MANUSCRIPT

Simulation-Based Prior Knowledge Elicitation for Parametric Bayesian Models

Authors:

Florence BOCKTING
Department of Statistics
TU Dortmund University, Germany

Stefan T. RADEV
Cognitive Science Department
Rensselaer Polytechnic Institute, NY, USA

Paul-Christian BÜRKNER
Department of Statistics
TU Dortmund University, Germany

Last update: Tuesday 22nd August, 2023

corresponding author: Florence Bockting, florence.bockting@tu-dortmund.de

Simulation-based prior knowledge elicitation for parametric Bayesian models

Abstract

A central characteristic of Bayesian statistics is the ability to consistently incorporate prior knowledge into various modeling processes. In this paper, we focus on translating domain expert knowledge into corresponding prior distributions over model parameters, a process known as prior elicitation. Expert knowledge can manifest itself in diverse formats, including information about raw data, summary statistics, or model parameters. A major challenge for existing elicitation methods is how to effectively utilize all of these different formats in order to formulate prior distributions that align with the expert's expectations, regardless of the model structure. To address these challenges, we develop a simulation-based elicitation method that can learn the hyperparameters of potentially any parametric prior distribution from a wide spectrum of expert knowledge using stochastic gradient descent. We validate the effectiveness and robustness of our elicitation method in four representative case studies covering linear models, generalized linear models, and hierarchical models. Our results support the claim that our method is largely independent of the underlying model structure and adaptable to various elicitation techniques, including quantile-based, moment-based, and histogram-based methods.

Keywords — Bayesian statistics, prior specification, prior elicitation, simulation-based inference, expert knowledge

1 Introduction

The essence of Bayesian statistics lies in the ability to consistently incorporate prior knowledge into the modeling process (Jaynes, 2003; Gelman et al., 2013). Accordingly, specifying sensible prior distributions over the parameters of Bayesian models can have multiple advantages. For instance, integrating theory-guided knowledge into otherwise data-driven models effectively constrains the model behavior within an expected range (Navarro, 2019; Mikkola et al., 2023; Manderson & Goudie, 2023). This integration ideally enhances both model faithfulness and computational aspects, including convergence and sampling efficiency. Moreover, well-specified prior distributions can positively influence various model criteria, such as parameter recoverability, predictive performance, and convergence (Bürkner et al., 2022).

However, despite these apparent advantages, prior specification poses a significant challenge for analysts as it is often unclear a priori what constitutes a "sensible" prior (Gelman et al., 2017). In this paper, we focus on one specific approach to prior specification, namely, the elicitation and translation of expert knowledge into prior distributions, also known as *prior elicitation* (Mikkola et al., 2023). Against this background, *a sensible prior is one that accurately reflects domain knowledge as elicited from an expert or a group of experts*. Still, achieving the latter criterion poses challenges on its own. Model parameters for which priors are needed might lack intuitive meaning for the domain expert, making it difficult to query the properties of these parameters directly (Albert et al., 2012). Moreover, the relationship between priors and the data may not be apparent from the model specification, especially for complex models, such as hierarchical models (da Silva et al., 2019). Indeed, constructing priors for every single model parameter in models with a large number of parameters might be inefficient or even infeasible. Finally, the domain expert might lack statistical expertise to translate their knowledge into appropriate prior distributions (Hartmann et al., 2020).

To address these challenges, several tools for prior elicitation have been developed in the past, as reviewed by Garthwaite et al. (2005) and O'Hagan et al. (2006). However, despite the widespread acceptance and routine application of Bayesian statistics nowadays, the field of prior elicitation still lags behind in terms of its routine implementation by practitioners (see Mikkola et al. (2023) for a recent comprehensive review). One contributing factor is that many existing methods primarily aim to elicit information about the model parameters directly. This approach makes these methods inherently model-specific and limits their widespread applicability. Additionally, as mentioned previously, the emphasis on parameters can present a challenge for experts in terms of interpretability (da Silva et al., 2019; Bedrick et al., 1996; Kadane et al., 1980).

In recent years, there has been an increasing focus on the development of model-agnostic approaches that center around the prior predictive distribution (Manderson & Goudie, 2023). These methods allow for the integration of expert knowledge regarding observed data patterns (i.e., elicitation in the observable space). In contrast to interpreting model parameters, domain experts can usually effectively interpret the scale and magnitude of observable quantities (Mikkola et al., 2023; Muandet et al., 2017; Akbarov, 2009; da Silva et al., 2019; Hartmann et al., 2020). Despite these recent developments, the general applicability as well as the actual application of elicitation methods remain limited (Mikkola et al., 2023). This lack of popularity persists, at least in part, because these methods are relatively complex, do not easily generalize to different types of expert information, or necessitate substantial tuning or other manual adjustments. In light of the preceding considerations, we introduce an elicitation method that seeks to overcome these challenges. Specifically, this work makes a contribution to prior elicitation research by proposing a method that satisfies the following criteria:

1. *Model Independence*: Our method is agnostic to the specific probabilistic model, as long as sampling from it is feasible and stochastic gradients can be computed.

2. *Effective Utilization of Expert Knowledge*: By incorporating diverse expert information on model parameters, observed data patterns, or other relevant statistics, our method maximizes the utility of expert knowledge and expands the diversity of information embedded in the model.

3. *Flexibility in Knowledge Formats and Elicitation Techniques*: Our method can accommodate various knowledge formats and adapt to different elicitation techniques, ensuring that

individual expert preferences are considered.

4. *Modular Design*: Having a modular structure, our method provides flexibility to analysts, allowing for easy adaptation, improvement, or replacement of specific components, both during method development and application.

2 Methods

We propose a new elicitation method for translating knowledge from a domain expert into an appropriate parametric prior distribution. In particular, our approach builds on recent contributions made by Hartmann et al. (2020), da Silva et al. (2019), and Manderson & Goudie (2023) (see Section 3 for details on these approaches). Their key commonality is the development of (more or less) model-agnostic methods in which the search for appropriate prior distributions is formulated as an optimization problem. Thus, the objective is to determine the optimal hyperparameters that minimize the discrepancy between model-implied and expert-elicited statistics. We also adopt this perspective and introduce a novel elicitation method that supports expert feedback in both the space of parameters and observable quantities (i.e., a *hybrid* approach) and minimizes human effort. The key ideas underlying our method are outlined as follows:

1. The analyst defines a generative model comprising a likelihood function $p(y \mid \theta)$ and a parametric prior distribution $p(\theta \mid \lambda)$ for the model parameters, where λ represents the prior hyperparameters to be inferred from expert knowledge.

2. The analyst selects a set of target quantities, which may involve queries related to observable quantities (data), model parameters, or anything else in between.

3. The domain expert is queried using a specific elicitation technique for each target quantity (*expert-elicited statistics*).

4. From the generative model implied by likelihood and prior and a given value of λ, parameters and (prior) predictive data are simulated, and the predefined set of target quantities is computed based on the simulations (*model-implied quantities*).

5. The discrepancy between the model-implied and the expert-elicited statistics is evaluated with a discrepancy measure (loss function).

6. Stochastic gradient descent is employed to update the hyperparameters λ so as to minimize the loss function.

7. Steps 4 to 6 are repeated iteratively until an optimal set of hyperparameters λ is found that minimizes the discrepancy between the model-implied and the expert-elicited statistics.

In the upcoming sections, we will delve into the details of the outlined approach. Following the terminology introduced in Mikkola et al. (2023), our prior elicitation procedure involves two key individuals: the *analyst* and the *domain expert*. The analyst has a dual role, acting both as a statistician responsible for formulating the generative model and selecting the target quantities to be queried from the expert, as well as a facilitator who extracts the requested information from the domain expert. On the other hand, the domain expert possesses valuable knowledge pertaining to the uncertain quantities of interest that the analyst aims to extract.

To provide a visual representation of all steps involved in our proposed elicitation method, Figure 1 presents a graphical overview. In addition, readers can find a symbol glossary in Appendix A for a quick reference.

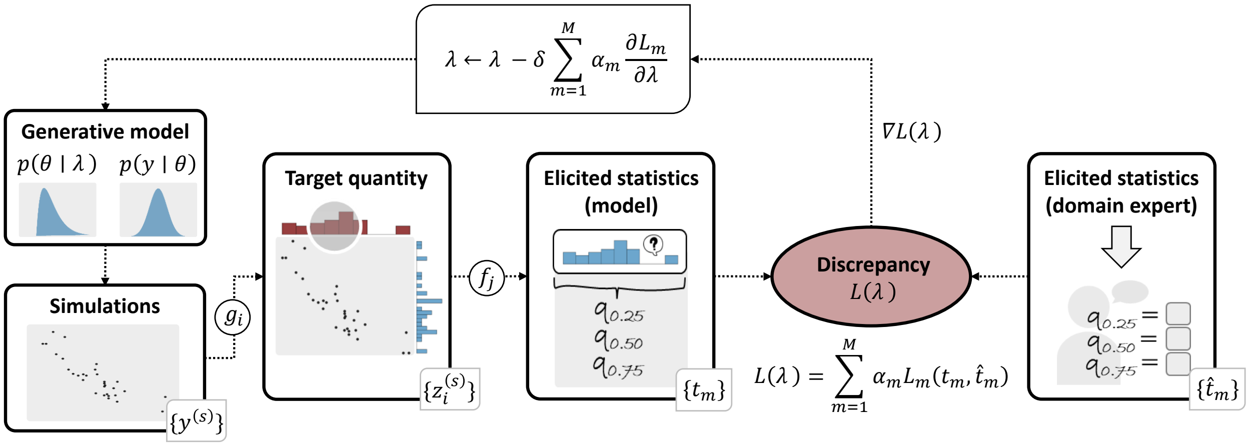

Figure 1: *Graphical illustration of our simulation-based elicitation method.* Step 1 involves employing elicitation techniques to extract target quantities from the domain expert. Subsequently, the objective is to minimize the discrepancy between model-implied and expert-elicited statistics by optimizing the hyperparameters λ. The optimization process iteratively simulates data using the current hyperparameters λ, computes model-implied elicited statistics, compares them with the expert-elicited statistics using a loss function (L_m), and updates λ to improve agreement between model-implied and expert-elicited statistics. Here, α_m is the weight of the m^{th} loss component and δ is the step size.

2.1 Elicited Statistics From the Expert

We assume that the analyst queries the domain expert regarding a predetermined set of I target quantities, represented as $\{z_i\} := \{z_i\}_{i=1}^{I}$. The set $\{z_i\}$ is selected by the analyst, whereby the choice of target quantities is influenced by the requirements of the statistical model (da Silva et al., 2019; Stefan et al., 2022). Additionally, the selected target quantities should align with the expert's knowledge, encompassing both potential knowledge about model parameters and/or observable data patterns (Mikkola et al., 2023). Once this set is defined, the expert is queried regarding each individual target quantity z_i, assuming that the expert possesses an implicit representation, denoted as $\hat{z}_i$, which can be accessed using expert elicitation techniques (Manderson & Goudie, 2023; Kadane & Wolfson, 1998).

The utilization of *elicitation techniques* for querying the expert is essential for effectively capturing the probabilistic nature of the expert's judgment (O'Hagan et al., 2006; Hartmann et al., 2020). While numerous elicitation techniques have been proposed in the literature (see e.g. Stefan et al., 2022), it can be argued that these techniques essentially represent different facets of the following three general method families: moment-based elicitation (e.g., mean and standard deviation), quantile-based elicitation (e.g., median, lower quartile, and upper quartile), and histogram elicitation (e.g., constructing a histogram by sampling from the distribution of z_i). Each target quantity z_i can be elicited through a distinct elicitation technique f_j. As emphasized by Falconer et al. (2022) and Albert et al. (2012), the selection of an appropriate elicitation technique depends on the specific context and must align with the expert's preferences. Hence, it is desirable to possess a methodology that accommodates different encoding formats. Within our notation, we represent the i^{th} target quantity elicited from the expert through the j^{th} elicitation technique as $\hat{t}_m = \hat{t}_{ij}$ and refer to it as *elicited statistics*:

$$\hat{t}_m = f_j(\hat{z}_i). \tag{1}$$

In this context, the index $m = 1, \ldots, M$ indicates the number of elicited statistics resulting from specific target-quantity × elicitation-technique combinations, as selected by the analyst. We use m instead of ij as subscript in order to emphasize that the set of elicited statistics consists of a *selection* rather than all possible combinations ij.

2.2 Model-Based Quantities

Considering the set of elicited statistics queried from the expert $\{\hat{t}_m\}$, it is possible to assess the extent to which a generative model, as specified by the analyst, aligns with the expert's expectations. A Bayesian model comprises a likelihood function denoted as $p(y \mid \theta)$, as well as parametric prior distributions $p(\theta \mid \lambda)$ for the model parameters θ where λ represents the prior hyperparameters to be inferred by our method. Here, y denotes a vector of observations. The prior distributions incorporate prior information into the model. Thus, the degree to which the model captures the expert's expectations relies on the specific values assigned to λ. Consequently, the objective is to identify an appropriate specification of λ that minimizes the discrepancy between the set of elicited statistics from the expert $\{\hat{t}_m\}$ and a corresponding set of elicited statistics derived from the model, $\{t_m\}$, to be discussed next.

First, we need to derive the set of model-implied target quantities $\{z_i\}$. As a target quantity can represent an observable, a parameter, or anything else in between, we define it in the most general form as a function of the model parameters θ, denoted as $z = g(\theta)$, where the function g

can take on various forms and be of deterministic or stochastic nature. In its simplest form, the target quantity directly corresponds to a parameter of interest in the data-generating model ($z_i = g(\theta) = \theta_i$; i.e., g would be a simple projection). Alternatively, g can be aligned with the generative model of the data, resulting in the target quantity being equivalent to the observations ($z_i = g(\theta) = y$).

Moreover, the function g can take on more complex forms. Suppose the domain expert provides prior knowledge about the coefficient of determination R^2. The R^2 measure expresses the proportion of variance explained by the model in relation to the total variance of the response y and is a commonly used to measure model fit in regression models (Gelman et al., 2019). To obtain the corresponding model-implied R^2, we first generate observations y using the specified generative model and then compute the R^2 value from the observations. Given the set of model-implied target quantities, we get the respective model-implied *elicited statistics*, denoted by $\{t_m\}$, by applying the elicitation technique f_j to the target quantity z_i:

$$t_m = f_j(z_i). \tag{2}$$

A challenge with this approach is that the distribution of $\{t_m\}$ may not be analytical or have a straightforward computational solution. For instance, consider the case where the target quantity is equivalent to the observations, $z_i = y$. In this case, the distribution of the predicted observations y gives rise to an integral equation known as the prior predictive distribution (PPD), denoted by $p(y \mid \lambda)$ and defined by averaging out the prior from the generative model:

$$p(y \mid \lambda) = \int_\Theta p(y \mid \theta) p(\theta \mid \lambda) d\theta. \tag{3}$$

Obtaining a closed-form expression for this integral is only feasible in certain special cases, such as when dealing with conjugate priors. This challenge extends to all situations where the target quantity is a function of the observations y. Additionally, as previously mentioned, the *elicited* statistic is defined as a function of the target quantity using an elicitation technique f_j which involves the computation of moments or quantiles, among other possibilities. Consequently, the resulting form of the elicited statistic may once again pose analytic challenges.

Our primary objective is to ensure the broad applicability of our elicitation method to a wide range of models, including those with intractable likelihood functions (Cranmer et al., 2020). To achieve this, we adopt a simulation-based approach that relies solely on the ability to generate samples from the relevant quantities. Bayesian models, by their very formulation, can simulate data from their prior and likelihood distributions, thereby enabling us to generate samples from the Bayesian probabilistic model (Aushev et al., 2023; Gelman et al., 2013). For example, in the case where $z_i = y$, the simulation-based procedure involves two steps: Firstly, we sample the model parameters from the prior distribution conditioned on hyperparameters λ: $\theta^{(s)} \sim p(\theta \mid \lambda)$. Subsequently, we generate data by sampling from the likelihood distribution, resulting in $y^{(s)} \sim p(y \mid \theta^{(s)})$. The superscript (s) is used to denote the s^{th} sample of the corresponding simulated quantity. By repeating these steps, we can generate a collection of S simulations $\{y^{(s)}\} := \{y^{(s)}\}_{s=1}^S$, where each element corresponds to a data point drawn from the PPD

$$y^{(s)} \sim p(y \mid \lambda). \tag{4}$$

This approach is equivalent to Monte Carlo approximation of the integral in the PPD (Eq. 3), since it yields random evaluation points $y^{(s)}$ from the "marginal" distribution $p(y \mid \lambda)$.

2.3 Illustrative Example

In this section, we illustrate our prior elicitation method using a toy linear regression model. The response variable y_n follows a normal distribution with mean μ_n and a fixed residual standard deviation $\sigma = 1.0$. The mean μ_n is predicted by a categorical grouping factor x_n, which takes the value 0 if the observation n belongs to group A, and 1 if it belongs to group B. The model parameters, β_0 (intercept) and β_1 (slope), have normal prior distributions. Our goal is to learn the hyperparameters $\lambda = (\mu_0, \sigma_0, \mu_1, \sigma_1)$ of these priors from expert knowledge.

Consider an analyst selecting three target quantities (z_1, z_2, z_3) for investigation: The first two quantities refer to the predictive group means of group A and B, respectively, and the third target quantity refers to the expected R^2. We assume that the expert possesses an implicit representation of these target quantities, $p(\hat{z}_i)$. In Figure 2, we visually depict each of the three target quantities (z_i) in columns 1-3 (from the left), with the dashed black density function illustrating the expert's implicit representation of the respective target quantity. To access the expert's implicit knowledge about the target quantities, the analyst employs quantile-based elicitation (f_{quan}) as well as moment-based elicitation (here, f_{mean} and f_{sd} for the mean and standard deviation, respectively). For the two predictive group means the analyst queries five quantiles Q_p with $p = 0.10, 0.25, 0.50, 0.75, 0.90$ from the expert, resulting in the elicited (vector-valued) statistics $\hat{t}_1 = f_{\text{quan}}(\hat{z}_1)$ and $\hat{t}_2 = f_{\text{quan}}(\hat{z}_2)$. In Figure 2, the expert's elicited quantiles are visually represented by blue arrows located at the bottom of each panel in columns 1 and 2. For accessing the expert's implicit knowledge about R^2, the analyst queries the expert for the mean M and standard deviation S, resulting in $\hat{t}_3 = f_{\text{mean}}(\hat{z}_3)$ and $\hat{t}_4 = f_{\text{sd}}(\hat{z}_3)$, respectively. The target quantity representing R^2 (i.e., z_3) is illustrated in the third column of Figure 2.

