}^* = \boldsymbol{\Lambda}^{(j)} + \mathbf{Q}$. Subsequently, $\boldsymbol{\Lambda}^*$ is accepted as the new value in the Markov chain under the following acceptance criterion:

$$\boldsymbol{\Lambda}^{(j+1)} = \begin{cases} \boldsymbol{\Lambda}^*, & \text{if } U < \alpha \\ \boldsymbol{\Lambda}^{(j)}, & \text{otherwise.} \end{cases}$$

Where the vector $\mathbf{Q} = [\mathbf{Q}_\theta{}', Q_{\sigma^2_\theta}]' \in \mathbb{R}^{S+1}$ denotes drawn values from proposal densities: $\mathbf{Q}_\theta \sim \mathcal{N}(\mathbf{0}, \sigma^2_{Q_\theta}\mathbf{I}_S)$ and $Q_{\sigma^2_\theta} \sim$ Inv-Gamma(a_Q, b_Q). Now, α is given by:

$$\begin{aligned}
\alpha &= \min\left(\frac{p(\boldsymbol{\Lambda}^*|\mathcal{D})}{p(\boldsymbol{\Lambda}^{(j)}|\mathcal{D})} \cdot \frac{Q(\boldsymbol{\Lambda}^{(j)}|\boldsymbol{\Lambda}^*)}{Q(\boldsymbol{\Lambda}^*|\boldsymbol{\Lambda}^{(j)})}, 1\right) \\
&= \min\left(\frac{p(\mathcal{D}|\boldsymbol{\Lambda}^*)p(\boldsymbol{\Lambda}^*)}{p(\mathcal{D}|\boldsymbol{\Lambda}^{(j)})p(\boldsymbol{\Lambda}^{(j)})} \cdot \frac{Q(\boldsymbol{\theta}^{(j)}, \sigma^2_{\theta(j)}|\boldsymbol{\theta}^*, \sigma^2_{\theta^*})}{Q(\boldsymbol{\theta}^*, \sigma^2_{\theta^*}|\boldsymbol{\theta}^{(j)}, \sigma^2_{\theta(j)})}, 1\right) \\
&= \min\left(\frac{p(\mathcal{D}|\boldsymbol{\theta}^*)p(\boldsymbol{\theta}^*, \sigma^2_{\theta^*})}{p(\mathcal{D}|\boldsymbol{\theta}^{(j)})p(\boldsymbol{\theta}^{(j)}, \sigma^2_{\theta(j)})} \cdot \frac{Q_\theta(\boldsymbol{\theta}^{(j)}|\boldsymbol{\theta}^*)Q_{\sigma^2_\theta}(\sigma^2_{\theta(j)}|\sigma^2_{\theta^*})}{Q_\theta(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(j)})Q_{\sigma^2_\theta}(\sigma^2_{\theta^*}|\sigma^2_{\theta(j)})}, 1\right) && \text{Likelihood not dependent on } \sigma^2_\theta \\
&= \min\left(\frac{p(\mathcal{D}|\boldsymbol{\theta}^*)p(\boldsymbol{\theta}^*|\sigma^2_{\theta^*})p(\sigma^2_{\theta^*})}{p(\mathcal{D}|\boldsymbol{\theta}^{(j)})p(\boldsymbol{\theta}^{(j)}|\sigma^2_{\theta(j)})p(\sigma^2_{\theta(j)})} \cdot \frac{Q_{\sigma^2_\theta}(\sigma^2_{\theta(j)}|\sigma^2_{\theta^*})}{Q_{\sigma^2_\theta}(\sigma^2_{\theta^*}|\sigma^2_{\theta(j)})}, 1\right) && \text{Symmetry of } Q_\theta \qquad (15)
\end{aligned}$$

Assuming $\sigma^2_\theta \sim$ Inv-Gamma (a, b) and still assuming $\boldsymbol{\theta}|\sigma^2_\theta \sim \mathcal{N}(\mathbf{0}, \sigma^2_\theta\mathbf{I}_S)$, Equation 15 simplifies to:

$$\alpha = \min\left(\frac{h\left(k(\boldsymbol{\theta}^*)\right) \cdot \frac{1}{\sqrt{\left(2\pi\sigma^2_{\theta^*}\right)^S}} \exp\left(-\frac{1}{2\sigma^2_{\theta^*}}\|\boldsymbol{\theta}^*\|^2\right) \cdot \frac{1}{\left(\sigma^2_{\theta^*}\right)^{a+1}} \exp\left(-\frac{b}{\sigma^2_{\theta^*}}\right) \cdot \frac{1}{\left(\sigma^2_{\theta(j)}\right)^{a_Q+1}} \exp\left(-\frac{b_Q}{\sigma^2_{\theta(j)}}\right)}{h\left(k(\boldsymbol{\theta}^{(j)})\right) \cdot \frac{1}{\sqrt{\left(2\pi\sigma^2_{\theta(j)}\right)^S}} \exp\left(-\frac{1}{2\sigma^2_{\theta(j)}}\|\boldsymbol{\theta}^{(j)}\|^2\right) \cdot \frac{1}{\left(\sigma^2_{\theta(j)}\right)^{a+1}} \exp\left(-\frac{b}{\sigma^2_{\theta(j)}}\right) \cdot \frac{1}{\left(\sigma^2_{\theta^*}\right)^{a_Q+1}} \exp\left(-\frac{b_Q}{\sigma^2_{\theta^*}}\right)}, 1\right)$$

