



Expert-elicitation method for non-parametric joint priors using normalizing flows

Florence Bockting¹ · Stefan T. Radev² · Paul-Christian Bürkner¹

Received: 24 November 2024 / Accepted: 6 June 2025

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2025

Abstract

We propose an expert-elicitation method for learning non-parametric joint prior distributions using normalizing flows. Normalizing flows are a class of generative models that enable exact, single-step density evaluation and can capture complex density functions through specialized deep neural networks. Building on our previously introduced simulation-based framework, we adapt and extend the methodology to accommodate non-parametric joint priors. Our framework thus supports the development of elicitation methods for learning both parametric and non-parametric priors, as well as independent or joint priors for model parameters. To evaluate the performance of the proposed method, we perform four simulation studies and present an evaluation pipeline that incorporates diagnostics and additional evaluation tools to support decision-making at each stage of the elicitation process.

Keywords Prior elicitation · Expert knowledge · Joint prior distribution · Normalizing flows · Non-parametric priors

1 Introduction

The Bayesian paradigm offers the possibility to incorporate *prior knowledge* into a *statistical model* through the specification of *prior distributions*. This possibility is a central advantage of the Bayesian paradigm (Mikkola et al. 2023), yet it also presents one of its most challenging aspects (Simpson et al. 2017; Igozata et al. 2015; Van Dongen 2006). In the following, we define prior knowledge as the expertise provided by a *domain expert* — an individual with extensive knowledge of a specific subject matter (Falconer et al. 2022). This knowledge can be represented in various forms, but to integrate it into a Bayesian model, we need to translate it into a formal mathematical language that can be expressed

as a prior distribution over the model parameters (Perepolkin et al. 2023; O'Hagan 2019; Martin et al. 2012; Garthwaite et al. 2005).

A distinct field of research, commonly known as *(expert) prior elicitation*, has developed around the question of how to gather expert knowledge and translate it into appropriate prior distributions. This area of study has a long history, dating back to the 1960s (Winkler 1967; Kadane et al. 1980; Kadane and Wolfson 1998), and continues to be an active area of research today (Stefan et al. 2022; Mikkola et al. 2023; Falconer et al. 2022). Garthwaite et al. (2005) identified four key stages in a *prior elicitation process*:

1. **Setup stage:** In this stage, the problem is defined, an expert is selected, and the quantities to be elicited from the expert (in this paper referred to as *target quantities*) are determined;
2. **Elicitation stage:** Here, the target quantities are queried from the expert using specific elicitation techniques, resulting in what we call *elicited statistics*;
3. **Fitting stage:** This involves fitting a (potentially joint) probability distribution based on the expert-elicited statistics;
4. **Evaluation stage:** Finally, the adequacy of the fitted probability distribution is assessed in collaboration with the expert.

✉ Florence Bockting
florence.bockting@tu-dortmund.de

Stefan T. Radev
stefan.ralev93@gmail.com

Paul-Christian Bürkner
paul.buerkner@gmail.com

¹ Department of Statistics, TU Dortmund University,
Vogelpothsweg 87, 44227 Dortmund, North
Rhine-Westphalia, Germany

² Cognitive Science Department, Rensselaer Polytechnic
Institute, 110 Eighth Street, 12180 Troy, NY, United States

In this context, *elicitation methods* aim to provide a systematic and formal procedure for deriving prior distributions based on expert-elicited statistics. Early elicitation methods primarily tackled the problem by seeking analytical solutions, such as conjugate models or problem-specific transformation functions resulting in highly model-specific prior elicitation methods (see Mikkola et al. 2023, for a recent review). This model dependence is partly due to the use of direct elicitation techniques, which involve asking experts directly about model parameters (Stefan et al. 2022; Falconer et al. 2022). This approach is problematic not only due to its inherent model dependence but also because model parameters are often challenging for domain experts to interpret meaningfully. Consequently, the quality of expert input can be questionable. In response to these limitations, substantial efforts have been made to develop methods that enable more flexible and interpretable prior specification for domain experts.

A recently proposed group of elicitation methods uses advances in machine learning to automate the process of translating expert knowledge into prior distributions (Hartmann et al. 2020; da Silva et al. 2019; Manderson and Goudie 2023; Bockting et al. 2024). These methods address both key challenges: they allow expert knowledge to be expressed in terms of observable quantities and ensure that the elicited information remains interpretable for domain experts. Additionally, since the quantities elicited from the expert are not tied to specific model parameters, these methods are highly versatile and model-independent. The underlying idea behind this approach is closely related to prior predictive checks, which are an integral part of the *Bayesian workflow* (Gelman et al. 2020; Gabry et al. 2019). The key concept is the *prior predictive distribution* (PPD), defined as $p(y) = \int p(y | \theta)p(\theta)d\theta$ with likelihood $p(y | \theta)$ and prior $p(\theta)$. The PPD establishes a formal relationship between the prior distributions of the model parameters and the model predictions, $p(y)$. In this way, the PPD provides a means to link expert knowledge, expressed in terms of observable quantities, to the latent model parameters. Since the PPD is often not analytically tractable, it is typically approximated using Monte Carlo integration (Mikkola et al. 2023) which involves first sampling $\theta' \sim p(\theta)$ and then $y' \sim p(y | \theta')$. The resulting model predictions, y' , are then compared to expert knowledge on the outcome variable, y^* , using an appropriate discrepancy loss function. The goal is to learn prior distributions, $p(\theta)$, that produce model predictions consistent with expert knowledge by means of minimizing the discrepancy loss (Garthwaite et al. 2005; Gelman et al. 2017; Simpson et al. 2017; Betancourt 2020).

The methods introduced so far that use this *simulation-based* approach focus on learning independent, parametric priors $p(\theta | \lambda)$, parameterized by a set of prior hyperparameters λ (Hartmann et al. 2020; da Silva et al. 2019; Manderson

and Goudie 2023). While assuming *independent* priors for model parameters can be reasonable in some cases due to theoretical considerations or model constraints, this approach may not be sufficient for more complex or high-dimensional problems (Gelman et al. 2017, 2020; Simpson et al. 2017). In such cases, it becomes important to account for the *joint distribution* of model parameters to capture dependencies and interactions that reflect the underlying structure more accurately. Similarly, the use of *parametric* prior distribution families can be well-justified in some cases, but may lead to model misspecification in others, prompting the need for *non-parametric* approaches. Elicitation methods that focus on non-parametric priors are less studied, but examples include the use of Gaussian processes (Oakley and O'Hagan 2007) and quantile-parameterized distributions (Perepolkin et al. 2023, 2024).

Building in particular on the work of Hartmann et al. (2020); da Silva et al. (2019); and Manderson and Goudie (2023), we previously introduced a simulation-based framework with a highly modular structure that closely aligns with the elicitation process described by Garthwaite et al. (2005). The advantage of this modular structure is its flexibility, allowing for easy adaptation to different types of methods. In our previous work (Bockting et al. 2024), we demonstrated how to use this framework to learn parametric prior distributions for the model parameters. In this paper, we show how the same simulation-based framework can be used to learn flexible (i.e., non-parametric) joint priors for the model parameters with only minor adjustments to the workflow. In both approaches, we employ mini-batch stochastic gradient descent as the optimization method.

Main Contributions Our motivation is two-fold: First, we aim to promote the development of a unifying, highly modular framework that supports a wide range of prior elicitation methods. We believe that such a framework would benefit both users and developers of prior elicitation methods by providing structure and clarity in a field currently populated with numerous coexisting methods, which can make comparisons challenging. Second, we want to demonstrate how our simulation-based framework, introduced in Bockting et al. (2024), can serve as a flexible approach for learning either parametric or non-parametric, as well as independent or joint, priors with only minor adjustments. While our previous work focused on learning parametric, independent priors, the present work emphasizes learning non-parametric, joint priors.

2 Methodology

In this paper, we propose an elicitation method that extends our recently introduced simulation-based framework (Bock-

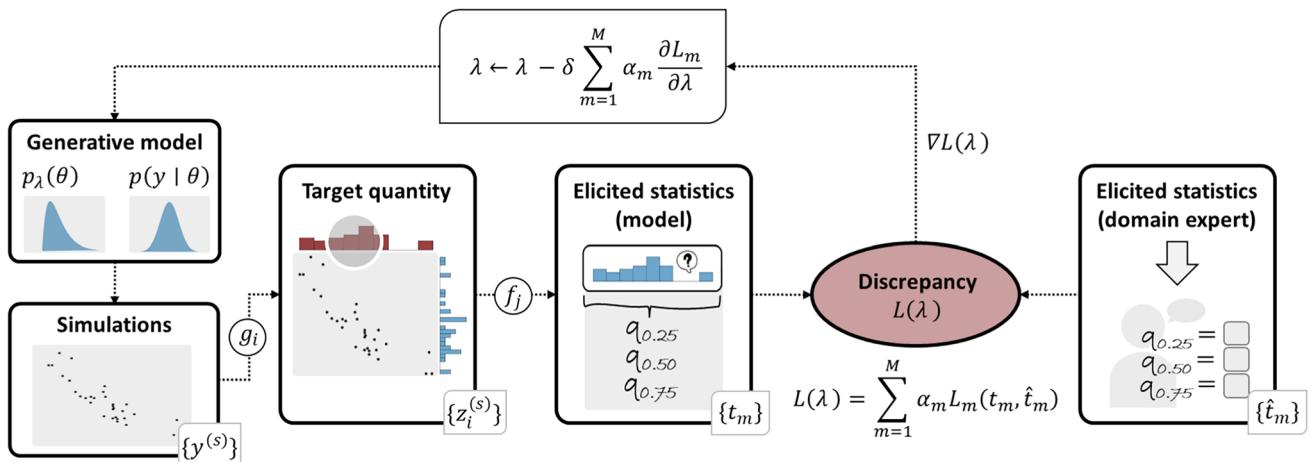


Fig. 1 Graphical illustration of our simulation-based prior-elicitation framework. The process begins by identifying target quantities to be elicited from the domain expert and selecting appropriate elicitation techniques, which result in (expert-)elicited statistics. Next, predictions are simulated from the generative model by sampling from the prior $p_\lambda(\theta)$ and computing the corresponding model-implied target quantities and elicited statistics. The consistency between model-and

expert-elicited statistics is assessed using a loss function L_m , where α_m is the weight of the m^{th} loss component. Hyperparameters λ that define the learned prior are adjusted based on this evaluation to reduce the loss and align model predictions more closely with expert knowledge. This iterative process continues until a prior is found that minimizes the discrepancy between model and expert-elicited statistics. In the updating rule, δ refers to the step size

ing et al. 2024) to support the learning of non-parametric joint priors. The overall workflow remains unchanged; the modifications primarily concern how the prior distributions are specified and learned. We will first provide a brief overview of the workflow to establish the background, followed by a more detailed discussion of the modification introduced to the original framework, which is the focus of the current paper.

The general workflow of the framework closely resembles the approach of *prior predictive checks* (Gelman et al. 2020): We simulate from the joint model $p(\theta, y)$ and assess how well the resulting prior predictions align with the expert's expectations. If there is a discrepancy between the expert's expectations and the model simulations, the prior specification needs to be adjusted accordingly. Figure 1 provides a graphical representation of the framework and Table 1 offers a summary of the symbols and notation used in the following sections. The general workflow of our framework can be summarized as follows:

1. *Define the generative model*: Define the generative model including dimensionality and parameterization of prior distribution(s). (Setup stage; Section 2.1)
2. *Identify variables and elicitation techniques for querying expert knowledge*: Select the set of variables to be elicited from the domain expert (target quantities) and determine which elicitation techniques to use for querying the selected variables from the expert (elicited statistics). (Setup stage; Section 2.2)

3. *Elicit statistics from expert and simulate corresponding predictions from the generative model*: Sample from the generative model and perform all necessary computational steps to generate model predictions (model-elicited statistics) corresponding to the set of expert-elicited statistics. (Elicitation stage; Section 2.3)
4. *Evaluate consistency between expert knowledge and model predictions*: Evaluate the discrepancy between the model- and expert-elicited statistics via a multi-objective loss function. (Fitting stage; Section 2.4)
5. *Adjust prior to align model predictions more closely with expert knowledge*: Use mini-batch stochastic gradient descent to adjust the prior so as to reduce the loss. (Fitting stage; Section 2.4)
6. *Find prior that minimizes the discrepancy between expert knowledge and model predictions*: Repeat steps 2 to 5 iteratively until a prior is found that minimizes the discrepancy between the model and expert-elicited statistics. (Fitting stage; Section 2.4)
7. *Evaluate the learned prior distributions*: Run the learning algorithm (steps 2 to 6) multiple times to obtain a set of prior distributions that can equally well represent the expert data. Select a plausible prior distribution in consultation with the domain expert or apply model averaging techniques. (Evaluation stage; Section 2.5)

The following sections discuss each step in greater detail.

Table 1 Notation and symbols used in this paper

Symbol	Description
Simulation-based prior elicitation	
y_n	data $n = 1, \dots, N$
θ_k	model parameter $k = 1, \dots, K$
λ	model hyperparameter
$p(y \theta)$	likelihood
$p(\theta \lambda)$	prior parameterized by λ
$p(y)$	prior predictive distribution
$z_i = c_i(\lambda)$	simulated model target quantity defined by function c_i
\hat{z}_i	expert representation of the target quantity
$t_m = f_j(z_i)$	model-elicited statistic defined by elicitation technique f_j
\hat{t}_m	expert-elicited statistic
$L_m(t_m(\lambda), \hat{t}_m)$	loss component
$L(\lambda) = \sum_{m=1}^M \alpha_m L_m(t_m(\lambda), \hat{t}_m)$	total loss as weighted sum of L_m with weights α_m
$p_{\lambda_1, \dots, \lambda_R}(\theta) = \sum_{r=1}^R w_r \cdot p_{\lambda_r}(\theta)$	averaged prior across R replications and weighted by w_r with $r = 1, \dots, R$
$\Delta_r(L) = L(\lambda_r) - L(\lambda_{\min})$	difference between total loss of replication r and replication with minimum total loss
$w_r = \frac{\exp\{-\gamma \Delta_r(L)\}}{\sum_{v=1}^V \exp\{-\gamma \Delta_v(L)\}}$	weights used for model averaging with γ being a scaling factor; in all simulation studies $\gamma = 1$.
Normalizing flow	
$u \sim p_U(u)$	samples from base distribution
$g_\lambda(\cdot)$	generator function
$u = g_\lambda(\theta)$	normalizing direction
$\theta = g_\lambda^{-1}(u)$	generative direction
$g_\lambda = g_{\lambda_H} \circ \dots \circ g_{\lambda_1}$	composition of H coupling layers

2.1 Prior Specification & Learning

In our previous work (Bockting et al. 2024), we introduced the simulation-based framework along with a method for learning a set of *independent parametric* prior distributions, $p(\theta_k | \lambda_k)$, for the model parameters θ_k with $k = 1, \dots, K$. In this paper, we extend our framework by introducing a method to learn a *joint non-parametric* prior distribution $p_\lambda(\theta) = p_\lambda(\theta_1, \dots, \theta_K)$ over all model parameters θ . In both the parametric and non-parametric approaches, the objective of the optimization process is to learn the hyperparameters λ that minimizes the discrepancy between the model simulations and the expert expectations. However, the interpretation of these hyperparameters differs between approaches, which we emphasize by using a different notation for the prior distributions. In the non-parameteric approach, we employ *normalizing flows* (NFs) to induce a flexible family of prior distributions which entail specialized deep neural networks with trainable parameters λ (Kobyzev et al. 2020). Thus, in the non-parametric approach, λ represents the weights of the deep neural networks within the NFs, whereas in the parametric approach, λ denotes the parameters of the associated prior distribution families (i.e., the prior hyperparameters).

A more detailed introduction to NFs will be provided in the upcoming section.

In summary, the main difference between the parametric approach (proposed by Bockting et al. (2024)) and the non-parametric approach (focus of the current paper) is that, in the former, the user must specify a parametric prior distribution family for each model parameter, and the corresponding prior hyperparameters are learned via stochastic optimization. In the new extension, the user no longer needs to specify a parametric prior for the model parameters. Instead, a complex non-parametric prior distribution is assumed. This non-parametric joint prior is represented and learned using deep neural networks (specifically, NFs), with the weights learned via stochastic optimization. Table 2 summarizes the main steps in the elicitation process and highlights where the extension modifies the existing workflow of our simulation-based framework. Furthermore, we provide in this paper additional diagnostics to support users in interpreting the simulation results.

Normalizing Flows NFs transform a simple probability distribution, known as the *base distribution* $p(u)$, into a more complex *target distribution*, which, in our case, is the joint

Table 2 Comparison between the deep-prior and parametric-prior approach on a conceptual level. Note: The symbol ‘%’ refers to the same content as provided in parametric-prior approach. NFs is an abbreviation for Normalizing Flows (see text for details)

Elicitation process	parametric-prior approach	deep-prior approach
Setup		
<i>Model</i>	specify generative model	%
<i>Priors</i>	specify parametric prior distribution families	assume non-parametric joint prior (specified via NFs)
<i>Data</i>	specify target quantities and elicited statistics (less expert input needed due to stronger regularized priors)	% (more expert input required (e.g., dependency information) due to flexible prior)
<i>Elicitation</i>	specify elicitation techniques used to query the domain expert	%
<i>Fitting</i>	apply mini-batch stochastic gradient descent	%
<i>Evaluation</i>	provide diagnostics, visualizations, etc. to evaluate simulation results	%

prior $p_\lambda(\theta)$. This transformation is achieved through a series of invertible and differentiable mappings g_λ , parameterized by λ (Kobyzev et al. 2020). The composition of invertible functions g_λ yields an explicit form of the density $p_\lambda(\theta)$ via the change of variables formula

$$p_\lambda(\theta) = p(u = g_\lambda(\theta)) \mid \det g'_\lambda(\theta) \mid,$$

where $g'_\lambda(\theta) = \frac{\partial}{\partial \theta} g_\lambda(\theta)$ is the Jacobian matrix of g_λ at θ and $\mid \det g'_\lambda(\theta) \mid$ is its absolute determinant. This formula allows us to easily evaluate the prior density at arbitrary points of interest θ (Dinh et al. 2016). Obtaining samples from the joint prior can be achieved by first sampling from the base distribution, $p(u)$, and then applying the inverse $g_\lambda^{-1}(u)$ to obtain samples from $p_\lambda(\theta)$:

$$\theta = g_\lambda^{-1}(u) \sim p_\lambda(\theta) \quad \text{for } u \sim p(u).$$

Typically, $p(u)$ is chosen to be a simple density, such as a standard Gaussian or a uniform distribution (Kobyzev et al. 2020).

For NFs to be practical, the mapping g_λ should be easily invertible, computationally efficient, and sufficiently expressive to model any target distribution of interest (Kobyzev et al. 2020). Several types of NFs that meet these requirements have been proposed, including affine (Dinh et al. 2014) and spline coupling flows (Durkan et al. 2019). A detailed discussion and introduction to coupling flows can be found in various sources (Draxler et al. 2024; Dinh et al. 2014, 2016; Radev et al. 2020; Kobyzev et al. 2020; Papamakarios et al. 2021; Bond-Taylor et al. 2021). Due to their simple architecture, affine coupling flows are easy to invert and computationally efficient, although they have restricted expressiveness (Bond-Taylor et al. 2021). However, the expressiveness of a coupling flow can be increased by stacking multiple affine coupling layers, $g_\lambda = g_{\lambda_H} \circ \dots \circ g_{\lambda_1}$ with $h = 1, \dots, H$ coupling blocks (Dinh et al. 2016). In fact,

Draxler et al. (2024) prove that affine coupling flows are distributional universal approximators, capable of representing any target distribution despite their seemingly restrictive architecture. In line with these considerations and the strong empirical performance of affine coupling flows, we focused on this type of normalizing flows in our simulation studies (see Section 4), using the implementation provided by BayesFlow (Radev et al. 2023).