Having acquired the elicited statistics from the expert, the next step involves obtaining the corresponding simulated model-based elicited statistics. This is accomplished by sampling from the forward model for $s = 1, \ldots, S$, as outlined below:

$$\begin{aligned}
\beta_0^{(s)} &\sim \text{Normal}(\mu_0, \sigma_0) \\
\beta_1^{(s)} &\sim \text{Normal}(\mu_1, \sigma_1) \\
\mu_n^{(s)} &= \beta_0^{(s)} + \beta_1^{(s)} x_n \\
y_n^{(s)} &\sim \text{Normal}(\mu_n^{(s)}, \sigma = 1.0).
\end{aligned}$$

We proceed by computing the three model-implied target quantities: the predictive group means, $z_1^{(s)} = \bar{y}_A^{(s)}$ and $z_2^{(s)} = \bar{y}_B^{(s)}$ (obtained by averaging across observations n per group), along with the

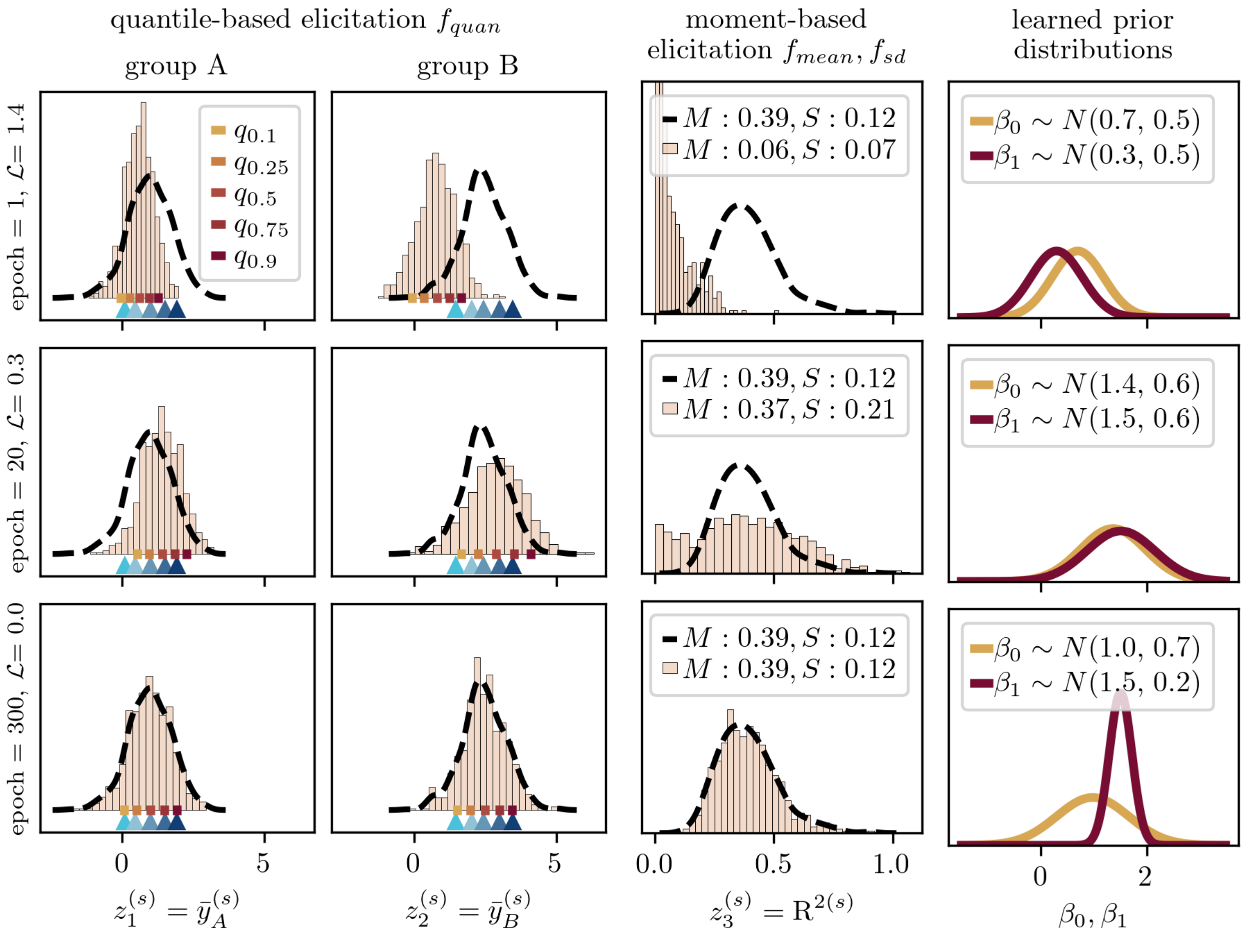

Figure 2: *Visualizing hyperparameter learning based on expert knowledge.* Columns 1 and 2 demonstrate quantile-based elicitation for two target quantities. The expert's implicit representation is shown by the dashed black line, while the simulated model-implied distribution is represented by brown histograms. Blue arrows indicate the expert-elicited quantiles, and brown lines show the model-implied quantiles. Column 3 exemplifies moment-based elicitation for the third target quantity (R^2), with the upper M-S value pair representing the expert elicitation, and the lower value pair the model-implied first moments. Here M denotes mean and S standard deviation. The hyperparameters λ are learned by minimizing the discrepancy between model simulations and expert predictions. The learning progression is shown row-wise, with the first row presenting the initial situation (epoch=1) and in the last row learning has finished. The rightmost column 4 depicts the prior distributions of the model parameters as implied by the learned hyperparameters.

predictive R^2: $z_3^{(s)} = \mathrm{R}^{2(s)}$, whereby the predictive R^2 is computed as the ratio of the variance of the modeled predictive means to the variance of the predictive data, incorporating random noise induced by the (fixed) residual standard deviation σ.

These simulated model-implied target quantities are represented by the brown histograms in the first three columns of Figure 2. Next, we apply the corresponding elicitation techniques to obtain the model-implied elicited statistics: $t_1 = f_{\text{quan}}(z_1^{(s)})$, $t_2 = f_{\text{quan}}(z_2^{(s)})$, and $t_3 = f_{\text{mean}}(z_3^{(s)})$, and $t_4 = f_{\text{sd}}(z_3^{(s)})$. In Figure 2, the model-implied quantiles are visually represented by brownish lines (or markers) at the bottom of each panel in columns 1 and 2. Additionally, the model-implied mean and standard deviation of the R^2 distribution are presented in column 3 by the lower M-S value pair.

When comparing the elicited statistics from the expert, $\{\hat{t}_m\}$, to the model-implied quantities, $\{t_m\}$, we can evaluate the discrepancy between the two. The objective is to learn the hyperparameters λ by minimizing this discrepancy, resulting in model simulations that are more consistent with the expert's predictions. In the first row of Figure 2, the initial situation is depicted, where the simulated model-implied quantities noticeably deviate from the quantities elicited from the expert. The expert-elicited quantiles (blue arrows) clearly differ from the model-implied quantiles (brown markers). Additionally, there is a disparity in the mean and standard deviation of R^2, as presented in column three of Figure 2. As the learning process progresses, the model-implied quantities gradually align with the expert-elicited statistics, as illustrated in the second row of Figure 2. The model simulations start to resemble the expert predictions more closely. In the final row of Figure 2, the learning process has finished and the model-implied quantities align precisely with the expert-elicited statistics. At this stage, the model has effectively learned the hyperparameters λ that result in a high level of agreement with the expert's knowledge. The corresponding prior distributions for the model parameters are presented in the rightmost column of Figure 2.

2.4 Multi-Objective Optimization Problem

Once the elicited statistics $\{\hat{t}_m\}$ from the expert and a procedure to compute the corresponding model-implied elicited statistics $\{t_m\}$ are chosen, the focus can be shifted towards the main objective: Determine the hyperparameters λ that minimize some discrepancy measure (loss function) $L(\lambda)$ between the expert-implied $\{\hat{t}_m\}$ and the model-implied statistics $\{t_m\} = \{t_m(\lambda)\}$, given on the current prior specifications determined by λ. Since the evaluation of the discrepancy extends to all elicited statistics $\{t_m\}$, $L(\lambda)$ has to be formulated as a multi-objective loss function. This loss function encompasses a linear combination of discrepancy measures L_m, with corresponding weights α_m (see Section 2.7). In the following, we will also use the term *loss components* to refer to the individual components in the weighted sum. The selection of the discrepancy measure L_m is contingent upon the elicited statistic, therefore different choices may be appropriate depending on the specific quantity to be compared (see Section 2.6). Independently of these specific choices, our main objective can be written as

$$\lambda^* = \arg\min_{\lambda} L(\lambda) = \arg\min_{\lambda} \sum_{m=1}^{M} \alpha_m L_m(t_m(\lambda), \hat{t}_m), \tag{5}$$

where λ^* denotes the optimal value of the hyperparameters λ given the provided expert knowledge.

2.5 Gradient-based Optimization

The optimization procedure for solving Equation (5) follows an iterative approach. In each iteration, we sample from the generative model, compute the model-implied elicited statistics, and update the hyperparameters λ. This update relies on calculating the gradient of the discrepancy loss with respect to the hyperparameters λ and adjusting them in the opposite direction of the gradient (Goodfellow et al., 2016). The procedure continues until a convergence criterion is met, usually when all elements of the gradient approach zero. We employ mini-batch stochastic gradient descent (SGD) with automatic differentiation, facilitated by the reparameterization trick (explicit or implicit; Kingma & Welling, 2022; Figurnov et al., 2018). In mini-batch SGD, the term *stochastic* indicates that each iteration involves randomly selecting a batch of data from the entire data set. In our case, stochasticity arises naturally as we simulate new model-implied quantities at each iteration step.

The reparameterization trick involves splitting the representation of a random variable into stochastic and deterministic parts. By differentiating through the deterministic part, we can compute gradients with respect to λ using automatic differentiation (Sjölund, 2023). To leverage backpropagation, it is essential that all operations and functions in the computational graph are differentiable with respect to λ. This requirement extends to the loss function and all computational operations in the generative model (Hartmann et al., 2020; da Silva et al., 2019).

However, dealing with discrete random variables poses a challenge due to the non-differentiable nature of discrete probability distributions, making gradient descent through such variables difficult. One approach to overcome this challenge is to use continuous relaxation of discrete random variables, which enables the estimation of gradients and thus the use of gradient-based optimization methods for models that involve discrete random variables (Tokui & Sato, 2016; Maddison et al., 2017; Jang et al., 2017). For instance, both Maddison et al. (2017) and Jang et al. (2017) independently proposed the Gumbel-Softmax trick, which approximates a categorical distribution with a continuous distribution. Let $x \sim \text{Categorical}(\pi)$, where π_i represents the probability of category i among n categories. The Gumbel-Max trick (Gumbel, 1962; Maddison et al., 2014) provides a simple and efficient way to draw samples from a categorical distribution by setting $x_i = \text{one-hot}(\arg\max_i[g_i + \log \pi_i])$, where g_i denotes an iid sample from a $\text{Gumbel}(0, 1)$ distribution (Jang et al., 2017). However, the derivative of the $\arg\max$ is 0 everywhere except at the boundary of state changes, where it is undefined. To overcome this limitation, Jang et al. (2017) and Maddison et al. (2017) independently proposed using the softmax function as a continuous, differentiable approximation to the $\arg\max$ operator, and generate $k-$dimensional vectors in a

$(k-1)$-dimensional simplex Δ^{k-1} where

$$x_i = \frac{\exp\left\{(\log \pi_i + g_i)\tau^{-1}\right\}}{\sum_j \exp\left\{(\log \pi_j + g_j)\tau^{-1}\right\}}. \tag{6}$$

Here, τ denotes the temperature hyperparameter, whereby small temperature values reduce the bias but increase the variance of gradients. In the case of $\tau \rightarrow 0$, the Gumbel-Softmax distribution is equivalent to the categorical distribution (Jang et al., 2017). On the other hand, higher temperature values lead to smoother samples, reducing the variance of gradients but introducing more bias. We observed robust results for $\tau = 1.0$ in our simulations and used it for all case studies.

The Gumbel-Softmax trick is applicable to discrete distributions with a limited (finite) number of categories. Recently, Joo et al. (2020) proposed an extension of the Gumbel-Softmax trick to arbitrary discrete distributions by introducing truncation for those distributions that lack upper and/or lower boundaries. This extension introduces an extra degree of freedom in the form of specifying the truncation level. A wider truncation range brings the approximate distribution closer to the original distribution but also increases the associated computational costs. In our study, we applied the truncation technique in various case studies and specifically investigated the use of different truncation thresholds for the Poisson distribution in Case Study 3 (Subsection 4.4). We observed that the results remain robust across reasonable threshold ranges. For a detailed algorithm outlining the Gumbel-Softmax trick with truncation, please refer to Joo et al. (2020).

2.6 Maximum Mean Discrepancy

A key aspect of the optimization problem, as expressed in Equation (5), is the selection of an appropriate discrepancy measure, L_m. This measure depends on the characteristics of the elicited statistics $\{t_m\}$ and $\{\hat{t}_m\}$. Given that our method entails the generation of $\{t_m\}$ through repeated sampling from the generative model, a loss function is needed that can quantify the discrepancy between samples. The *Maximum Mean Discrepancy* (MMD; Gretton et al., 2006; Gretton, 2017) is a kernel-based method designed for comparing two probability distributions when only samples are available, making it suitable for our specific requirements. We utilize the MMD for all loss components in our applications. This decision is based on the robust simulation results and excellent performance reported in the Case Studies section. That said, our method does not strictly require the MMD, but allows analysts to choose a different discrepancy measure for each loss component, if desired.

Let $x = \{x_1, \ldots, x_n\}$ and $y = \{y_1, \ldots, y_m\}$ be independently and identically distributed draws from the distributions p and q, respectively. The MMD is defined as the distance between the two distributions p and q (Gretton et al., 2006, Definition 2) and can be expressed as follows

$$\text{MMD}[\mathcal{F}, p, q] = \sup_{f \in \mathcal{F}} \left(\mathbb{E}_{x\sim p}[f(x)] - \mathbb{E}_{y\sim q}[f(y)]\right) \tag{7}$$

where $\mathcal{F}$ is the set containing all continuous functions. If this set is a unit ball in the universal *reproducing kernel Hilbert space* (RKHS) $\mathcal{H}$ with associated reproducing kernel $k(\cdot,\cdot)$, the $\text{MMD}[\mathcal{H},p,q]$ is a strictly proper divergence, so it equals zero if and only if $p = q$. Intuitively, $\text{MMD}[\mathcal{H},p,q]$ is expected to be small if $p \approx q$, and large if the distributions are far apart (Gretton, 2017). The (biased) empirical estimate of the squared-MMD is defined as

$$\text{MMD}_b^2[\mathcal{H},p,q] = \frac{1}{n^2}\sum_{i,j=1}^{n} k(x_i,x_j) + \frac{1}{m^2}\sum_{i,j=1}^{m} k(y_i,y_j) - \frac{2}{mn}\sum_{i,j=1}^{m,n} k(x_i,y_j) \tag{8}$$

where $k(\cdot,\cdot)$ is a continuous and characteristic kernel function. One popular choice for a characteristic kernel is the *Gaussian* kernel (Muandet et al., 2017), defined as $k(x,y) = \exp\{-||x-y||^2/(2\sigma^2)\}$ where σ^2 represents the bandwidth. To mitigate sensitivity to different choices of the bandwidth σ^2, a common approach is to use a mixture of G Gaussian kernels that cover a range of bandwidths, given by $k(x,y) = \sum_{i=1}^{G} k_{\sigma_i}(x,y)$ where σ_i denotes the bandwidth for the respective Gaussian kernel (see e.g. Li et al., 2015; Schmitt et al., 2022; Feydy, 2020). An alternative option, as proposed by Feydy (2020) and Feydy et al. (2019), is the *energy distance* kernel $k(x,y) = -||x-y||$, which does not require an extra hyperparameter for tuning. In our simulations, presented in the case studies section, we examined both kernels. However, due to the robustness of the results obtained with the energy kernel, along with its additional advantages of faster learning and the absence of an extra hyperparameter to be specified, we employed the energy kernel exclusively for all experiments presented in this paper.

2.7 Dynamic Weight Averaging

Formulating the multi-objective optimization problem as a weighted sum of loss functions necessitates an important consideration regarding the choice of weights α_m in Equation (5). One possibility is for the user to customize the choice of α_m, signifying the varying degrees of importance for each loss component in a particular application (Deb, 2011). However, another consideration refers to the *task balancing problem*. When employing stochastic gradient descent to minimize the objective outlined in Equation (5), the hyperparameters λ are updated according to the following rule $\lambda \leftarrow \lambda - \delta \sum_{m=1}^{M} \alpha_m \frac{\partial L_m}{\partial \lambda}$, where δ is the step size (i.e., learning rate). The equation suggests that the hyperparameter update may not yield optimal results if one loss component significantly outweighs the others (Deb, 2011).

Consequently, a strategy is needed to dynamically modify the weights α_m to ensure effective learning of all loss components. The *Dynamic Weight Averaging* (DWA) method proposed by Liu, Johns, & Davison (2019) determines the weights based on the learning speed of each component, aiming to achieve a more balanced learning process. Specifically, the weight of a component exhibiting a slower learning speed is increased, while it is decreased for faster learning components (Crawshaw, 2020). Here, we employ a slightly modified variant of DWA, where the weight α_m for loss component m at current step t_{curr} is set as

$$\alpha_m^{(t_{\text{curr}})} = \frac{M \cdot \exp(\gamma_m^{t_{\text{prev}}}/a)}{\sum_{m'=1}^{M} \exp(\gamma_{m'}^{t_{\text{prev}}}/a)} \quad \text{with} \quad \gamma_m^{t_{\text{prev}}} = \frac{L_m^{t_{\text{prev}}}}{L_m^{t_{\text{start}}}}. \tag{9}$$

Here, M is the total number of loss components L_m, t_{prev} indexes the previous iteration step and t_{start} refers to the initial iteration step, γ_m calculates the relative rate of descent, and a is a temperature parameter that controls the softness of the loss weighting in the softmax operator. Setting the temperature a to a large value results in the weights α_m approaching unity. In our case studies (see Section Case Studies), we obtained favorable outcomes when employing a value of around $a = 1.6$, which we utilized consistently across all case studies.

The definition of DWA in Equation (9) includes a slight modification compared to its original formulation, specifically concerning the rate of change of individual loss components, L_m. While Liu, Johns, & Davison (2019) compare the loss ratio between two consecutive iteration steps, we have adopted the definition from the *Loss-Balanced Task Weighting* method proposed by Liu, Liang, & Gitter (2019), where the loss ratio is calculated by comparing the current loss to the initial loss. This modification is motivated by the stochastic fluctuations of the loss value which renders the loss ratio between consecutive steps less informative. However, the loss ratio between the current and initial loss can be a useful proxy for assessing the model's training progress for a given task (Liu, Liang, & Gitter, 2019). While the loss values of two consecutive SGD iterations may exhibit strong fluctuations, the overall trend should follow a decrease in the loss value as the algorithm converges. By comparing the current loss to the initial loss, we can then gain insights into how well a particular task has been learned. Tasks that are poorly learned would have ratios close to 1, indicating their larger contribution to the overall loss and gradient.

3 Related Work

The process of prior elicitation involves the extraction and translation of knowledge from domain experts with the goal to specify appropriate prior distributions for the parameters in probabilistic models. For a comprehensive overview of this field, we refer interested readers to the recent review provided by Mikkola et al. (2023). In prior elicitation, a distinction can be made between techniques for knowledge *extraction*, as reviewed by Garthwaite et al. (2005); O'Hagan et al. (2006); Falconer et al. (2022), and methods for *translating* knowledge into suitable prior distributions. We explicitly aim to contribute to the second.

Historically, elicitation methods have primarily focused on prior elicitation in a model's parameter space, entailing direct inquiries to experts regarding the model parameters (Mikkola et al., 2023). However, there has been a recent shift towards methods that facilitate elicitation in the observable space (Manderson & Goudie, 2023). The current vision of a (yet to be achieved) "gold standard" is an elicitation method that includes both a model's parameter and observable space, exhibits model-agnostic characteristics, and prioritizes sample efficiency to minimize the human effort involved (Mikkola et al., 2023). Taking these desiderata into consideration, our method builds upon recent advancements in prior elicitation, specifically on the works of Hartmann et al. (2020); da Silva et al. (2019), and Manderson & Goudie (2023) who proposed model-agnostic elicitation methods. All three methods have in common that they learn hyperparameters of prior distributions by minimizing the discrepancy between model-implied and expert-elicited statistics. These quantities are associated with the observable space, leading to the model-agnostic characteristic of these approaches. Differences among these methods arise from the specification of target quantities, discrepancy measures, and the specific optimization procedure.