and by taking the log:

$$\log(\alpha) = \min\Bigg(\log\left(h(k(\boldsymbol{\theta}^*))\right) - \frac{S}{2}\log(2\pi\sigma^2_{\theta^*}) - \frac{\|\boldsymbol{\theta}^*\|^2}{2\sigma^2_{\theta^*}} - (a+1)\log\left(\sigma^2_{\theta^*}\right) - \frac{b}{\sigma^2_{\theta^*}}$$
$$- (a_Q+1)\log\left(\sigma^2_{\theta^{(j)}}\right) - \frac{b_Q}{\sigma^2_{\theta^{(j)}}} - \log\left(h(k(\boldsymbol{\theta}^{(j)}))\right) + \frac{S}{2}\log(2\pi\sigma^2_{\theta^{(j)}})$$
$$+ \frac{\|\boldsymbol{\theta}^{(j)}\|^2}{2\sigma^2_{\theta^{(j)}}} + (a+1)\log\left(\sigma^2_{\theta^{(j)}}\right) + \frac{b}{\sigma^2_{\theta^{(j)}}} + (a_Q+1)\log\left(\sigma^2_{\theta^*}\right) + \frac{b_Q}{\sigma^2_{\theta^*}}, 0\Bigg)$$

Appendix B Genetic Algorithm

Define $\boldsymbol{\theta}^{(n,m)} \in \mathbb{R}^p$ to be the n^{th} solution (individual) from the m^{th} generation such that $n = \{1,\ldots,N\}$ and $m = \{1,\ldots,M\}$. We present the N-size population as $\boldsymbol{\Theta}^{(m)} = \left[\boldsymbol{\theta}^{(1,m)}, \boldsymbol{\theta}^{(2,m)}, \ldots, \boldsymbol{\theta}^{(N,m)}\right]_{p\times N}$ for the m^{th} generation. We initialize by setting $\boldsymbol{\Theta}^{(0)}$ where $\boldsymbol{\theta}^{(n,0)} \sim \mathcal{U}_p$ for $n = \{1,\ldots,N\}$. At the termination of the algorithm, we return the solution $\boldsymbol{\theta}^{(n^\star,m^\star)}$ that achieved the highest objective value across all individuals and generations:

$$\boldsymbol{\theta}^{(n^\star,m^\star)} = \underset{n\in\{1,\ldots,N\},\, m\in\{1,\ldots,M\}}{\arg\max} \text{Obj}\left(\boldsymbol{\theta}^{(n,m)}\right).$$

Hence for generation $m = 1,\ldots,M$:

Fitness

For $n = 1,\ldots,N$, we compute the fitness for each n^{th} individual of the m^{th} generation as $f_{n,m} = \text{Obj}\left(\boldsymbol{\theta}^{(n,m-1)}\right)$.

Selection (Roulette Wheel)

1. Compute selection probabilities:
$$p_{n,m} = \frac{f_{n,m}}{\sum_{i=1}^N f_{i,m}}, \quad n = 1,\ldots,N.$$
2. Compute cumulative probabilities:
$$C_{n,m} = \sum_{i=1}^{n} p_{i,m}, \quad n = 1,\ldots,N.$$
3. For each selection $i = 1,\ldots,N$:
   (a) Sample $r \sim \mathcal{U}(0,1)$.
   (b) Find the smallest n such that $C_{n,m} \geq r$.
   (c) Select parent $\boldsymbol{\theta}^{(n,m)} = \tilde{\boldsymbol{\theta}}^{(n,m)}$
4. To form the mating pool $\boldsymbol{\mathcal{M}}^{(m)} = \left[\tilde{\boldsymbol{\theta}}^{(1,m)}, \tilde{\boldsymbol{\theta}}^{(2,m)}, \ldots, \tilde{\boldsymbol{\theta}}^{(N,m)}\right]_{p\times N}$

Recombination (Blend-α crossover)

Select two parents $\tilde{\boldsymbol{\theta}}^{(i^*,m)}$ and $\tilde{\boldsymbol{\theta}}^{(j^*,m)}$ from the mating pool $\boldsymbol{\mathcal{M}}^{(m)}$, to create offspring (with a fixed α):

1. Repeat until N offspring are created:

   For each gene $r = 1,2,\ldots,p$:

   i. Sample $u_r \sim \mathcal{U}(0,1)$

   ii. Compute blend weight:
   $$v_r = (1+2\alpha)u_r - \alpha$$

   iii. Generate offspring gene:
   $$\hat{\theta}^{(n,m)}_r = v_r \cdot \tilde{\theta}^{(i^\star,m)}_r + (1-v_r)\cdot \tilde{\theta}^{(j^\star,m-1)}_r$$

   to create offspring $\boldsymbol{\mathcal{O}}^{(m)} = \left[\hat{\boldsymbol{\theta}}^{(1,m)}, \hat{\boldsymbol{\theta}}^{(2,m)}, \ldots, \hat{\boldsymbol{\theta}}^{(N,m)}\right]_{p\times N}$

