Statistics and Computing (2026) 36:27
https://doi.org/10.1007/s11222-025-10781-w

ORIGINAL PAPER

A note on auxiliary mixture sampling for Bayesian Poisson models

Aldo Gardini[1] · **Fedele Greco**[1] · **Carlo Trivisano**[1]

Received: 7 February 2025 / Accepted: 13 November 2025

Abstract
Bayesian hierarchical Poisson models are an essential tool for analyzing count data. However, designing efficient algorithms to sample from the posterior distribution of the target parameters remains a challenging task. Auxiliary mixture sampling algorithms have been proposed to this aim. They involve two steps of data augmentation: the first leverages the theory of Poisson processes, and the second approximates the residual distribution of the resulting model through a mixture of Gaussian distributions. In this way, an approximate Gibbs sampler can be implemented. This strategy is particularly beneficial for latent Gaussian models, as it allows one to exploit the sparsity of the precision matrix associated with the random effects and to efficiently incorporate linear constraints. In this paper, we focus on the accuracy of the approximation step, highlighting scenarios where the mixture fails to represent accurately the true underlying distribution, leading to a lack of convergence in the algorithm. We outline key features to monitor, in order to assess if the approximation performs as intended. Building on this, we propose a robust version of the auxiliary mixture sampling algorithm. Our approach includes mechanisms for detecting approximation failures and introduces an enhanced approximation of the right tail of the auxiliary variable distribution, supplemented by a Metropolis-Hastings correction step when needed. Finally, we evaluate the proposed algorithm together with the original mixture sampling algorithms on both simulated and real datasets.

Keywords Data augmentation · Gaussian mixture · Latent Gaussian models · Metropolis-Hastings

1 Introduction

In this paper, we consider Markov Chain Monte Carlo (MCMC, Robert and Casella 1999) sampling of Bayesian hierarchical models with Poisson likelihood. In these models, the expected value of the response variable, which consists of a vector of observed counts $\mathbf{y} = (y_1, \ldots, y_i, \ldots, y_n)^\top \in \mathbb{N}_0^n$, is modelled as

$$y_i|\lambda_i \overset{ind}{\sim} \mathcal{P}(t_i\lambda_i), \quad i = 1, \ldots, n; \tag{1}$$

where t_i represents an offset and the intensity parameter λ_i is linked to a linear predictor η_i by the canonical link function $\lambda_i = \exp(\eta_i)$, $i = 1, \ldots, n$. The primary focus of the paper is on a widespread subclass of hierarchical models known as Latent Gaussian Models (LGMs). They are characterized by a linear predictor $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_i, \ldots, \eta_n)^\top$ constituted by a priori independent additive components distributed as Gaussian random variables conditionally on model hyperparameters:

$$\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta} + \sum_{q=1}^{Q} \mathbf{Z}_q \boldsymbol{\gamma}_q. \tag{2}$$

The design matrix $\mathbf{X} \in \mathbb{R}^{n\times(P+1)}$ is associated with an overall intercept β_0 and P fixed effects collected in the vector $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_P)^\top$. Random components are expressed as the product of a design matrix for random effects $\mathbf{Z}_q \in \mathbb{R}^{n\times m_q}$, with $m_q \leq n$ and a random vector $\boldsymbol{\gamma}_q \in \mathbb{R}^{m_q}$, $q = 1, \ldots, Q$, which in LGMs follows a Gaussian distribution.

In a Bayesian framework, specification of the Poisson regression mixed model outlined in Equations (1)-(2) can be completed by a Gaussian prior on the coefficients vector $\boldsymbol{\beta} \sim \mathcal{N}_{P+1}(\mathbf{0}, \mathbf{V}_0)$, while the prior for the random effects is $\boldsymbol{\gamma}_q|\sigma_q^2 \sim \mathcal{N}_{m_q}\left(\mathbf{0}, \sigma_q^2\mathbf{K}_q^{-1}\right)$, where σ_q^2 is a scale parameter and $\mathbf{K}_q$ is a possibly structured precision matrix, that is sparse when employing Gaussian Markov Random Fields (GMRF) priors. Linear constraints having form $\mathbf{A}_q\boldsymbol{\gamma}_q = \mathbf{e}_q$ are often required, particularly, though not exclusively, when the pre-

✉ Aldo Gardini
aldo.gardini@unibo.it

[1] Department of Statistical Science 'P. Fortunati', University of Bologna, Bologna 40126, BO, Italy

Published online: 24 November 2025

cision matrix $\mathbf{K}_q$ is rank-deficient as in the notable case of Intrinsic GMRF priors (Rue and Held 2005). The model hierarchy is completed by assigning priors to the scale parameters σ_q^2; however, the core aspects of the algorithms presented in this paper are not dependent on this specific prior choice.

The problem of estimating Poisson regression models in the Bayesian paradigm has led to several research studies in computational statistics which propose efficient MCMC algorithms. Indeed, due to the lack of conditional conjugacy under Gaussian and conditional Gaussian priors for the coefficients, the use of Metropolis-Hastings (MH) or Hamiltonian Monte Carlo algorithms are required, and some strategies for the Poisson model have been proposed for particular contexts; for example Knorr-Held and Rue (2002) focused on a block-updating sampler for spatial models.

Alternatively, a popular line of research focuses on the adoption of data augmentation schemes, pioneered by Albert and Chib (1993) in the framework of probit regression. In the context of Poisson LGMs, the methods proposed by Frühwirth-Schnatter and Wagner (2006) and Frühwirth-Schnatter et al. (2009) are appealing. The authors proposed a two-step data augmentation approach: the first step reformulates the model in terms of inter-arrival times, leveraging the theory of Poisson processes, while the second step approximates the distribution of the residuals of the augmented model using a mixture of Gaussian distributions.

This approach is particularly convenient for estimating Poisson LGMs, as it facilitates the adaptation of efficient MCMC samplers designed for linear LGMs to the Poisson case. Specifically, these strategies enable the use of Gibbs samplers, as the full conditionals of the coefficients and random effects in LGMs reduce to Gaussian distributions. We emphasize the importance of the availability of such algorithms, as they can be tailored to accommodate specific features of LGMs, preserving the sparsity of the precision matrices $\mathbf{K}_q$ and enabling coefficient sampling under linear constraints defined by $\mathbf{A}_q$. These capabilities become increasingly critical as model structures grow in complexity (see Rue and Held 2005, for a general overview of computation al aspects of LGMs). Note that Frühwirth-Schnatter et al. (2009) show that their algorithm outperforms the block-updating MH approach that uses a proposal based on GMRF approximation (Knorr-Held and Rue 2002). General-purpose probabilistic programming languages, such as `Stan` (Stan Development Team 2024), have certain limitations in efficiently implementing samplers that can exploit the sparsity of precision matrices and incorporate linear constraints. The algorithms based on the aforementioned data augmentation strategies are relatively straightforward to implement. Moreover, several R packages, such as `MCMCpack` (Martin et al. 2011) and `pogit` (Dvorzak and Wagner 2016), employ these methods to fit Poisson models. From a different perspective, an approximation-based approach was proposed by D'Angelo and Canale (2023), who leveraged the convergence of the Negative Binomial distribution to the Poisson distribution to incorporate the Pólya-Gamma augmentation scheme of Polson et al. (2013). This scheme was used to construct a proposal distribution within an MH algorithm, and the authors considered only the case of fixed-effect models.