Manderson & Goudie (2023) adopt a two-stage global optimization process that incorporates multi-objective Bayesian optimization, while our approach aligns more with the methods proposed by da Silva et al. (2019) and Hartmann et al. (2020), who employ stochastic gradient-based optimization. However, similarly to our approach, Manderson & Goudie (2023) assume that an expert is queried about observable or model-derived quantities at various quantiles. Their method then uses these elicited statistics to fit a parametric distribution whose cumulative density function becomes the input to the discrepancy measure. In contrast, our method imposes no constraints on the domain of the discrepancy measure; quantiles, histograms, or moments can all serve as inputs.

The utilization of quantile-based elicitation can also be found in Hartmann et al. (2020), where the authors directly model the probability of the outcome data using a Dirichlet process. However, the work by da Silva et al. (2019) is most closely related to our method, as the authors propose a generic methodology that in principal supports an arbitrary choice of target quantities based on the prior predictive distribution and discrepancy measures. They illustrate this by selecting the expected mean and variance as inputs for a discrepancy measure that simultaneously optimizes both. However, it is not apparent how the method can be readily applied to arbitrary target quantities without the need for considerable customization. In contrast, due to the generality of the MMD criterion, our method is broadly applicable to various target quantities without the need for extensive customization. Furthermore, it differs from previous methods by allowing the elicited information to refer to both parameters *and* observables or other model-derived quantities. This flexibility is achieved by treating the model-implied target quantities in their most general form as a function of the model parameters.

Finally, an essential feature of our method is the use of simulations to obtain prior hyperparameter inference, which classifies it as a variant of *simulation-based inference* (SBI), also known as likelihood-free inference (Cranmer et al., 2020). SBI circumvents the analytic intractability of complex models through the use of various simulation-based techniques (e.g., training a deep neural network to recover model parameters from data). Analogously, our method utilizes simulations to approximate the potentially intractable distribution of the target quantities given the hyperparameters.

4 Case Studies

Below, we present four case studies exemplifying the application of our method. Each study concentrates on a distinct statistical model, highlighting specific facets of our approach. To introduce the method, in Case Study 1 we adopt a classic approach: a normal linear regression model with two factors. Proceeding to Case Study 2, we demonstrate the performance of our method for discrete distributions using a binomial distribution with a logit link. Given that the Softmax-Gumbel trick necessitates a double-bounded distribution, we probe the application of our method to a Poisson distribution with varying upper truncation thresholds in Case Study 3. Finally, in Case Study 4, we employ our method within hierarchical models, learning prior distributions for both overall and group-specific (varying) effects under multiple likelihood assumptions as well as partially inconsistent expert information. For an overview of the employed models and the core motivations driving each case study, consult Table 1.

Table 1: *Overview Case Studies.*

Case Study	*Model*	*Core motivation*
CS 1	Normal	Introduce method
CS 2	Binomial	Double-bounded discrete RV
CS 3	Poisson	Lower-bounded discrete RV with upper truncation threshold
		Add-on: Varying truncation thresholds
CS 4	Hierarchical	Normal model incl. varying effects
		Add-on: Replace Normal with Weibull likelihood
		Add-on: Inconsistent expert information

4.1 General Setup

Learning Algorithm In each case study, we utilize mini-batch SGD to learn all model hyperparameters. The optimization process employs the Adam optimizer with an exponential decay. The specific settings, including the initial learning rate, decay rate, and other relevant algorithm parameters, vary across case studies and are fully described in the respective sections. All case studies were implemented in Python, utilizing the *TensorFlow* library (Abadi et al., 2016), and optimization was performed on a CPU-only machine. All code and material can be found in OSF `https://osf.io/rxgv2`.

Ideal Expert Model In the case studies presented here, we employ an expert model to simulate responses, rather than querying real experts. This choice is motivated by our aim to demonstrate the *validity* of our method. By *validity*, we refer to our method's ability in accurately recovering the "true" hyperparameter values. This objective presupposes our knowledge of these true hyperparameter values, which is possible by the use of expert simulations. An important future step will involve assessing our method with actual experts.

In this paper, we introduce in each case study a specific data-generating model corresponding to an exemplary application scenario. The expert model mirrors this data-generating model precisely, with a distinct set of hyperparameter values λ^* that fully defines the statistical model. Consequently, we assume that the expert has exact knowledge of the generative model, which is why we refer to it as the "ideal" expert. The specific values of λ^* are introduced in each case study. From this ideal expert model, we simulate all predefined target quantities (discussed next) and compute the corresponding elicited statistics. For instance, we simulate a distribution of group means (target quantity) and derive the corresponding quantiles (elicited statistic using quantile-based elicitation). These elicited quantiles serve as input, representing the "expert knowledge", for our method to learn the hyperparameters λ. It is important to recognize that the elicited statistics are generally interrelated since they encompass information about the same aspects in the data. Employing an ideal expert leads to the derived expert-elicited statistics always being mutually consistent. However, this assumption might be less realistic in a real-world context. Therefore, in Case Study 4, we explicitly explore our method's behavior when dealing with inconsistent expert information. The expert data is simulated only *once before* the learning process, reflecting a one-shot elicitation scenario where the expert is queried only once. The use of the ideal expert approach enables us to assess the validity of our method by evaluating its ability to recover the true hyperparameter values λ^*.

Selection of Target Quantities & Elicited Statistics In the selection of target quantities and elicited statistics, we consider several key aspects. First, we aim to represent the variability of target quantities, which includes diverse statistics (e.g., R^2), predictions for the outcome variable (e.g., predictions for specific design points), and information about model parameters. Second, we account for the variability of elicitation techniques by employing histogram, quantile-based, and moment-based elicitation to compute the elicited statistics. Third, we prioritize interpretability by choosing target quantities that can be readily understood by experts in realistic scenarios. Finally, the selected number of elicited statistics should guarantee model identification. Addressing the issue of identification is crucial for our objective of assessing the validity of our method through the successful recovery of the true hyperparameters λ^*. To achieve this, the model must possess sufficient relevant information to uniquely identify the hyperparameter values. Otherwise, a failure to recover the hyperparameters could be attributed to either methodological limitations or insufficient information available from the elicited statistics. In our case studies, we strive to achieve perfect model identification by carefully selecting a "sufficiently large" set of queried quantities. Specifically, in the quantile-based elicitation, we query the expert with respect to nine quantiles Q_p with $p = (0.1, 0.2, \ldots, 0.9)$. For moment-based elicitation, the expert provides the mean M and standard deviation S of the target quantity. In histogram elicitation, the expert is asked to provide a histogram comprising S_E observations, with the specific value of S_E specified in each case study.

We acknowledge that in practical applications of elicitation methods, a trade-off arises between the desired information sought from the expert to achieve model identification and the limitations imposed by the expert's available resources and knowledge. Since the goal of the case studies is to establish the validity of our method, our focus lies on achieving *complete* prior hyperparameter identification. In Section 5, we will discuss future directions and steps, including the exploration of settings where prior hyperparameters are not fully identified by the provided expert information.

4.2 Case Study 1: Normal Linear Regression

Setup In the first case study, we present our method using a normal linear regression model, along with an example from the field of social cognition. Specifically, we focus on an experiment conducted by Unkelbach & Rom (2017), where participants are presented with general knowledge statements in two consecutive phases. During the second phase, they are required to indicate whether each statement is true or false. The response variable of interest is the proportion of true judgments (PTJs). The main objective of the study is to investigate the influence of two factors on PTJs: (1) repetition (ReP), which involves presenting some statements from the first phase again in the second phase, and (2) encoding depth (EnC), where participants are divided into groups varying in the level of elaboration required for the sentences in the first phase. We employ a 2 (ReP: *repeated, new*) × 3 (EnC: *shallow, standard, deep*) between-subject factorial design with treatment contrasts used for both factors. The baseline levels are set as new for ReP and shallow for EnC. Following Unkelbach & Rom (2017), we employ a linear model to describe the data-generating process [1]

$$\begin{aligned}
y_i &\sim \text{Normal}(\theta_i, s) \\
\theta_i &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 \\
\beta_k &\sim \text{Normal}(\mu_k, \sigma_k) \quad \text{ for } k = 0, \ldots, 5 \\
s &\sim \text{Exponential}(\nu).
\end{aligned} \tag{10}$$

Here, y_i represents the responses for each observation $i = 1, \ldots, N$, assumed to follow a normal distribution with mean θ_i and standard deviation s. The expected value θ_i is modeled as a linear function of two categorical predictors, ReP and EnC. The regression coefficients β_k for $k = 0, \ldots, 5$ are assigned normal prior distributions. Specifically, β_0 represents the PTJs for new statements in the shallow-encoding condition, while β_1 represents the difference in PTJs (ΔPTJ) between repeated and new statements. Additionally, β_2 captures the ΔPTJ for new statements in the shallow- vs. standard-encoding condition, β_3 represents the ΔPTJ for new statements in the shallow- vs. deep-encoding condition, and β_4 and β_5 account for the interaction effects between ReP and EnC.

[1] We acknowledge that the linear model might not be the most suitable model for modelling proportions. Nevertheless, we adopt the original approach of the authors for the sake of illustrating our methodology. Our primary objective centers on the application of our method to learn the hyperparameters, with this specific example serving as an illustration rather than an attempt to capture the data's characteristics perfectly.

The standard deviation s of the normal likelihood follows an Exponential prior distribution with rate parameter ν.[2] Our method will learn the hyperparameters $\lambda = (\mu_k, \sigma_k, \nu)$ based on expert knowledge.

Elicitation procedure For this example, we assume that the analyst elicits the following target quantities from the expert: (1) the expected PTJ for the marginal distribution of factor EnC, (2) as well as of factor ReP, (3) the expected ΔPTJ between repeated and new statements for each EnC level, (4) the expected R^2 defined as $R^2 = \text{var}(\theta_i)/\text{var}(y_i)$[3], and (5) the expected overall (grand) mean. To elicit this information from the expert, the analyst uses quantile-based elicitation for (1-3) and histogram elicitation for (4-5). To model the ideal expert, we assume the following "true" hyperparameters $\lambda^* = (\mu_0 = 0.12, \sigma_0 = 0.02, \mu_1 = 0.15, \sigma_1 = 0.02, \mu_2 = -0.02, \sigma_2 = 0.06, \mu_3 = -0.03, \sigma_3 = 0.06, \mu_4 = -0.02, \sigma_4 = 0.03, \mu_5 = -0.04, \sigma_5 = 0.03, \nu = 9.00)$.

Figure 3 depicts the expert's implicit representation of each *target* quantity (black dashed line) as well as the corresponding elicited statistics: In column 1-3 the expert's elicited quantiles are indicated by the blue arrows and in column 4 the elicited histograms are depicted in blue.

Optimization Once the elicited statistics from the expert are obtained, we can proceed with the model simulations and the optimization procedure. First, the hyperparameters λ are randomly initialized to enable simulations from the forward model and computing the corresponding model-implied target quantities as well as elicited statistics. In Figure 3 the model-implied target quantities and elicited statistics based on the final learned hyperparameters λ are visualized by brown histograms and brownish lines/markers, respectively. Second, we determine the hyperparameters of the learning algorithm, which we will refer to as *algorithm parameters* to avoid confusion with the term *prior hyperparameters* λ. The algorithm parameters include the batch size (B), the number of epochs (E), the number of samples from both the expert and generative model (S_E and S_M, respectively) and the initial as well as minimum learning rates (ϕ^0 and $\phi^{\min}$) for the learning rate schedule used with the Adam optimizer. The respective algorithm parameters for each model (i.e., case study) are summarized in the appendix. For the current example the specification of the algorithm parameters can be found in Appendix B.1.

The learning process is considered completed once the maximum number of epochs has been reached. To assess whether learning was successful, we first check the *convergence diagnostics* as summarized in Figure 4. A preliminary convergence check is conducted by examining the loss function, as shown in the leftmost column. The total loss demonstrates the desired decreasing behavior, which is also observed for the individual loss components in the panel below. An additional convergence check involves analyzing the gradients. The norm of the individual gradients

[2] In our implementation, we employ a reparameterization trick for $s \sim \text{Exponential}(\nu)$, achieved by expressing a prior on the mean of N replicated s as $1/N \sum_n s_n \sim \text{Gamma}(N, N \cdot \nu)$. Our simulation results demonstrate improved learning behavior for this alternative formulation.

[3] Here $\text{var}(\theta_i)$ is the variance of the modeled predictive means and $\text{var}(y_i)$ the variance of the predictive observations including the residual variance.

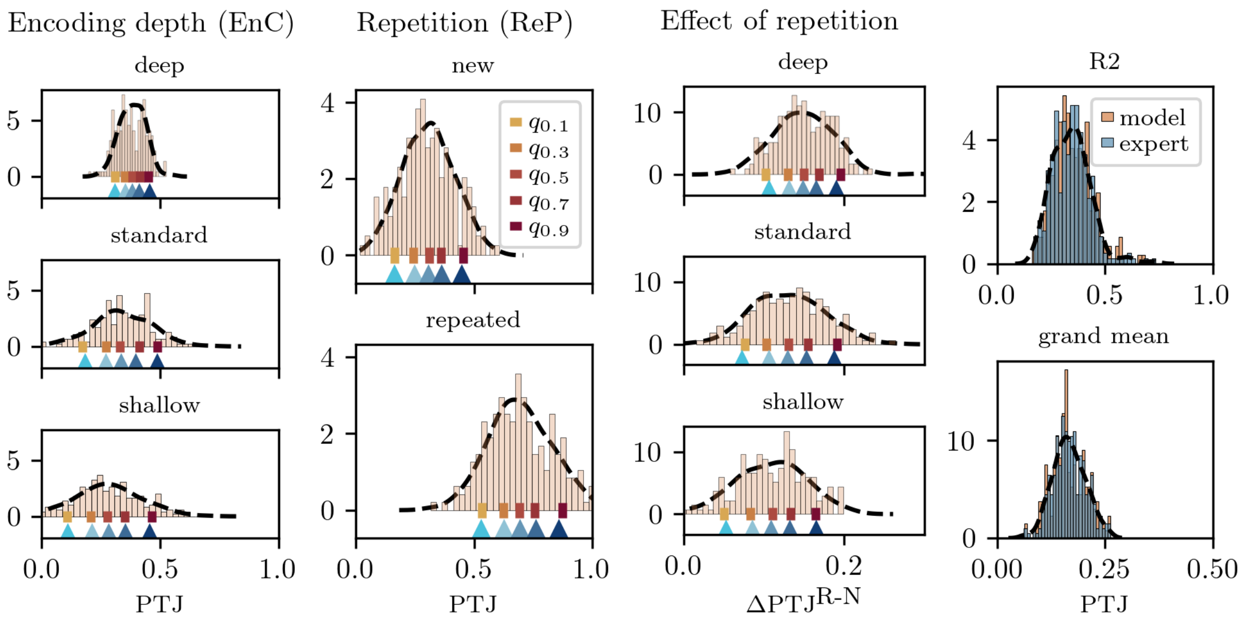

Figure 3: *Model-based and expert-elicited target quantities.* The first three columns (from the left) depict all target quantities obtained through quantile-based elicitation from the expert. To enhance clarity, we have presented only a subset of all model-implied and expert-elicited quantiles, specifically Q_p, with $p = (0.1, 0.3, 0.5, 0.7, 0.9)$. The last column depicts the two target quantities that were queried from the expert using histogram elicitation. All simulated model-implied quantities are based on the final learned hyperparameters λ. For a detailed explanation on interpreting each graphical element, please refer to the Illustrative Example in Section 2.3.

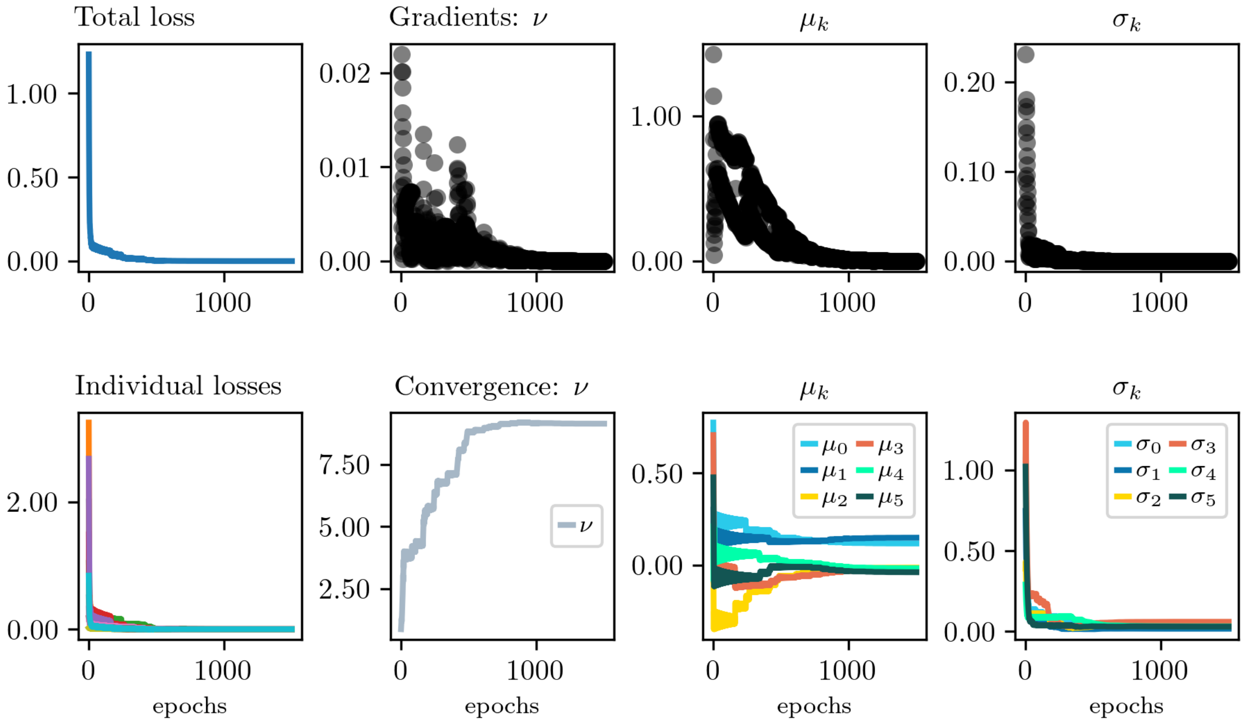

Figure 4: *Convergence diagnostics for Case Study 1.* The leftmost column represents the loss value across epochs, demonstrating the desired decreasing trend of all loss values, the total loss as well as the individual loss components, as depicted in the lower row. The upper three right panels display the expected decreasing trend towards zero of the gradient norm for each learned hyperparameter λ. The three lower right panels illustrate the update values of each learned hyperparameter after each iteration step (epoch), stabilizing in the long run at a specific value.

distributed over time (epochs) is presented in the upper, right row and demonstrates the expected behavior of the gradient norms decreasing towards zero. Furthermore, the convergence of each hyperparameters λ during the learning process towards a specific value is illustrated in the lower, right row.