Mutation (Gaussian)

Randomly perturb a subset of the newly created offspring in $\boldsymbol{\mathcal{O}}^{(m)}$:

1. Sample a subset of indices $\mathcal{K} \subset \{1, \ldots, N\}$ uniformly at random.
2. For each $k \in \mathcal{K}$, apply a Gaussian perturbation:

$$\hat{\boldsymbol{\theta}}^{(k,m)} \leftarrow \hat{\boldsymbol{\theta}}^{(k,m)} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$$

Replacement (Elitisim)

Preserve the best-performing individuals from generation m and newly created offspring:

1. Compute $2N$ fitness values $f_{i,m} = \text{Obj}\left(\boldsymbol{\Theta}^{(m)}, \boldsymbol{\mathcal{O}}^{(m)}\right)$ for $i = 1, \ldots, 2N$
2. Identify the index set of the top N individuals from $\{\boldsymbol{\Theta}^{(m)}, \boldsymbol{\mathcal{O}}^{(m)}\}$:

$$\mathcal{E} = \{i \in \{1, \ldots, 2N\} : f_{i,m} \text{ is among the top } N \text{ values in } \{f_{1,m}, \ldots, f_{2N,m}\}\}$$

3. The $(m+1)^{th}$ generation is defined as $\boldsymbol{\Theta}^{(m+1)} = \left[\boldsymbol{\theta}^{(1,m+1)}, \boldsymbol{\theta}^{(2,m+1)}, \ldots, \boldsymbol{\theta}^{(N,m+1)}\right]_{p \times N}$ where every $\boldsymbol{\theta}^{(n,m+1)}$ is such that $n \in \mathcal{E}$.

Appendix C Specifications

Navigation Problem

Parameter	Value
R_{inner}	0.25
R_{outer}	1
R_{crash}	0.05
K	250
J	50
T	100
δ	0.01
P_{lower}	$2K$
P_{upper}	$3K$
sf	1
ω_0^{Train}	2024
$\{\omega_j^{\text{Test}}\}_{j=1}^{1000}$	$\{1, 2, \ldots, 1000\}$

Table 4: Specifications for Navigation Problem

Overall Specifications

Parameter	Value
Neural Network	
L (# layers in NN excluding input layer)	3
d_1, d_2 (# nodes in each hidden layer)	3
$\sigma_1(\cdot), \sigma_2(\cdot)$ (activation functions for both hidden layers)	$\tanh(\cdot)$
Metropolis-Hastings	
σ^2_{Init} (unless otherwise stated)	1
δ	1000
Δ	100
$(s^2)^{(0)}$	1
κ	0.6

Table 5: Overall Specifications

Appendix D Random Search

Define the objective function $\text{Obj}(\boldsymbol{\theta})$, search space $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^p$ and number of samples/iterations S.

1. Initialize by setting best score $f^\star \leftarrow -\infty$, best parameter $\boldsymbol{\theta}^\star \leftarrow$ null.
2. For $s = 1, \ldots, S$:
   (a) Sample $\boldsymbol{\theta}^{(s)} \sim \mathcal{U}_p(\Theta)$.
   (b) Evaluate objective:
   $$f_s = \text{Obj}\left(\boldsymbol{\theta}^{(s)}\right)$$
   (c) If $f_s > f^\star$, then update:
   $$f^\star \leftarrow f_s, \quad \boldsymbol{\theta}^\star \leftarrow \boldsymbol{\theta}^{(s)}$$
3. Our solution is the best found parameter $\boldsymbol{\theta}^\star$ with corresponding score $f^\star$.

References

Amari, S.-i. (1993). Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4-5):185–196.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.

Dobson, A. J. and Barnett, A. G. (2018). *An introduction to generalized linear models*. Chapman and Hall/CRC.

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741.

Guo, B.-N., Qi, F., Zhao, J.-L., and Luo, Q.-M. (2015). Sharp inequalities for polygamma functions. *Mathematica Slovaca*, 65(1):103–120.

Haario, H., Saksman, E., and Tamminen, J. (2001). An adaptive metropolis algorithm. *Bernoulli*, 7(2):223–242.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd edition.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications.

Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.

Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092.

Painter, M. and Brunskill, E. (2018). Cs234 notes - lecture 1: Introduction to reinforcement learning. Accessed: 2025-02-10.

Roberts, G. O., Gelman, A., and Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk metropolis algorithms. *The Annals of Applied Probability*, 7(1):110–120.

Roberts, G. O. and Rosenthal, J. S. (2007). Coupling and ergodicity of adaptive markov chain monte carlo algorithms. *Journal of Applied Probability*, 44(2):458–475.

Vapnik, V. (1991). Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4.