In this paper, we focus on the data augmentation schemes proposed by Frühwirth-Schnatter and Wagner (2006) and Frühwirth-Schnatter et al. (2009), for their convenience in estimating LGMs. In this context, we demonstrate that carefully assessing the accuracy of the mixture approximation is essential, as poor approximation can lead to failure to converge to the target posterior distribution. We conduct a detailed analysis of the approximation accuracy and show that the primary discrepancies arise in the tails of the auxiliary variable's distribution. In particular, the Gaussian mixture fails to accurately capture the true tail behavior, exhibiting slower decay in the left tail and faster decay in the right tail. To address these limitations, we propose a robust sampling strategy that combines an adjusted mixture density approximation with an MH step. We demonstrate that applying the MH step alone does not ensure convergence and generally leads to a less efficient algorithm compared to the approach based on the adjusted mixture. Given the increased computational cost of the robust method, we also introduce an automatic detection procedure that activates the robust sampler only when samples from the tails of the auxiliary variable's distribution are needed, and otherwise defaults to the algorithm proposed by Frühwirth-Schnatter et al. (2009). The effectiveness of our approach is demonstrated through both a simulated example and two real-data case studies. The proposed algorithm is implemented in the R package `SamplerPoisson` (https://github.com/agardini/SamplerPoisson), and the R code used to reproduce all results in this paper is provided as supplementary material.

The remainder of the paper is structured as follows. Section 2 reviews the two auxiliary mixture sampling algorithms considered. Section 3 highlights issues related to the accuracy of the mixture approximation step, while Section 4 introduces a robust version of the sampling algorithms designed to address these approximation challenges. Section 5 compares the performance of the discussed algorithms on both simulated and real-world data. Finally, Section 6 provides concluding remarks.

2 Augmentation schemes

Exploiting the properties of the Poisson process, Frühwirth-Schnatter and Wagner (2006) expressed the observations in a Poisson regression model as a sequence of inter-arrival times, leading to a data augmentation scheme that is described in Section 2.1. This algorithm was improved by

Frühwirth-Schnatter et al. (2009), leading to an appealing and computationally convenient algorithm that we summarise in Section 2.2.

2.1 Auxiliary mixture sampling

To express the Poisson distribution in terms of inter-arrival times, Frühwirth-Schnatter and Wagner (2006) introduced $y_i + 1$ independent Exponential random variables with parameter λ_i for each sampling count $y_i, i = 1, \ldots, n$. This sequence of random variables is defined as

$$\tau_{ij}|\lambda_i = \frac{\zeta_{ij}}{\lambda_i};\ j = 1, \ldots, y_i + 1;\ i = 1, \ldots, n; \tag{3}$$

where $\zeta_{ij} \overset{ind}{\sim} \text{Exp}(1)$, $\forall(i, j)$. Taking the negative of the logarithm, the model can be expressed through the following linear relationship:

$$y^*_{ij} = \mathbf{x}_i^\top \boldsymbol{\beta} + \sum_{q=1}^{Q} \mathbf{z}_{qi}^\top \boldsymbol{\gamma}_q + \varepsilon_{ij},\ \forall(i, j); \tag{4}$$

where $y^*_{ij} = -\log(\tau_{ij})$ is the auxiliary variable determining the linear model and the error term $\varepsilon_{ij} = -\log\left(\zeta_{ij}\right)$ follows a Negative Log-Gamma (NLG) distribution. Such distribution plays a fundamental role in both augmentation schemes: the density function of a random variable $U \sim \text{NLG}(\psi, 1)$ is

$$f_U(u) = \frac{\exp\left\{-u\psi - e^{-u}\right\}}{\Gamma(\psi)}. \tag{5}$$

Under this augmentation scheme, the $n^* = n + \sum_{i=1}^{n} y_i$ auxiliary variables follow a NLG distribution with $\psi = 1$:

$$\varepsilon_{ij} \overset{ind}{\sim} \text{NLG}(1, 1), \qquad \forall(i, j).$$

Frühwirth-Schnatter and Wagner (2006) proposed to introduce a second step of data augmentation by approximating the density function (5) with $\psi = 1$ through a mixture of K Gaussian distributions

$$g_\varepsilon(z) = \sum_{k=1}^{K} w_k \phi\left(z; m_k, s_k^2\right). \tag{6}$$

In this expression, $\phi(z; m_k, s_k^2)$ indicates the density function of the k-th Gaussian component, evaluated in z with mean m_k, variance s_k^2, and having weight w_k. Under this approximation for the distribution of the error term, a Gaussian linear model emerges conditionally on the labels $r_{ij} \in \{1, \ldots, K\}$ which identify the mixture component:

$$y^*_{ij}|r_{ij} = \mathbf{x}_i^\top \boldsymbol{\beta} + \sum_{q=1}^{Q} \mathbf{z}_{qi}^\top \boldsymbol{\gamma}_q + m_{r_{ij}} + \varepsilon_{ij},$$
$$\varepsilon_{ij}|r_{ij} \sim \mathcal{N}(0, s^2_{r_{ij}}).$$

To express the model in matrix form, let $\mathbf{r} \in \{1, \ldots, K\}^{n^*}$ denote the n^*-dimensional vector of mixture labels, and define the vectors containing the means and variances of the mixture components conditionally on the specific labels $\mathbf{m}(\mathbf{r}) = (m_{r_{11}}, \ldots, m_{r_{n,y_n+1}})^\top$ and $\mathbf{s}^2(\mathbf{r}) = (s^2_{r_{11}}, \ldots, s^2_{r_{n,y_n+1}})^\top$. Storing the auxiliary variables into $\mathbf{y}^*$ and the corresponding design matrices in $\mathbf{X}^* \in \mathbb{R}^{n^* \times P+1}$ and $\mathbf{Z}^*_q \in \mathbb{R}^{n^* \times M_q}$, $\forall q$, the linear model conditional on the latent labels indicating the mixture components can be expressed in matrix form as:

$$\mathbf{y}^*|\mathbf{r} = \mathbf{X}^*\boldsymbol{\beta} + \sum_{q=1}^{Q} \mathbf{Z}^*_q \boldsymbol{\gamma}_q + \mathbf{m}(\mathbf{r}) + \boldsymbol{\varepsilon},$$
$$\boldsymbol{\varepsilon}|\mathbf{r} \sim \mathcal{N}_{n^*}\left(\mathbf{0}, \mathbf{D}(\mathbf{r})\right),$$

where $\mathbf{D}(\mathbf{r}) = \text{diag}\left(\mathbf{s}^2(\mathbf{r})\right)$.