Results After confirming successful convergence, we shift our focus to the simulation results. Referring back to Figure 3, the simulated model-based quantiles (brown markers), obtained by using the final learned hyperparameters λ, align perfectly with the expert-elicited quantiles indicated by the blue arrows. Additionally, the simulated model-based histograms depicted in brown in the rightmost column, precisely match the blue histograms obtained from the expert. Thus, simulation results indicate a close match between the prior predictions obtained from our final learned model and the expert's elicited statistics. This is further supported by the visualization of the final learned

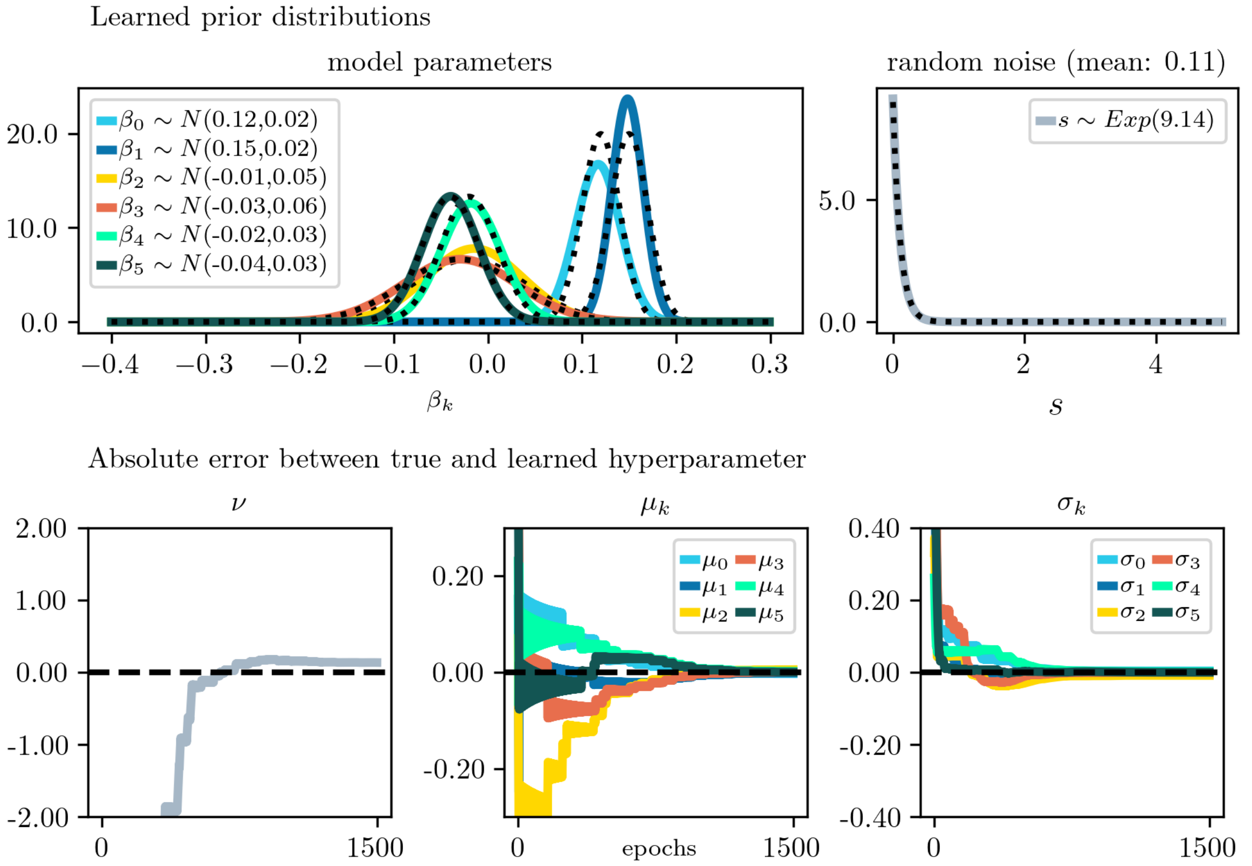

Figure 5: *Results of Case Study 1.* The upper row presents the final learned prior distributions per model parameter β_k and s. The dotted lines indicate the true priors according to the ideal expert, while the solid colored lines represent the learned priors. The lower row depicts the error between the learned and true hyperparameter values (ν, μ_k, σ_k) over time. The error diminishes towards zero across epochs indicating excellent hyperparameter recovery.

prior distributions in Figure 5. In the upper row of Figure 5, the *learned*[4] prior distributions are depicted with solid lines, while the *true* priors (according to the ideal expert) are represented by dotted lines. The substantial overlap between these distributions indicates a successful learning process. This is further emphasized in the second row, where the error between the learned and true hyperparameter values gradually decreases towards zero.

The case study illustrates that our method effectively recovers true hyperparameters in a linear model, which is a special case within the broader framework of *generalized linear models* (GLMs). In GLMs, a non-linear function can be used to transform the expected value, accounting for natural range restrictions in parameters like probabilities or counts. In the upcoming case studies we further explore the method's performance across various response distributions and link functions.

4.3 Case Study 2: GLMs – Binomial Model

Setup In Case Study 2 we utilize a Binomial response distribution with a logit-link function for the probability parameter. As an accompanying example, we use the Haberman's survival dataset from the UCI machine learning repository (Dua & Graff, 2017). The dataset contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer. In the following, we use the detected number of axillary lymph nodes that contain cancer (i.e., (positive) axillary nodes) as numerical predictor X which consists in total of 31 observations ranging between 0 and 59 axillary nodes. The dependent variable y is the number of patients who died within five years out of $T = 100$ trials for each observation $i = 1, \dots, N$. We consider a simple Binomial regression model with one continuous predictor

$$\begin{aligned} y_i &\sim \text{Binomial}(T, \theta_i) \\ \text{logit}(\theta_i) &= \beta_0 + \beta_1 x_i \\ \beta_k &\sim \text{Normal}(\mu_k, \sigma_k) \quad \text{for} \quad k = 0, 1. \end{aligned} \tag{11}$$

The probability parameter θ_i is predicted by a continuous predictor x with an intercept β_0 and slope β_1. We assume normal priors for the regression coefficients, with mean μ_k and standard deviation σ_k for $k = 0, 1$. Through the logit-link function, the probability θ_i is mapped to the scale of the linear predictor. The objective is to learn the hyperparameters $\lambda_k = (\mu_k, \sigma_k)$ based on expert knowledge.

Elicitation Procedure & Optimization The analyst selects as target quantities the expected number of patients who died within five years for different numbers of axillary nodes x_i, with $i = 0, 5, 10, 15, 20, 25, 30$. Furthermore, we define the ideal expert by the following true hyperparameters $\lambda^* = (\mu_0 = -0.51, \sigma_0 = 0.06, \mu_1 = 0.26, \sigma_1 = 0.04)$. The specification of the algorithm parameters for the optimization procedure can be found in Appendix B.2. The convergence diag-

[4] The final, learned hyperparameter λ is the average over the last 30 epochs.

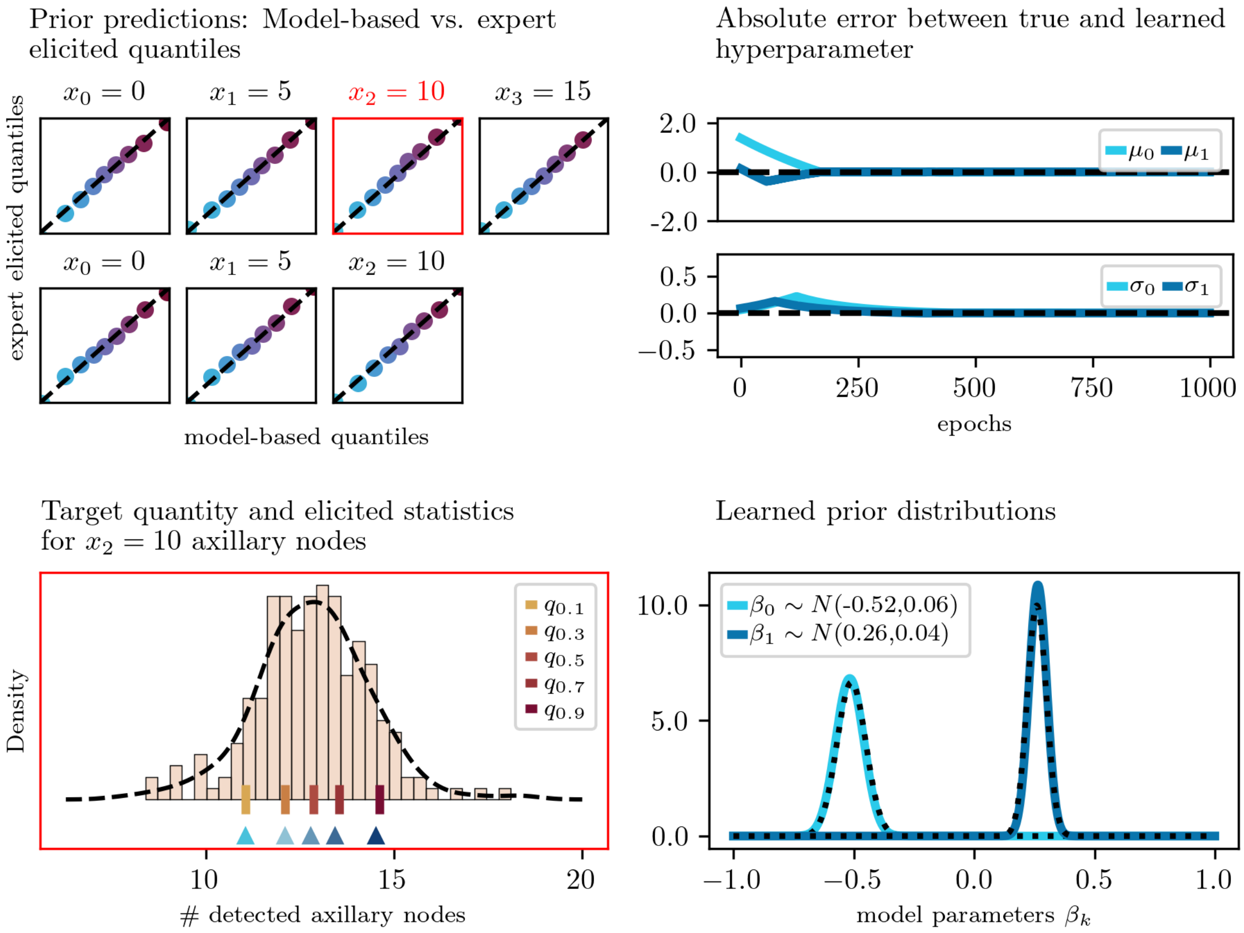

Figure 6: *Results of Case Study 2*: The upper left panel presents a comparison between model-based (x-axis) and expert-elicited quantiles (y-axis) for each x_i, where $i = 1, \dots, 7$, whereby the model simulations are based on the final learned hyperparameter λ. The lower left panel highlights x_2 (in red) and includes not only the quantiles (elicited statistic) but also the corresponding target quantity (for interpretation, please refer to the Illustrative Example in Section 2.3). The two upper right panels illustrate the learning of hyperparameters across epochs, showcasing the difference between the true and learned values. The lower right panel displays the true (dotted line) and final learned (solid line) prior distributions of the model parameters. Overall, the simulation results demonstrate a successful recovery of the true hyperparameters.

nostics check follows the same procedure as discussed for Case Study 1, and showed successful convergence. A summary of the diagnostic results can be found in Appendix B.2.1.

Results The simulation results, based on the final learned hyperparameters λ, are presented in Figure 6. The upper left panel shows a comparison between the expert-elicited quantiles for each queried number of axillary nodes x_i, revealing an almost perfect match between both quantities. For x_2 (highlighted in red), the panel below presents not only the quantiles (blue arrows for the expert and brown lines for the model) but also the distribution of the target quantity. The expert's implicit representation of the target quantity is shown as a dashed black line, and the model-based simulation is displayed as a brown histogram. Moving to the upper two right panels, it becomes evident that the error between the true and learned hyperparameters tends towards zero. Additionally, the lower right panel presents the true (dotted line) and final learned (solid line) prior distributions. Overall, the results indicate excellent performance of our method for a Binomial regression model.

4.4 Case Study 3: GLMs – Poisson Model

Setup In Case Study 3, we extend our examination of count data likelihoods, with a specific focus on the Poisson distribution. Unlike the Binomial distribution, the Poisson distribution lacks an upper bound. This distinction becomes important, since we employ the Gumbel-Softmax trick during gradient-based learning of discrete random variables (see Section 2.5 for details). The Gumbel-Softmax trick treats discrete distributions as categorical distributions with a finite number of categories, necessitating a double-bounded count distribution. To meet this requirement, we adopt the approach proposed by Joo et al. (2020) and introduce a truncation threshold t^u as the upper bound for the Poisson distribution. In the following section, we illustrate and discuss the implications of various truncation thresholds on learning performance. Importantly, we demonstrate the robustness of our results for different (reasonable) choices of t^u.

For demonstration purposes, we adapt an example from Johnson et al. (2022), which investigates the number of LGBTQ+ anti-discrimination laws in each US state. The distribution of these laws is assumed to follow a Poisson distribution, with the rate of such laws being influenced by demographic and voting trend. The demographic trend is quantified by the percentage of a state's residents living in urban areas, ranging from 38.7% to 94.7%. Additionally, the voting trend is represented by historical voting patterns in presidential elections, categorizing each state as consistently voting for the Democratic or Republican candidate or being a Swing state. We employ a Poisson regression model including one treatment-coded categorical predictor: the *voting trend*. This predictor has three levels: Democrats, Republicans, and Swing, with Democrats serving as the reference category. Furthermore, the model incorporates one continuous predictor: the *demo-*

graphic trend, measured as a percentage. The Poisson regression model is represented as follows:

$$\begin{aligned} y_i &\sim \text{Poisson}(\theta_i) \\ \log(\theta_i) &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 \\ \beta_k &\sim \text{Normal}(\mu_k, \sigma_k) \quad \text{for } k = 0, \ldots, 3. \end{aligned} \tag{12}$$

Here, y_i is the number of counts for observation $i = 1, \ldots, N$. The counts follow a Poisson distribution with rate θ_i. The rate parameter is predicted by a linear combination of two predictors: the continuous predictor x_1 with slope β_1 and a three-level factor represented by the coefficients β_2 and β_3 for both contrasts x_2 and x_3. The logged average count y is denoted by β_0. All regression coefficients are assumed to have normal prior distributions with mean μ_k and standard deviation σ_k for $k = 0, \ldots, 3$. The log-link function maps θ_i to the scale of the linear predictor. The main goal is to learn the hyperparameters $\lambda = (\mu_k, \sigma_k)$ based on expert knowledge.

Elicitation procedure To elicit this knowledge, the analyst queries the expert regarding the following target quantities: the predictive distribution of the group means for states categorized as Democrats, Republicans, and Swing, and the expected number of LGBTQ+ anti-discrimination laws for selected US states x_i, where $i = 1, 11, 17, 22, 35, 44$. Quantile-based elicitation is used for the distribution of group means, while histogram elicitation is utilized for the observations per US state. Furthermore, the expert is queried about the expected maximum number of LGBTQ+ anti-discrimination laws in one US state. This information is essential for setting the truncation threshold for the upper bound of the Poisson distribution t^u. To account for higher values than expected and reduce potential biases resulting from excluding relevant information, t^u is set twice as high. In the current example, the truncation threshold is set to $t^u = 110$. The ideal expert is defined by the following true hyperparameters $\lambda^* = (\mu_0 = 2.91, \sigma_0 = 0.07, \mu_1 = 0.23, \sigma_1 = 0.05, \mu_2 = -1.51, \sigma_2 = 0.135, \mu_3 = -0.61, \sigma_3 = 0.105)$. The specification of algorithm parameters for the optimization procedure as well as a Figure summarizing the convergence diagnostics can be found in Appendix B.3.

Results The learned hyperparameters' results are presented in Figure 7. In the upper right panel, a comparison between model-based and expert-elicited statistics is presented. The level of agreement between model predictions and expert expectations is high for both elicitation formats: quantile-based elicitation for the voting groups in the first row and histogram elicitation for single states in the second and third rows. Exemplary for the voting group *Swing* (highlighted in red), the lower left panel provides not only the quantiles but also the distribution of the target quantity. The expert's implicit representation of the target quantity is shown as the dashed black line, while the model-based simulations are depicted as the brown histogram. The elicited quantiles are displayed at the bottom of the panel, with brown representing model-based quantiles and blue representing expert elicited quantiles. The upper right panel further demonstrates the success of the learning process, as the error between learned and true hyperparameter values converges towards zero over time.

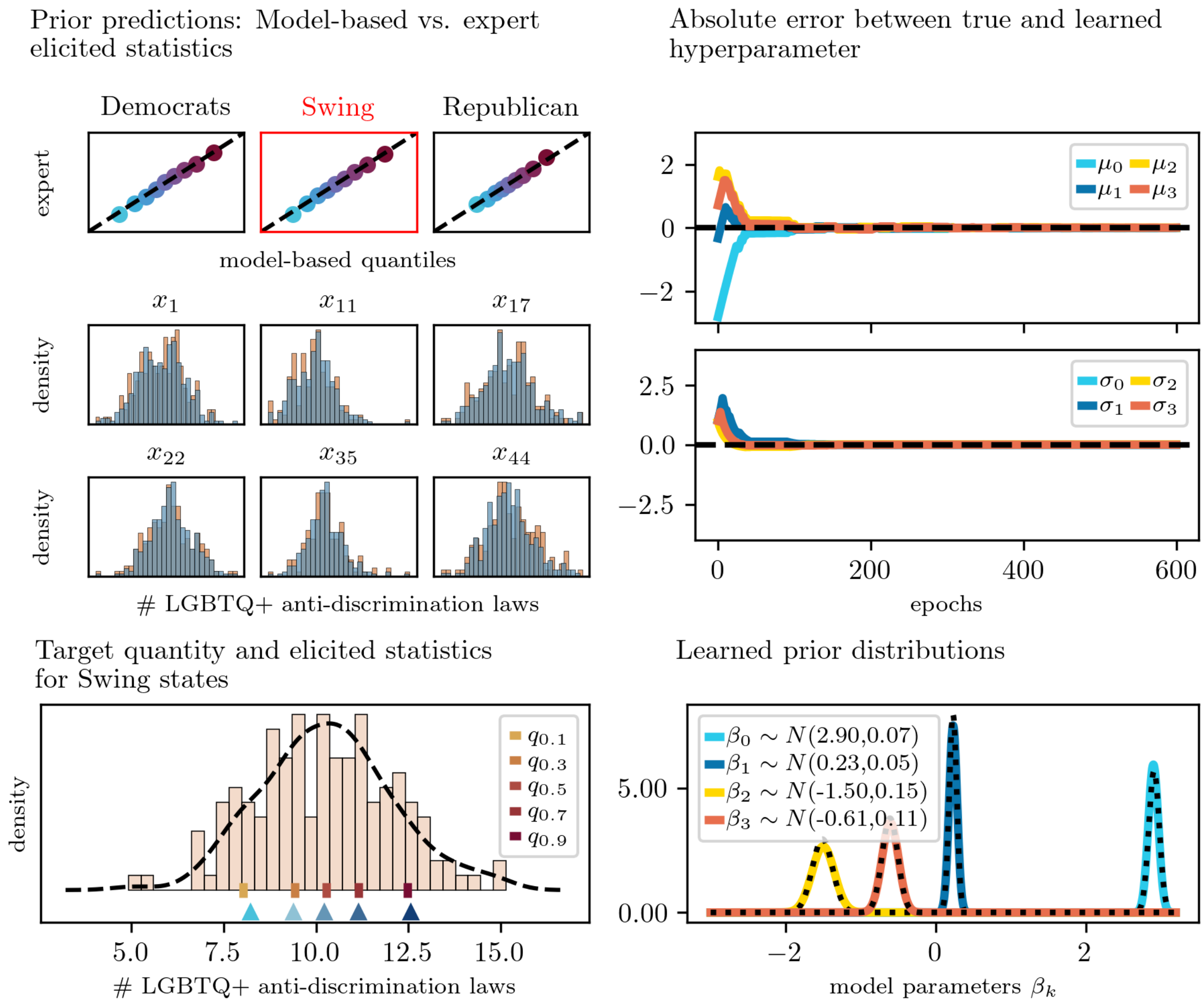

Figure 7: *Results of Case Study 3*: The upper left panel presents a comparison between model-based and expert-elicited statistics, whereby the model simulations are based on the final learned hyperparameter λ. The first row depicts quantile-based elicitation and the others histogram elicitation for each state x_i. The lower left panel shows for the voting group *Swing* as an example, the distribution of the target quantity and the corresponding elicited quantiles (for interpretation of the depicted details, please refer to the Illustrative Example in Section 2.3). The two upper right panels illustrate the learning of hyperparameters across epochs, showcasing the difference between the true and learned values. The lower right panel displays the true (dotted line) and final learned (solid line) prior distributions of the model parameters. Overall, the simulation results demonstrate a successful recovery of the true hyperparameters.