Finally, it is worth noting that, since our focus is solely on the mapping from the base distribution to the target distribution (i.e., the generative direction of NFs), virtually any other generative model family could be used to learn how to generate samples from a joint prior. Examples of common generative architecture alternatives include diffusion models (Cao et al. 2024) and flow matching (Lipman et al. 2022). However, we used NFs here because they perform well in practice and are computationally efficient. In particular, they do not require solving differential equations during sampling (Draxler et al. 2024).

2.2 Target Quantities & Elicitation Techniques

A key task in the setup stage of the elicitation process is the selection of target quantities and elicitation techniques to effectively gather the expert information. Two main criteria should guide this selection: *interpretability* and *informativeness* (Mikkola et al. 2023; Crowder 1992; da Silva et al. 2019; Garthwaite et al. 2005).

Selecting Target Quantities *Interpretability* refers to the selection of target quantities that allows domain experts to meaningfully express their knowledge based on their experience and expertise (Garthwaite et al. 2013). Target quantities on the same scale as the outcome variable (cf., *observable* quantities) are considered to be highly interpretable (da Silva et al. 2019; Manderson and Goudie 2023). In contrast, target quantities that refer to model parameters are more challenging to interpret, especially when the outcome variable is not

on the same scale as the model parameters (Kadane and Wolfson 1998; Kadane et al. 1980). Therefore, it is sometimes advocated asking experts exclusively about observable quantities (Kadane and Wolfson 1998). We share the view of Denham and Mengersen (2007) that the appropriateness of either approach depends on the specific problem being addressed and the type of expert providing the information.

Guided by this idea, we define a target quantity within our framework in the most general sense as a function of the trainable hyperparameters of the prior: $z_i = c_i(\lambda)$ for $i = 1, \dots, I$. Typically, multiple target quantities (I in total) are elicited from the user to inform different aspects of the data-generating process. For instance, if we want to query the expert regarding the outcome variable (i.e., the observable space), one of the target quantities can be simply defined as $z_i = y$. This is obtained by defining the function c_i as a query to the prior predictive distribution $p(y) = \int p(y | \theta)p_\lambda(\theta)d\theta$, where the likelihood is marginalized over the prior. Note that this expression is not just a constant as it refers to the prior predictive distribution and should not be confused with the marginal likelihood of observed data (see for this point also Hartmann et al. (2020)). As another example, expert knowledge may be elicited about the parameter space. In this case, the target quantity is defined as $z_i = \theta_k$, which is obtained by defining c_i as a simple projection onto θ_k . In this way, we can also accommodate any other target quantity that can be derived from the model parameters or the data (e.g., R^2 ; the proportion of variance explained by the model; (Gelman et al. 2019)).

Selecting Elicitation Techniques Once the set of target quantities has been selected, we need to determine how to elicit this information from the expert, which requires choosing an appropriate *elicitation technique*. While experts can be asked to provide the full distribution of a target quantity (e.g., in the form of a histogram (Johnson et al. 2010)), another common technique is to focus on specific summary statistics (Morris et al. 2014). Research on prior elicitation suggests that experts can reasonably be asked to describe a target quantity using proportions, the mode, the median, and generally quantiles (Garthwaite et al. 2005; Stefan et al. 2022; O'Hagan et al. 2006), with quantile-based elicitation techniques being particularly recommended (Kadane and Wolfson 1998). In contrast, asking about the mean is less advisable when the distribution is skewed (Peterson and Miller 1964), and similarly, querying variances is in general discouraged (Garthwaite et al. 2005; Kadane and Wolfson 1998). Furthermore, assessing dependencies between variables is challenging, making the elicitation of joint priors especially difficult (Garthwaite et al. 2005). Elicitation of correlations in the context of joint priors has been studied among others by Gokhale and Press (1982); Clemen et al. (2000), and Dickey et al. (1985) as reviewed by Mikkola et al. (2023).

In our elicitation method, we represent an elicitation technique as a function f_j for $j = 1, \dots, J$, of a target quantity z_i . The resulting set of *elicited* target quantities is referred to as *elicited statistics*, $\{t_m(\lambda)\}$, where $t_m(\lambda) = f_j(z_i)$ and m refers to the corresponding $i \times j$ combination. In the following, we sometimes omit the explicit dependence on λ and simply write t_m instead of $t_m(\lambda)$. Note also that, depending on the elicitation technique employed, t_m may accept either a single value or a set of values as input. To illustrate this point more clearly, consider the case in which we elicit expert knowledge about the predictive distribution conditional on a category of a categorical predictor, i.e., $z = y | \text{cat}$. To elicit this information, we query the expert regarding the 25%, 50%, and 75% quantiles of this predictive distribution. Thus, we have $t = \{Q_{25\%}(y | \text{cat}), Q_{50\%}(y | \text{cat}), Q_{75\%}(y | \text{cat})\}$.

Sensitivity Analysis In addition to interpretability, the second criterion for selecting target quantities and elicitation techniques is *informativeness*, which refers to the relevance of the elicited statistics for learning the prior distributions. Evaluating this criterion is nontrivial because the relationship between the elicited statistics and the model (hyper-)parameters is complex and often analytically intractable. One computational approach to assess informativeness is to conduct a *sensitivity analysis* (Depaoli et al. 2020). In this approach, we systematically vary one aspect of the prior distribution at a time while keeping all other aspects constant. For each specific change in the prior, we run the generative model in forward mode by sampling from the prior, then from the likelihood, and subsequently computing the target quantities and elicited statistics. Sensitivity analysis enables us to assess how changes in one aspect of the prior distribution impact each elicited statistic. If we observe no variation in a particular elicited statistic for a given prior variation, this statistic does not provide any useful information for the learning algorithm to determine the corresponding aspect of the prior. This approach can help inform the selection of elicited statistics and identify which aspects of the prior are most likely to remain unidentifiable given the current set of expert information.

2.3 Model-implied & Expert-elicited Statistics

After determining the set of target quantities and corresponding elicitation techniques, we can proceed to gather information from the expert through specific *prior elicitation protocols* (Gosling 2018; Cooke 1991; European Food Safety 2014). This process may be preceded by a training phase to familiarize the expert with the elicitation tasks (Stefan et al. 2022). In prior elicitation research, an entire subfield has focused on determining how to conduct optimal expert interviews, taking into account factors such as cognitive biases

and heuristics. While it is beyond the scope of this paper to review this literature, several comprehensive reviews are available (e.g., Mikkola et al. 2023; Stefan et al. 2022; Falconer et al. 2022). At this point, we will skip the actual interrogation process and assume that the expert information has been successfully gathered, providing us with a set of expert-elicited statistics, individually denoted by \hat{t}_m and together as $\{\hat{t}_m\}$.

Given the set of expert-elicited statistics, $\{\hat{t}_m\}$, we can evaluate the discrepancy between the expert expectations and the model-implied statistics, $\{t_m\}$. The latter are computed by simulating from the model. This proceeds as follows: We start with an arbitrary initial prior distribution from which we sample model parameters $\theta^{(s)} \sim p_{\lambda_0}(\theta)$, where λ_0 represent the initial hyperparameter values that define the prior distribution. The superscript s denotes the s^{th} sample out of a total of S samples. Subsequently, we perform all necessary computational steps to derive samples from the target quantities $z_i^{(s)}$. Finally, we apply the corresponding elicitation technique to obtain the model-implied statistics $t_m = f_j(\{z_i^{(s)}\})$ from the simulations.

2.4 Discrepancy Evaluation & Training

To evaluate the discrepancy between the model- and expert-elicited statistics, an appropriate loss function is used for each statistic, $L_m(t_m(\lambda), \hat{t}_m)$. Different loss functions may be preferable depending on the type of the respective elicited statistic. Since the discrepancy is computed for each statistic within the set of elicited statistics, the total loss comprises multiple *loss components*, which are combined into a single total loss using a weighted sum:

$$L(\lambda) = \sum_{m=1}^M \alpha_m L_m(t_m(\lambda), \hat{t}_m), \quad (1)$$

where α_m denote the weights. The choice of weights α_m can be guided by different considerations: One approach involves manually adjusting the weights, guided either by the expert's assessment of the importance of individual loss components or by ensuring that the effects of different magnitudes, arising from varying scales, are balanced, such that no single loss component dominates the others (Wang et al. 2011). Alternatively, automated weighting methods, known as loss-balancing methods, can be used to optimally balance the contribution of each individual term to the total gradient (Liu et al. 2019; Crawshaw 2020; Bischof and Kraus 2021).

After computing the total loss, $L(\lambda)$, the gradients with respect to the hyperparameters λ are calculated. These gradients are then used to adjust the hyperparameters in the opposite direction:

$$\lambda' \leftarrow \lambda - \delta \sum_{m=1}^M \alpha_m \frac{\partial L_m}{\partial \lambda}, \quad (2)$$

with δ being the step size (Goodfellow et al. 2016). We employ mini-batch stochastic gradient descent (SGD) with automatic differentiation, facilitated by the (explicit or implicit) reparameterization trick (Kingma and Welling 2014; Figurnov et al. 2018). With the adjusted hyperparameters λ' , new samples from the prior $p_{\lambda'}(\theta)$ can be generated and the updated model-implied statistics $t_m(\lambda')$ computed. We then reassess the discrepancy between the model-implied and expert-elicited statistics and continue to adjust the hyperparameters as needed. This iterative process continues until a convergence criterion is met or a stopping rule is applied.

2.5 Evaluation of Training Results

Convergence Checks Training is considered successful if the loss is minimized and no further learning occurs. With a proper discrepancy measure as loss function, it is guaranteed that the total loss (and the individual loss components) approach zero as learning progresses. The learning progress can be tracked by examining the loss behavior across epochs, where a decreasing trend is expected until the loss stabilizes at a value close to zero. Further insights into the learning progress can be obtained by tracking the convergence of additional quantities of interest beside the loss, for instance the mean and standard deviation of the marginal priors.

A further method to assist in the examination of convergence is to compute the slope of a linear regression fitted to the loss values over the last m epochs, whereby m should not be taken too large such that the linearity assumption holds. Typically, the overall loss function exhibits exponential decay, but once the loss stabilizes, a linear trend is a good approximation. Ideally, a slope of zero would indicate perfect convergence, though some variation is expected in practice. To evaluate whether the slope remains acceptable, the training algorithm can be run repeatedly with random seeds and the absolute slopes across all replications can be compared. For training runs with the highest final slopes, we can perform a visual inspection of the trajectories of key quantities across epochs, such as the total loss, loss components, and prior means and standard deviations. If these runs show successful convergence, we can reasonably conclude that other runs (with smaller final slopes) have also converged.

Non-uniqueness of Learned Priors Once the elicited statistics have been learned accurately, we can examine the corresponding learned prior distribution. As there are often multiple optimal values λ^* that align closely with the same set of elicited statistics, multiple replications will yield different prior distributions (da Silva et al. 2019; Manderson and Goudie 2023; Stefan et al. 2022). This lack of uniqueness

is anticipated, given the limited information provided by the set of elicited statistics compared to the complexity of the generative model (Manderson and Goudie 2023). Furthermore, when target quantities refer solely to the observable space, we only gain access to the model's overall uncertainty with limited ability to differentiate between aleatoric and epistemic uncertainty (Nemani et al. 2023; Perepolkin et al. 2024). Aleatoric uncertainty, by definition, is irreducible; however, several methods exist to reduce epistemic uncertainty. These include (i) selecting more informative target quantities, (ii) eliciting more detailed expert information (e.g., increasing the number of quantiles in quantile-based elicitation), (iii) adding theory-informed regularization terms to constrain the search space (as illustrated in Manderson and Goudie (2023)), (iv) improving the initialization method to achieve better starting points, and (v) employing more advanced optimization algorithms when dealing with a large number of model parameters (Nemani et al. 2023).

Model Averaging Given the (potential) non-uniqueness of the learned prior distributions, it is advisable to run the training multiple times on the same expert data but with different random seeds. This approach helps provide insight into the variability among the learned prior distributions. From the resulting set of candidate priors $p_{\lambda_r}(\theta)$ for $r = 1, \dots, R$, we can, in consultation with the domain expert, exclude prior distributions that do not meet the expert's expectations. For the remaining set of prior distributions, which may be considered practically equivalent from the expert's perspective, either a single prior distribution can be chosen, or alternatively, model averaging techniques can be applied. In the case of model averaging, the combined prior would take the form:

$$p_{\lambda_1, \dots, \lambda_R}(\theta) = \sum_{r=1}^R w_r \cdot p_{\lambda_r}(\theta) \quad (3)$$

where the weights w_r sum to one. The weights w_r can either be assigned by the user, based on theoretical assumptions, or selected according to one of several proposed weighting schemes (Claeskens and Hjort 2008). Here, we apply an approach suggested by Buckland et al. (1997) for the general purpose of model averaging and selection, where the weights w_r are defined as

$$w_r = \frac{\exp\{-\gamma I_r\}}{\sum_{v=1}^V \exp\{-\gamma I_v\}}.$$

In the above expression, γ is a scaling factor, and I represents the performance criterion by which the models are weighted. This definition ensures that models with identical performance receive equal weight (Buckland et al. 1997). In our setting, it is sensible to use the total loss $L(\lambda_r)$ of each

replication r as the performance criterion. Consequently, the "best" model is the one with the smallest total loss, $L(\lambda_{\min})$ (Bissiri et al. 2016). This yields the following definition of weights

$$w_r = \frac{\exp\{-\gamma \Delta_r(L)\}}{\sum_{v=1}^V \exp\{-\gamma \Delta_v(L)\}}. \quad (4)$$

where $\Delta_r(L) = L(\lambda_r) - L(\lambda_{\min})$. The difference is used primarily for computational reasons to ensure numerical stability (Claeskens and Hjort 2008). The formulation in Eq. (4) can be interpreted as the probability of replication r representing the best prior, given the collection of learned prior distributions (Wagenmakers and Farrell 2004). The scaling factor γ controls the influence of each replication's relative performance $\Delta_r(L)$. When $\gamma > 1$, discrepancies in model performance are amplified, with better models receiving higher weights and worse models receiving lower weights. As $\gamma \rightarrow \infty$, this weighting would approach a one-hot vector, effectively performing model selection. Conversely, when $\gamma < 1$, the performance differences between models are smoothed out, reducing the influence of performance discrepancies.

Simulating From an "Oracle" The simulation-based framework enables us to evaluate the degree of non-identifiability and determine if further regularization is required, even before querying any data from the expert. An effective approach involves using an *oracle*, where a pre-defined prior distribution serves as the *ground truth*. From this "true" prior, we perform forward simulations: sampling first from the prior, then from the likelihood, and carrying out all necessary computational steps to generate a set of model-implied statistics corresponding to the ground truth. This set of simulated statistics can then serve as a proxy for the "expert-elicited" statistics.

This *oracle* approach allows us to assess the self-consistency of the algorithm, the degree of non-identifiability (i.e., how informative are the elicited statistics), and the suitability of the chosen algorithm parameters used for training NPs with mini-batch SGD (such as the learning rate or the number of epochs). However, there is one important caveat when using an *oracle*: the user might be tempted to compare the learned prior distributions against the true prior distributions (which are now available). Specifically, if there is an unsatisfactory match between the true and learned prior distributions, the user might interpret this mismatch as a failure in learning. It is important to remember, however, that the method only knows about the elicited statistics, which are incorporated into the loss function. Thus, when the elicited statistics do not incorporate information about the model parameters but instead pertain to observable quantities, there is, from the method's perspective, no concept of a true prior. In particular, if the true and learned elicited statistics match

but there is a mismatch in the learned prior, this is likely a result of an underidentified problem, and the objective function requires additional information to properly constrain the problem.

3 Implementation

In all simulation studies, we use a batch size of 128. To simulate from an oracle (discussed in Section 4.1), we draw once $S = 10,000$ samples from the true joint prior and then compute the corresponding set of “expert”-elicited statistics. During training with mini-batch SGD we draw $S = 200$ samples from the joint prior. To evaluate the discrepancy between the model-implied and expert-elicited statistics, we use a squared, biased *Maximum Mean Discrepancy* (MMD) loss with an energy kernel (Gretton et al. 2006; Feydy et al. 2019), following our previous success with this metric (Bockting et al. 2024). For the correlation between parameters, however, we apply a squared-error loss (L2 loss; for details see Section 4.1). The total loss is computed as a weighted sum of the individual loss components. The values of the respective weights are discussed together with the selection of the elicited statistics whereby each elicited statistic represents one loss component (see Section 4.1).

The architecture of the NFs is rather simple compared to typical applications in deep learning (e.g., Kingma and Dhariwal 2018,) and consists of a standard multivariate Gaussian as a base distribution and three affine coupling blocks. Each coupling block consists of two dense layers with 128 units and ReLU activation functions. In Simulation Study 1, the number of learnable hyperparameters is $|\lambda| = 202,776$, while in the case studies corresponding to Simulation Study 2, it is $|\lambda| = 205,872$.

In Simulation Study 1, we assume a discrete likelihood and use the gumble-softmax trick to compute the gradients of a discrete random variable (for more information, see Bockting et al. 2024). The gradients of the total loss with respect to the hyperparameters are used to update the hyperparameters λ . Optimization was performed with the Adam optimizer for 500 epochs in Simulation Study 1 and 800 epochs in Simulation Study 2. We used a fixed learning rate of 0.00025 for Simulation Study 2, Scenario 1 & 2, and 0.0001 for Simulation Study 1 and Stimulation Study 2, Scenario 3. These algorithm (hyper-)parameter values were obtained by manual tuning until good performance across all of our case studies was achieved.

All simulations and plots were performed in Python 3.11 using our Python package *elicito* 0.3.1 (Bockting and Bürkner 2025), which relies, among others, on the following libraries: TensorFlow 2.14 (Abadi et al. 2016), TensorFlow Probability 0.23 (Dillon et al. 2017), NumPy 1.26 (Harris et al. 2020), pandas 2.2.3 (McKinney et al. 2010), and

BayesFlow 1.1.6 (Radev et al. 2023). To ensure the reproducibility of our results, the specific version of *elicito* used in this paper has been archived on Zenodo and is accessible via the following DOI <https://doi.org/10.5281/zenodo.15241853>. Simulations were executed on a YOGA Pro 9i laptop equipped with GPU acceleration.

4 Simulation Studies

4.1 General Setup

In the following, we present four case studies organized across two simulation studies to evaluate the performance of our elicitation method. The selection of case studies is motivated by the desire to choose scenarios that are simple and allow us to study the behavior of the method in a well-controlled setting. Since this paper presents the extension with NFs for the first time, it is important for us to first develop an understanding of the method’s behavior. In future work, we will examine the behavior of our method in more complex models that are closer to real-world applications.

In Simulation Study 1, we introduce a simple Binomial regression model with one continuous predictor. In Simulation Study 2, we present three scenarios based on a normal regression model with a three-level categorical predictor. The three normal scenarios differ only in the pre-defined joint prior representing the ground truth. Table 3 provides an overview of the generative models and the joint priors representing the ground truth used in the simulation studies. Before introducing the simulation studies, we will discuss the decisions made in the simulation setup in greater detail. All code and results can be found on GitHub (<https://github.com/florence-bockting/non-parametric-prior>) and OSF (<https://osf.io/xrzh6/>), respectively. Information about the training time for each simulation study is provided in Appendix A.1.

Simulated Expert as Ground Truth For the simulation studies in this paper, we simulate expert input using synthetic data generated from a predefined joint prior representing the ground truth (i.e., oracle, see Section 2.5 for details). For simplicity, we use a parametric prior with fixed hyperparameter values as ground truth for each case study. Further details on the exact specification of the “true” joint prior is provided in the respective section and in Table 3.