The MCMC algorithm, denoted as the AMS (Auxiliary Mixture Sampling) algorithm in what follows, is implemented by iterating through the following steps. Note that, to simplify notation, we use $\boldsymbol{\gamma} = [\boldsymbol{\gamma}_1^\top \cdots \boldsymbol{\gamma}_Q^\top]^\top$ and $\mathbf{Z}^* = [\mathbf{Z}^*_1 \cdots \mathbf{Z}^*_Q]$ when it is not necessary to distinguish between specific random effects.

- *Step 1.* For each i, $y_i + 1$ auxiliary variables τ_{ij}, $j = 1, \ldots, y_i + 1$ are generated.
  - For $j = 1, \ldots y_i$, auxiliary variables are constituted by a sorted sample of size y_i drawn from a Uniform distribution on the interval $[0, 1]$ since the arrival times of a Poisson process conditioned on having observed a given number of jumps are distributed as the order statistics of a $\mathcal{U}(0, 1)$ distribution.
  - The $(y_i + 1)-$th auxiliary variable is drawn exploiting that $\tau_{i(y_i+1)} = 1 - \sum_{j=1}^{y_i} \tau_{ij} + \zeta_i/\lambda_i$, where $\zeta_i \sim \text{Exp}(1)$. This generates a realization of the n^* auxiliary variables $\mathbf{y}^*$.
- *Step 2.* The mixture component indicators r_{ij} are sampled with probabilities proportional to $\mathbb{P}[r_{ij} = k|y^*_{ij}, \boldsymbol{\beta}, \boldsymbol{\gamma}]$, obtaining $\mathbf{r}$.
- *Step 3.* Fixed effects $\boldsymbol{\beta}$ are drawn from the approximated full conditional:

$$\boldsymbol{\beta}|\mathbf{y}^*, \mathbf{r}, \boldsymbol{\gamma} \sim N\left(\boldsymbol{\mu}_\beta(\mathbf{r}), \boldsymbol{\Sigma}_\beta(\mathbf{r})\right), \tag{7}$$

where

$$\boldsymbol{\Sigma}_\beta(\mathbf{r}) = \left(\mathbf{X}^{*\top}\mathbf{D}(\mathbf{r})^{-1}\mathbf{X}^* + \mathbf{V}_0^{-1}\right)^{-1},$$

and

$$\boldsymbol{\mu}_{\beta}(\mathbf{r}) = \boldsymbol{\Sigma}_{\beta}\mathbf{X}^{*\top}\mathbf{D}(\mathbf{r})^{-1}\left(\mathbf{y}^{*} - \mathbf{Z}^{*}\boldsymbol{\gamma} - \mathbf{m}(\mathbf{r})\right).$$

- *Step 4.* Random effects $\boldsymbol{\gamma}_q$, $\forall q$, are drawn from the approximate full conditional distributions:

$$\boldsymbol{\gamma}_q|\mathbf{y}^{*}, \mathbf{r}, \boldsymbol{\beta}, \sigma^2_{\gamma_q} \sim N\left(\boldsymbol{\mu}_{\gamma_q}(\mathbf{r}), \boldsymbol{\Sigma}_{\gamma_q}(\mathbf{r})\right), \tag{8}$$

where

$$\boldsymbol{\Sigma}_{\gamma_q}(\mathbf{r}) = \left(\mathbf{Z}_q^{*\top}\mathbf{D}(\mathbf{r})^{-1}\mathbf{Z}_q^{*} + \sigma_{\gamma_q}^{-2}\mathbf{K}_q\right)^{-1},$$

and

$$\boldsymbol{\mu}_{\gamma_q}(\mathbf{r}) = \boldsymbol{\Sigma}_{\gamma_q}\mathbf{Z}_q^{*\top}\mathbf{D}(\mathbf{r})^{-1}\left(\mathbf{y}^{*} - \mathbf{X}^{*}\boldsymbol{\beta} - \sum_{j\neq q}\mathbf{Z}_j^{*}\boldsymbol{\gamma}_j - \mathbf{m}(\mathbf{r})\right).$$

- *Step 5.* Scale parameters σ_q^2, $\forall q$, are either drawn from their full conditional distributions or updated using an MH step.

Note that Steps 3-5 are standard steps for sampling a LGM with Gaussian likelihood that can be implemented thanks to the auxiliary variables introduced in Steps 1-2. The ability to apply efficient techniques in Steps 3-5 represents one of the main advantages of the auxiliary variable framework. In particular, sampling from the full conditional distribution (8) can be carried out by taking advantage of the sparsity of the precision matrix $\mathbf{K}_q$. Moreover, sampling under linear constraints from the distribution $\boldsymbol{\gamma}_q \mid \mathbf{A}_q\boldsymbol{\gamma}_q = \mathbf{0}$ can make use of efficient methods such as conditioning by Rue and Held (2005).

2.2 Improved auxiliary mixture sampling

The main drawback of the AMS algorithm is the large number of latent variables introduced, particularly when the total observed count $\sum_{i=1}^{n} y_i$ is large. To address this issue, Frühwirth-Schnatter et al. (2009) propose a more parsimonious scheme, requiring for each observation two auxiliary variables if $y_i > 0$, and one auxiliary variable if $y_i = 0$. Denoting the number of observations equal to zero as n_0, the total number of latent variables is $\widetilde{n}^{*} = 2n - n_0$, which can be much smaller than n^{*}.

In this improved scheme, a Poisson process is defined over the time interval $t \in [0, 1]$. For each observation i, two latent variables are defined conditionally on $y_i > 0$ jumps occurring before $t = 1$. The variable $\widetilde{\tau}_{i2}$ represents the arrival time after the y_i jumps before $t = 1$, while $\widetilde{\tau}_{i1}$ denotes the inter-arrival time of the first jump after $t = 1$. Consequently, the sum $\widetilde{\tau}_{i2} + \widetilde{\tau}_{i1}$ gives the arrival time of the first jump after $t = 1$. For observations where $y_i = 0$, $\widetilde{\tau}_{i2}$ is known to be zero, so only the latent variable $\widetilde{\tau}_{i1}$ is required.