The final prior distributions are depicted in the lower right panel, with solid lines representing the learned priors and dotted lines the true priors. Overall, the simulation results for the Poisson model exhibit excellent performance, showcasing the effectiveness of our method for models of count data without natural bounds.

Examination of varying threshold values We proceed by examining the sensitivity of our results to varying threshold values t^u for the upper bound of the Poisson distribution. We start the examination by first simulating the *true* distribution of law counts for each of the 49 US states to gain insights into the underlying distribution of the expected count values. Figure 8 displays the distribution per US state in brown, while the black density line represents all count values together, disregarding the grouping by US state. It is evident that the majority of simulated count values fall within the range of $[0, 30)$. To comprehensively examine the model's behavior, we consider different scenarios with thresholds that clearly underestimate, overestimate, or match the simulated data. The selected threshold values are: $t^u = 5, 15$ (clear underestimation), $t^u = 30, 55$ (more or less approximate to the actual simulated threshold), and $t^u = 110, 210$ (clear overestimation). The dotted vertical lines in Figure 8 depict the thresholds $t^u = 5, 15, 30$, and 55. Higher threshold values are excluded from visualization for clarity.

A graphical representation of the simulation results is provided in Appendix 7. The *accuracy* of results is evaluated through the absolute error between the learned hyperparameters (λ) and the true hyperparameters (λ^*). Underestimation of the threshold ($t^u = 5, 15$) leads to increasing inaccuracies as the model does not consider the full range of possible count values. However, despite the clear underestimation when $t^u = 15$, the error in accuracy remains relatively minor. A threshold that matches the simulated data ($t^u = 30$) results in an error approaching zero, indicating successful learning. Successful learning is also observed for slight or relatively high overestimation of the upper threshold ($t^u = 55$ and 110). However, substantial overestimation of the maximum threshold ($t^u = 210$) leads to decreased accuracy.

Efficiency of learning is evaluated through the run time for a single epoch in seconds. The run time tends to increase with higher threshold settings, but this increase is gradual within the range of considered t^u values. Overall, the simulation results demonstrate that the learning algorithm performs well under varying threshold settings and remains robust against a certain degree of under- and overestimation of the actual threshold. However, significant misspecification of the threshold setting leads to reduced performance, affecting both accuracy and efficiency.

4.5 Case Study 4: Hierarchical Models

Setup In this concluding case study, we investigate the performance of our elicitation method when applied to hierarchical models. This specific model class poses a distinct challenge for analysts and domain experts alike due to the inherent complexity of the model and the non-intuitive nature of varying effects (i.e., varying intercepts and slopes). Consequently, setting expectations for these model parameters from the perspective of domain experts becomes a difficult task. This

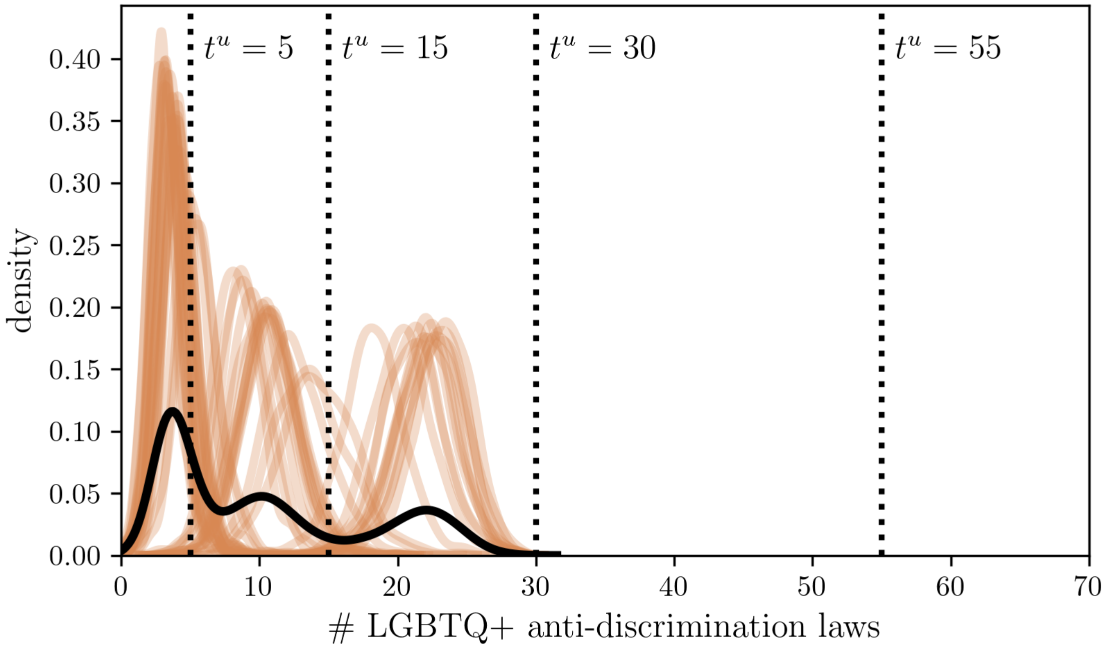

Figure 8: *Simulated distributions of law counts and threshold selection.* For each of the 49 US states, the simulated distribution of law counts based on the true hyperparameters λ is depicted in brown. The black density function represents all simulated observations, disregarding the grouping factor of US states. Dotted vertical lines indicate different selected threshold values, with $t^u = 5$ and 15 considerably underestimating the upper threshold, $t^u = 30$ precisely depicting the upper bound, and $t^u = 55$ overestimating the upper threshold. The method was executed with all thresholds (as well as $t^u = 110, 210$) and the results were evaluated with respect to run time and accuracy of learned λ.

complexity is especially pronounced when link functions are employed that require the interpretation of parameters on a scale different from the outcome variable. To illustrate this aspect, we first introduce a hierarchical model with a normal likelihood and an identity link. Subsequently, we transition from the normal to a Weibull likelihood with a log link. We demonstrate that we can use the same expert-elicited statistics to learn the hyperparameter λ on different scales. To the best of our knowledge, our elicitation method is the first to learn prior distributions for model parameters within a hierarchical framework, relying on expert knowledge that is articulated in terms of interpretable target quantities.

The accompanying example in this case study draws inspiration from the *sleepstudy* dataset (Belenky et al., 2003) that comes along with the R-package *lme4* (Bates et al., 2015). This dataset contains information about the average reaction time (RT) in milliseconds for N individuals who undergo sleep deprivation for nine consecutive nights (with less than three hours of sleep per night). In order to construct a model for this data, we consider a hierarchical model with days serving as a continuous predictor denoted as x,

$$\begin{aligned}
y_{ij} &= \text{Normal}(\theta_{ij}, s)\\
\theta_{ij} &= \beta_{0,j} + \beta_{1,j}x_{ij}\\
\beta_{0,j} &= \beta_0 + u_{0,j}\\
\beta_{1,j} &= \beta_1 + u_{1,j}\\
(u_{0,j}, u_{1,j}) &\sim \text{MvNormal}\left(\mathbf{0}, \Sigma_u\right)\\
\Sigma_u &= \begin{pmatrix} \tau_0^2 & \rho_{01}\tau_0\tau_1 \\ \rho_{01}\tau_0\tau_1 & \tau_1^2 \end{pmatrix}\\
\beta_k &\sim \text{Normal}(\mu_k, \sigma_k) \qquad \text{for } k = 0, 1\\
\tau_k &\sim \text{TruncatedNormal}(0, \omega_k) \qquad \text{for } k = 0, 1\\
\rho_{01} &\sim \text{LKJ}(\alpha_{\text{LKJ}})\\
s &\sim \text{Exponential}(\nu).
\end{aligned} \tag{13}$$

Here y_{ij} represents the average RT for the j^{th} participant at the i^{th} day with $j = 1, \ldots, 100$[5] and $i = 0, \ldots, 9$. The RT data is assumed to follow a normal distribution with local mean θ_{ij} and within-person standard deviation s. Here, θ_{ij} is predicted by a linear combination of the continuous predictor x with overall slope β_1 and intercept β_0. Given the potential variation in both baseline and change in RT across participants, the model incorporates varying (i.e., "random") intercepts $u_{0,j}$ and varying slopes $u_{1,j}$. These varying intercepts and slopes follow a multivariate normal distribution, centered at a mean vector of zero and with a covariance matrix Σ_u. This encodes the variability (τ_0, τ_1) and the correlation (ρ_{01}) between $u_{0,j}$ and $u_{1,j}$. For the resulting set of model parameters, the

[5] The original *sleepstudy* dataset comprises only 18 participants. However, our objective is to assess the validity of our elicitation method by recovering each model (hyper)parameter accurately. To achieve this goal and capture the precise variability indicated by the varying effects, it was necessary to employ a larger sample size.

following prior distributions are assumed: A normal distribution for the overall (i.e., "fixed") effects β_k $(k = 0, 1)$ with mean μ_k and standard deviation σ_k. A truncated normal distribution centered at zero with a standard deviation of ω_k, is employed for the person-specific variation τ_k, which is constrained to be positive. The correlation parameter ρ_{01} follows a Lewandowski-Kurowicka-Joe (LKJ; Lewandowski et al., 2009) distribution with scale parameter α_{LKJ}. In the subsequent context, we set α_{LKJ} to 1. Additionally, an Exponential prior distribution with rate ν is used for the within-person (error) standard deviation s. The goal is to learn the hyperparameters $\lambda = (\mu_k, \sigma_k, \omega_k, \nu)$ based on expert knowledge.

Elicitation procedure & Optimization We elicit expert knowledge within both the parameter and observable space, underscoring the inherently *hybrid* nature of our elicitation method. The analyst queries the expert regarding the following target quantities: the expected average RT for specific days x_i, where $i = 0, 2, 4, 6, 8$, the within-person standard deviation s, and the expected distribution of R^2 for the initial and final days $(i = 0, 9)$. We assume that the analyst employs quantile-based elicitation for the expected average RT per chosen day x_i. For the within-person standard deviation s, a moment-based elicitation approach is used, asking the expert regarding both the expected mean and standard deviation of s. Additionally, histogram-elicitation is utilized to query the expected R^2 distribution for each day. To determine the true hyperparameter values of the ideal expert, we fitted a corresponding hierarchical model in *brms* (Bürkner, 2017) using the *sleepstudy* dataset and used the estimated parameter values as a reference point. Consequently, the ideal expert is defined by the following true hyperparameters: $\lambda^* = (\mu_0 = 250.40, \mu_1 = 30.26, \sigma_0 = 7.27, \sigma_1 = 4.82, \omega_0 = 33.00, \omega_1 = 23.00, \nu = 0.04)$. Please refer to Appendix B.4 for detailed information about the algorithm parameters of the optimization procedure together with a figure summarizing the convergence diagnostics indicating successful convergence.

Results Figure 9 presents the results derived from the optimization process. The upper left quadrant depicts the congruence between simulation-based and expert-elicited statistics, effectively highlighting successful learning. The first and second rows illustrate the alignment between expert-derived quantiles for the six chosen days x_i and the corresponding simulated quantiles generated by the final trained model. Furthermore, a perfect match between the distributions of R^2 as indicated by the expert and predicted by the model for day 0 and 9 is evident in the first two panels of the last row. Finally, the queried moments (i.e., mean and standard deviation) for the model parameter s, which signifies the within-person standard deviation, exhibit perfect correspondence between expert expectations and model simulations. This correspondence can be seen in the lower right plot, characterized by the diagonal alignment of the red and blue scatter points, representing the mean and standard deviation values of s, respectively.

The finally learned prior distributions for each model parameter are depicted in the lower row of Figure 9. The high overlap between true (dashed lines) and learned (solid lines) prior distributions indicates an additional instance of successful learning. This is further supported by the assessment

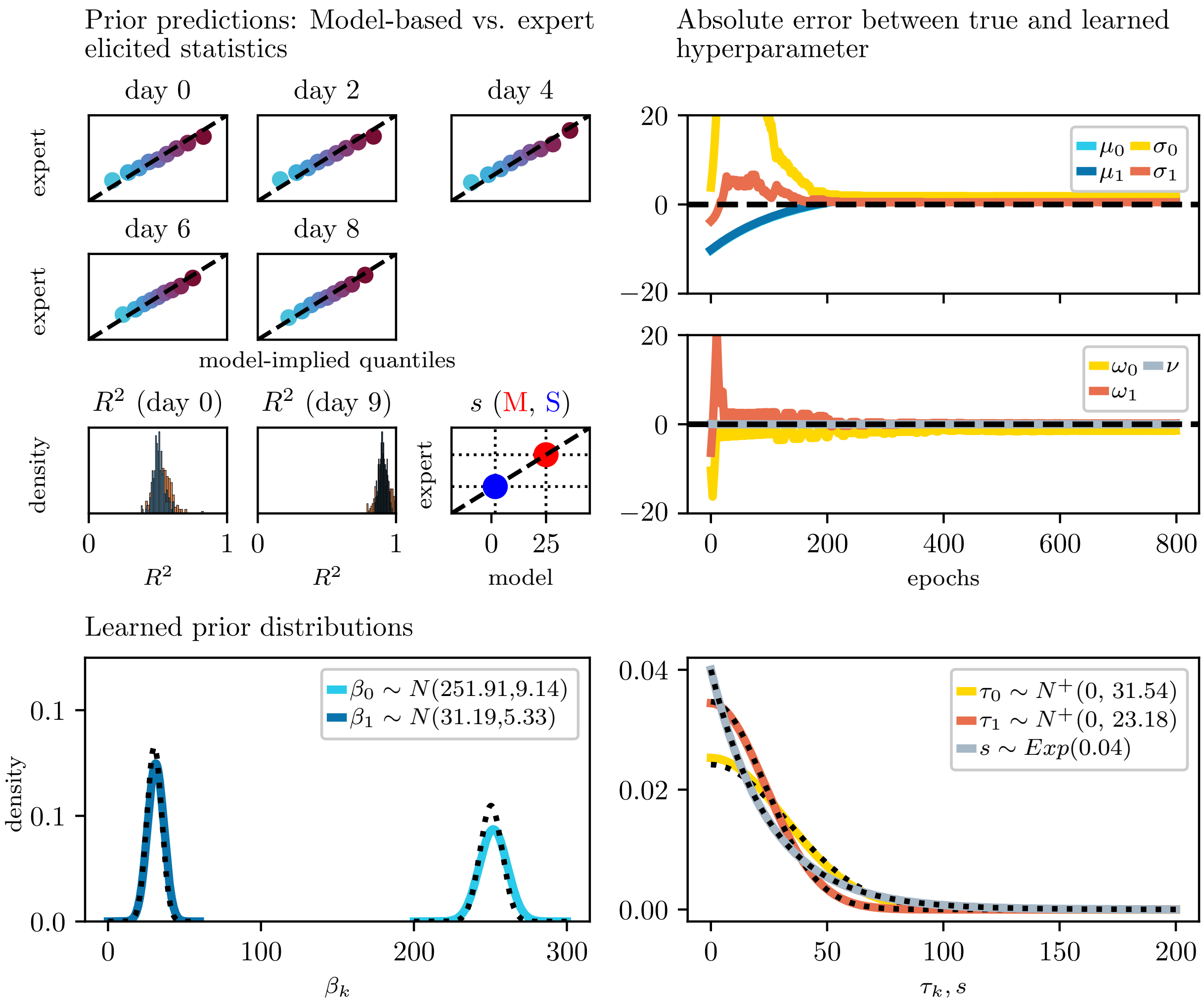

Figure 9: *Results of Case Study 4 - Hierarchical normal model*: The upper left panel presents a comparison between model-based and expert-elicited statistics, whereby the model simulations are based on the final learned hyperparameter λ. The first two rows depict quantile-based elicitation for each day x_i. In the last row histogram-based elicitation is used for R^2 and moment-based elicitation for the parameter s (within-person standard deviation). The two upper right panels illustrate the learning of hyperparameters across epochs, showcasing the difference between the true and learned values. In the two lower panels the true (dotted line) and final learned (solid line) prior distributions of each model parameter is depicted.

of the absolute error between true and learned hyperparameters in the upper right panel, revealing a progressive convergence towards zero across epochs.

4.5.1 Hierarchical Weibull Model

In the following analysis, we replace the normal with a Weibull likelihood, employing a log link on the likelihood mean to account for the fact that the Weibull distribution only allows for positive responses. The Weibull distribution is defined by a positive shape parameter α and a scale parameter $\beta = \frac{\theta_{ij}}{\Gamma(1+1/\alpha)}$, where θ_{ij} corresponds to the mean of the distribution and Γ represents the gamma function. We assume an Exponential prior distribution for α with hyperparameter ν. To underscore the modifications to the model, we present the revised model formulation as follows:

$$\begin{aligned}
y_{ij} &\sim \text{Weibull}(\alpha, \beta_{ij}) \\
\beta_{ij} &= \frac{\theta_{ij}}{\Gamma(1+1/\alpha)} \\
\log(\theta_{ij}) &= \beta_{0,j} + \beta_{1,j} x_{ij} \\
\alpha &\sim \text{Exponential}(\nu) \\
\ldots & \quad \text{(identical to Eq. 13)}
\end{aligned}$$

Our primary objective remains learning the hyperparameters $\lambda = (\mu_k, \sigma_k, \omega_k, \nu)$ based on expert knowledge.

For the modified model, we aim to employ the same expert information that was previously elicited for the normal model (see above). Since the expert information pertains to the observable domain, apart from the parameter information on the within-person standard deviation s, we can conveniently use the same data to train a different model. However, we must translate the provided information about s to properly inform the parameters of the Weibull model. Formally, this relationship can be expressed as $s = \sqrt{\beta^2 \left[\Gamma\left(1+\frac{2}{\alpha}\right) - \left(\Gamma\left(1+\frac{1}{\alpha}\right)\right)^2\right]}$. Equivalent to the normal model, we used the *sleepstudy* data set to fit a hierarchical model with Weibull likelihood in *brms* and used the estimated parameters as reference for the true hyperparameters: $\lambda^* = (\mu_0 = 5.52, \mu_1 = 0.10, \sigma_0 = 0.03, \sigma_1 = 0.02, \omega_0 = 0.15, \omega_1 = 0.09, \nu = 0.069)$. The specifications of the algorithm parameters for the learning process, along with a summary of the convergence diagnostics, are available in Appendix B.5. The simulation results are depicted in Figure 10 and affirm that our method is able to recover the true hyperparameter values in the Weibull model, notably by using the same expert information as used for the normal model.