Selection of Target Quantities and Elicitation Techniques Following recommendations in the field, we tried to select mainly observable quantities as target quantities for our simulation studies (see Section 2.2 for details). For the Binomial regression model, we used the expected observations of the outcome variable conditional on two observations of the continuous predictor, $y | x_0$ and $y | x_1$. For the normal regression model, we considered the prior predictive observations con-

Table 3 Overview of generative models and ground truth used in each simulation study. Symbols: \mathbf{R} denotes the correlation matrix and s the vector of standard deviations

Model	Generative Model	Ground Truth
Binomial — M1 (Section 4.2)	$y_i \sim \text{Binomial}(p_i, 30)$ $p_i = \text{sigmoid}(\beta_0 + \beta_1 x_i)$	$\beta_0 \sim \text{Normal}(0.1, 0.1)$ $\beta_1 \sim \text{Normal}(-0.1, 0.3)$
Normal models		
Scenario 1 — M2 (Section 4.3.1)	$y_i \sim \text{Normal}(\mu_i, \sigma)$ $\mu_i = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i}$	$\beta_0 \sim \text{Normal}(10, 2.5)$ $\beta_1 \sim \text{Normal}(7, 1.3)$ $\beta_2 \sim \text{Normal}(2.5, 0.8)$ $\sigma \sim \text{Gamma}(5, 2)$
Scenario 2 — M3 (Section 4.3.2)	see M2	$\beta_0 \sim \text{Normal}(10, 2.5)$ $\beta_1 \sim \text{SkewNormal}(7, 1.3, 4)$ $\beta_2 \sim \text{SkewNormal}(2.5, 0.8, 4)$ $\sigma \sim \text{Gamma}(5, 2)$
Scenario 3 — M4 (Section 4.3.3)	see M2	$\beta \sim \text{Mv-Normal} \left(\begin{bmatrix} 10 \\ 7 \\ 2.5 \end{bmatrix}, D(s) \mathbf{R} D(s) \right)$ $\mathbf{R} = \begin{bmatrix} 1. & 0.3 & -0.3 \\ 0.3 & 1. & -0.2 \\ -0.3 & -0.2 & 1. \end{bmatrix}$ $s = (2.5, 1.3, 0.8)$ $\sigma \sim \text{Gamma}(5, 2)$

Table 4 Overview of the analysis workflow used in the simulation studies

Task	Approach	Insights
Setup & Elicitation Stage		
Select the set of elicited statistics and evaluate its informativeness.	(1) Simulate elicited statistics by defining a true joint prior (i.e., the oracle). (2) Conduct a sensitivity analysis.	Can the set of elicited statistics provide sufficient information to learn the key aspects of the joint prior?
Fitting Stage		
Evaluate variability in learned priors to assess non-identifiability.	Run the training algorithm several times with different random seeds.	Do different training runs yield different results?
Verify convergence of the training algorithm.	(1) Examine the slope of final loss values across replications. (2) Visually check the convergence of key quantities for selected seeds.	Will training the model for additional epochs lead to a practically relevant improvement in performance or a reduction in total loss?
Verify accurate learning of expert data.	Visually compare model-implied and expert-elicited statistics.	Was the method successful in accurately capturing the expert data?
Evaluation Stage		
Examine the learned priors across the different seeds	Visually inspect the marginal priors and correlations between model parameters.	How much do the learned priors vary between different seeds? Were any degenerate prior distributions learned? Do we observe any form of multi-modality? Can any of the learned priors be ruled out as unreasonable post hoc?
Obtain the final learned prior for subsequent analysis.	Perform model averaging using weights based on relative loss performance.	Instead of selecting a single prior, compute a model average over practically equivalent priors to account for additional variability due to non-identifiability.

ditional on three groups of the categorical predictor, $y \mid \text{gr}_i$ for $i = 1, 2, 3$.

However, we had to deviate from the recommendation to select observable target quantities in two respects. First, we included R^2 as an additional target quantity for the normal model (Simulation Study 2), motivated by the following reasoning: if we only have information about the outcome variable for each of the three groups, we provide the learning algorithm with no means to differentiate between parameter uncertainty and data uncertainty. The coefficient of determination, R^2 , represents the ratio of parameter uncertainty to total uncertainty (which encompasses both data and parameter uncertainty). Consequently, it offers more information for the learning algorithm to differentiate between these two types of uncertainty, even though it may be somewhat more challenging for a domain expert to interpret. However, as R^2 is commonly reported as a metric in a regression context, we would argue that this quantity is still well understood by domain experts.

Second, we included the pairwise correlation between model parameters in the set of target quantities for each case study. If independence between model parameters is assumed, all correlations are set to zero; otherwise, we use the exact correlation structure indicated by the ground truth. We acknowledge that this information cannot be reasonably demanded from a domain expert in most cases. However, we currently lack suitable elicitation methods for asking domain experts about interpretable quantities that provide sufficient information regarding the correlation structure between the model parameters. A future task will be to develop such elicitation methods and to then incorporate these methods into our computational framework.

Regarding the selection of *elicitation techniques*, we follow the recommendation to query quantiles from a domain expert (Kadane and Wolfson 1998). For each target quantity, except for the correlation information, we elicit five quantiles: 5%, 25%, 50%, 75%, 95%. For the correlation values, we assume that the expert provides a single point estimate corresponding to a moment-based elicitation approach.

In summary, the elicited statistics for Simulation Study 1 consist of five quantiles each for $y \mid x_0 (t_1)$ and $y \mid x_1 (t_2)$, as well as the correlation information $\rho(\beta_0, \beta_1) (t_3)$. Each of these three elicited statistics corresponds to a distinct loss component in the objective function. The elicited statistics for Simulation Study 2 consist of five quantiles each for $R^2 (t_1)$ and $y \mid \text{gr}_i$ with $i = 1, 2, 3 (t_2, t_3, t_4)$, as well as the correlation information between the model parameters (t_5). The discrepancy in the correlation values is measured using an L2 loss, whereas the discrepancies for the remaining components are computed using the MMD² loss. To account for differences in scale-arising both from the use of different discrepancy measures and from the varying domains of the elicited statistics-we weight the individual loss components

as follows: the correlation loss component is scaled by a factor of 0.1, the R^2 loss component by a factor of 10.0, and all other loss components are uniformly weighted with a factor of 1.0.

To assess whether the selected set of elicited statistics is sufficiently informative to determine a prior distribution for the model parameters, we conduct a *sensitivity analysis* for each case study.

Evaluation of Simulation Results We run each simulation study 30 times with different random seeds. As preliminary convergence checks, we calculate the slope of the loss trajectory over the last 100 epochs and perform a visual convergence check for the five replications with the highest absolute slopes (i.e., worst cases). Additionally, we examine the trajectories of the total loss, the individual loss components related to each elicited statistic, as well as the mean and standard deviation for each marginal prior across epochs. Furthermore, to visually assess whether the learning criterion has been accurately captured, we compare the final model-implied statistics with the expert-elicited statistics. Finally, we examine the learned prior distributions for each replication and compute the model average across the 30 learned prior distributions. Table 4 provides a summary of the analysis workflow followed in each of the subsequent simulation studies.

Additional simulation results For each case study, we conduct two additional sets of simulations, which are not part of the main analysis but are included in the supplementary material for completeness and further comparison. The supplementary material can be found on GitHub at <https://github.com/florence-bockting/non-parametric-prior>. As these simulations are complementary to the main focus of the paper, we do not report the results in the main text.

In the first additional set of simulations, we re-ran all case studies using the *model parameters* as the “elicited statistics” (instead of an indirect quantity derived from the model parameters). In this case, the method trains directly on the model parameters, enabling us to assess whether it is, in principle, capable of recovering the true joint prior distribution when complete information is available. Thus, this set of simulations serves as a validation set for our method.

In the second additional set of simulations, we applied the parametric-prior method proposed in Bockting et al. (2024) to all case studies. This allows for a direct comparison between the performance of the proposed non-parametric approach and the previously introduced parametric variant within the same simulation-based framework. The only exception is Scenario 3 of Simulation Study 2, where we introduce a multivariate normal prior. In this case, a comparison is currently not possible, as the optimization of a full covariance matrix is not yet implemented in *elicto* (Bockt-

ing and Bürkner 2025), but it is planned for a (near) future release.

4.2 Simulation Study 1: Binomial Likelihood with Independent Normal Priors

We introduce the general idea of our method using a Binomial model with a logit link (sigmoid response function) and a single continuous predictor, x . The predictor was generated by first constructing a sequence from 1 to 50, which was then scaled by its standard deviation. In the following we refer to this model as M1:

$$\begin{aligned} y_i &\sim \text{Binomial}(p_i, 30) \\ p_i &= \text{sigmoid}(\beta_0 + \beta_1 x_i) \\ \beta_0, \beta_1 &\sim p_\lambda(\beta_0, \beta_1) \\ \theta &\equiv (\beta_0, \beta_1) \end{aligned} \quad (\text{M1})$$

The goal is to learn a joint prior for the model parameters β_0 (intercept) and β_1 (slope), assuming independence between these parameters.

Setup & Elicitation Stage As elicited statistics, we select prior predictions for two values of the continuous predictor, $y \mid x_0$ and $y \mid x_1$. The two values of x correspond to the 25% and 75% quantiles of the predictor. We select specific quantiles rather than randomly sampling two observations from the predictor to avoid the observations being too close together, which would reduce informativeness. From the distribution of the two selected observations, we compute five quantiles which represent the elicited statistics (see Section 4.1 for details). Note that $y \mid x_0$ and $y \mid x_1$ each have an associated distribution, resulting from simulating $y^{(s)}$ for S samples drawn from the joint prior and conditioning on the respective x value. The distribution over $y \mid x$ encodes the parameter uncertainty.

To obtain the “expert”-elicited statistics, we define a true prior that represents the ground truth and simulate from the generative model in forward mode, computing the corresponding true-elicited statistics. The *true* joint prior is defined by independent normal distributions for each model parameter: $\beta_0 \sim \text{Normal}(0.1, 0.1)$ and $\beta_1 \sim \text{Normal}(-0.1, 0.3)$.

To assess the *informativeness* of the selected set of elicited statistics, we conduct a sensitivity analysis, with the results shown in Figure 2. In each row, a single hyperparameter is varied across the range shown on the x-axis, while all other hyperparameters are held constant at their true values, indicated by the red vertical line. The columns present the two elicited statistics, specifically the five quantiles (in shades of blue and green) for $y \mid x_0$ and $y \mid x_1$. The upper two rows show results for the intercept parameter β_0 and the lower two rows for the slope parameter β_1 . The results of the sensitivity

analysis indicate that the selected elicited statistics provide sufficient informativeness for the model hyperparameters.

Fitting Stage We ran the training algorithm 30 times with different random seeds and conducted an initial convergence check by inspecting the slope of the last 100 epochs for each replication. In Figure 3, the upper row shows the absolute slope (scaled by a factor of 100) for each replication. In the lower row, the learning trajectory of the total loss is shown for the best-performing (leftmost) and the four worst-performing replications. We observe a near-zero slope for the best model, indicating convergence, while the negative slopes for the worst-performing seeds suggest that the learning algorithm continues to make progress.

However, a visual inspection of the learning trajectories of the total loss and individual loss components (Figure 4) and the means and standard deviations of the marginal priors (Figure 5), indicates satisfactory convergence. Therefore, although the slope analysis indicates minor ongoing learning progress, no substantial changes in the results are expected if the model is trained for additional epochs. Overall, we conclude that the training process was successful.

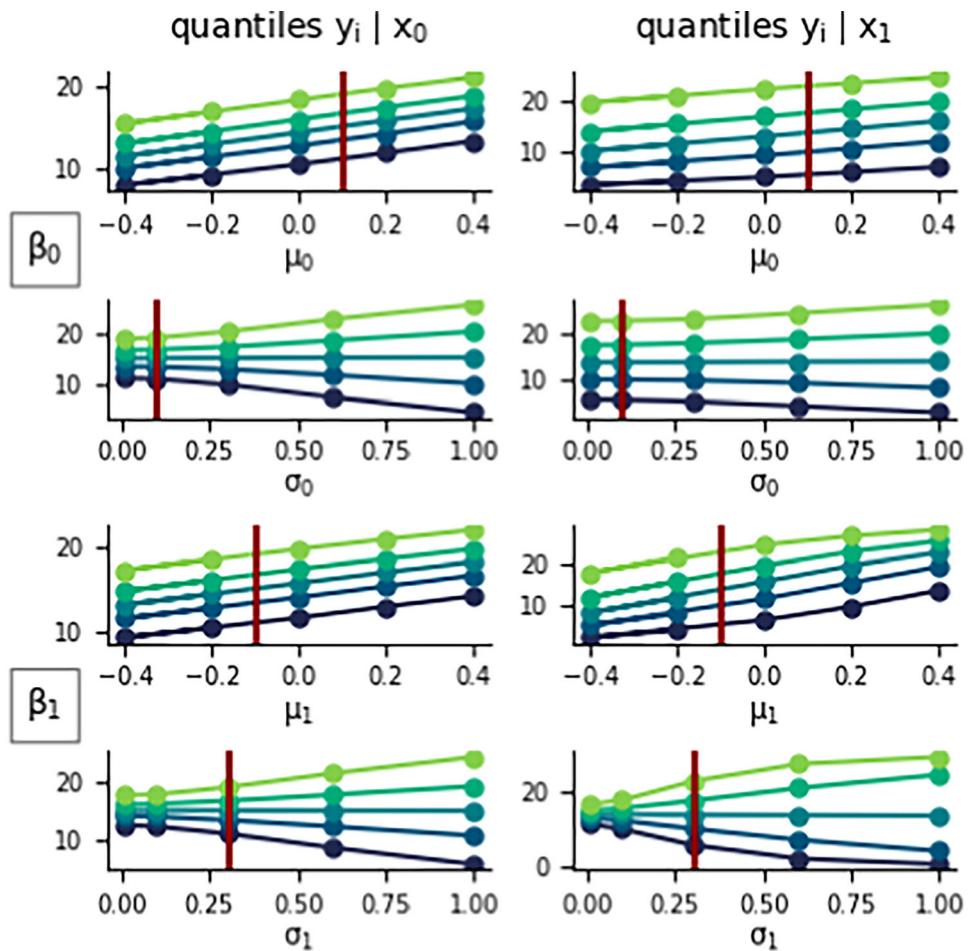
The corresponding final learned elicited statistics for each replication are shown in Figure 6. In the first two plots, each true quantile (x-axis) is plotted against its corresponding learned quantile (y-axis) for each replication. Points that align perfectly with the diagonal dashed line represent a perfect match between true and learned quantiles. Points above the diagonal indicate that the learned quantiles are higher than the true ones (and vice versa for lower points). Accuracy of the learned correlation information is presented in Figure 6. The true (red) correlation matches perfectly with the learned (black) correlations.

Evaluation Stage Finally, we examine the learned prior distributions. Figure 7 shows the learned marginal priors for each replication, alongside the results from the model averaging approach. The weights used for model averaging are shown in the upper plot of Figure 7 and reflect the relative performance of each replication based on their total loss (see Section 2.5 for details). In this case, all replications perform similarly well, resulting in an almost uniform weighting scheme. The resulting averaged prior is represented by the solid red line in the lower plot, shown together with the learned priors from each replication.

4.3 Simulation Studies 2: Normal Likelihood

In the remaining simulation studies, we aim to evaluate the method’s performance using a model with slightly increased complexity, specifically a normal regression model with standard deviation σ and mean μ_i , where μ_i depends on a three-level grouping variable encoded via the dummy variables x_1 and x_2 :

Fig. 2 Binomial model—Sensitivity Analysis: Rows represent hyperparameters of each model parameter. Columns represent elicited statistics: five quantiles for $y | x_0$ and $y | x_1$. Quantiles are depicted in different colors. In each row, the corresponding hyperparameter, is varied across the range shown on the x-axis, while all other hyperparameters are held constant at their true values, indicated by the red vertical line. The upper two rows indicate results for the intercept parameter β_0 and the lower two rows for the slope parameter β_1



$$y_i \sim \text{Normal}(\mu_i, \sigma)$$

$$\mu_i = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i}$$

$$\beta_0, \beta_1, \beta_2, \sigma \sim p_\lambda(\beta_0, \beta_1, \beta_2, \sigma)$$

$$\theta \equiv (\beta_0, \beta_1, \beta_2, \sigma)$$

with β_0 , β_1 , and β_2 being the regression coefficients representing the intercept and the two group contrasts, respectively. The goal is to learn a joint prior for the model parameters θ . To systematically assess our method, we use the same model across all upcoming simulation studies, varying only the *true* joint prior in each scenario:

- **Scenario 1 (M2):** Independent normal priors for the regression coefficients are defined as $\beta_0 \sim \text{Normal}(10, 2.5)$, $\beta_1 \sim \text{Normal}(7, 1.3)$, $\beta_2 \sim \text{Normal}(2.5, 0.8)$, with a Gamma prior for the random noise, $\sigma \sim \text{Gamma}(5, 2)$
- **Scenario 2 (M3):** Identical to M2 but introduces skewed normal priors for β_1 and β_2 , where $\beta_1 \sim \text{SkewNormal}(7, 1.3, 4)^1$ and $\beta_2 \sim \text{SkewNormal}(2.5, 0.8, 4)$.

¹ For the skewed normal distribution, we use the implementation from TensorFlow Probability (Abadi et al. 2016), which features the two-

- **Scenario 3 (M4):** Identical to M2 but introduces a dependency structure among the model coefficients β using a correlated multivariate normal distribution, with marginal distributions identical to those in M2. The correlated prior is specified as:

$$\beta \sim \text{Mv-Normal} \left(\begin{bmatrix} 10 \\ 7 \\ 2.5 \end{bmatrix}, D(s) \mathbf{R} D(s) \right) \text{ with} \\ \mathbf{R} = \begin{bmatrix} 1. & 0.3 & -0.3 \\ 0.3 & 1. & -0.2 \\ -0.3 & -0.2 & 1. \end{bmatrix}, s = (2.5, 1.3, 0.8)$$

where \mathbf{R} denotes the correlation matrix and s the vector of standard deviations.

piece normal distribution (Fernández and Steel 1998). This distribution is parameterized by location, scale, and shape parameters. The shape parameter controls the skewness, with values above one resulting in a right-skewed distribution.