The set of latent variables $\widetilde{\tau}_{i1}$ is defined as in Equation (3), while the distribution of $\widetilde{\tau}_{i2}$ follows from the fact that the arrival time is the sum of y_i independent Exponential random variables with rate λ_i, hence:

$$\widetilde{\tau}_{i2} = \frac{\widetilde{\zeta}_{i2}}{\lambda_i}, \quad \widetilde{\zeta}_{i2} \sim \text{Gamma}(y_i, 1), \ \forall i.$$

Likewise to the AMS data augmentation scheme, it is possible to express the model through a linear relationship as in Equation (4):

$$\widetilde{y}_{ij}^{*} = \mathbf{x}_i^{\top}\boldsymbol{\beta} + \sum_{q=1}^{Q}\mathbf{z}_{qi}^{\top}\boldsymbol{\gamma}_q + \widetilde{\varepsilon}_{ij}, \tag{9}$$

where $\widetilde{y}_{ij}^{*} = -\log\left(\widetilde{\tau}_{ij}\right)$ and $\widetilde{\varepsilon}_{ij} = -\log\left(\widetilde{\zeta}_{ij}\right)$, with $j = 1$ if $y_i = 0$ and $j \in \{1, 2\}$ if $y_i > 0$. Concerning the error term, $\widetilde{\varepsilon}_{ij} \sim \text{NLG}(1, 1)$ for $j = 1$ while, if $y_i > 0$, $\widetilde{\varepsilon}_{ij} \sim \text{NLG}(y_i, 1)$ when $j = 2$. The distribution of $\widetilde{\varepsilon}_{i1}$ is approximated using the same mixture as in Equation (6). Conversely, the approximation of the distribution of $\widetilde{\varepsilon}_{i2}$ depends on y_i. Frühwirth-Schnatter et al. (2009) discuss and tabulate the different vectors of weights, means and variances that characterize the mixture approximation in Equation (6) for $y \in \mathbb{N}_0$.

The sampling scheme derived from this data augmentation strategy will be referred to as the IAMS (Improved Auxiliary Mixture Sampling) algorithm. It essentially follows the same steps as the AMS algorithm, with the only difference occurring in Step 1:

- *Step 1.* Sample $\widetilde{\zeta}_i$ from an Exp(1) distribution.
  - If $y_i = 0$, set $\widetilde{\tau}_{i1} = 1 + \widetilde{\zeta}_i/\lambda_i$
  - If $y_i > 0$, draw $\widetilde{\tau}_{i2}$ from a Beta$(y_i, 1)$ distribution and set $\widetilde{\tau}_{i1} = 1 - \widetilde{\tau}_{i2} + \widetilde{\zeta}_i/\lambda_i$.
- *Steps 2-5.* As AMS algorithm.

3 Investigating the accuracy of the mixture approximation

In this section, our goal is to point out some criticisms related to the second data augmentation step of AMS and IAMS algorithms, namely the approximation of the residuals distributions through a mixture of Gaussian random variables. As highlighted by Frühwirth-Schnatter and Wagner (2006) and Frühwirth-Schnatter et al. (2009), the accuracy of the

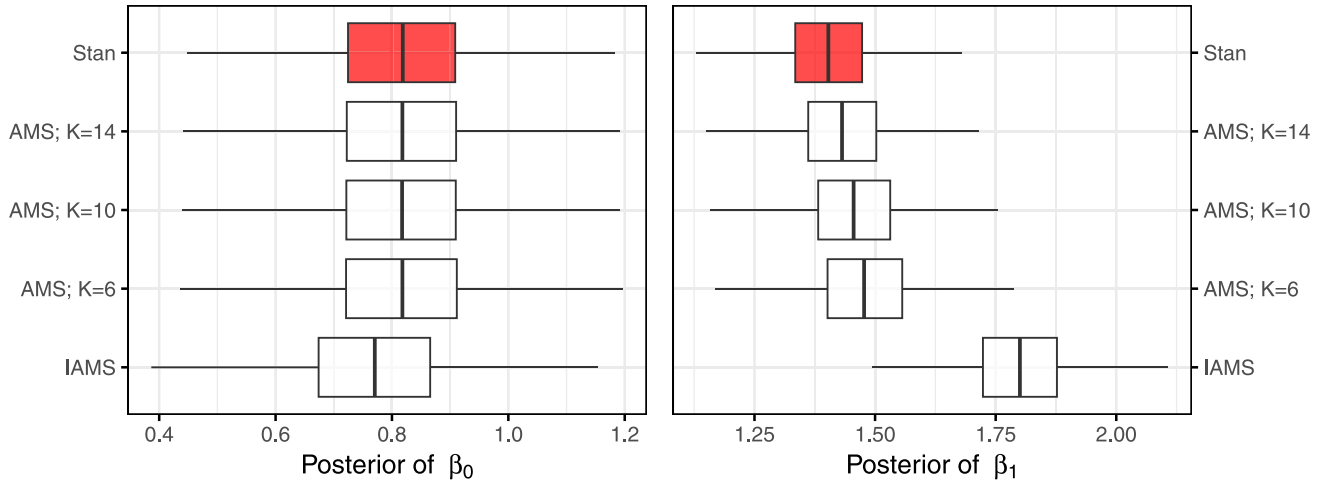

Fig. 1 Boxplot of the posterior distributions for the regression coefficients β_0 and β_1

approximation is generally excellent. However, some issues may arise in the tails, where the decay of the NLG distribution is significantly faster in the left tail and slower in the right tail compared to the Gaussian distributions used in the approximation. It is important to note that in the vast majority of applications where we tested the AMS and IAMS algorithms, they have successfully converged. Nonetheless, there may be instances where the accuracy of the mixture approximation is insufficient to ensure convergence of the algorithms to the target posterior distribution. Such cases can occur when large residuals arise during the first data augmentation process, often due to factors like model misspecification or the presence of outlying or extreme observations. In any case, an MCMC algorithm should reliably converge to the target posterior regardless of difficulties that the peculiar data structure might pose.

To better understand the issues affecting the AMS and IAMS algorithms, we consider a synthetic toy dataset referring to a simple Poisson regression model with fixed covariates. In particular, a sample of size $n = 30$ is generated, with two covariates, x_1 and x_2, drawn from a standard Gaussian distribution. These covariates are then used to generate the response, which follows a Poisson distribution with log-rate given by $\log(\lambda_i) = 0.1 + x_{1i} + 1.2x_{2i}, i = 1, \ldots, 30$. Results presented in what follows come from the estimation of the Poisson regression model with linear predictor

$$\log(\lambda_i) = \beta_0 + \beta_1 x_{1i}, \quad i = 1, \ldots, 30,$$

i.e. we mimic a model misspecification setting by omitting the covariate x_2.