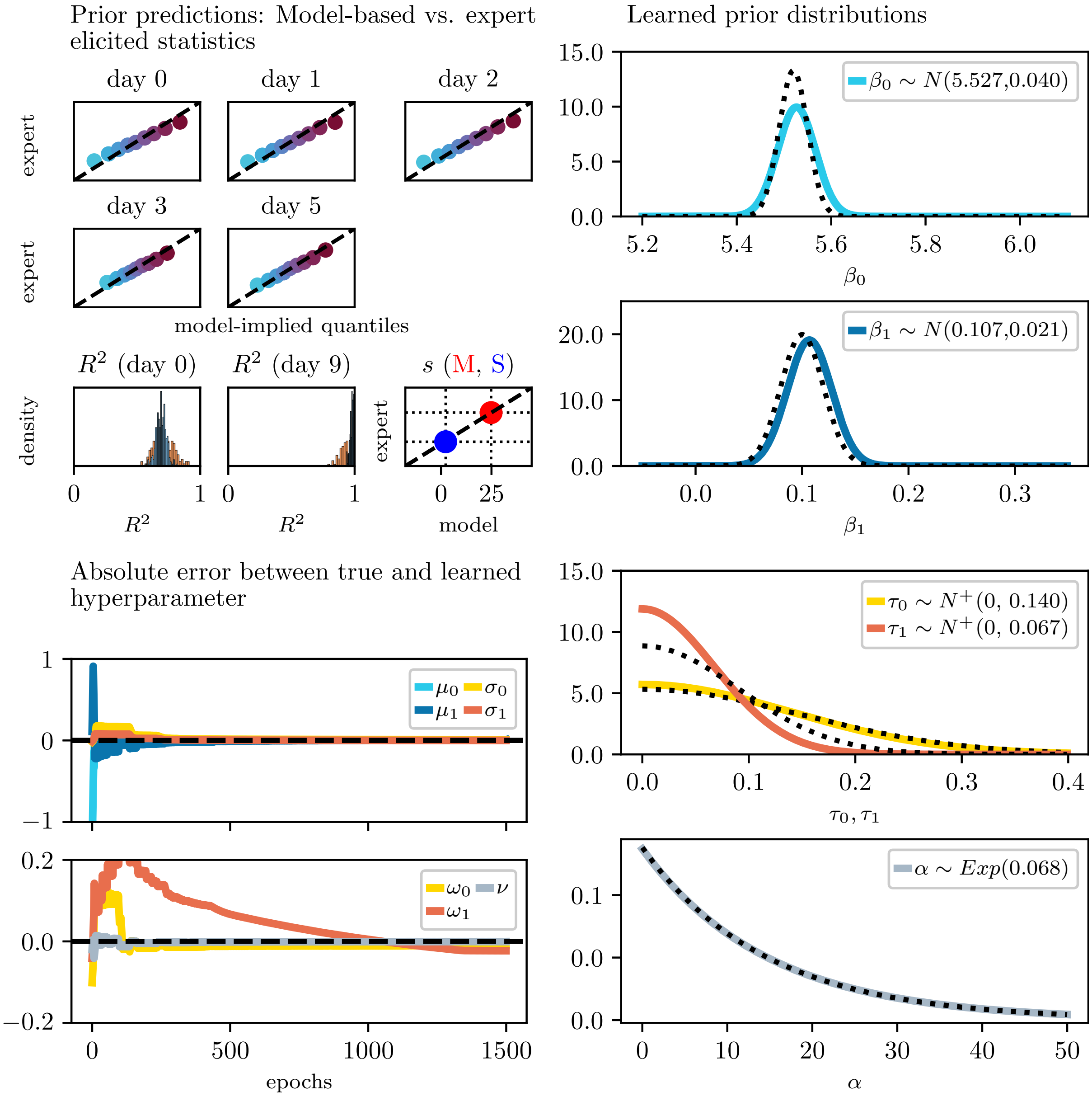

Figure 10: *Results of Case Study 4 - Weibull MLM*: The upper left panel presents a comparison between model-based and expert-elicited statistics, whereby the model simulations are based on the final learned hyperparameter λ. The first two rows depict quantile-based elicitation for each day x_i. In the last row histogram-based elicitation is used for R^2 and moment-based elicitation for the parameter s (within-person standard deviation). The two lower left panels illustrate the learning of hyperparameters across epochs, showcasing the difference between the true and learned values. In the right column the true (dotted line) and final learned (solid line) prior distributions of each model parameter is depicted.

4.5.2 Inconsistent expert information

In each of the preceding case studies, we simulated an ideal expert that mirrors the data-generating model with distinct hyperparameter values λ^*. A consequence of this decision is the mutual consistency of the elicited statistics derived from the expert model. However, the target quantities established by the analyst are not independent; for instance, in the previous normal model, the within-person standard deviation (s) directly influences the computation of R^2. In real-world scenarios, assuming consistency of expert knowledge is rather unrealistic. Human judgment is susceptible to heuristics and biases, potentially leading experts to provide conflicting information. This situation prompts us to consider how such inconsistencies might impact the performance of our elicitation method.

To analyze the learning from inconsistent expert information, we take the ideal expert from the normal multilevel model as a benchmark scenario in which expert information is consistent. The true hyperparameter values λ^* defining the ideal expert of the benchmark scenario are presented in Table 2 under the column labeled *Consistent*. Furthermore, we create two scenarios with inconsistent expert information, altering a single elicited statistic in each while keeping all other elicited statistics constant, thereby introducing conflicting information. In Scenario 1, we double the provided value for the within-person standard deviation (s) and in Scenario 2, we halve the values for both R^2 measures. The process of learning the hyperparameters λ based on the modified expert-elicited statistics adheres to the same algorithm parameter setup as described for the normal multilevel model. Appendix B.6 provides a graphical representation of the convergence diagnostics for each scenario. The learned hyperparameter values λ per scenario are depicted in the two last columns of Table 2. Furthermore, a comparison of simulated and expert-elicited statistics is provided in Figure 11, whereby the simulated elicited statistics are based on the final learned hyperarameters as depicted in Table 2.

Table 2: Model performance when input information for the learning algorithm is inconsistent

	Consistent	Inconsistent I	Inconsistent II
	(benchmark)	(double s)	(halve R^2)
λ	*true*	*learned*	
μ_0	251.865	251.935	251.938
μ_1	30.962	31.238	31.163
σ_0	9.367	**14.439** ↑	7.784
σ_1	5.991	**6.631** ↑	4.651
ω_0	32.356	**62.962** ↑	**18.100** ↓
ω_1	22.766	**44.764** ↑	**4.684** ↓
ν	0.040	**0.020** ↓	0.040

The elicited statistics of Scenario 1 are depicted on the left side of Figure 11, while the learned hyperparameter values are displayed in the middle column of Table 2. It becomes evident

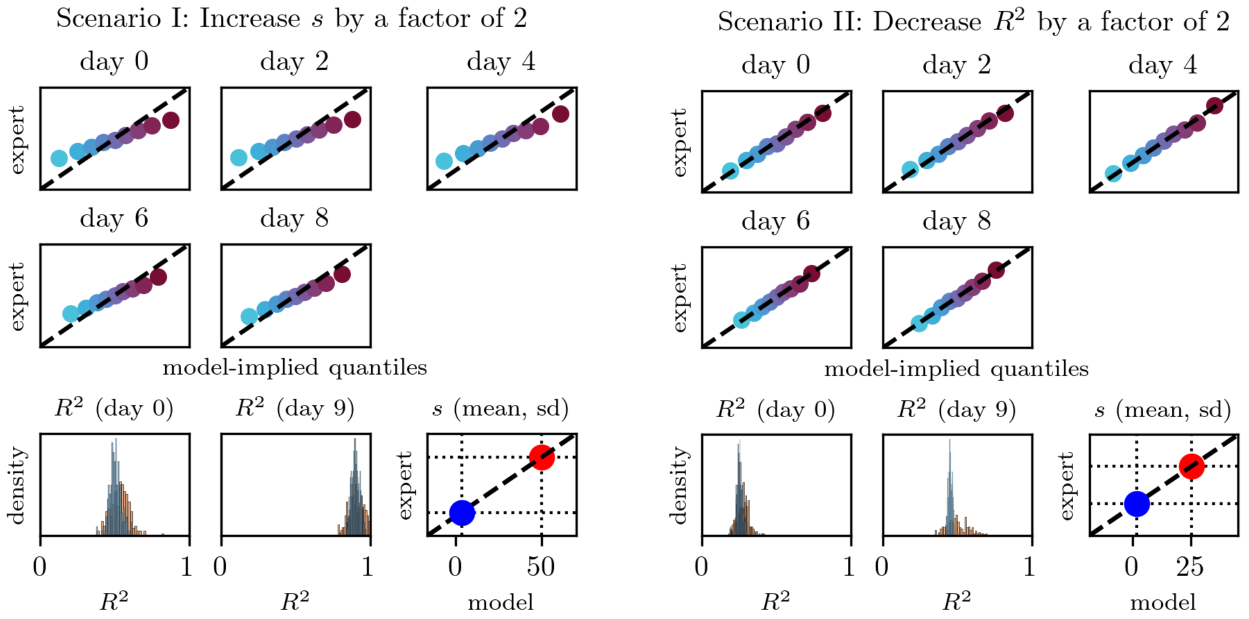

Figure 11: *Inconsistent expert information.* Starting with the ideal expert of the normal multilevel model, two scenarios were created with inconsistent expert information. In each scenario, one target quantity was altered while all other quantities remained unchanged. In Scenario 1 (left), the within-person standard deviation s was doubled and in Scenario 2 (right), R^2 was halved. The learning algorithm aims to strike the best possible balance between, on one hand, accurately learning the expert-elicited statistics, and on the other hand, maintaining (or restoring) consistency among the model parameters. In Scenario 1, s and R^2 are learned correctly. However, higher values for the standard deviation of the β coefficients and random effects are learned compared to the ideal expert, which has an effect on the expected distribution of RT values per day as can be seen in the quantile plots on the left side. In Scenario 2, all elicited statistics are learned correctly. However, the learned value for the varying effects is smaller (refer to Table 2) compared to that of the ideal expert, in order to maintain consistency between model parameters.

from both the learned value of the hyperparameter ν and the first two elicited moments of the model parameter s, in the lower-right plot, that the doubling of the within-person standard deviation s has been accurately learned. Furthermore, the histogram plots in the last row of Figure 11 illustrate a match between the simulated and the expert elicited R^2 values. Here, R^2 is defined as the ratio between the variance of the modeled predictive means and the total variance, including the residual variance s^2 (Gelman et al., 2019). As both the R^2 value and the residual variance are accurately learned, the model must compensate for the introduced inconsistency in the expert information by increasing the other variance components to restore consistency between the hyperparameters. This adjustment is indeed observable in the changes of σ_k and ω_k in Table 2, as well as through the differences between model-implied and expert-elicited quantiles in the upper two rows of Figure 11.

For Scenario II, the elicited statistics are presented on the right side of Figure 11, and the learned hyperparameters are displayed in the rightmost column of Table 2. The plots in Figure 11 indicate a match between the learned model-implied and expert-elicited statistics. Except for the variation in the R^2 which is reduced for the model-implied elicited statistic as the model reduces the between-person variation (i.e., varying effects) in order to account for the introduced inconsistency in the expert information, as evident in Table 2. In summary, learning from inconsistent expert information can be characterized as a balance between achieving the highest possible alignment between the model and expert-elicited statistics, while simultaneously compensating for conflicting information to arrive at a final, consistent set of hyperparameter values.

5 Discussion

When developing Bayesian models, analysts face the challenge of specifying appropriate prior distributions for each model parameter, involving both the choice of the distributional family as well as the corresponding hyperparameter values. We proposed an elicitation method that assists analysts in identifying the hyperparameter values of given prior distribution families based on expert knowledge. Our method accommodates various types of expert knowledge, including information about model parameters, observable data patterns, and other relevant statistics (e.g., R^2). Additionally, it can handle common expert data formats resulting from histogram, moment-, or quantile-based elicitation. Our method is agnostic to the specific probabilistic model and offers a modular design, providing analysts with the flexibility to modify or replace specific building blocks, such as the discrepancy measure or the loss weighting method. In our case studies, we demonstrated the excellent performance of our method for various modeling tasks, combining different kinds of expert knowledge in a consistent and flexible manner. Despite these highly promising results, some relevant limitations remain, which are discussed below together with ideas for future research.

Our method employs gradient-based optimization to learn hyperparameter values. This choice offers the advantage of requiring only the ability to sample from the generative model, simplifying the learning process. However, it comes with the prerequisite that all operations and functions in the computational graph must be differentiable or admit a reparameterization whose gradients can be approximated with sufficient accuracy. Consequently, for discrete random variables, specific techniques, such as the Softmax-Gumbel trick that we used in our case studies, are necessary to obtain approximate gradients. Alternatively, one could opt for optimization methods that entirely forego gradient computations. For instance, Manderson & Goudie (2023) propose a two-stage optimization process based on Bayesian optimization (Frazier, 2018), which avoids the need for gradients altogether. Nevertheless, this choice has its own limitations, notably in terms of scalability, as Bayesian optimization does not perform optimally in higher-dimensional spaces (Eriksson & Jankowiak, 2021). Given the ongoing active research into the development of optimization techniques that can scale effectively in higher dimensions and handle discrete random variables, we acknowledge the potential for further advancement and refinement of our proposed method.

Having a suitable optimization method is fundamental for learning hyperparameters based on expert knowledge. However, there are cases where hyperparameters cannot be uniquely determined from available expert data, leading to different learned hyperparameters upon multiple replications of the learning process. This situation raises the question of how to choose between prior distributions that represent the elicited expert knowledge equally well. Initial approaches, such as incorporating a regularization term in the loss function to favor priors with higher entropy, have been proposed to address this challenge (Manderson & Goudie, 2023). Another avenue to achieve model identification involves the model architecture. For instance, statistical models that adopt joint priors for their parameters and thus keep the number of hyperparameters low, are expected to exhibit better model identification compared to models that assume independent priors for the parameters (e.g., Aguilar & Bürkner, 2023). Nevertheless, further research is needed to develop techniques that can efficiently handle unidentified models. Recently, Sameni (2023) emphasized the importance of establishing quantitative metrics for assessing model identification. We fully support this notion, as we also recognize the necessity for such metrics in our proposed method. Considering it as a future task, these metrics would provide valuable guidelines for analysts in making informed decisions regarding the selection among expert data collection strategies and model architectures.

Finally, we need to discuss an important aspect shared by all gradient-based optimization methods, including our own: the objective of finding an optimal *point estimate* for the hyperparameters λ. By adopting this approach, any uncertainties surrounding the value of λ are neglected, despite the potential introduction of uncertainty during the prior elicitation process. For instance, an expert may possess uncertainties about the true value of a queried quantity or face uncertainty while quantifying implicit knowledge. The adoption of fixed point estimates for hyperparameters fails to account for these uncertainties and may result in an overly confident representation of the prior knowledge.

To address this limitation, it would be advantageous to adopt a probabilistic approach that explicitly accounts for uncertainty in the hyperparameters. Some initial attempts to tackle this issue exist. For instance, Hartmann et al. (2020) proposed modeling the uncertainty in quantifying expert knowledge using a Dirichlet distribution, where the precision parameter controls the variance. However, this approach is problem-specific, lacking generalizability to arbitrary models. An alternative probabilistic perspective to address this concern was suggested by Mikkola et al. (2023), who advocate for a *Bayesian treatment of the expert*. The central idea is to incorporate a user model for the expert and assume that the analyst holds a prior belief about the expert's knowledge, which is updated using Bayes' rule after each query. Given the flexibility of our method, it can readily accommodate this concept, offering a promising avenue for future development and next steps.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., … others (2016). Tensorflow: a system for large-scale machine learning. In *Osdi* (Vol. 16, pp. 265–283).

Aguilar, J. E., & Bürkner, P.-C. (2023). Intuitive joint priors for Bayesian linear multilevel models: The R2D2M2 prior. *Electronic Journal of Statistics*, *17*(1), 1711–1767. doi: 10.1214/23-EJS2136

Akbarov, A. (2009). *Probability elicitation: Predictive approach* (Unpublished doctoral dissertation). University of Salford.

Albert, I., Donnet, S., Guihenneuc-Jouyaux, C., Low Choy, S., Mengersen, K. L., & Rousseau, J. (2012). Combining expert opinions in prior elicitation. *Bayesian Analysis*, *7*(3), 503–532. doi: https://doi.org/10.1214/12-BA717

Aushev, A., Putkonen, A., Clarte, G., Chandramouli, S., Acerbi, L., Kaski, S., & Howes, A. (2023). Online simulator-based experimental design for cognitive model selection. *arXiv preprint*. Retrieved from https://doi.org/10.48550/arXiv.2303.02227

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, *67*(1), 1–48. doi: https://doi.org/10.18637/jss.v067.i01

Bedrick, E. J., Christensen, R., & Johnson, W. (1996). A New Perspective on Priors for Generalized Linear Models. *Journal of the American Statistical Association*, *91*(436), 1450–1460. doi: https://doi.org/10.1080/01621459.1996.10476713

Belenky, G., Wesensten, N. J., Thorne, D. R., Thomas, M. L., Sing, H. C., Redmond, D. P., … Balkin, T. J. (2003). Patterns of performance degradation and restoration during sleep restriction and subsequent recovery: A sleep dose-response study. *Journal of Sleep Research*, *12*(1), 1–12. doi: https://doi.org/10.1046/j.1365-2869.2003.00337.x

Bürkner, P.-C., Scholz, M., & Radev, S. T. (2022). Some models are useful, but how do we know which ones? Towards a unified Bayesian model taxonomy. *arXiv preprint*. doi: https://doi.org/10.48550/arXiv.2209.02439

Bürkner, P.-C. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, *80*(1), 1–28. doi: 10.18637/jss.v080.i01

Cranmer, K., Brehmer, J., & Louppe, G. (2020). The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, *117*(48), 30055–30062. doi: https://doi.org/10.1073/pnas.1912789117

Crawshaw, M. (2020). Multi-Task Learning with Deep Neural Networks: A Survey. *arXiv preprint*. doi: https://doi.org/10.48550/arXiv.2009.09796

da Silva, E. d. S., Ku'smierczyk, T., Hartmann, M., & Klami, A. (2019). Prior specification via prior predictive matching: Poisson matrix factorization and beyond. *arXiv preprint*. doi: https://doi.org/10.48550/arXiv.1910.12263

Deb, K. (2011). Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction. In L. Wang, A. H. C. Ng, & K. Deb (Eds.), *Multi-objective evolutionary optimisation for product design and manufacturing* (pp. 3–34). Springer. doi: https://doi.org/10.1007/978-0-85729-652-8_1

Dua, D., & Graff, C. (2017). *UCI Machine Learning Repository.* Retrieved from http://archive.ics.uci.edu/ml

Eriksson, D., & Jankowiak, M. (2021). High-dimensional Bayesian optimization with sparse axis-aligned subspaces. In C. de Campos & M. H. Maathuis (Eds.), *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence* (Vol. 161, pp. 493–503). Retrieved from https://proceedings.mlr.press/v161/eriksson21a.html

Falconer, J. R., Frank, E., Polaschek, D. L., & Joshi, C. (2022). Methods for Eliciting Informative Prior Distributions: A Critical Review. *Decision Analysis*, *19*(3), 189–204. doi: https://doi.org/10.1287/deca.2022.0451

Feydy, J. (2020). *Geometric data analysis, beyond convolutions* (Unpublished doctoral dissertation). Université Paris-Saclay Gif-sur-Yvette, France.