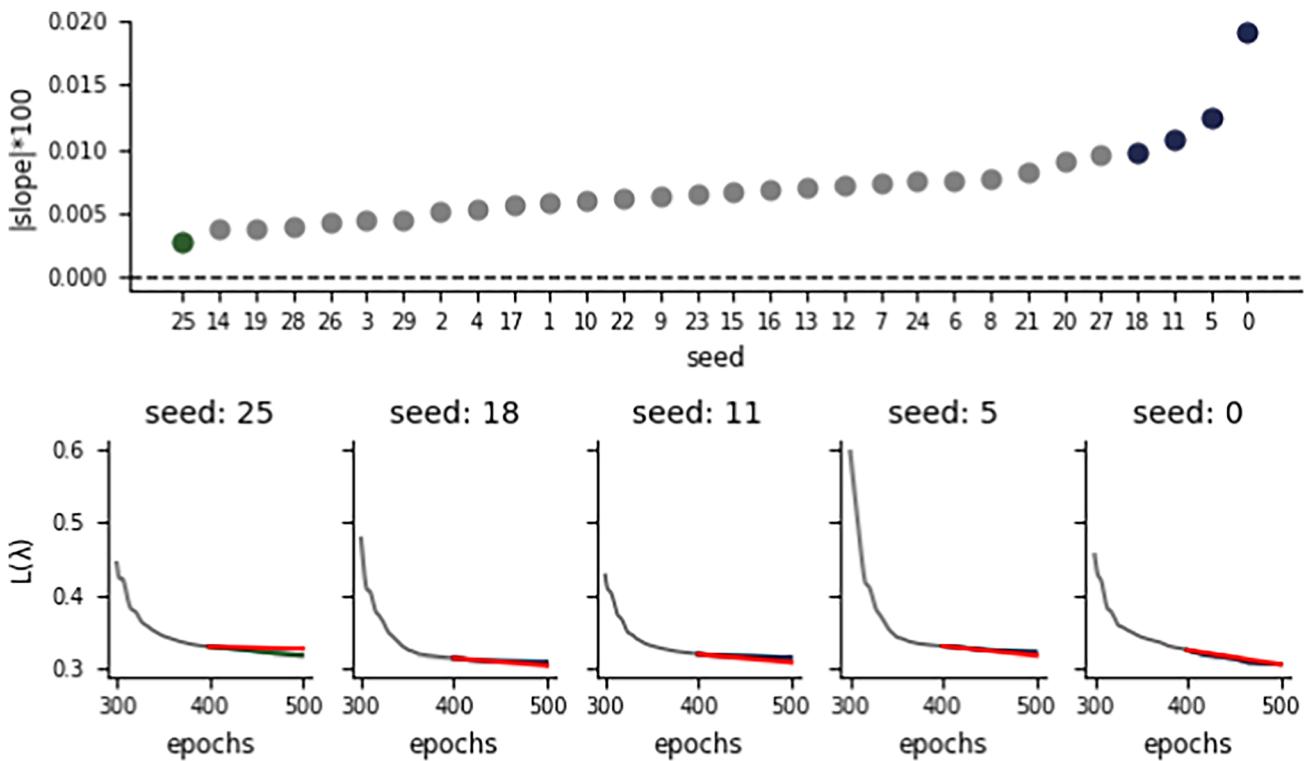


Fig. 3 Binomial model—Preliminary convergence check for all 30 replications: Upper plot: Absolute slope of the total loss trajectory (scaled by a factor of 100) over the last 100 epochs on the y-axis, with each replication shown along the x-axis. Replications (i.e., seeds)

are arranged in ascending order based on the magnitude of the absolute slope. Best-performing model is seed 25, while the worst-performing models are seeds 18, 11, 5, and 0. Lower plot: Loss trajectories of the best- and worst-performing seeds over the last 200 epochs

Fig. 4 Binomial model—Visual convergence check 1: Updating trajectory across epochs for all 30 replications. On the left, trajectory of the total loss and on the right, trajectory of the weighted individual loss components

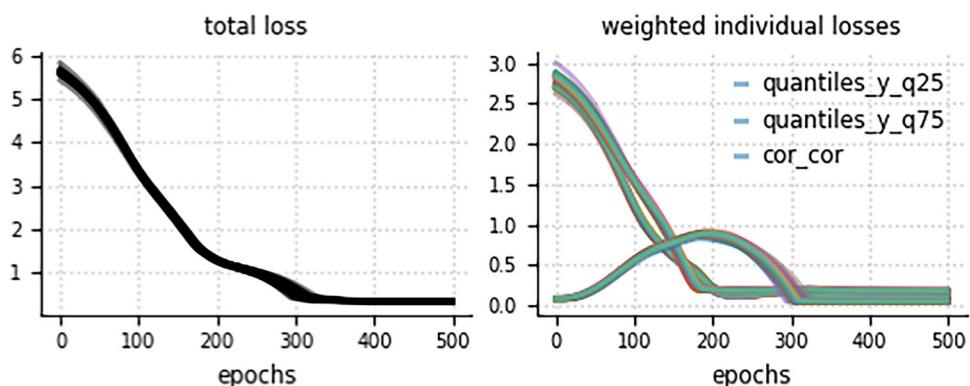
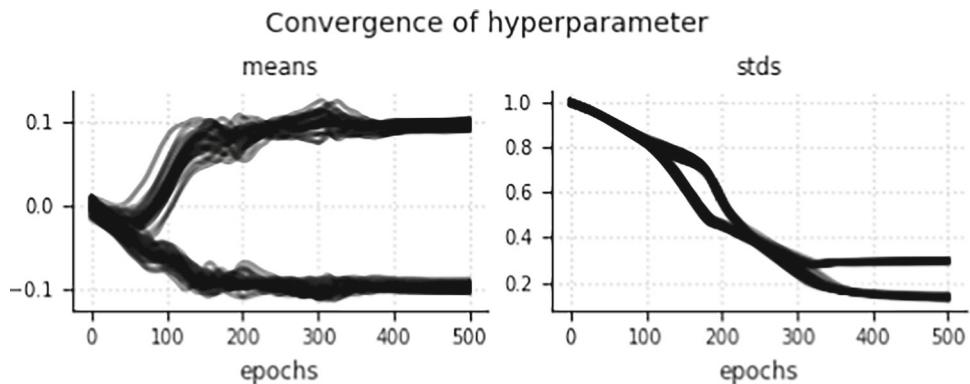


Fig. 5 Binomial model—Visual convergence check 2: Updating trajectory across epochs for all 30 replications. Trajectory of the mean (left plot) and standard deviation (right plot) of the marginal priors



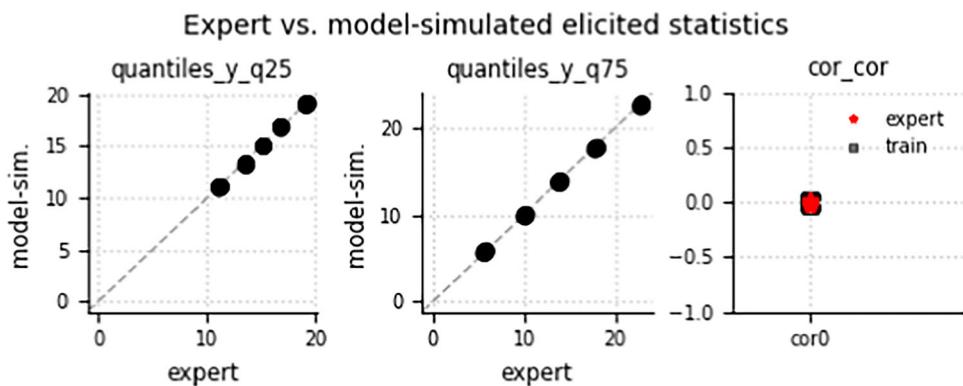
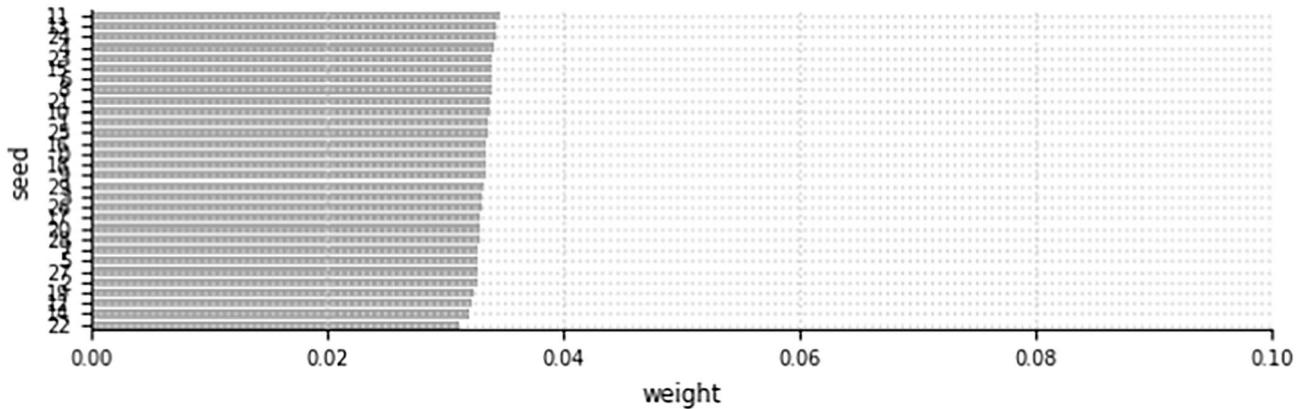


Fig. 6 Binomial model—Learned elicited statistics for all 30 replications: The two leftmost plots show the learned quantiles (y-axis) vs. the true quantiles (x-axis) of each replication for the two target quantities, $y \mid x_0$ and $y \mid x_1$. The diagonal line serves as a reference, where

points lying on the line indicate a perfect match between learned and true quantiles. The rightmost plot displays the learned (black) vs. true (red) correlations. Given that we assume independence between model parameters in this scenario, the true correlation is zero

Prior averaging

Prior averaging (weights)



Prior distributions

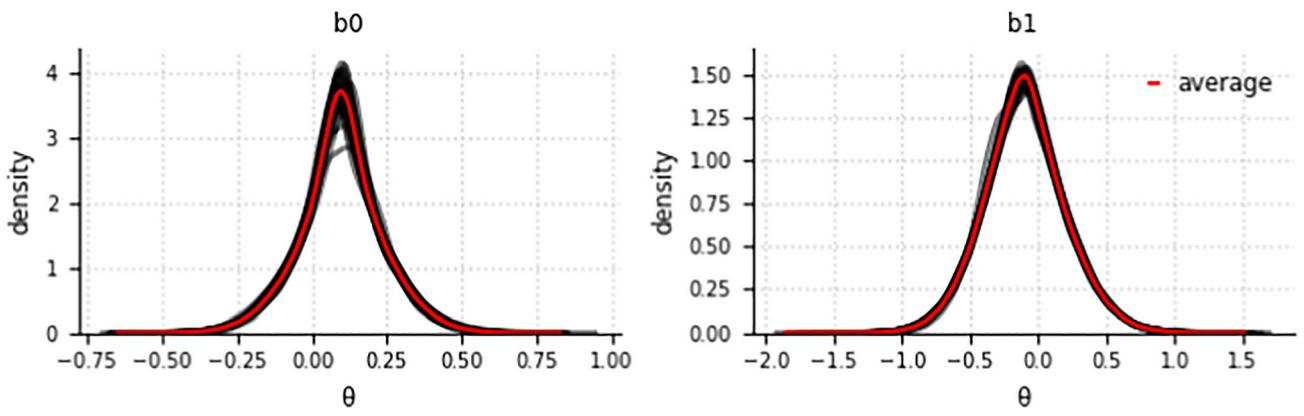


Fig. 7 Binomial model—Learned marginal priors for all 30 replications: Upper plot: Computed weights per replication used for model averaging, reflecting the relative performance of each model based on the final total loss. In this scenario, all replications perform equally well, resulting in almost uniform weights. Lower plot: The learned marginal

priors for each replication and in red the prior average. In this case, the elicited statistics are sufficiently informative to effectively identify the generative model, resulting in only minor variations in the learned priors across replications

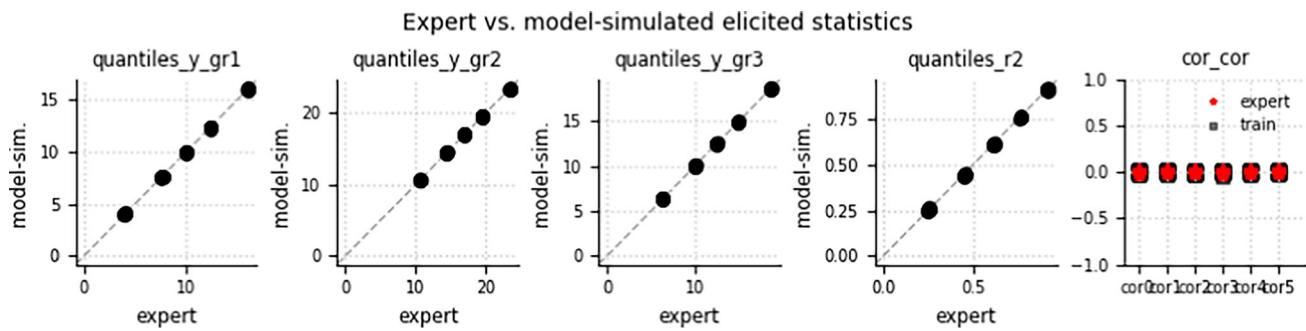


Fig. 8 Normal model / Scenario 1 (M2) — Learned elicited statistics for all 30 replications: First three plots (from the left) show the learned quantiles (y-axis) vs. the true quantiles (x-axis) of each replication for the four target quantities $y \mid gr_i$ with $i = 1, 2, 3$ and R^2 . The diagonal line serves as a reference, where points lying on the line indicate a

perfect match between learned and true quantiles. The rightmost plot displays the learned (black) vs. true (red) correlations between the model parameters. Given the independence assumption, the true correlation is zero

4.3.1 Scenario 1: Independent Normal Priors

Setup & Elicitation Stage We begin with a brief assessment of the selected elicited statistics, with sensitivity analysis results detailed in Appendix A.2. While the intercept coefficient, β_0 , is well-informed by all three target quantities related to the predictive distributions of the groups, the slope coefficients, β_1 and β_2 , are each informed by only one target quantity, specifically those corresponding to group 2 and group 3, respectively. Consequently, we anticipate greater difficulty in accurately learning these coefficients. Additionally, we included R^2 among the target quantities to aid the learning algorithm in distinguishing parameter uncertainty from data uncertainty.

Fitting Stage To verify successful convergence, we first examined the absolute slopes across all 30 replications, followed by a detailed visual inspection of the learning trajectories for the loss components and additional quantities of interest (see Appendix A.2 for results). The analysis indicates that the learning process was successful. The model-implied statistics are shown in Figure 8 and demonstrate a strong match between the learned and true elicited statistics, supporting the assumption of successful convergence.

Evaluation Stage The corresponding learned marginal priors of each replication alongside with the model averaging results are depicted in Figure 9. In this scenario, we observe noticeably greater variation in the learned priors across different replications, particularly for β_1 and β_2 . This variation is also reflected in the model averaging weights, which now deviate clearly from the uniform distribution observed in Simulation Study 1. These results suggest that while the elicited statistics provide sufficient information to learn the marginal priors of β_0 and σ , they are insufficient to uniquely identify the priors for β_1 and β_2 . It is important to emphasize that the only information available to the method for

learning the prior distributions is based on the elicited statistics shown in Figure 8. Given this limited set of information, learning a non-parametric joint prior becomes a highly under-constrained problem, where multiple prior distributions can be consistent with the same set of elicited statistics. However, this should not be interpreted as a limitation of the method itself, but rather as a consequence of the limited information available to the method for learning the joint prior.

4.3.2 Scenario 2: Skewed Normal Priors

Setup & Elicitation Stage Scenario 2 differs from the previous one only in the ground truth, where we introduce a skew-normal marginal prior for the two slope coefficients, β_1 and β_2 . In this scenario, we examine whether the method can learn this additional skewness property using the same “elicited statistics” as those introduced in the previous case study. The sensitivity analysis (see Appendix A.3) already suggests that there is likely insufficient information to disentangle the variance and skewness components for the two slope coefficients; therefore, we expect to observe more variation in the learned prior distributions.

Fitting & Evaluation Stage The convergence diagnostics are depicted in Appendix A.3 and indicate successful convergence. The corresponding match between the model-implied and true “elicited” statistics is shown in Figure 10, demonstrating overall good performance.

Figure 11 shows the corresponding learned marginal priors together with the results from prior averaging. The results indicate relatively stable learning of the prior distributions across all replications. As before, the priors for the slope coefficients exhibit higher variation. Most importantly, however, the method is able to learn the introduced skewness.

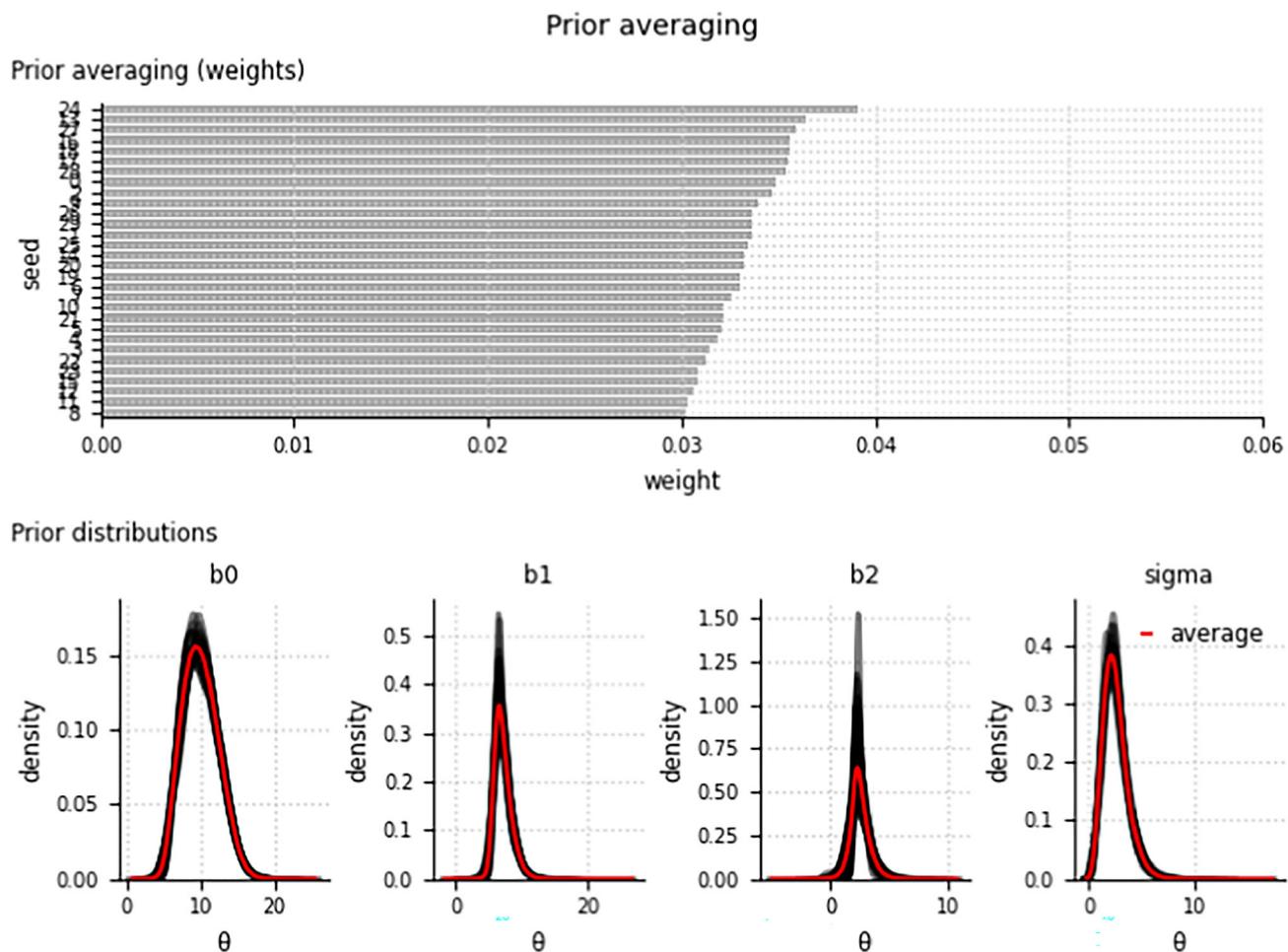


Fig. 9 Normal model / Scenario 1 (M2) — Learned marginal priors across replications: Upper plot: Computed weights per replication used for model averaging, reflecting the relative performance of each model

based on the final total loss. Lower plot: The learned marginal priors for each replication (in black) and the prior average (in red)

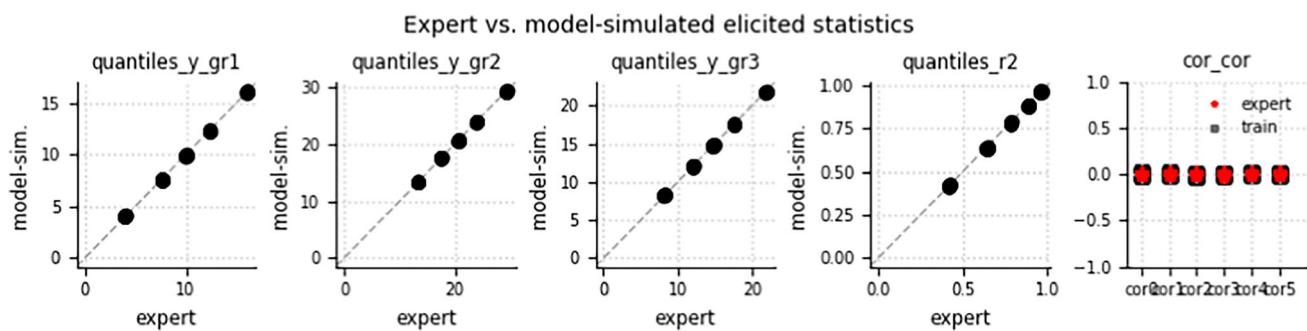


Fig. 10 Normal model / Scenario 2 (M3) — Learned elicited statistics for all 30 replications: First four plots from the left depict the learned quantiles (y -axis) vs. the true quantiles (x-axis) of each replication for the four target quantities $y \mid gr_i$ with $i = 1, 2, 3$ and R^2 . The diagonal line serves as a reference, where points lying on the line indicate a

perfect match between learned and true quantiles. The rightmost plot displays the learned (black) vs. true (red) correlations between the model parameters. Given the independence assumption, the true correlation is zero