The model is first fitted in `Stan` through its `R` interface `rstan` (Stan Development Team 2024), to get a reliable benchmark. Then, the same model is also fitted by exploiting the AMS and IAMS algorithms. Concerning AMS, in addition to the suggested mixture approximation with $K = 10$ components, we also include the mixtures with $K = 6$ and $K = 14$ components, to show the effect of different approximations on the convergence of the algorithm. The mixture parameters are retrieved following the procedure outlined in Section 2.3 of Frühwirth-Schnatter and Frühwirth (2007).

The posterior distributions of the regression coefficients, based on $B = 100,000$ iterations for each algorithm, are compared in Figure 1, where the results obtained using `Stan` are highlighted in red to underscore that they represent the benchmark. The most noticeable discrepancies are observed using the IAMS algorithm, which leads to miscalibrated posterior distributions for both parameters. On the other hand, under the AMS algorithm the posteriors of β_0 appear to be well calibrated. Conversely, lack of convergence is observed for the posteriors of β_1. In this case, the discrepancies between the obtained posterior and the results from `Stan` gradually decrease when passing from $K = 6$ to $K = 14$, through the recommended $K = 10$ setting, highlighting the role of the approximation in these results. We emphasize that standard convergence diagnostics, such as traceplots and autocorrelation plots (not shown), indicate convergence of the AMS and IAMS algorithms to a stationary distribution, even though they actually converge to a distribution that differs from the target posterior.

Insight into the lack of convergence of the AMS and IAMS algorithms can be obtained by monitoring the differences between the logarithms of the true density of the residuals ε_{ij} (indicated with f_ε) and their mixture approximation (g_ε). In particular, Figure 2 shows the average of this discrepancy over the B MCMC iterations, defined as:

$$\Delta_{ij} = \frac{1}{B} \sum_{b=1}^{B} \left(\log g_\varepsilon \left(y_{ij}^{*(b)} - \mathbf{x}_i^\top \boldsymbol{\beta}^{(b)} \right) \right.$$

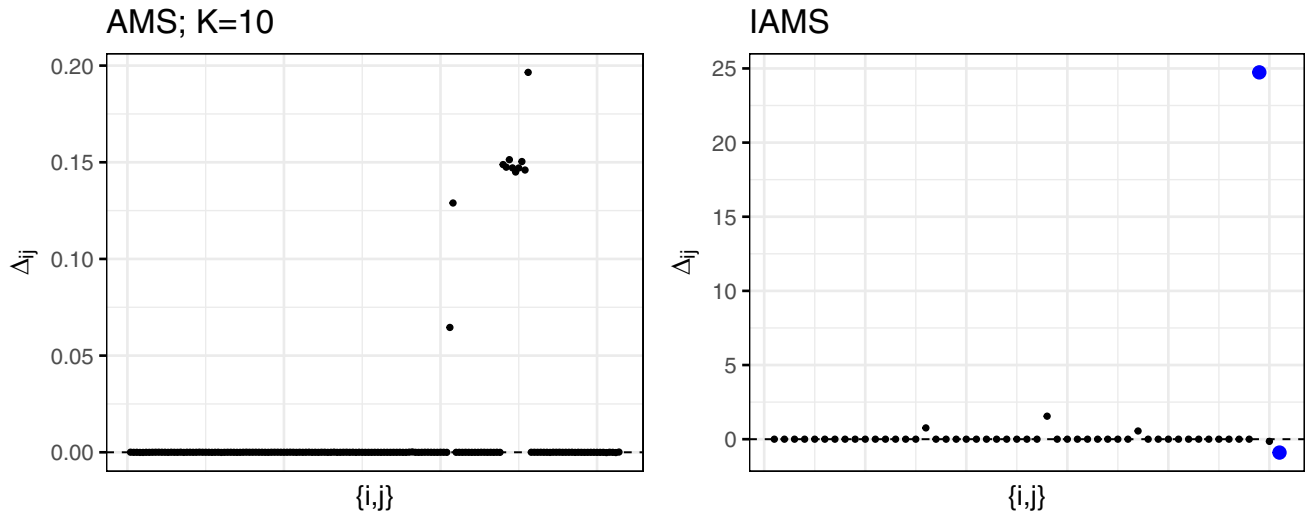

Fig. 2 Average differences between the true log-likelihood and its mixture approximation. The blue dots highlight the larger values

$$-\log f_{\varepsilon}\left(y_{ij}^{*(b)}-\mathbf{x}_i^{\top}\boldsymbol{\beta}^{(b)}\right)\Big), \tag{10}$$

where $y_{ij}^{*(b)}$ should be replaced by $\widetilde{y}_{ij}^{*(b)}$ when dealing with the IAMS algorithm. By comparing the results under the AMS and IAMS algorithms, an explanation for the greater robustness of AMS is that the deviations of Δ_{ij} from zero are much smaller compared to those from the IAMS algorithm.

The blue dots in the right panel of Figure 2 refer to the latent variable residuals for which the highest positive and negative differences are observed for IAMS. Figure 3 focuses on these particular latent variable residuals: in the bottom row of plots, the log-density of the true distribution (red solid line) is compared to its mixture approximation (black dashed line), revealing substantial differences in the tail decay. The top row of plots reports the posterior distributions of the residuals related to the selected auxiliary variables, comparing the posteriors obtained using `Stan` (red lines) with those obtained using IAMS (black lines). Note that, while the distributions of the auxiliary variables are a by-product of the IAMS algorithm, they are not directly available in `Stan`. However, given the posterior distributions of λ_i, Step 1 of the RIAMS algorithm can be used to sample from the distributions of the auxiliary variables and, consequently, to obtain the distribution of residuals. Negative values of Δ_{ij} are observed when the residuals are located in the right tail of the distribution (such as residual #51), and, in this situation, the approximation leads to a left shift of the posterior of the residual. For positive values of Δ_{ij}, as in the case of residual #49, a left shift is observed because the left tail of the mixture approximation is heavier than the NLG distribution: in fact, the Gaussian density cannot mimic the double exponential decay of the NLG density.

4 Improving convergence of the IAMS algorithm

A possible way to design an MCMC algorithm that is robust with respect to potential issues caused by approximating a distribution is the inclusion of a rejection step. Indeed, as it is discussed in Frühwirth-Schnatter and Wagner (2006), the full conditional of the coefficients in Equations (7) and (8) can also be exploited as proposal distributions within an MH algorithm. In this section, we first discuss some aspects related to setting an MH algorithm within the IAMS, then we propose a robust version of the IAMS algorithm.