Feydy, J., Séjourné, T., Vialard, F.-X., Amari, S.-i., Trouvé, A., & Peyré, G. (2019). Interpolating between Optimal Transport and MMD using Sinkhorn Divergences. In K. Chaudhuri & M. Sugiyama (Eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics* (Vol. 89, pp. 2681–2690). Retrieved from https://proceedings.mlr.press/v89/feydy19a.html

Figurnov, M., Mohamed, S., & Mnih, A. (2018). Implicit Reparameterization Gradients. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 31). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2018/file/92c8c96e4c37100777c7190b76d28233-Paper.pdf

Frazier, P. I. (2018). A Tutorial on Bayesian Optimization. *arXiv preprint*. doi: https://doi.org/10.48550/arXiv.1807.02811

Garthwaite, P. H., Kadane, J. B., & O'Hagan, A. (2005). Statistical Methods for Eliciting Probability Distributions. *Journal of the American Statistical Association*, *100*(470), 680–701. doi: https://doi.org/10.1198/016214505000000105

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian Data Analysis* (3rd ed.). Chapmann and Hall/CRC press. doi: https://doi.org/10.1201/b16018

Gelman, A., Goodrich, B., Gabry, J., & Vehtari, A. (2019). R-squared for Bayesian Regression Models. *The American Statistician*, 307–309. doi: https://doi.org/10.1080/00031305.2018.1549100

Gelman, A., Simpson, D., & Betancourt, M. (2017). The Prior Can Often Only Be Understood in the Context of the Likelihood. *Entropy*, *19*(10), 555. doi: https://doi.org/10.3390/e19100555

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. (http://www.deeplearningbook.org)

Gretton, A. (2017, Jun). *Notes on the Cramer GAN.* Towards Data Science. Retrieved from https://towardsdatascience.com/notes-on-the-cramer-gan-752abd505c00

Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., & Smola, A. (2006). A Kernel Method for the Two-Sample-Problem. In B. Schölkopf, J. Platt, & T. Hoffman (Eds.), *Advances in Neural Information Processing Systems* (Vol. 19). MIT Press. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2006/file/e9fb2eda3d9c55a0d89c98d6c54b5b3e-Paper.pdf

Gumbel, E. (1962). Statistical Theory of Extreme Values. *Belgian Journal of Operations Research, Statistics, and Computer Science*, *3*(2), 3–11.

Hartmann, M., Agiashvili, G., Bürkner, P., & Klami, A. (2020). Flexible Prior Elicitation via the Prior Predictive Distribution. In J. Peters & D. Sontag (Eds.), *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)* (Vol. 124, pp. 1129–1138). Retrieved from https://proceedings.mlr.press/v124/hartmann20a.html

Jang, E., Gu, S., & Poole, B. (2017). Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations.* Retrieved from https://openreview.net/forum?id=rkE3y85ee

Jaynes, E. T. (2003). *Probability Theory: The Logic of Science*. Cambridge University Press. doi: https://doi.org/10.1017/CBO9780511790423

Johnson, A. A., Ott, M. Q., & Dogucu, M. (2022). *Bayes Rules!: An Introduction to Applied Bayesian Modeling* (1st ed.). Chapman and Hall/CRC Press. doi: https://doi.org/10.1201/9780429288340

Joo, W., Kim, D., Shin, S., & Moon, I.-C. (2020). Generalized Gumbel-Softmax Gradient Estimator for Generic Discrete Random Variables. *arXiv preprint*. doi: 10.48550/arXiv.2003.01847

Kadane, J., Dickey, J. M., Winkler, R. L., Smith, W. S., & Peters, S. C. (1980). Interactive Elicitation of Opinion for a Normal Linear Model. *Journal of the American Statistical Association*, *75*(372), 845–854. doi: https://doi.org/10.2307/2287171

Kadane, J., & Wolfson, L. J. (1998). Experiences in Elicitation. *Journal of the Royal Statistical Society Series D: The Statistician*, *47*(1), 3–19. doi: https://doi.org/10.1111/1467-9884.00113

Kingma, D. P., & Welling, M. (2022). *Auto-Encoding Variational Bayes.* doi: https://doi.org/10.48550/arXiv.1312.6114

Lewandowski, D., Kurowicka, D., & Joe, H. (2009). Generating random correlation matrices based on vines and extended onion method. *Journal of Multivariate Analysis*, *100*(9), 1989–2001. doi: https://doi.org/10.1016/j.jmva.2009.04.008

Li, Y., Swersky, K., & Zemel, R. (2015). Generative Moment Matching Networks. In F. Bach & D. Blei (Eds.), *Proceedings of the 32nd International Conference on Machine Learning* (Vol. 37, pp. 1718–1727). PMLR. Retrieved from https://proceedings.mlr.press/v37/li15.html

Liu, S., Johns, E., & Davison, A. J. (2019). End-To-End Multi-Task Learning With Attention. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1871–1880). doi: https://doi.org/10.1109/CVPR.2019.00197

Liu, S., Liang, Y., & Gitter, A. (2019). Loss-Balanced Task Weighting to Reduce Negative Transfer in Multi-Task Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 33, pp. 9977–9978). doi: https://doi.org/10.1609/aaai.v33i01.33019977

Maddison, C. J., Mnih, A., & Teh, Y. W. (2017). The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In *International Conference on Learning Representations.* Retrieved from https://openreview.net/forum?id=S1jE5L5gl

Maddison, C. J., Tarlow, D., & Minka, T. (2014). A* Sampling. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K. Weinberger (Eds.), *Advances in Neural Information Processing Systems* (Vol. 27). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2014/file/309fee4e541e51de2e41f21bebb342aa-Paper.pdf

Manderson, A. A., & Goudie, R. J. (2023). Translating Predictive Distributions into Informative Priors. *arXiv preprint*. doi: https://doi.org/10.48550/arXiv.2303.08528

Mikkola, P., Martin, O. A., Chandramouli, S., Hartmann, M., Pla, O. A., Thomas, O., … others (2023). Prior Knowledge Elicitation: The Past, Present, and Future. *Bayesian Analysis*, *1*(1), 1–33. doi: https://doi.org/10.1214/23-BA1381

Muandet, K., Fukumizu, K., Sriperumbudur, B., Schölkopf, B., et al. (2017). Kernel Mean Embedding of Distributions: A Review and Beyond. *Foundations and Trends® in Machine Learning*, *10*(1-2), 1–141. doi: http://dx.doi.org/10.1561/2200000060

Navarro, D. J. (2019). Between the Devil and the Deep Blue Sea: Tensions Between Scientific Judgement and Statistical Model Selection. *Computational Brain & Behavior*, *2*(1), 28–34. doi: https://doi.org/10.1007/s42113-018-0019-z

O’Hagan, A., Buck, C. E., Daneshkhah, A., Eiser, J. R., Garthwaite, P. H., Jenkinson, D. J., … Rakow, T. (2006). *Uncertain Judgements: Eliciting Experts’ Probabilities*. John Wiley & Sons. doi: https://doi.org/10.1002/0470033312

Sameni, R. (2023). Beyond Convergence: Identifiability of Machine Learning and Deep Learning Models. *arXiv print*. doi: https://doi.org/10.48550/arXiv.2307.11332

Schmitt, M., Bürkner, P.-C., Köthe, U., & Radev, S. T. (2022). Detecting Model Misspecification in Amortized Bayesian Inference with Neural Networks. *arXiv print*. doi: https://doi.org/10.48550/arXiv.2112.08866

Sjölund, J. (2023). A Tutorial on Parametric Variational Inference. *arXiv preprint*. doi: https://doi.org/10.48550/arXiv.2301.01236

Stefan, A. M., Evans, N. J., & Wagenmakers, E.-J. (2022). Practical challenges and methodological flexibility in prior elicitation. *Psychological Methods*, *27*(2), 177–197. doi: https://doi.org/10.1037/met0000354

Tokui, S., & Sato, I. (2016). Reparameterization Trick for Discrete Variables. *arXiv preprint*. doi: https://doi.org/10.48550/arXiv.1611.01239

Unkelbach, C., & Rom, S. (2017). A Referential Theory of the Repetition-induced Truth Effect. *Cognition*, *160*, 110–126. doi: https://doi.org/10.1016/j.cognition.2016.12.016

Appendix

A Method

A.1 Symbol Glossary

Table A.1.1: Overview of model and algorithm (hyper-)parameters

Workflow task	Notation	Label	Comment
General	seed	seed	seed = 2023
Generative Model	λ	model hyperparameter	hyperparameter of prior distributions
	θ	model parameter	model parameter
Dynamic Weight Averaging	a	temperature	$a = 1.6$
Gumbel-Softmax Trick	τ	temperature	$\tau = 1.0$
Gradient Descent	E	epochs / iterations	number of iterations used for learning the model hyperparameter
	B	batch size	$B = 2^8$, number of simulations within one epoch; fix for all case studies
	S_E	repetition-expert	number of model-parameter repetitions within one batch for simulating the expert data
	S_M	repetition-expert	number of model-parameter repetitions within one batch for simulating the model-implied data
Adam Optimizer (with exponential decay)	lr_0	initial learning rate	learning rate of the initial iterations
	lr_{min}	minimum learning rate	lower bound for the decay
	lr_{decay}	decay	rate of exponential decay
	decay-step	decay step	number iterations (steps) until the next decay is applied

B Case Studies: Convergence diagnostics and Learning Results

B.1 Case Study 2: LM — Normal linear regression model

The algorithm parameters for the optimization procedure in order to learn the model hyperparameters λ are: $B = 2^8$, $E = 1\,500$, $S_E = 300$, $S_M = 200$, and an exponential learning rate schedule that decays every 5 steps with a base of 0.97, an initial learning rate $\phi^0 = 0.1$, and $\phi^{\min} = 10^{-5}$. Figure with convergence diagnostics of the optimization results can be found in the Case Studies section in the main text.

B.2 Case Study 2: GLMs — Binomial model

To setup the learning algorithm, the following algorithm parameters are used: $B = 2^8$, $E = 1,000$, $S_E = 300$, $S_M = 200$, and an exponential learning rate schedule that decays every 18 steps with a base of 0.95. The initial learning rate is set to $\phi^0 = 0.01$, and the minimum learning rate is $\phi^{\min} = 10^{-3}$. To improve gradient-based learning, we divide the continuous predictor by its standard deviation and use this scaled predictor in the optimization process. Figure B.2.1 depicts the convergence diagnostics of the optimization results.

B.3 Case Study 3: GLMs — Poisson model

For the learning algorithm, we standardize (z-transform) the continuous predictor to improve learning and we set the algorithm parameters as follows: $B = 2^8, E = 600$, $S_E = 300$, $S_M = 150$, and an exponential learning rate schedule that decays every 7 steps with a base of 0.95, an initial learning rate $\phi^0 = 0.1$, and $\phi^{\min} = 10^{-4}$. The convergence diagnostics inspection shows successful convergence. Figure B.3.1 depicts the convergence diagnostics of the optimization results.

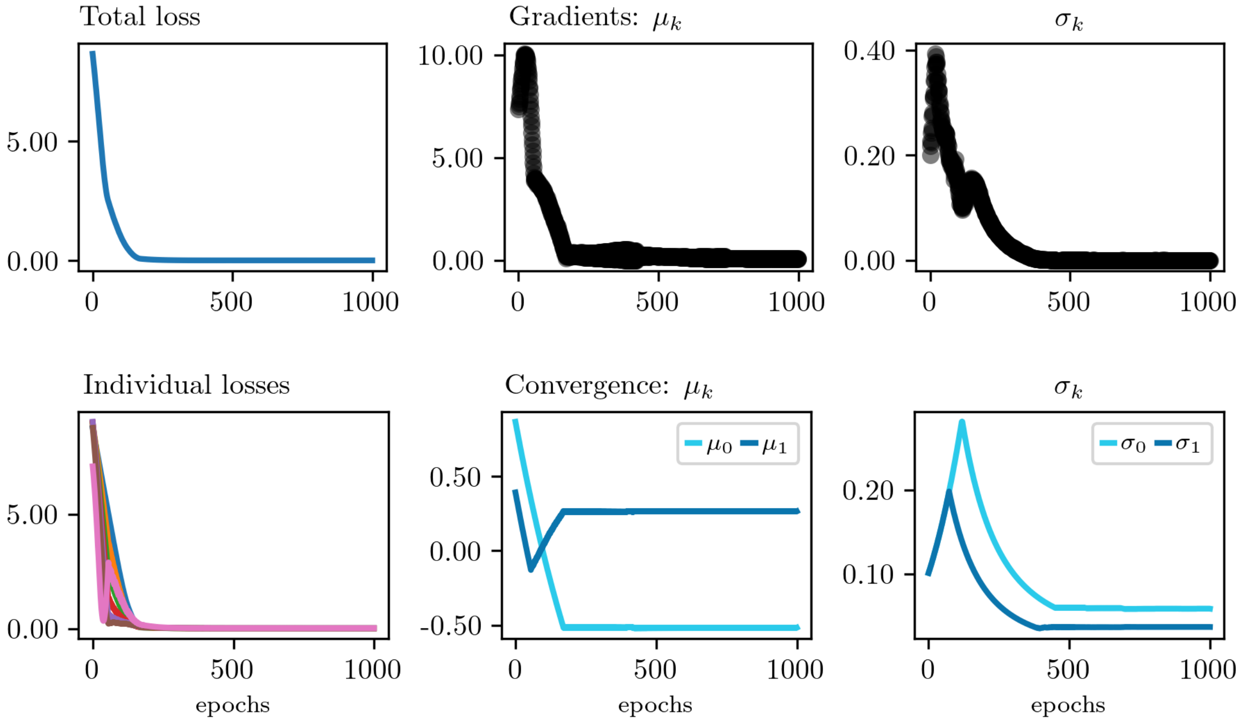

Figure B.2.1: *Convergence diagnostics Binomial model.* The leftmost column represents the loss value across epochs, demonstrating the desired decreasing trend of all loss values (i.e., total loss as well as individual loss components). The upper two right panels display the expected decreasing trend towards zero of the gradient norm for each learned hyperparameter λ. The two lower right panels illustrate learning of each hyperparameter across epochs, stabilizing in the long run at a specific value.

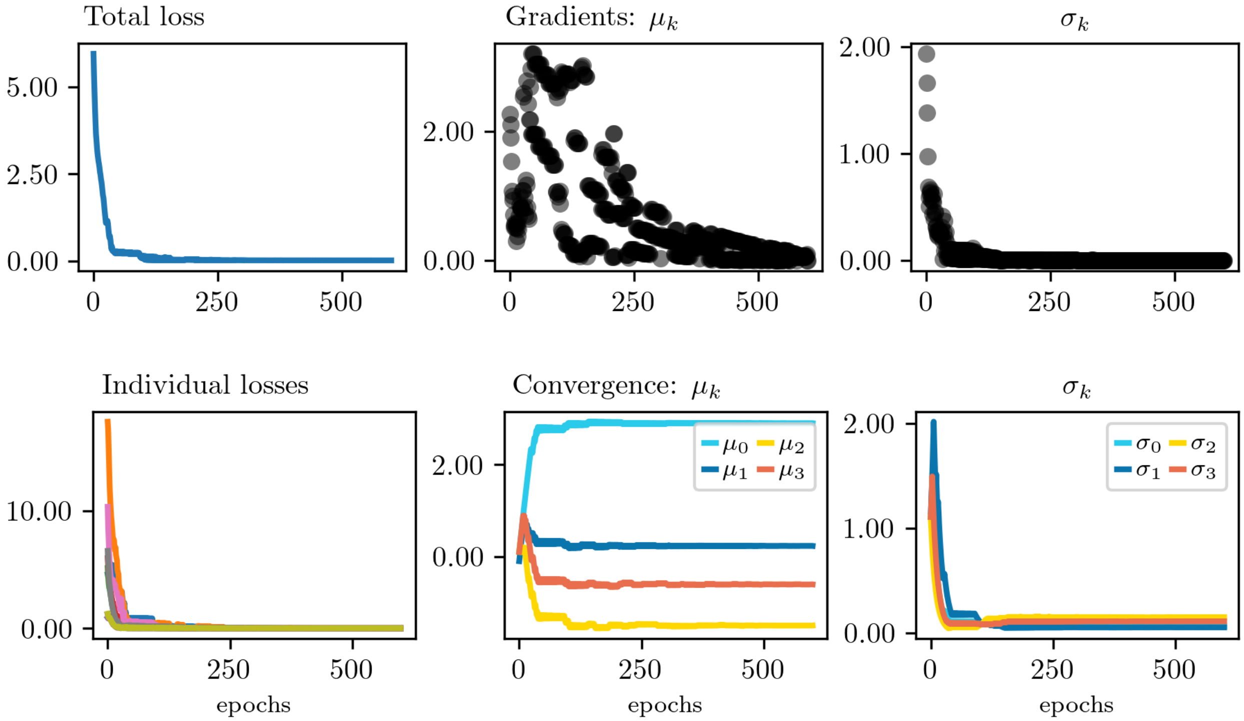

Figure B.3.1: *Convergence diagnostics for Poisson model.* The leftmost column represents the loss value across epochs, demonstrating the desired decreasing trend of all loss values (i.e., total loss as well as individual loss components). The upper two right panels display the expected decreasing trend towards zero of the gradient norm for each learned hyperparameter λ. The two lower right panels illustrate learning of each hyperparameter across epochs, stabilizing in the long run at a specific value.

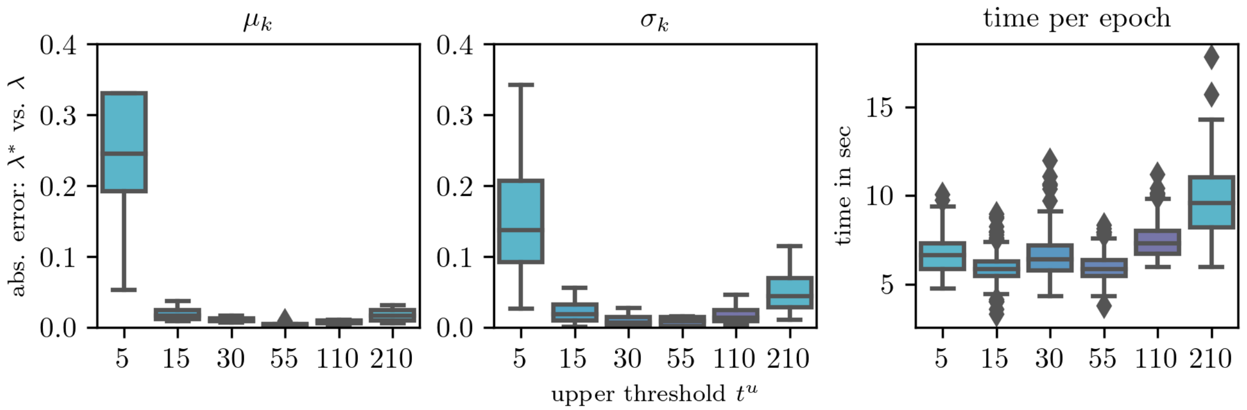

Figure B.3.2: *Effect of different t^u on accuracy of hyperparameter learning and run time (Poisson model).* The first two panels from the left illustrate the absolute error between the true and learned hyperparameters λ (μ_k in the first panel and σ_k in the second panel) for different selected upper thresholds ($t^u = 5, 15, 30, 55, 110, 210$). The error is considerably increased for thresholds that clearly under- and, interestingly, overestimate the upper threshold ($t^u = 5, 15$, and 210). On the right panel, the run time per epoch in seconds for the selected t^u is shown. The run time exhibits the expected pattern, with higher thresholds resulting in increased run time, thus decreasing the method's efficiency.