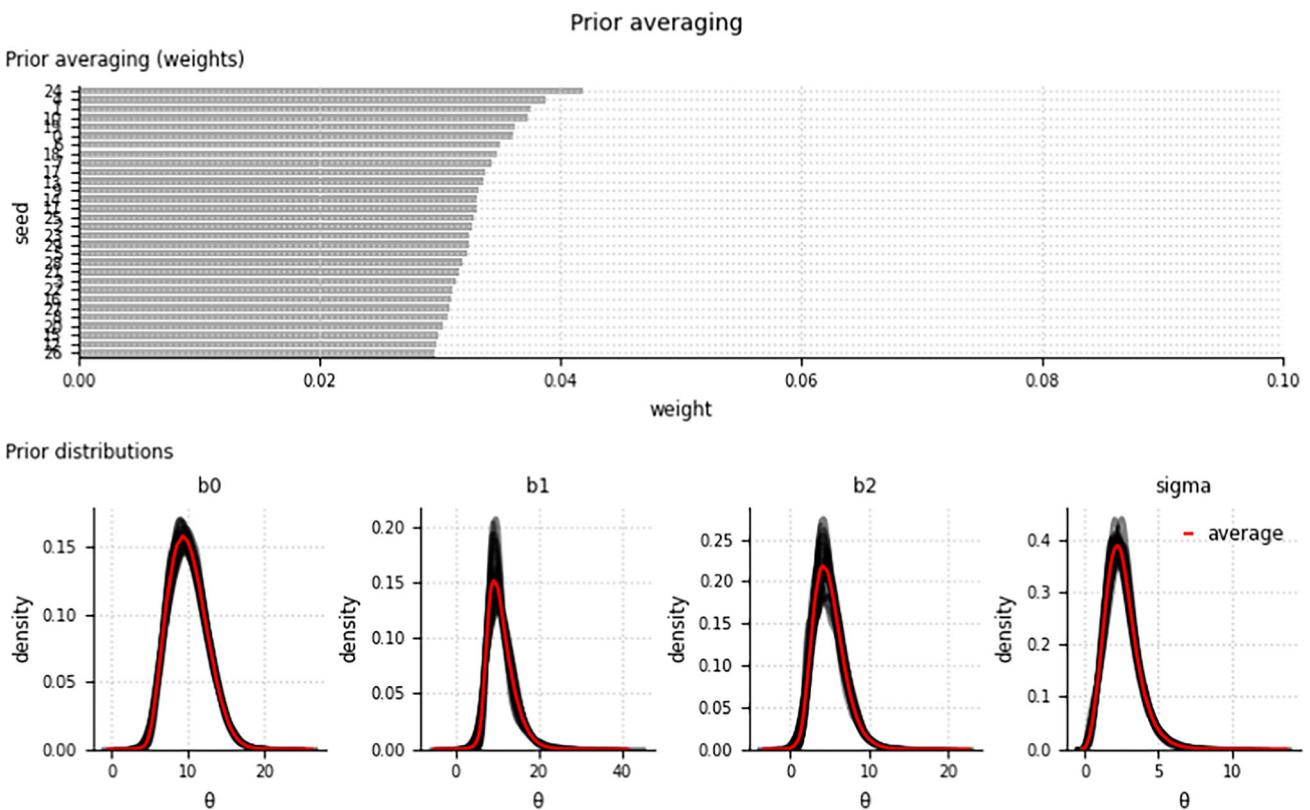


Fig. 11 Normal model / Scenario 2 (M3) — Learned marginal priors for all 30 replications: Upper plot: Computed weights per replication used for model averaging, reflecting the relative performance of each

model based on the final total loss. Lower plot: The learned marginal priors for each replication (in black) and the prior average (in red)

4.3.3 Scenario 3: Correlated Normal Priors

In this final scenario, we introduced a dependency structure among the regression coefficients, while all other aspects remained identical to Scenario 1 (M1). The results of the sensitivity analysis and the convergence checks are provided in Appendix A.4 and show successful convergence. The model-implied and true “elicited” statistics are shown in Figure 12 and indicate a close match for all elicited statistics.

In Figure 13, the final learned marginal priors for each replication are shown alongside the results from prior averaging. Results show substantial variation in the learned priors for β_2 , whereas the priors for all other parameters show relatively high consistency across replications. Since the focus of this case study is specifically on the correlation structure of the regression coefficients, it is informative to also examine the joint prior distribution, which is shown in Figure 14. The joint priors from different replications are shown in different colors. Although there is clear variation in the learned joint priors, the method is able to capture the underlying correlational pattern. Although this result is not particularly surprising, as we provided the method with the exact correlation information between the model parameters, it

nonetheless indicates that the method is capable of learning the correlational structure of a joint prior when provided with the relevant information. Therefore, future work is needed to develop elicitation methods that enable the method to obtain relevant information by querying domain experts indirectly.

5 Discussion & Outlook

In this paper, we introduce an extension of our simulation-based framework (Bockting et al. 2024) for learning *non-parametric joint priors*. This extension builds on using normalizing flows (Kobyzev et al. 2020) to learn flexible joint priors for the model parameters. Although we employ normalizing flows in this work, other generative models, such as flow matching (Lipman et al. 2022) or diffusion models (Cao et al. 2024) are also compatible with our framework and can be readily used as a drop-in replacement.

Across four simulation studies, we demonstrated successful learning of joint priors based primarily on interpretable quantities that can be meaningfully provided by domain experts. One exception is the use of correlation information, which the current method requires to reasonably constrain the

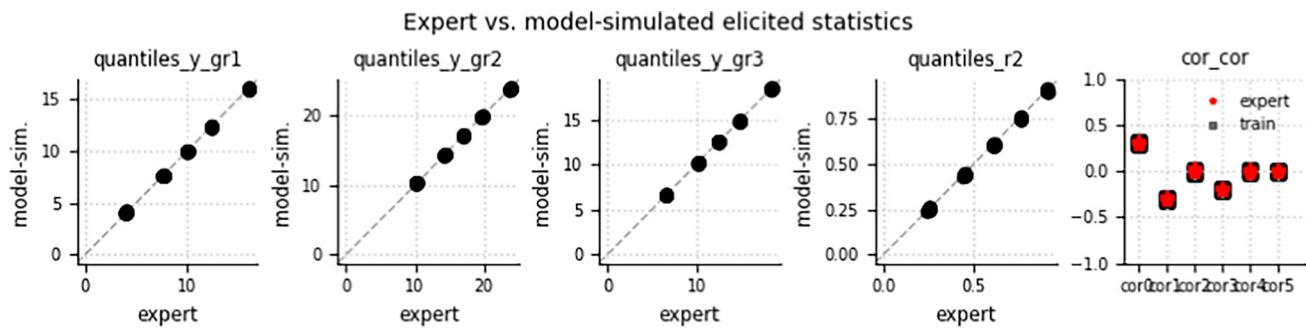


Fig. 12 Normal model / Scenario 3 (M4) — Learned elicited statistics for all 30 replications: First four plots show the learned quantiles (y-axis) vs. the true quantiles (x-axis) of each replication for the four target quantities $y \mid gr_i$ with $i = 1, 2, 3$ and R^2 . The diagonal line

serves as a reference, where points lying on the line indicate a perfect match between learned and true quantiles. The rightmost plot displays the learned (black) vs. true (red) correlations between the model parameters

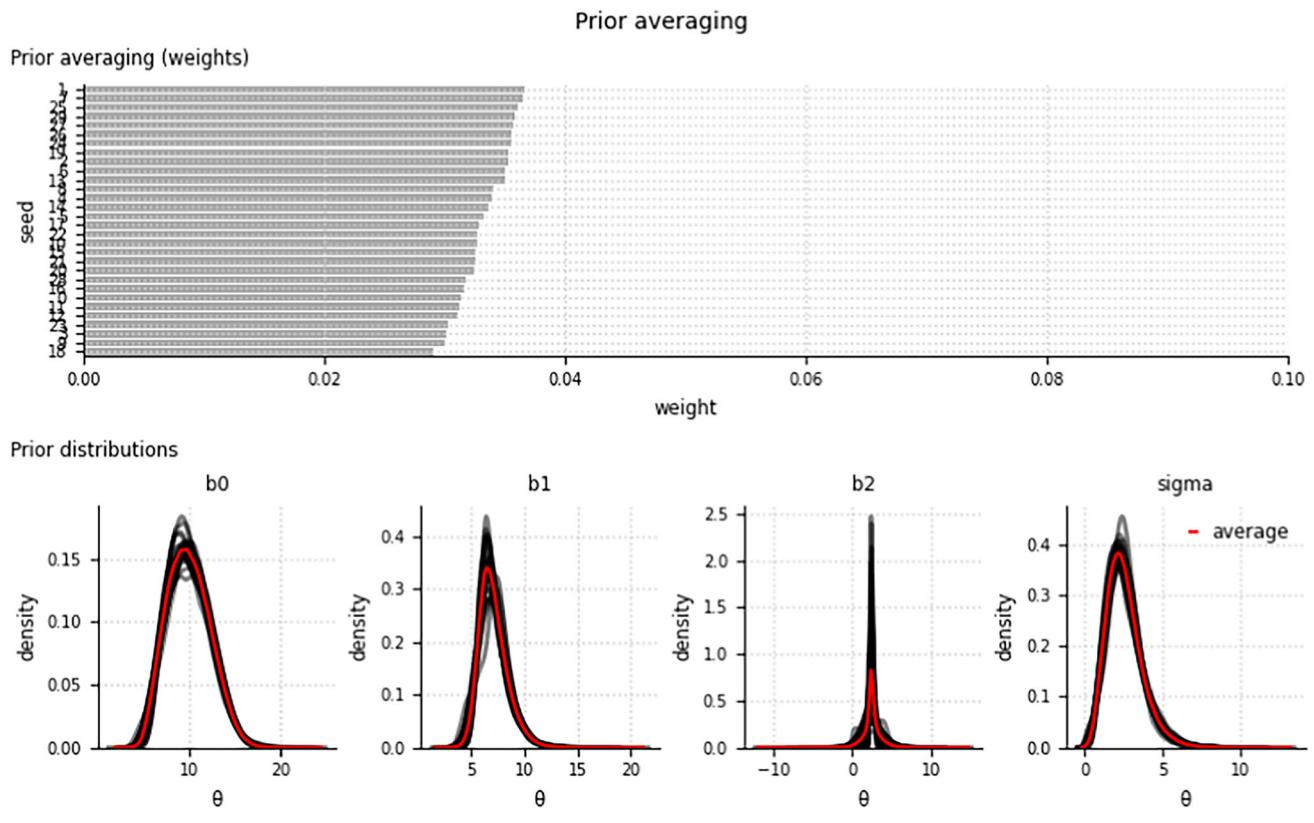


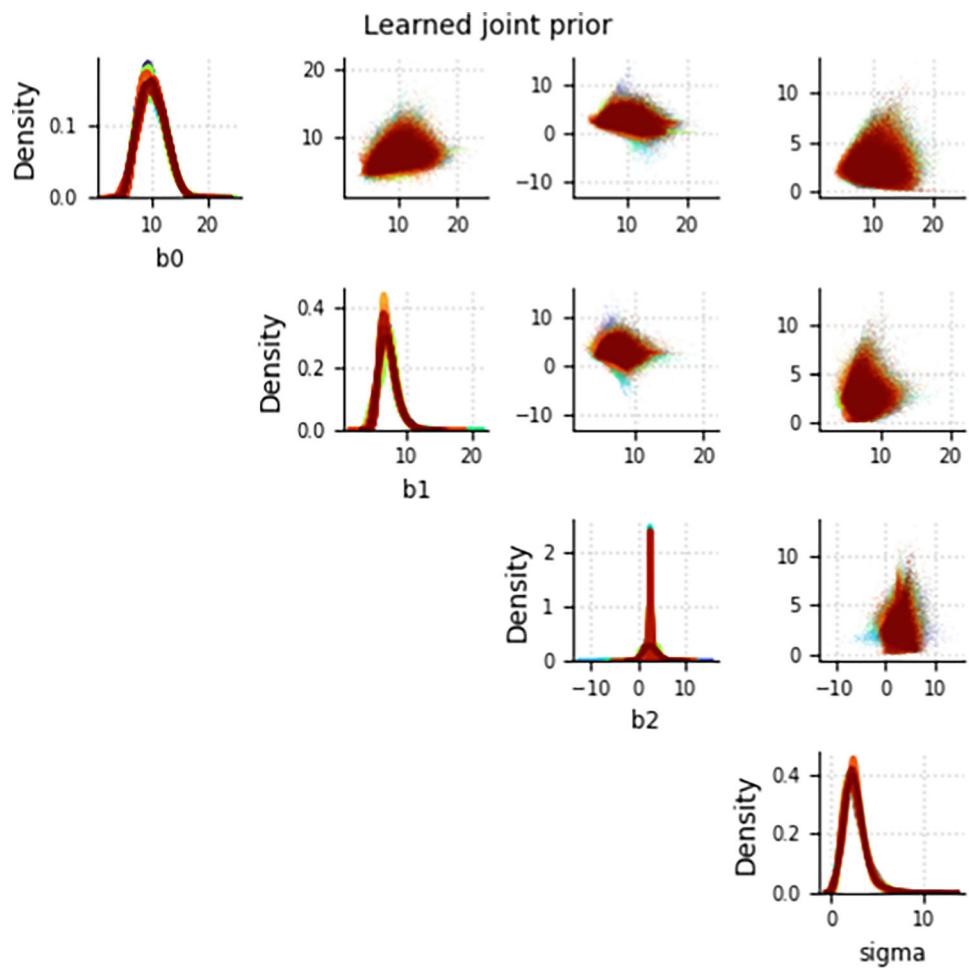
Fig. 13 Normal model / Scenario 3 (M4) — Learned marginal priors for all 30 replications: Upper plot: Computed weights per replication used for model averaging, reflecting the relative performance of each

model based on the final total loss. Lower plot: The learned marginal priors for each replication (black) and the prior average (red)

optimization problem. Future work is required to integrate in our method an elicitation technique that infers the correlation indirectly from the expert by asking about interpretable quantities. Recently, Mikkola et al. (2024) proposed a method that uses pairwise comparisons or ranking techniques to learn a multivariate prior density, which could serve as an interesting avenue for future research.

With the extension of our originally proposed simulation-based framework to support non-parametric prior distributions, we offer an elicitation framework, implemented in *elicito* (Bockting and Bürkner 2025), which can accommodate various types of elicitation methods within a single, unified structure. This facilitates the development of a stan-

Fig. 14 Normal model / Scenario 3 (M4)—Learned joint prior for all 30 replications
Each replication is shown in a different color. The correlation structure of the regression coefficients used for learning the prior is $\rho(\beta_0, \beta_1) = 0.3$, $\rho(\beta_1, \beta_2) = -0.2$, and $\rho(\beta_0, \beta_2) = -0.3$. Independence between σ and correlation coefficients is assumed



dardized set of diagnostics and analysis tools, addressing a current gap in the literature (Mikkola et al. 2023).

In addition to accommodating different types of prior distributions, another strength of the framework is that *expert knowledge* can be specified in various ways. This ranges from querying model parameters to observable quantities or any derived statistics, allowing the queried information to be tailored to the specific problem and the expertise of the domain expert. This design choice provides users with maximum flexibility in specifying the set of quantities to be elicited from the domain expert. However, it also places the responsibility on the user to define a valid and appropriate set of quantities to query from the expert (Simpson et al. 2017). We would argue that this flexibility is in general a desirable property of a method, as it avoids imposing specific modeling choices on the user. Thus, instead of requiring that the problem is framed in a predefined way, the method allows users to adapt it to their specific needs (Regenwetter and Cavagnaro 2019; Simpson et al. 2017; Scott 2017). However, in practice, this flexibility is only effective if good recommendations, guidelines, and diagnostics are available to help users define an

appropriate set of quantities to query from the expert (Garthwaite et al. 2013).

A significant challenge in this context is assessing whether the chosen set of statistics is *sufficiently informative* for the corresponding model. The goal is not necessarily to identify a unique prior that fully defines the generative model. Instead, the aim is to determine a set of equivalently plausible priors, supporting the assumption that a domain expert is unlikely to have a single “true” prior in mind (Winkler 1967). This introduces at least two additional requirements for an elicitation method. First, it should include an efficient implementation that allows the training algorithm to be executed multiple times with different seeds, enabling the assessment of variability in the learned prior distributions. Second, the elicitation method should ideally offer additional diagnostics or advanced techniques, such as model selection or model averaging, to assist the user (Falconer et al. 2022). In this work, we propose a model averaging method in which the weights calculated for averaging can also serve as diagnostics, helping to identify potential outliers within the set of learned priors. Identifying these edge cases could provide a

valuable starting point for evaluating the plausibility of the learned priors in collaboration with a domain expert.

If the set of learned prior distributions includes degenerate cases, this signals the need for additional constraints. These constraints could involve eliciting further or alternative quantities (i.e., elicited statistics) from the expert and/or adding a regularization term to the total loss function to better guide the learning process. To evaluate the informativeness of the elicited statistics, we use a sensitivity analysis as a preliminary evaluation tool (O'Hagan and Oakley 2004). Additionally, we advocate for employing an oracle (i.e., simulating data from a known ground truth) and conducting multiple training replications with varying random seeds. These strategies provide a solid initial check to determine if the learned priors require further constraints. Overall, this discussion highlights the importance of developing comprehensive tutorials and resources to guide users effectively through the elicitation workflow as a key task for future work.

Other important design choices within our framework that require appropriate default settings include the specification of the discrepancy measure and the selection of the optimization approach, among other aspects. So far, in all simulation studies, we have used the maximum mean discrepancy (MMD, Gretton et al. 2006) to evaluate the discrepancy between the model-implied and expert-elicited statistics. The only exception is the correlation information, for which we employed an L2 loss. One key advantage of the MMD is its flexibility in handling a wide range of statistics, from simple point estimates to complex histogram data. However, a notable limitation of the MMD is its high (i.e., quadratic) computational cost. In contrast, the L2 loss is computationally efficient but lacks the capability to process histogram data. Despite the higher computational cost, the MMD consistently showed very good performance and robustness across all simulation studies so far. Therefore, it appears to be a solid choice as a default discrepancy measure. Future work should study the performance of different loss functions depending on different elicited statistics, with the goal to develop user guidelines for selecting an appropriate loss function.

Furthermore, in all simulation studies conducted so far, we used mini-batch SGD to optimize the prior network parameters. Given the generally good performance observed, we believe that this optimization method is a solid default choice. However, we aim to explore additional optimization approaches within our framework, such as Bayesian optimization (as employed by Manderson and Goudie (2023)). Comparing these approaches in terms of performance and the level of hyperparameter tuning required could offer valuable insights into the conditions under which optimization method might be more advantageous than the other.

Finally, while our current methodology learns prior distributions based on expert data, it does currently not account for additional ‘meta’ uncertainty that arises during the elicitation process. This includes uncertainties introduced by the elicitation technique itself or the expert’s uncertainty regarding the ‘true’ quantity (see e.g., Falconer et al. 2022, 2023; Stefan et al. 2022). We acknowledge that these are important sources of information that should be incorporated in future developments of our method. One promising approach is to view this problem from a Bayesian perspective and treat the proposed workflow as a Bayesian inference problem. This idea was already proposed by Mikkola et al. (2023) and referred to as the *supra-Bayesian approach*. We believe this is an important and valuable direction for the next steps in our elicitation method.

Appendix A Simulation Studies

A.1 Training time

The median training time across the 30 replications (\pm sd of training time):

- **Binomial model:** 6.87 minutes (\pm 1.45).
- **Independent normal scenario:** 10.18 minutes (\pm 1.68).
- **Skewed normal scenario:** 14.45 minutes (\pm 2.80).
- **Correlated normal scenario:** 10.10 minutes (\pm 0.17).