To clarify the integration of an MH step into the IAMS algorithm, we consider the case of sampling from the posterior of $\boldsymbol{\beta}$, though the same considerations easily extend to the random coefficients $\boldsymbol{\gamma}$. In more detail, at iteration b, the acceptance probability for a proposed vector of coefficients $\boldsymbol{\beta}^{\text{prop}}$ depends on the previous value $\boldsymbol{\beta}^{(b-1)}$ and the auxiliary variables $\mathbf{y}^{*(b)}$, regardless of the specific scheme under consideration. This acceptance probability is defined as $\alpha = \min\left(1, \pi\left(\boldsymbol{\beta}^{\text{prop}}, \boldsymbol{\beta}^{(b-1)}, \mathbf{y}^{*(b)}, \boldsymbol{\gamma}^{(b-1)}\right)\right)$, where

$$\pi\left(\boldsymbol{\beta}^{\text{prop}}, \boldsymbol{\beta}^{(b-1)}, \mathbf{y}^{*(b)}, \boldsymbol{\gamma}^{(b-1)}\right) = \frac{p\left(\boldsymbol{\beta}^{\text{prop}}|\mathbf{y}^{*(b)}, \boldsymbol{\gamma}^{(b-1)}\right)}{p\left(\boldsymbol{\beta}^{(b-1)}|\mathbf{y}^{*(b)}, \boldsymbol{\gamma}^{(b-1)}\right)} \times \frac{\mathcal{K}\left(\boldsymbol{\beta}^{(b-1)}|\boldsymbol{\beta}^{\text{prop}}\right)}{\mathcal{K}\left(\boldsymbol{\beta}^{\text{prop}}|\boldsymbol{\beta}^{(b-1)}\right)}. \tag{11}$$

This ratio is a function of the target posterior

$$p\left(\boldsymbol{\beta}|\mathbf{y}^{*}, \boldsymbol{\gamma}\right) \propto \mathcal{L}\left(\boldsymbol{\beta}, \boldsymbol{\gamma}; \mathbf{y}^{*}\right) p(\boldsymbol{\beta}) \tag{12}$$

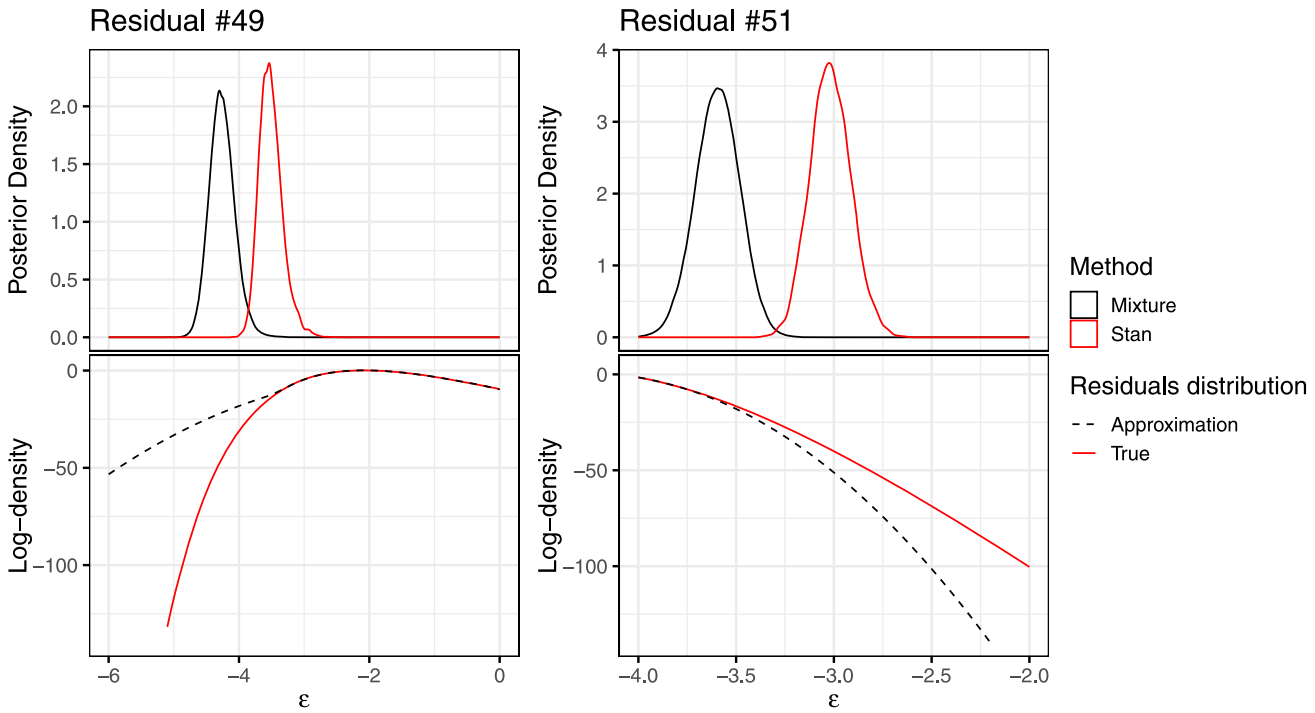

Fig. 3 Upper panel plots: posterior density of residuals related to two auxiliary variables obtained using `Stan` (red) and the IAMS (black) sampler. Bottom panel plots: log density of the true auxiliary variable distribution (red) and its mixture approximation (black)

and the transition kernel

$$\mathcal{K}\left(\boldsymbol{\beta}^{(b-1)}|\boldsymbol{\beta}^{\text{prop}}\right) = \mathcal{L}_a\left(\boldsymbol{\beta}^{(b-1)}, \boldsymbol{\gamma}^{(b-1)}; \mathbf{y}^{*(b)}\right) p\left(\boldsymbol{\beta}^{(b-1)}\right); \tag{13}$$

which, in turn, depend on the augmented likelihood $\mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\gamma}; \mathbf{y}^*) = \prod_{i=1}^n \prod_j f_\varepsilon\left(y_{ij}^* - \mathbf{x}_i^\top \boldsymbol{\beta} - \mathbf{z}_i^\top \boldsymbol{\gamma}\right)$ and its approximation $\mathcal{L}_a(\boldsymbol{\beta}, \boldsymbol{\gamma}; \mathbf{y}^*) = \prod_{i=1}^n \prod_j g_\varepsilon\left(y_{ij}^* - \mathbf{x}_i^\top \boldsymbol{\beta} - \mathbf{z}_i^\top \boldsymbol{\gamma}\right)$. Note that the range of index j is left unspecified as it depends on the data augmentation scheme. Finally, by plugging Equations (12) and (13) into (11), it follows that

$$\pi\left(\boldsymbol{\beta}^{\text{prop}}, \boldsymbol{\beta}^{(b-1)}, \mathbf{y}^{*(b)}, \boldsymbol{\gamma}^{(b-1)}\right) = \frac{\mathcal{L}\left(\boldsymbol{\beta}^{\text{prop}}, \boldsymbol{\gamma}^{(b-1)}; \mathbf{y}^{*(b)}\right)}{\mathcal{L}\left(\boldsymbol{\beta}^{(b-1)}, \boldsymbol{\gamma}^{(b-1)}; \mathbf{y}^{*(b)}\right)} \times \frac{\mathcal{L}_a\left(\boldsymbol{\beta}^{(b-1)}, \boldsymbol{\gamma}^{(b-1)}; \mathbf{y}^{*(b)}\right)}{\mathcal{L}_a\left(\boldsymbol{\beta}^{\text{prop}}, \boldsymbol{\gamma}^{(b-1)}; \mathbf{y}^{*(b)}\right)}. \tag{14}$$