B.4 Case Study 4: Hierarchical model — Normal likelihood

For the learning algorithm, we divided the continuous predictor x_i by its standard deviation. Moreover, per loss component we normalized each elicited quantity t_m and $\hat{t}_m$ before computing the Maximum Mean discrepancy loss. this was done in order to reduce differences in the order of magnitude between loss components such that no loss component overshadows the other loss components during learning. Normalization was done by first computing the minimum and maximum value for the model-implied elicited quantity t_m and using these values in order to scale it into a 0-1 range. Then, we used the same computed minimum and maximum values to scale the expert elicited quantities $\hat{t}_m$ accordingly. For learning, we set the algorithm parameters as follows: $B = 2^8, E = 800$, $S_E = 200$, $S_M = 200$, and an exponential learning rate schedule that decays every 7 steps with a base of 0.95, an initial learning rate $\phi^0 = 0.1$, and $\phi^{\text{min}} = 10^{-3}$. The convergence diagnostics inspection shows successful convergence. Figure B.4.1 depicts a summary of convergence diagnostics.

B.5 Case Study 4: Hierarchical model — Weibull likelihood

For the learning algorithm, we divided the continuous predictor x_i by its standard deviation. Moreover, per loss component we normalized each elicited quantity t_m and $\hat{t}_m$ before computing the Maximum Mean discrepancy loss. this was done in order to reduce differences in the order of magnitude between loss components such that no loss component overshadows the other loss components dur-

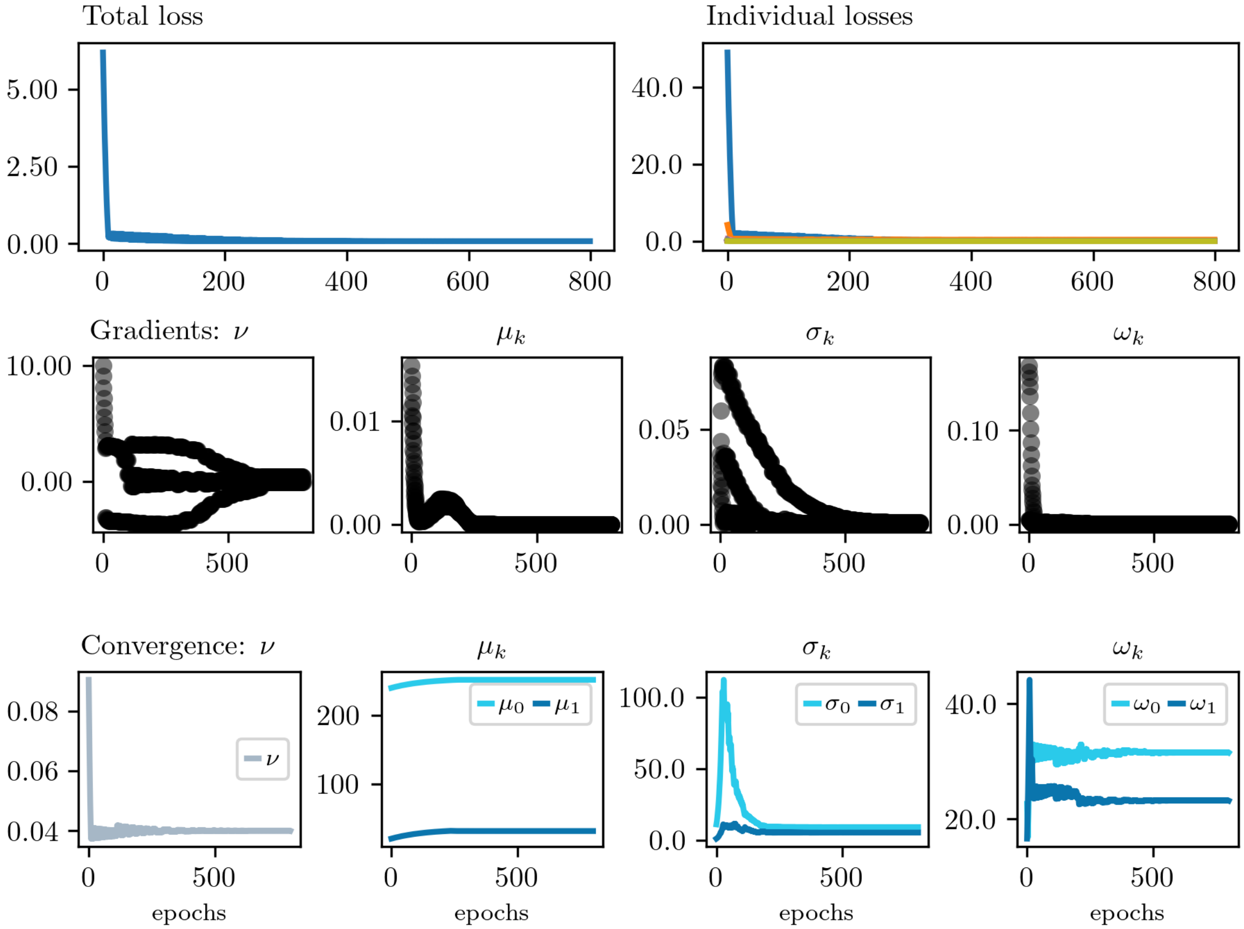

Figure B.4.1: *Convergence diagnostics for multilevel model with normal likelihood.* The first row represents the loss value across epochs, demonstrating the desired decreasing trend of all loss values (i.e., total loss on the left as well as individual loss components on the left). The second row displays the expected decreasing trend towards zero of the gradient norm for each learned hyperparameter λ. The last row illustrates learning of each hyperparameter across epochs, stabilizing in the long run at a specific value.

ing learning. Normalization was done by first computing the minimum and maximum value for the model-implied elicited quantity t_m and using these values in order to scale it into a 0-1 range. Then, we used the same computed minimum and maximum values to scale the expert elicited quantities $\hat{t}_m$ accordingly. For learning, we set the algorithm parameters as follows: $B = 2^8, E = 400$, $S_E = 200$, $S_M = 200$, and an exponential learning rate schedule that decays every 7 steps with a base of 0.90, an initial learning rate $\phi^0 = 0.1$, and $\phi^{\min} = 10^{-4}$. The convergence diagnostics inspection shows successful convergence. Figure B.5.2 depicts a summary of convergence diagnostics.

B.6 Case Study 4: Hierarchical model — Inconsistent Expert Information

For both scenarios with inconsistent expert information the same algorithm parameters as for the hierarchical model with normal likelihood were used. See description in Appendix B.4.

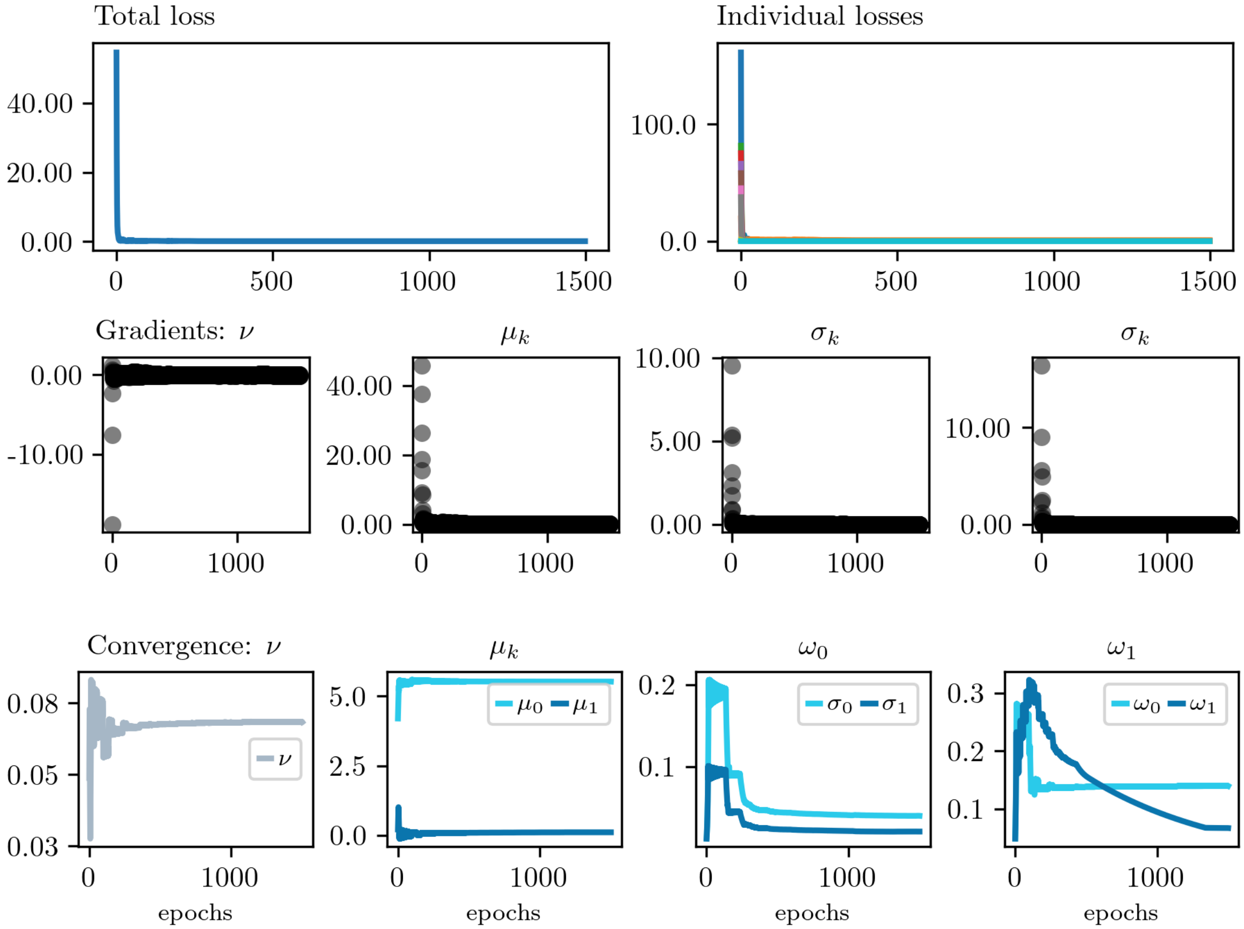

Figure B.5.2: *Convergence diagnostics for multilevel model with Weibull likelihood.* The first row represents the loss value across epochs, demonstrating the desired decreasing trend of all loss values (i.e., total loss on the left as well as individual loss components on the left). The second row displays the expected decreasing trend towards zero of the gradient norm for each learned hyperparameter λ. The last row illustrates learning of each hyperparameter across epochs, stabilizing in the long run at a specific value.

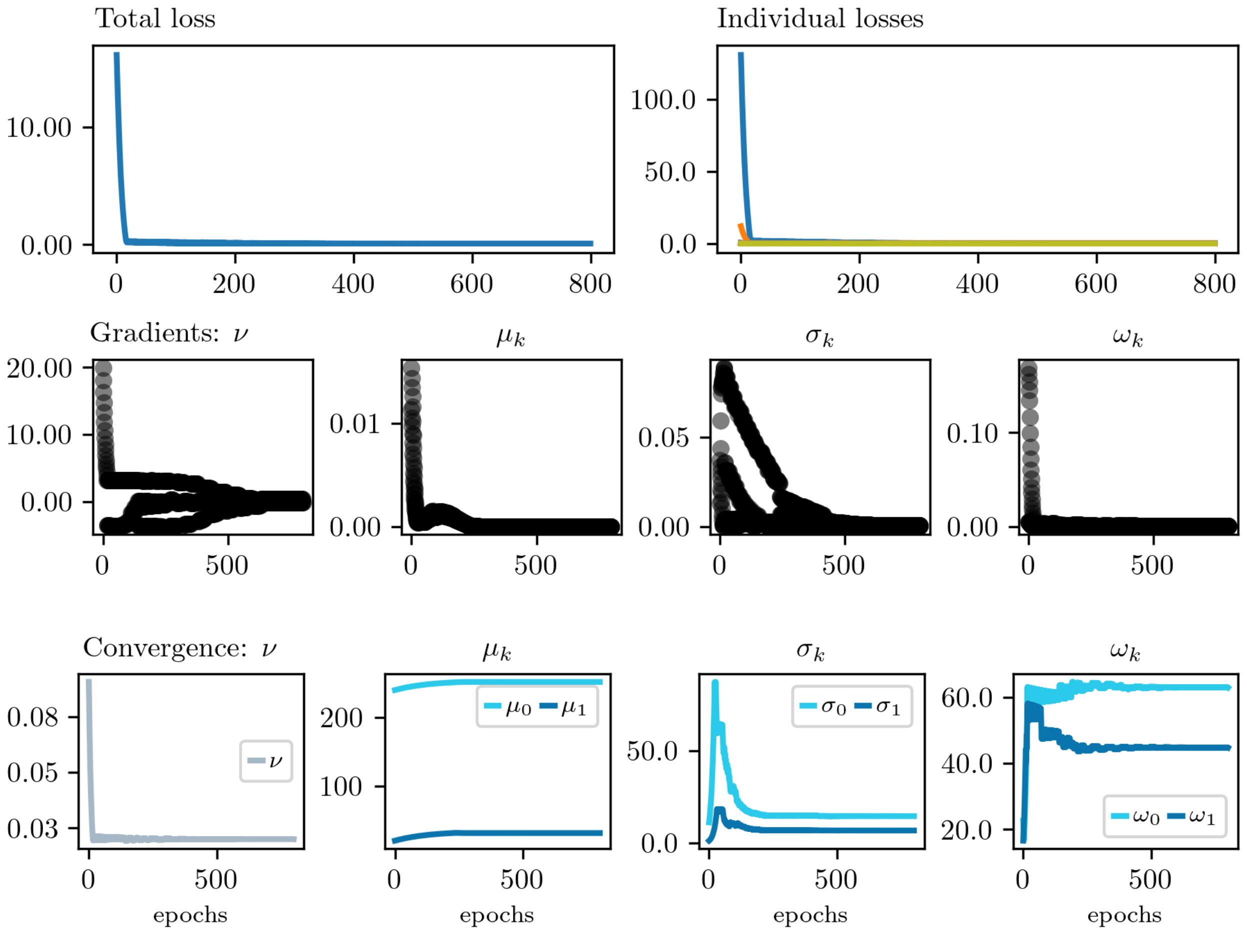

Figure B.6.3: *Convergence diagnostics for inconsistent expert information for Scenario 1.* The first row represents the loss value across epochs, demonstrating the desired decreasing trend of all loss values (i.e., total loss on the left as well as individual loss components on the left). The second row displays the expected decreasing trend towards zero of the gradient norm for each learned hyperparameter λ. The last row illustrates learning of each hyperparameter across epochs, stabilizing in the long run at a specific value.

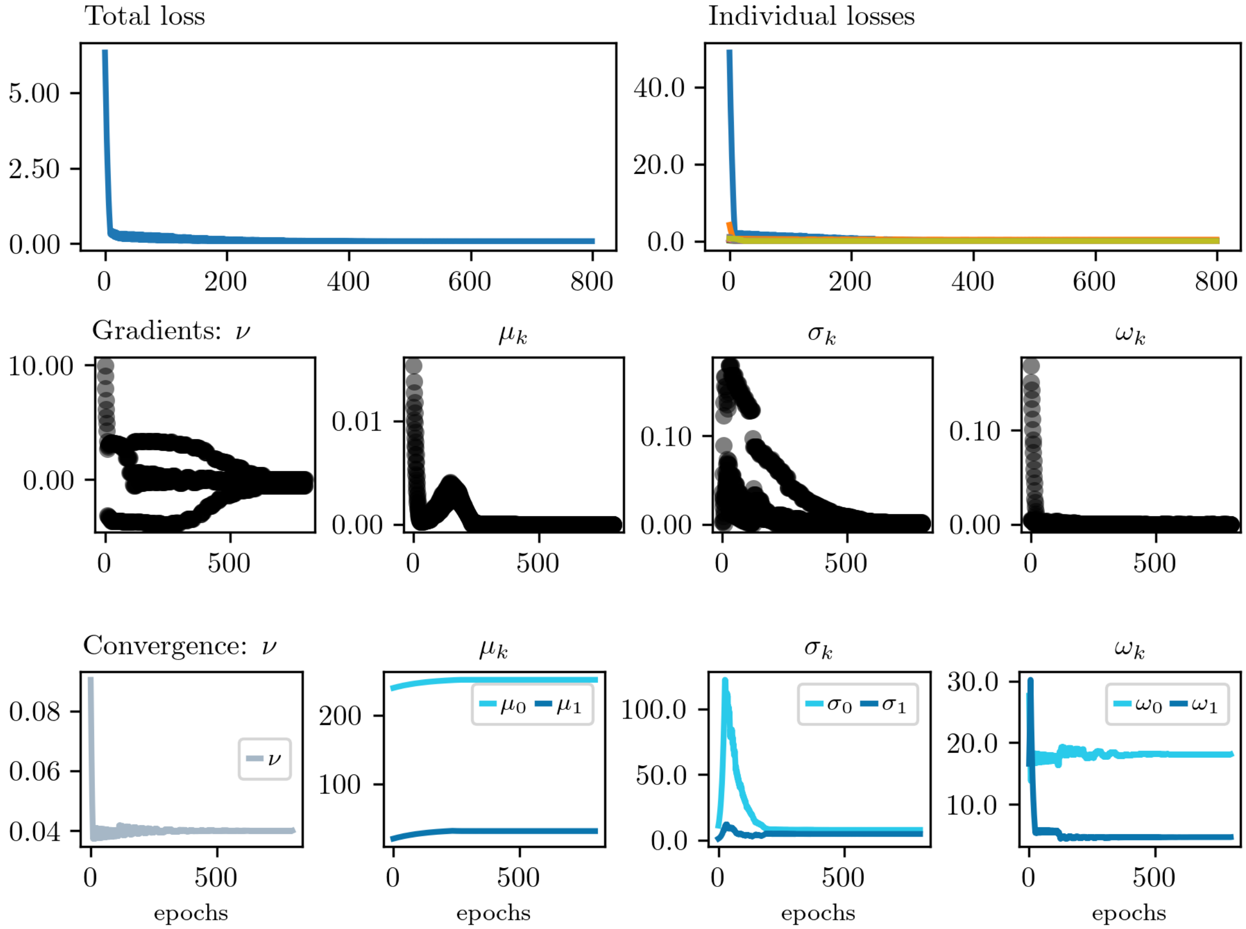

Figure B.6.4: *Convergence diagnostics for inconsistent expert information for Scenario 2.* The first row represents the loss value across epochs, demonstrating the desired decreasing trend of all loss values (i.e., total loss on the left as well as individual loss components on the left). The second row displays the expected decreasing trend towards zero of the gradient norm for each learned hyperparameter λ. The last row illustrates learning of each hyperparameter across epochs, stabilizing in the long run at a specific value.