A.2 Independent normal scenario

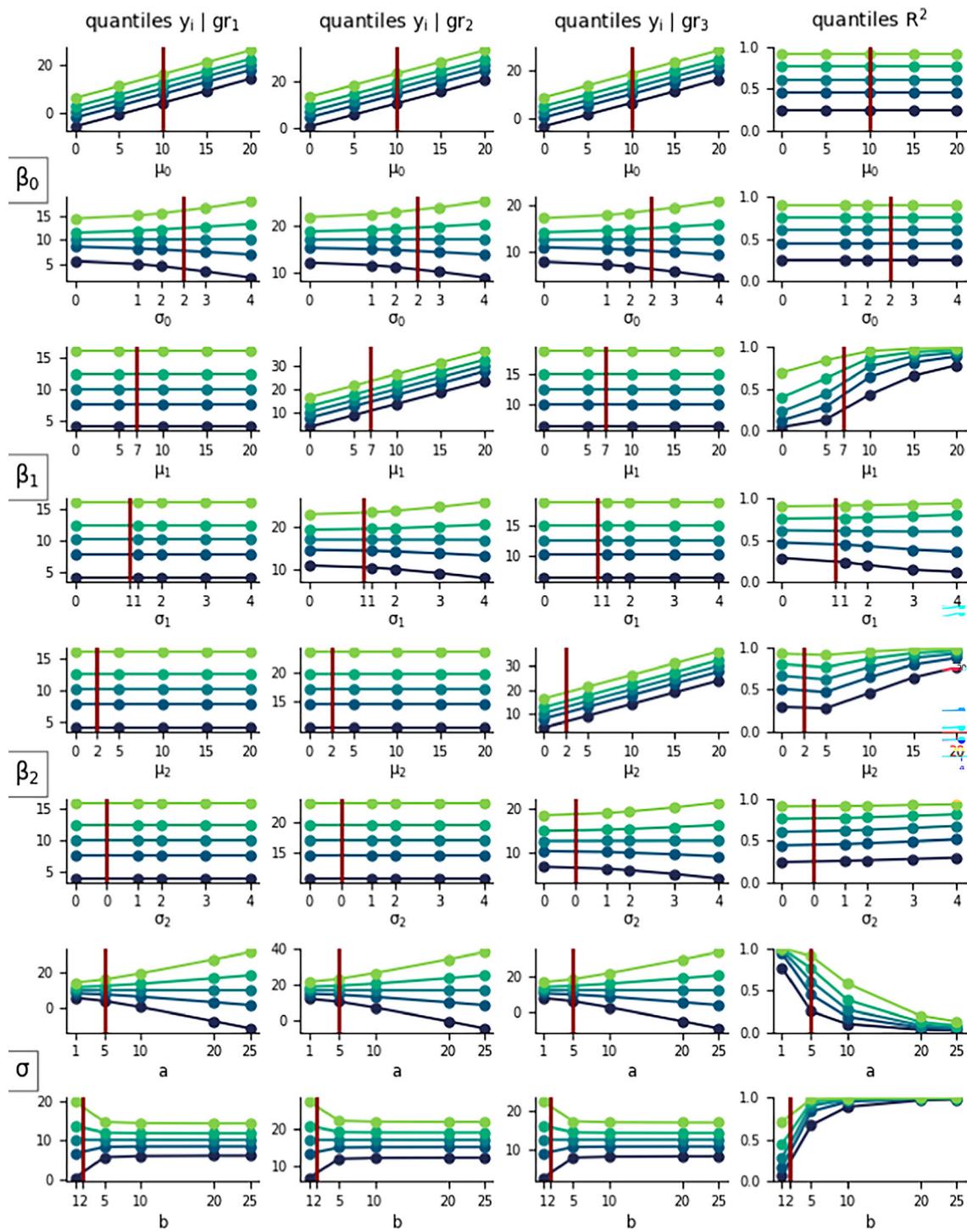


Fig. 15 Normal model / Scenario 1 (M2)—Sensitivity Analysis: Rows represent hyperparameters of each model parameter. Columns represent elicited statistics: five quantiles for $y_i | gr_i$ with $i = 1, 2, 3$ and for R^2 . Quantiles are depicted in different colors. In each row, the correspond-

ing hyperparameter, is varied across the range shown on the x-axis, while all other hyperparameters are held constant at their true values, indicated by the red vertical line

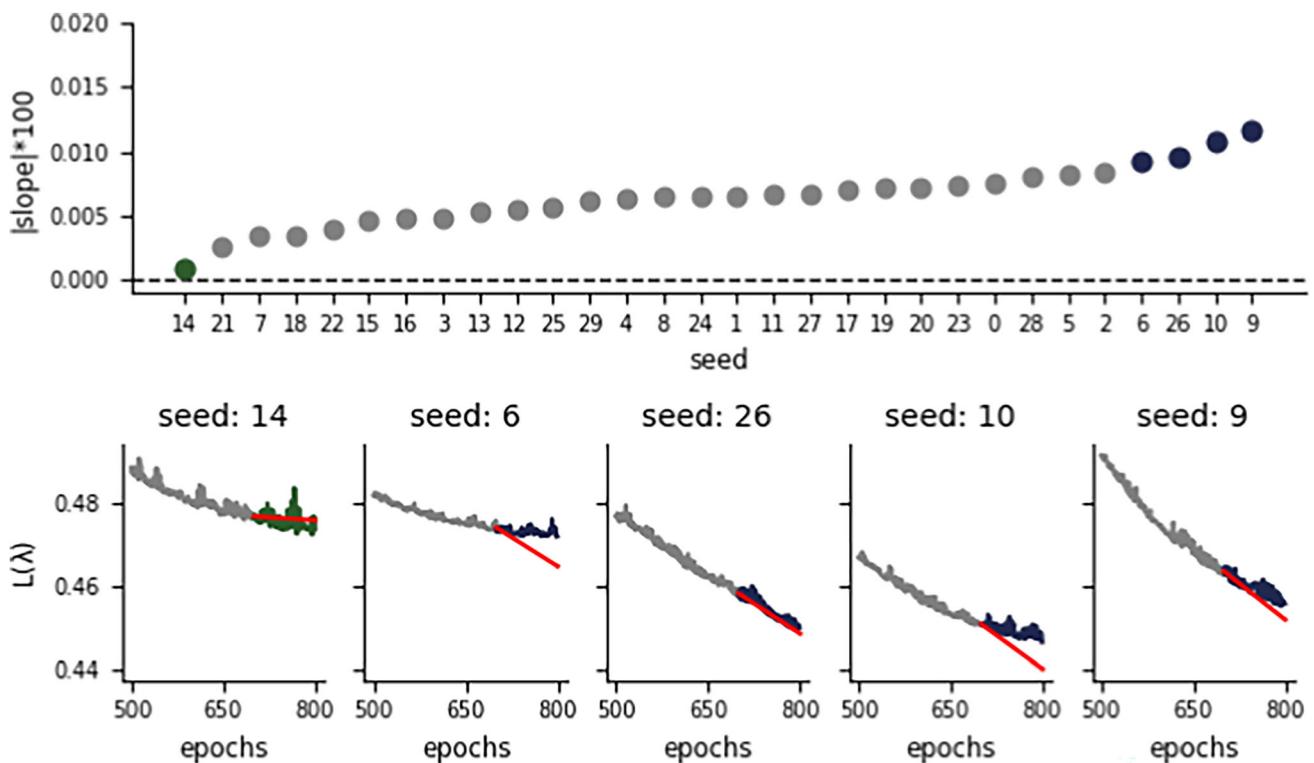


Fig. 16 Normal model / Scenario 1 (M2) — Preliminary Convergence Check: Upper plot displays the absolute slope of the total loss trajectory (scaled by a factor of 100) over the last 100 epochs on the y-axis, with each replication shown along the x-axis. The replications (i.e., seeds) are arranged in ascending order based on the magnitude of the absolute slope. The best-performing model is seed 14, while the worst-

performing models are seeds 6, 26, 10, and 9. The lower plot illustrates the loss trajectories of the best- and worst-performing seeds over the last 300 epochs. The loss updates for the final 100 epochs are highlighted in green (best model) and blue (worst models), with a red line segment indicating the respective slope

Fig. 17 Normal model / Scenario 1 (M2) — Visual convergence check 1: Updating trajectory across epochs for the total loss (left) and the individual (weighted) loss components (right)

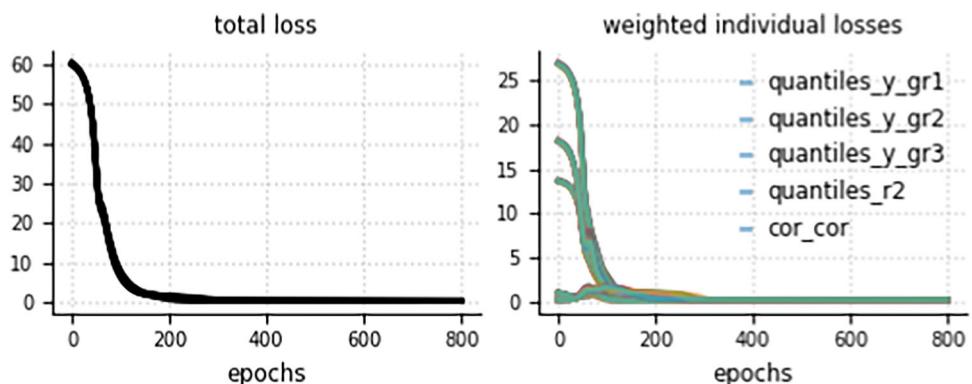


Fig. 18 Normal model / Scenario 1 (M2) — Visual convergence check 2: Updating trajectory across epochs for the mean (left) and standard deviation (right) of the final learned marginal priors for all 30 replications

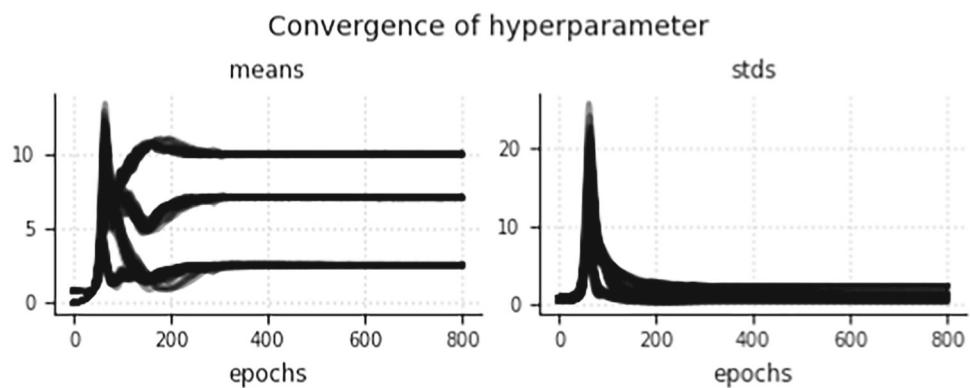
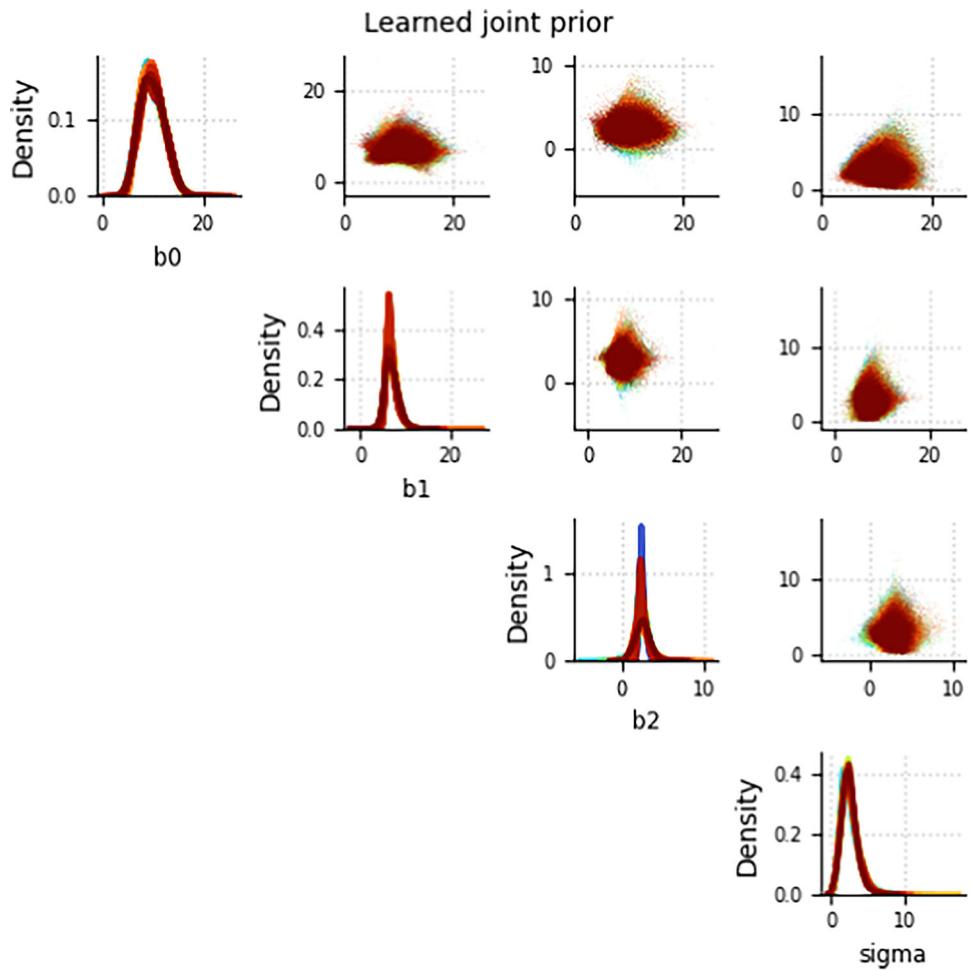


Fig. 19 Normal model / Scenario 1 (M2) — Joint prior: Each replication is visualized in a different color



A.3 Skewed normal scenario

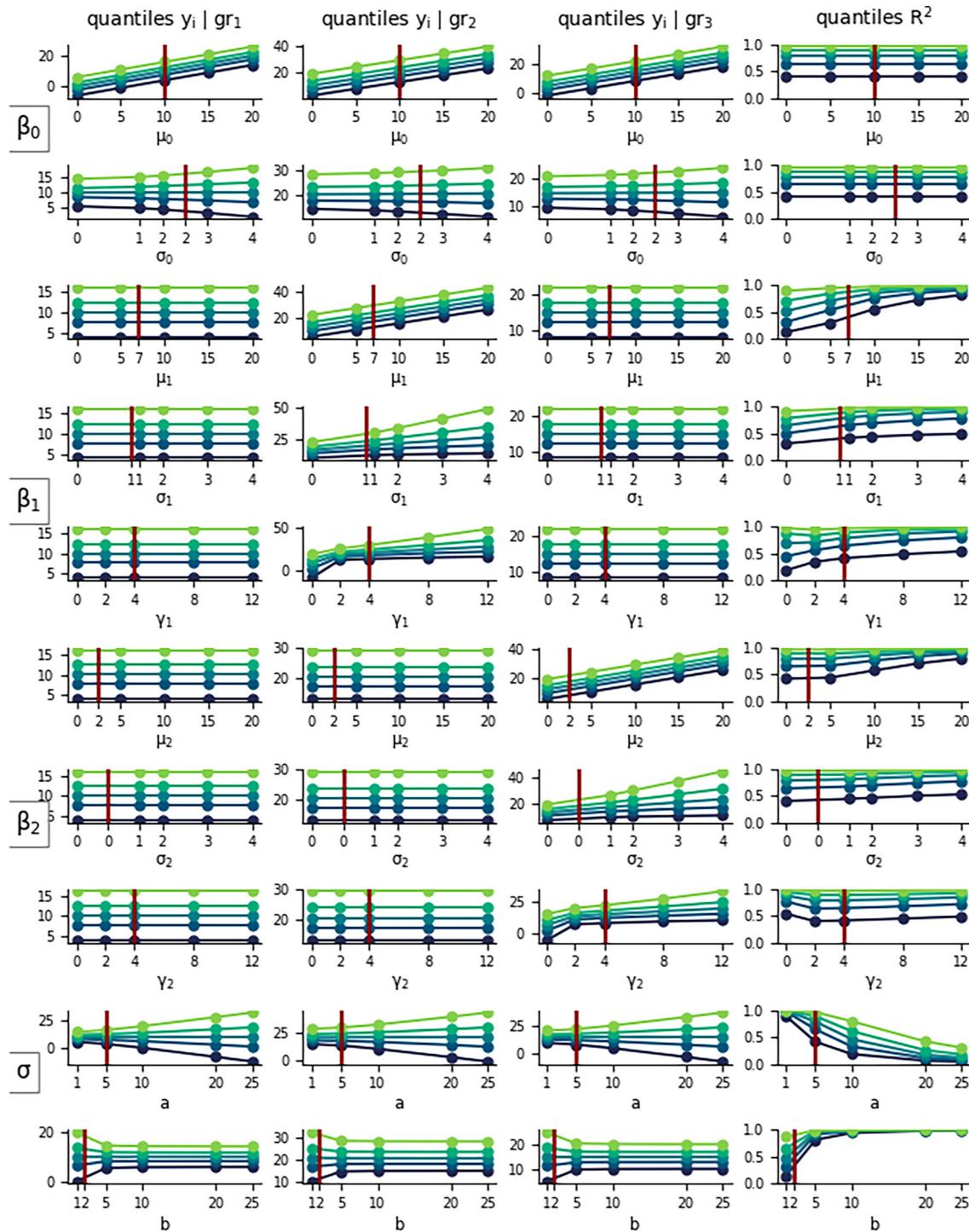


Fig. 20 Normal model / Scenario 2 (M3) — Sensitivity Analysis: Rows represent hyperparameters of each model parameter. Columns represent elicited statistics: five quantiles for $y_i | gr_i$ with $i = 1, 2, 3$ and for R^2 . Quantiles are depicted in different colors. In each row, the cor-

responding hyperparameter, is varied across the range shown on the x-axis, while all other hyperparameters are held constant at their true values, indicated by the red vertical line