From the transition kernel expression (13), we can point out that an MH algorithm with independent proposal distribution is being considered, as $\mathcal{K}\left(\boldsymbol{\beta}^{(b-1)}|\boldsymbol{\beta}^{\text{prop}}\right) = \mathcal{K}\left(\boldsymbol{\beta}^{(b-1)}\right)$ due to the independence on $\boldsymbol{\beta}^{\text{prop}}$. In addition, the full conditional distributions of the coefficients involved in the AMS and IAMS algorithms play the role of proposal distribution in the MH sampler.

When dealing with an MH algorithm with independent proposal, it is important to verify the conditions for convergence, as the ratio between the proposal and the target strongly affects the acceptance rate. According to Theorem 2.1 in Mengersen and Tweedie (1996), the convergence of an independent MH algorithm requires that there exists a number $t > 0$ such that

$$\frac{\mathcal{K}(\boldsymbol{\beta})}{p(\boldsymbol{\beta}|\mathbf{y}^*)} \geq t, \ \boldsymbol{\beta} \in \mathbb{R}^p. \tag{15}$$

In other words, the proposal distribution must have heavier tails than the target posterior.

When a Gaussian finite mixture is used to approximate the residuals distribution, some issues may arise in this context. Indeed, as noted in the bottom line plots of Figure 3, the left tail of the approximation dominates that of the NLG distribution, while its right tail is dominated by the NLG. This feature can prevent convergence of the MH algorithm. To verify whether condition (15) holds, we consider the simple model:

$$y_i|\mu \overset{ind}{\sim} \mathcal{P}(t_i \exp\{\mu\}), \ i = 1, \ldots, n,$$

which leads to the following model in terms of auxiliary variables, regardless of the specific augmentation scheme used:

$$y_{ij}^* = \mu + \varepsilon_{ij},\ \forall(i,j).$$

Under this model, the ratio in Equation (15) can be expressed as

$$\frac{\mathcal{K}(\boldsymbol{\beta})}{p(\boldsymbol{\beta}|\mathbf{y}^*)} = \frac{\prod_{i=1}^n \prod_j g_\varepsilon\left(y_{ij}^* - \mu\right)}{\prod_{i=1}^n \prod_j f_\varepsilon\left(y_{ij}^* - \mu\right)}.$$

Plugging the expressions of the densities (5) and (6) into this formula, we obtain:

$$\begin{aligned}\frac{\mathcal{K}(\boldsymbol{\beta})}{p(\boldsymbol{\beta}|\mathbf{y}^*)} &= \frac{\prod_{i=1}^n \prod_j \left(\sum_{k=1}^K w_k (2\pi\sigma_k)^{-1/2} \exp\left\{-\frac{1}{2\sigma_k^2}\left(y_{ij}^*-\mu\right)^2\right\}\right)}{\prod_{i=1}^n \prod_j \Gamma(y_i)^{-1} \exp\left\{-(y_{ij}^*-\mu)y_i - \exp\{-y_{ij}^*+\mu\}\right\}} \\ &= \frac{\sum_{k=1}^K \left(\frac{w_k}{\sqrt{2\pi}\sigma_k}\right)^{n^*} \exp\left\{-\frac{1}{2\sigma_k^2}\left(\sum_{i,j} y_{ij}^{*2} - 2\mu\sum_{i,j} y_{ij}^* + n^*\mu^2\right)\right\}}{\left[\prod_i \Gamma(y_i)^{-n_i^*}\right] \exp\left\{-\sum_{i,j} y_{ij}^* y_i + \mu\sum_i n_i^* y_i - \sum_{i,j} \frac{e^\mu}{e^{y_{ij}^*}}\right\}},\end{aligned} \tag{16}$$

where n_i^* is the number of auxiliary variables associated with the i-th observation. From this result, we observe that

$$\lim_{\mu\to+\infty} \frac{\mathcal{K}(\boldsymbol{\beta})}{p(\boldsymbol{\beta}|\mathbf{y}^*)} = 0, \tag{17}$$

i.e. the right tail of the proposal for μ is lighter than the right tail of the target distribution, leading to the violation of the condition (15). This means that an MH sampler adopting the mixture approximation used in the AMS and IAMS algorithms is not guaranteed to converge because of the fast decay of the right tail.

The above considerations lead us to design an algorithm that incorporates both a rejection step and an adjusted mixture approximation, as detailed in the next subsection. We specifically focus on the IAMS algorithm because of its computational efficiency and suitability for managing large observed counts.

4.1 The Robust IAMS (RIAMS) algorithm

The first step in developing a version of the IAMS algorithm that is robust with respect to large residuals is to find an adjusted mixture approximation of the NLG distribution that preserves the decay of the NLG density in the right tail. While, in principle, using a finite mixture of Gaussian distributions cannot alter the limiting behavior described by Equation (17), it is possible to add mixture components to significantly mitigate the issue. Since obtaining a more accurate approximation by using the algorithm proposed in in Section 2.3 of Frühwirth-Schnatter and Frühwirth (2007) is infeasible for numeric problems, we implement a rough approximation that serves only as a proposal distribution in an MH algorithm.

The adjusted approximation, denoted as $g_\varepsilon^\star$ in what follows, starts by identifying the point where the difference between the NLG density and the original mixture density (6) become, as a rule of thumb, greater than 1 in the log scale: this point is denoted as $\xi_U(y)$ in what follows, since it depends on y, $y \in \mathbb{N}$. Then, we add a new mixture component centered at $m_{K+1}(y) = \xi_U(y)$, followed by 29 equally spaced knots between $m_{K+1}(y)$ and $2.5 \times q_{1-10^{-16}}(y) + 1.5 \times \log(y)$, where $q_{1-10^{-16}}(y)$ is the quantile of order $1 - 10^{-16}$ of the NLG$(y, 1)$ distribution. The dependence on y of the rightmost point where the density is approximated allows to take account of the different decay rate of the right tail of the approximated density as a function of y. The variance of each mixture component is computed by minimizing the difference between the true density and the mixture density at the successive knot. At each iteration, the sum of weights is normalized. Note that, as weights of the new component are very small, the accuracy of the approximation for lower quantiles is preserved, as shown in Figure 4. At the same time, the adjusted mixture more effectively captures the decay of the right tail of the NLG distribution.