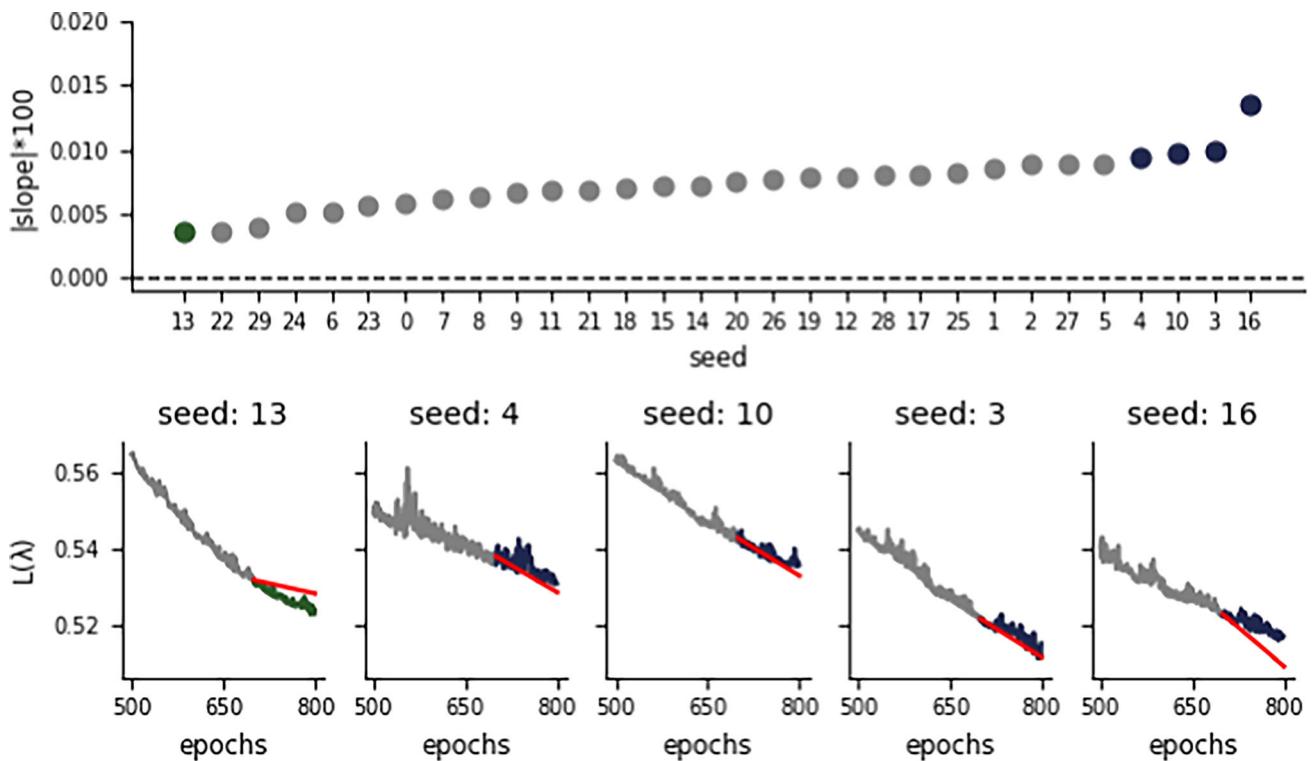
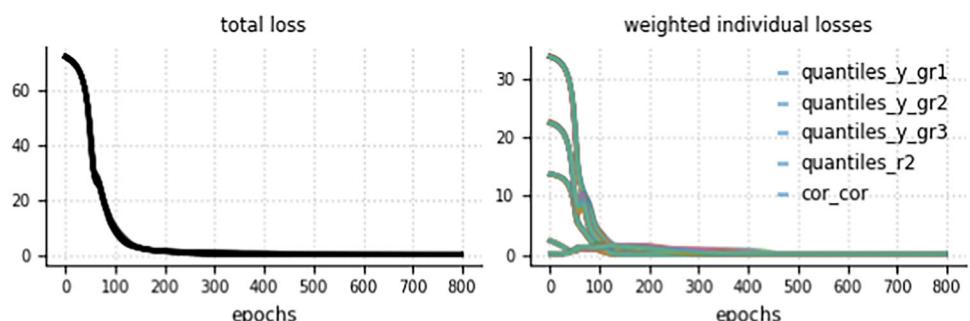


Fig. 21 Normal model / Scenario 2 (M3) — Preliminary Convergence Check for all 30 replications: The upper plot displays the absolute slope of the total loss trajectory (scaled by a factor of 100) over the last 100 epochs on the y-axis, with each replication shown along the x-axis. The replications (i.e., seeds) are arranged in ascending order based on the magnitude of the absolute slope. The best-performing model is seed

13, while the four worst-performing models are seeds 4, 10, 3, and 16. The lower plot illustrates the loss trajectories of the best- and worst-performing seeds over the last 300 epochs. The loss updates for the final 100 epochs are highlighted in green (best seed) or blue (worst seeds). The red line segment indicates the respective slope

Fig. 22 Normal model / Scenario 2 (M3) — Visual convergence check 1: Updating trajectory across epochs for the total loss (left) and the individual (weighted) loss components (right)



Convergence of hyperparameter

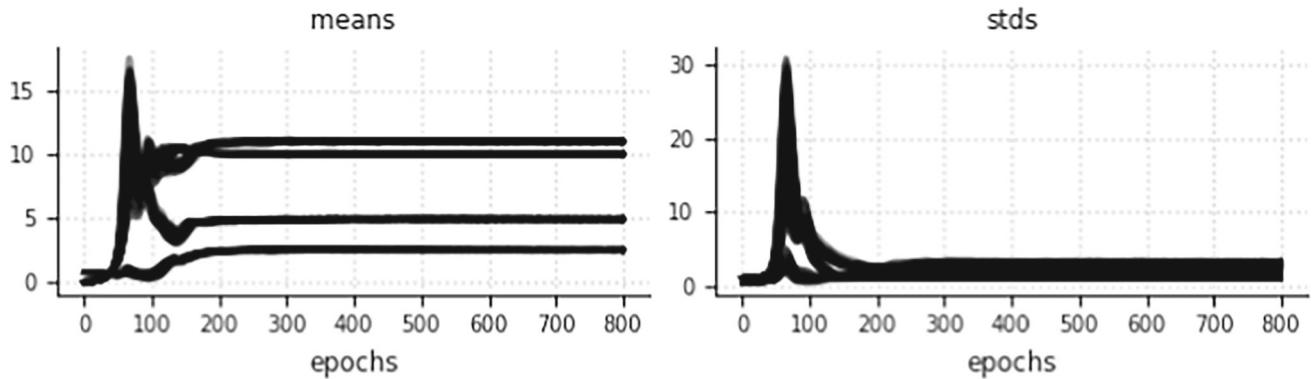
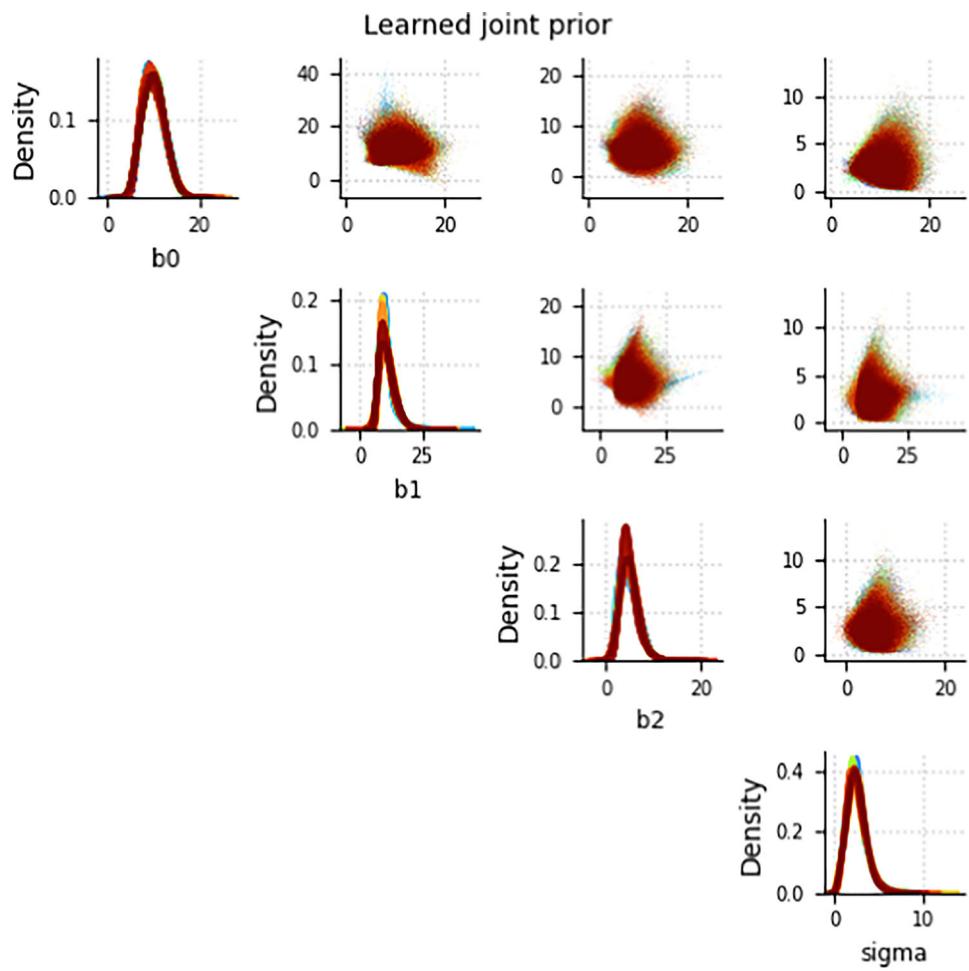


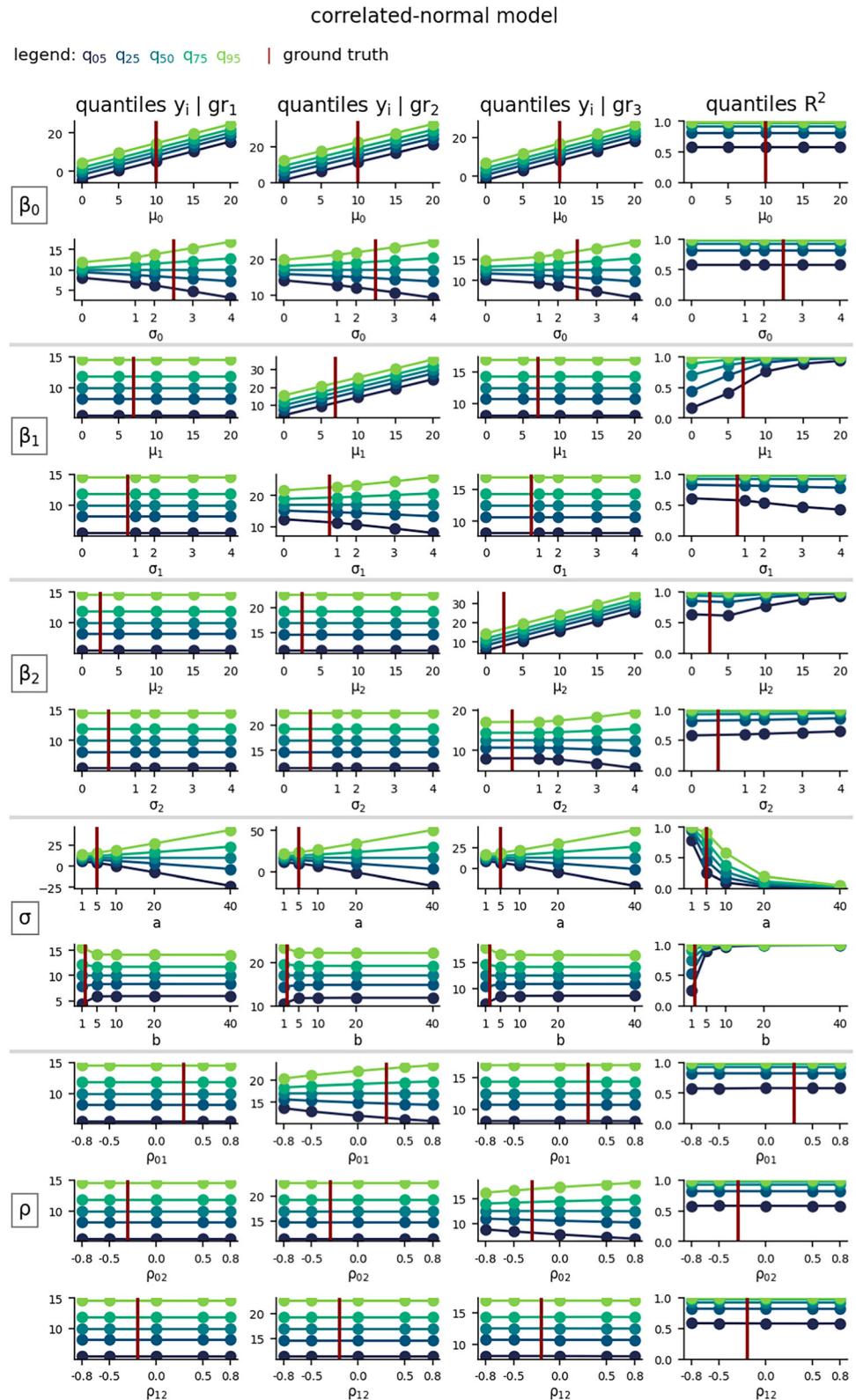
Fig. 23 Normal model / Scenario 2 (M3) — Visual convergence check 2: Updating trajectory across epochs for the mean (left) and standard deviation (right) of the marginal priors

Fig. 24 Normal model / Scenario 2 (M3) — Visual convergence check 2: Joint prior distribution. Each replication is visualized in a different color



A.4 Correlated normal scenario

Fig. 25 Normal model / Scenario 3 (M4)—Sensitivity Analysis: Rows represent hyperparameters of each model parameter. Columns represent elicited statistics: five quantiles for $y_i | gr_i$ with $i = 1, 2, 3$ and for R^2 . Quantiles are depicted in different colors. In each row, the corresponding hyperparameter is varied across the range shown on the x-axis, while all other hyperparameters are held constant at their true values, indicated by the red vertical line



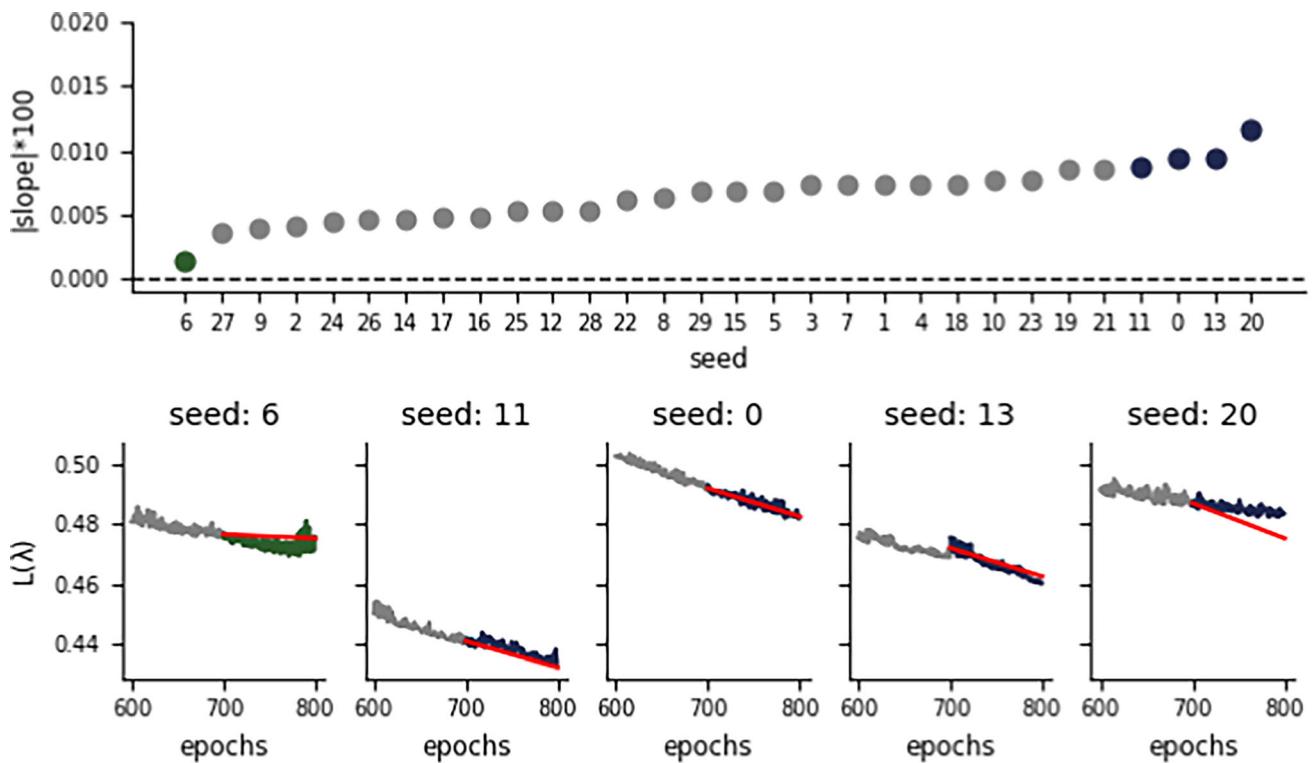


Fig. 26 Normal model / Scenario 3 (M4) — Preliminary Convergence Check for all 30 replications: The upper plot displays the absolute slope of the total loss trajectory (scaled by a factor of 100) over the last 100 epochs on the y-axis, with each replication shown along the x-axis. The replications (i.e., seeds) are arranged in ascending order based on the magnitude of the absolute slope. The best-performing model is seed

6, while the worst-performing models are seeds 11, 0, 13, and 20. The lower plot illustrates the loss trajectories of the best- and worst-performing seeds over the last 200 epochs. The loss updates for the final 100 epochs are highlighted in green for the best model and in blue for the worst models, with a red line segment indicating the respective slope

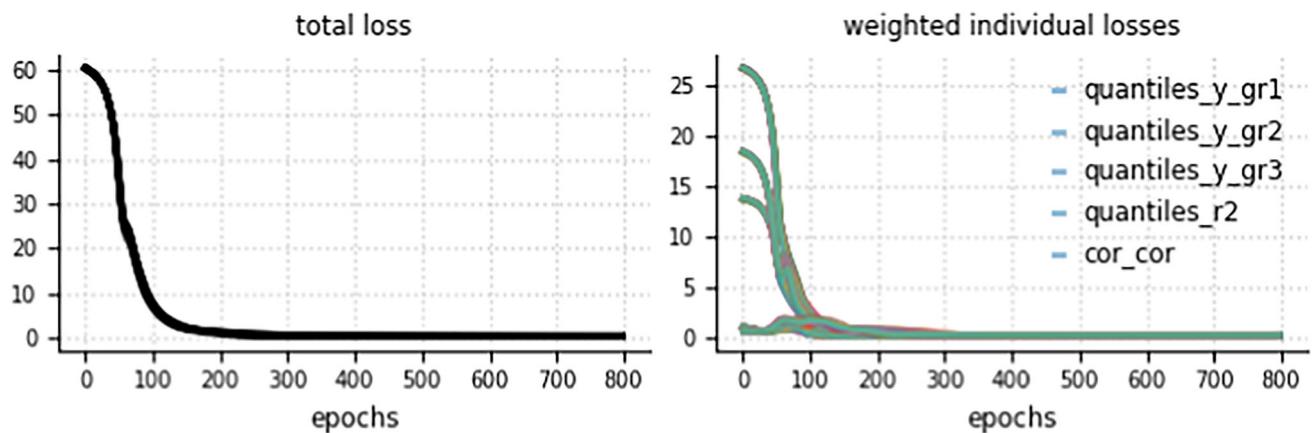


Fig. 27 Normal model / Scenario 3 (M4) — Visual convergence check 1: Updating trajectory across epochs for the total loss (left) and the individual (weighted) loss components (right)

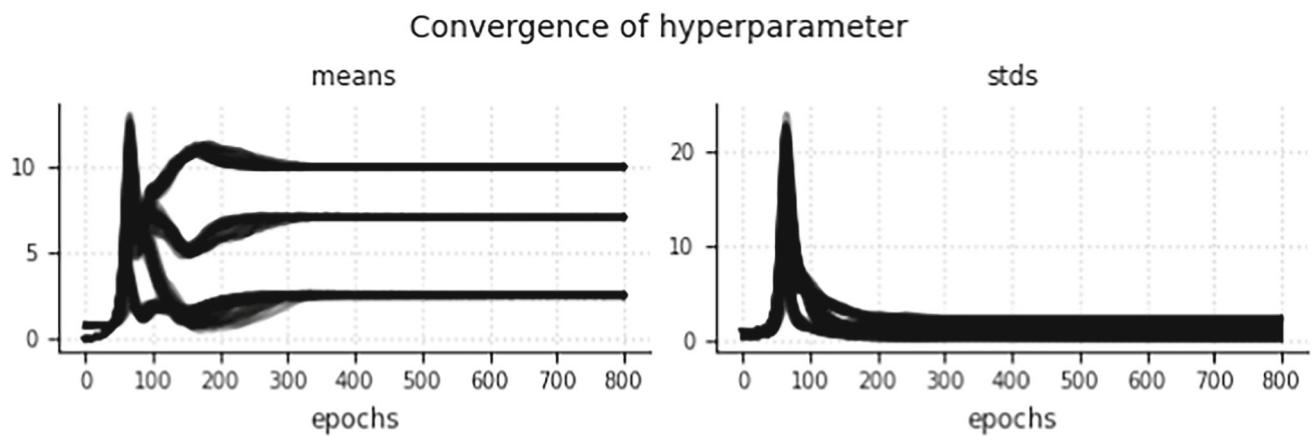
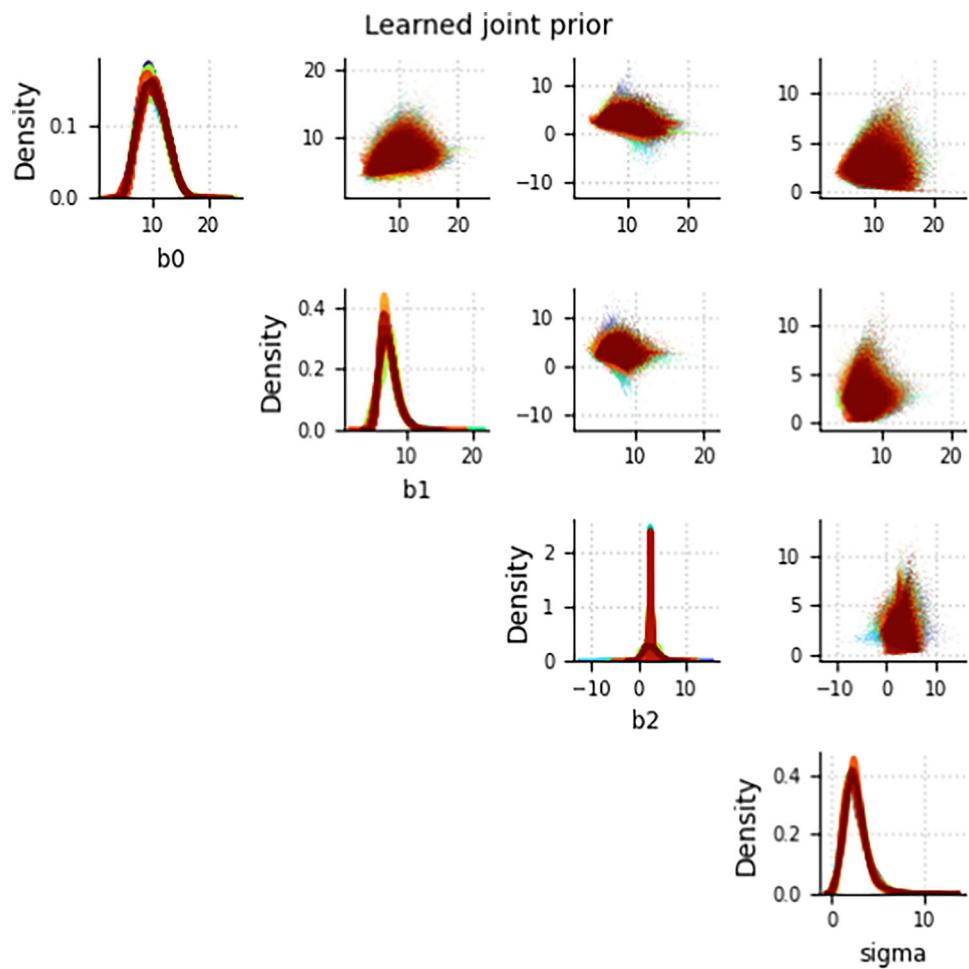


Fig. 28 Normal model / Scenario 3 (M4) — Visual convergence check 1: Updating trajectory across epochs for the mean (left) and standard deviation (right) of the marginal priors

Fig. 29 Normal model / Scenario 3 (M4) — Visual convergence check 2: Joint prior distribution. Each replication is visualized in a different color



Acknowledgements We thank Dmytro Perepolkin and Luna Fazio for their valuable comments and suggestions that greatly improved the manuscript.

Author Contributions Conceptualization: PB, SR, FB, Methodology: PB, FB, Software: FB, Writing (Original Draft): FB, Writing (Editing): FB, SR, PB, Writing (Review): FB, PB, Visualization: FB, Supervision: PB, Funding acquisition: PB

Funding Not applicable.

Data Availability The simulation results are available on OSF at <https://osf.io/xrzh6/>.

Declarations

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

Consent to participate Not applicable.

Consent for publication Not applicable.

Code availability Code for running the simulations and plotting the results is available on GitHub at <https://github.com/florence-bockting/non-parametric-prior>. Simulations and most plots are based on/provided by our Python package *elicito* (v0.3.1 Bockting and Bürkner 2025). To ensure the reproducibility of our results, the specific version of *elicito* used in this study has been archived on Zenodo and is accessible via the following DOI <https://doi.org/10.5281/zenodo.15241853>.

Materials availability Supplementary material is available on GitHub at <https://github.com/florence-bockting/non-parametric-prior>.

References

- Abadi, M., Barham, P., Chen, J., et al.: TensorFlow: A system for large-scale machine learning. In: 12th USENIX symposium on operating systems design and implementation (OSDI 16), pp 265–283, (2016) <https://doi.org/10.5555/3026877.3026899>
- Betancourt, M.: Towards a principled bayesian workflow. Blog Post, (2020) https://betanalpha.github.io/assets/case_studies/principled_bayesian_workflow.html
- Bischof, R., Kraus, M.: Multi-objective loss balancing for physics-informed deep learning. (2021) arXiv preprint <https://doi.org/10.48550/arXiv.2110.09813>
- Bissiri, P.G., Holmes, C.C., Walker, S.G.: A general framework for updating belief distributions. *J. R. Stat. Soc. Series. B Stat. Methodol.* **78**(5), 1103–1130 (2016). <https://doi.org/10.1111/rssb.12158>
- Bockting, F., Bürkner, P.C.: elicito - a tool for expert prior elicitation. (2025) <https://github.com/florence-bockting/elicito>
- Bockting, F., Radev, S.T., Bürkner, P.C.: Simulation-based prior knowledge elicitation for parametric bayesian models. *Sci. Rep.* **14**(1), 17330 (2024). <https://doi.org/10.1038/s41598-024-68090-7>
- Bond-Taylor, S., Leach, A., Long, Y., et al.: Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(11), 7327–7347 (2021). <https://doi.org/10.1109/TPAMI.2021.3116668>
- Buckland, S.T., Burnham, K.P., Augustin, N.H.: Model selection: an integral part of inference. *Biometrics* pp 603–618. (1997) <https://doi.org/10.2307/2533961>
- Cao, H., Tan, C., Gao, Z., et al.: A survey on generative diffusion models. *IEEE Trans. Knowl. Data Eng.* **36**(7), 2814–2830 (2024). <https://doi.org/10.1109/TKDE.2024.3361474>
- Claeskens, G., Hjort, N.L.: Model selection and model averaging. Cambridge books (2008)
- Clemen, R.T., Fischer, G.W., Winkler, R.L.: Assessing dependence: Some experimental results. *Manage. Sci.* **46**(8), 1100–1115 (2000). <https://doi.org/10.1287/mnsc.46.8.1100.12023>
- Cooke, R.M.: Experts in uncertainty: opinion and subjective probability in science. Oxford University Press, USA (1991)
- Crawshaw, M.: Multi-task learning with deep neural networks: A survey. (2020) arXiv preprint <https://doi.org/10.48550/arXiv.2009.09796>
- Crowder, M.: Bayesian priors based on a parameter transformation using the distribution function. *Ann. Inst. Stat. Math.* **44**, 405–416 (1992). <https://doi.org/10.1007/BF00050695>
- Denham, R., Mengersen, K.: Geographically assisted elicitation of expert opinion for regression models. *Bayesian Anal.* **2**(1), 99–136 (2007). <https://doi.org/10.1214/07-BA205>
- Depaoli, S., Winter, S.D., Visser, M.: The importance of prior sensitivity analysis in bayesian statistics: demonstrations using an interactive shiny app. *Front. Psychol.* **11**, 608045 (2020). <https://doi.org/10.3389/fpsyg.2020.608045>
- Dickey, J., Lindley, D.V., Press, S.J.: Bayesian estimation of the dispersion matrix of a multivariate normal distribution. *Commun. Stat. Theory Methods* **14**(5), 1019–1034 (1985). <https://doi.org/10.1080/03610928508828960>
- Dillon, J.V., Langmore, I., Tran, D., et al.: Tensorflow distributions. arXiv preprint (2017) <https://doi.org/10.48550/arXiv.1711.10604>
- Dinh, L., Krueger, D., Bengio, Y.: Nice: Non-linear independent components estimation. In: ICLR Workshop, (2014) <https://doi.org/10.48550/arXiv.1410.8516>
- Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real nvp. (2016) arXiv preprint <https://doi.org/10.48550/arXiv.1605.08803>
- Draxler, F., Wahl, S., Schnörr, C., et al.: On the universality of coupling-based normalizing flows. (2024) arXiv preprint <https://doi.org/10.48550/arXiv.2402.06578>
- Durkan, C., Bekasov, A., Murray, I., et al.: Neural spline flows. *Adv. Neural. Inf. Process. Syst.* **32**, (2019). <https://doi.org/10.5555/3454287.3454962>
- European Food Safety Authority: Guidance on expert knowledge elicitation in food and feed safety risk assessment. *EFSA J.* **12**(6), 3734 (2014). <https://doi.org/10.2903/j.efsa.2014.3734>
- Falconer, J.R., Frank, E., Polaschek, D.L., et al.: Methods for eliciting informative prior distributions: A critical review. *Decis. Anal.* **19**(3), 189–204 (2022). <https://doi.org/10.1287/deca.2022.0451>
- Falconer, J.R., Frank, E., Polaschek, D.L., et al.: Eliciting informative priors by modeling expert decision making. *Decis. Anal.* **21**(2), 77–90 (2023). <https://doi.org/10.1287/deca.2023.0046>
- Fernández, C., Steel, M.F.: On bayesian modeling of fat tails and skewness. *J. Am. Stat. Assoc.* **93**(441), 359–371 (1998). <https://doi.org/10.2307/2669632>
- Feydy, J., Séjourné, T., Vialard, F.X., et al.: Interpolating between optimal transport and mmd using sinkhorn divergences. In: The 22nd International Conference on Artificial Intelligence and Statistics, PMLR, pp 2681–2690, (2019) <https://proceedings.mlr.press/v89/feydy19a.html>
- Figurnov, M., Mohamed, S., Mnih, A.: Implicit reparameterization gradients. *Adv. Neural. Inf. Process. Syst.* **31**, (2018). <https://doi.org/10.5555/3326943.3326984>

- Gabry, J., Simpson, D., Vehtari, A., et al.: Visualization in bayesian workflow. *J. R. Stat. Soc. Ser. A Stat. Soc.* **182**(2), 389–402 (2019). <https://doi.org/10.1111/rssa.12378>
- Garthwaite, P.H., Kadane, J.B., O'Hagan, A.: Statistical methods for eliciting probability distributions. *J. Am. Stat. Assoc.* **100**(470), 680–701 (2005). <https://doi.org/10.1198/016214505000000105>
- Garthwaite, P.H., Al-Awadhi, S.A., Elfadaly, F.G., et al.: Prior distribution elicitation for generalized linear and piecewise-linear models. *J. Appl. Stat.* **40**(1), 59–75 (2013). <https://doi.org/10.1080/02664763.2012.734794>
- Gelman, A., Simpson, D., Betancourt, M.: The prior can often only be understood in the context of the likelihood. *Entropy* **19**(10), 555 (2017). <https://doi.org/10.3390/e19100555>
- Gelman, A., Goodrich, B., Gabry, J., et al.: R-squared for bayesian regression models. *Am. Stat.* pp 307–309. (2019) <https://doi.org/10.1080/00031305.2018.1549100>
- Gelman, A., Vehtari, A., Simpson, D., et al.: Bayesian workflow. (2020) arXiv preprint <https://doi.org/10.48550/arXiv.2011.01808>
- Gokhale, D., Press, S.J.: Assessment of a prior distribution for the correlation coefficient in a bivariate normal distribution. *J. R. Stat. Soc. Ser. A Stat. Soc.* **145**(2), 237–249 (1982). <https://doi.org/10.2307/2981537>
- Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016)
- Gosling, J.P.: Shelf: the sheffield elicitation framework, pp. 61–93. The science and art of structuring judgement, Elicitation (2018)
- Gretton, A., Borgwardt, K., Rasch, M., et al.: A kernel method for the two-sample-problem. *Adv. Neural. Inf. Process. Syst.* **19**, (2006). <https://doi.org/10.5555/2976456.2976521>
- Harris, C.R., Millman, K.J., van der Walt, S.J., et al.: Array programming with NumPy. *Nature* **585**(7825), 357–362 (2020). <https://doi.org/10.1038/s41586-020-2649-2>
- Hartmann, M., Agiashvili, G., Bürkner, P., et al.: Flexible prior elicitation via the prior predictive distribution. In: Peters J, Sontag D (eds) Proc. Conf. UAI, PMLR, pp 1129–1138, (2020) <https://proceedings.mlr.press/v124/hartmann20a.html>
- Johnson, S.R., Tomlinson, G.A., Hawker, G.A., et al.: Methods to elicit beliefs for Bayesian priors: a systematic review. *J. Clin. Epidemiol.* **63**(4), 355–369 (2010). <https://doi.org/10.1016/j.jclinepi.2009.06.003>
- Kadane, J., Wolfson, L.J.: Experiences in elicitation. *Statistician* **47**(1), 3–19 (1998). <https://doi.org/10.1111/1467-9884.00113>
- Kadane, J.B., Dickey, J.M., Winkler, R.L., et al.: Interactive elicitation of opinion for a normal linear model. *J. Am. Stat. Assoc.* **75**(372), 845–854 (1980). <https://doi.org/10.1080/01621459.1980.10477562>
- Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems* **31** (2018)
- Kingma, D.P., Welling, M.: Auto-encoding variational bayes. stat 1050:1. (2014) <https://doi.org/10.48550/arXiv.1312.6114>
- Kobyzev, I., Prince, S.J., Brubaker, M.A.: Normalizing flows: An introduction and review of current methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**(11), 3964–3979 (2020). <https://doi.org/10.1109/TPAMI.2020.2992934>
- Lipman, Y., Chen, R.T., Ben-Hamu, H., et al.: Flow matching for generative modeling. (2022) arXiv preprint <https://doi.org/10.48550/arXiv.2210.02747>
- Liu, S., Johns, E., Davison, A.J.: End-to-end multi-task learning with attention. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1871–1880, (2019) <https://doi.org/10.1109/CVPR.2019.00197>
- Manderson, A.A., Goudie, R.J.: Translating predictive distributions into informative priors. (2023) arXiv preprint <https://doi.org/10.48550/arXiv.2303.08528>
- Martin, T.G., Burgman, M.A., Fidler, F., et al.: Eliciting expert knowledge in conservation science. *Conserv. Biol.* **26**(1), 29–38 (2012). <https://doi.org/10.1111/j.1523-1739.2011.01806.x>
- McKinney, W., et al.: Data structures for statistical computing in python. In: Proceedings of the 9th Python in Science Conference, Austin, TX, pp 51–56, (2010) <https://doi.org/10.25080/Majora-92bf1922-00a>
- Mikkola, P., Martin, O.A., Chandramouli, S., et al.: Prior knowledge elicitation: The past, present, and future. *Bayesian Anal.* **1**(1), 1–33 (2023). <https://doi.org/10.1214/23-BA1381>
- Mikkola, P., Acerbi, L., Klami, A.: Preferential normalizing flows. (2024) arXiv preprint <https://doi.org/10.48550/arXiv.2410.08710>
- Morris, D.E., Oakley, J.E., Crowe, J.A.: A web-based tool for eliciting probability distributions from experts. *Environ. Model. Softw.* **52**, 1–4 (2014). <https://doi.org/10.1016/j.envsoft.2013.10.010>
- Nemani, V., Biggio, L., Huan, X., et al.: Uncertainty quantification in machine learning for engineering design and health prognostics: A tutorial. *Mech. Syst. Signal Process.* **205**, 110796 (2023). <https://doi.org/10.1016/j.ymssp.2023.110796>
- Oakley, J.E., O'Hagan, A.: Uncertainty in prior elicitation: a nonparametric approach. *Biometrika* **94**(2), 427–441 (2007). <https://doi.org/10.1093/biomet/asm031>
- O'Hagan, A., Oakley, J.E.: Probability is perfect, but we can't elicit it perfectly. *Reliab. Eng. Syst. Saf.* **85**(1–3), 239–248 (2004). <https://doi.org/10.1016/j.ress.2004.03.014>
- O'Hagan, A., Buck, C.E., Daneshkhah, A., et al.: Uncertain judgements: eliciting experts' probabilities. John Wiley & Sons (2006)
- O'Hagan, A.: Expert knowledge elicitation: subjective but scientific. *Am. Stat.* **73**(sup1), 69–81 (2019). <https://doi.org/10.1080/00031305.2018.1518265>
- Papamakarios, G., Nalisnick, E., Rezende, D.J., et al.: Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn.* **22**(57), 1–64 (2021). (<http://jmlr.org/papers/v22/19-1028.html>)
- Perepolkin, D., Lindström, E., Sahlin, U.: Quantile-parameterized distributions for expert knowledge elicitation. (2023) OSF preprint <https://doi.org/10.31219/osf.io/tq3an>
- Perepolkin, D., Goodrich, B., Sahlin, U.: Hybrid elicitation and quantile-parametrized likelihood. *Stat. Comput.* **34**(1), 11 (2024). <https://doi.org/10.1007/s11222-023-10325-0>
- Peterson, C., Miller, A.: Mode, median, and mean as optimal strategies. *J. Exp. Psychol.* **68**(4), 363 (1964). <https://doi.org/10.1037/h0040387>
- Radev, S.T., Mertens, U.K., Voss, A., et al.: Bayesflow: Learning complex stochastic models with invertible neural networks. *IEEE Trans. Neural. Netw. Learn. Syst.* **33**(4), 1452–1466 (2020). <https://doi.org/10.1109/TNNLS.2020.3042395>
- Radev, S.T., Schmitt, M., Schumacher, L., et al.: Bayesflow: Amortized bayesian workflows with neural networks. *J. Open Source Softw.* **8**(89), 5702 (2023). <https://doi.org/10.21105/joss.05702>
- Regenwetter, M., Cavagnaro, D.R.: Tutorial on removing the shackles of regression analysis: How to stay true to your theory of binary response probabilities. *Psychol. methods* **24**(2), 135 (2019). <https://doi.org/10.1037/met0000196>
- Igorzata, Roos, M., Martins, T.G., Held, L., et al.: Sensitivity analysis for bayesian hierarchical models. *Bayesian Anal.* **10**(2), 321–349 (2015). <https://doi.org/10.1214/14-BA909>
- Scott, J.G.: Prior specification is engineering, not mathematics. *Stat. Sci.* **32**(1), 29–32. (2017) DOI: <https://doi.org/10.1214/16-STS576>
- da Silva, EdS, Kuśmierczyk, T., Hartmann, M., et al.: Prior specification via prior predictive matching: Poisson matrix factorization and beyond. (2019) Preprint at <https://doi.org/10.48550/arXiv.1910.12263>
- Simpson, D., Rue, H., Riebler, A., et al.: Penalising model component complexity: A principled, practical approach to constructing

- priors. *Stat. Sci.* **32**(1), 1–28 (2017). <https://doi.org/10.1214/16-STS576>
- Stefan, A.M., Evans, N.J., Wagenmakers, E.J.: Practical challenges and methodological flexibility in prior elicitation. *Psychol. Methods* **27**(2), 177–197 (2022). <https://doi.org/10.1037/met0000354>
- Van Dongen, S.: Prior specification in bayesian statistics: three cautionary tales. *J. Theor. Biol.* **242**(1), 90–100 (2006). <https://doi.org/10.1016/j.jtbi.2006.02.002>
- Wagenmakers, E.J., Farrell, S.: AIC model selection using akaike weights. *Psychon. Bull. Rev.* **11**, 192–196 (2004). <https://doi.org/10.3758/BF03206482>
- Wang, L., Ng, A.H., Deb, K.: Multi-objective evolutionary optimisation for product design and manufacturing. Springer (2011)
- Winkler, R.L.: The assessment of prior distributions in bayesian analysis. *J. Am. Stat. Assoc.* **62**(319), 776–800 (1967). <https://doi.org/10.1080/01621459.1967.10500894>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.