The approximation problem we face forces us to adopt a coarse representation of the tail behavior, as it is not possible to precisely follow the slow decay of the NLG distribution with a Gaussian mixture with a finite number of components, as Gaussian densities are characterized by an exponential decay. To address this, the proposed RIAMS algorithm refines the approximation by introducing an MH step within the IAMS algorithm, substituting the original mixture g_ε with $g_\varepsilon^\star$ when necessary, as detailed in the algorithm description that follows. In fact, a preliminary step is included with a twofold purpose: (i) to run warm-up iterations using the IAMS algorithm, thereby reducing sensitivity to initial values far from the posterior distribution, and (ii) to identify residuals falling in the right tail, for which the adjusted mixture $g_\varepsilon^\star$ must be adopted. The RIAMS algorithm consists of the following steps:

- *Step 0a.* Perform T_1 iterations of the IAMS algorithm.
- *Step 0b.* Perform T_2 iterations of the IAMS algorithm and compute, for each residual, the proportions of iterations that exceed $\xi_U(y_i)$:

$$\kappa_{ij}^U = \frac{1}{T_2} \sum_{b=T_1+1}^{T_1+T_2} \mathbf{1}\left(\varepsilon_{ij}^{(b)} > \xi_U(y_i)\right), \qquad \forall(i,j). \tag{18}$$

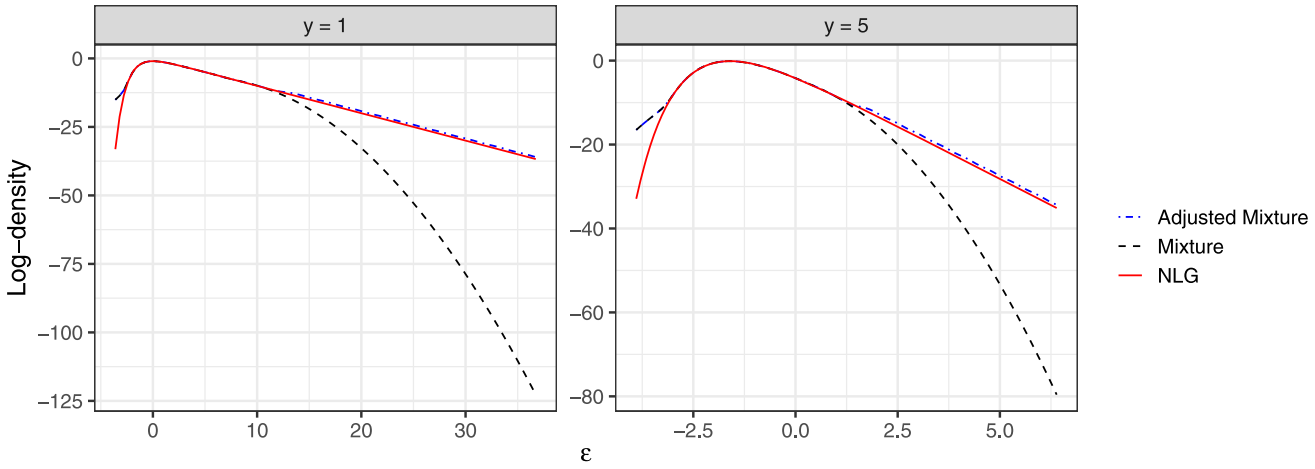

Fig. 4 Mixture approximation g_ε (red dashed line), adjusted mixture approximation $\widetilde{g}_\varepsilon$ (blue dashed line) and exact NLG log-density (black line) for $y = 1$ (left panel) and $y = 5$ (right panel)

At the end of the T_2 iterations, we determine the mixture $\widetilde{g}_\varepsilon(\varepsilon_{ij})$ that is effectively used as approximation:

$$\widetilde{g}_\varepsilon(\varepsilon_{ij}) = \begin{cases} g_\varepsilon(\varepsilon_{ij}) \text{ for } \varepsilon_{ij} \text{ such that } \kappa_{ij}^U \leq p_U; \\ g_\varepsilon^\star(\varepsilon_{ij}) \text{ for } \varepsilon_{ij} \text{ such that } \kappa_{ij}^U > p_U. \end{cases} \tag{19}$$

where the default value of p_U is set to .05 in the `SamplerPoisson` package.

- *Steps 1-2.* As IAMS algorithm.
- *Step 3.* Draw the proposed value $\boldsymbol{\beta}^{\text{prop}}$ from the full conditional as (7), but using the adjusted mixture $\widetilde{g}_\varepsilon$. Accept the proposed value with probability

$$\pi\left(\boldsymbol{\beta}^{\text{prop}}, \boldsymbol{\beta}^{(b-1)}, \mathbf{y}^{*(b)}, \boldsymbol{\gamma}^{(b-1)}\right) = \frac{\mathcal{L}\left(\boldsymbol{\beta}^{\text{prop}}, \boldsymbol{\gamma}^{(b-1)}; \mathbf{y}^{*(b)}\right)}{\mathcal{L}\left(\boldsymbol{\beta}^{(b-1)}, \boldsymbol{\gamma}^{(b-1)}; \mathbf{y}^{*(b)}\right)} \times \frac{\widetilde{\mathcal{L}}_a\left(\boldsymbol{\beta}^{(b-1)}, \boldsymbol{\gamma}^{(b-1)}; \mathbf{y}^{*(b)}\right)}{\widetilde{\mathcal{L}}_a\left(\boldsymbol{\beta}^{\text{prop}}, \boldsymbol{\gamma}^{(b-1)}; \mathbf{y}^{*(b)}\right)}; \tag{20}$$

where $\widetilde{\mathcal{L}}_a\left(\boldsymbol{\beta}, \boldsymbol{\gamma}; \mathbf{y}^*\right) = \prod_{i=1}^n \prod_j \widetilde{g}_\varepsilon\left(y_{ij}^* - \mathbf{x}_i^\top \boldsymbol{\beta} - \mathbf{z}_i^\top \boldsymbol{\gamma}\right)$.

- *Step 4.* Draw the proposed value $\boldsymbol{\gamma}_q^{\text{prop}}$, $\forall q$, from the full conditional as (8), but using the adjusted mixture $\widetilde{g}_\varepsilon$. Accept the proposed value with probability

$$\pi\left(\boldsymbol{\gamma}_q^{\text{prop}}, \boldsymbol{\gamma}_q^{(b-1)}, \mathbf{y}^{*(b)}, \boldsymbol{\beta}^{(b)}, \left\{\boldsymbol{\gamma}_{q'}^\bullet\right\}_{q' \neq q}\right) = \frac{\mathcal{L}\left(\boldsymbol{\beta}^{(b)}, \boldsymbol{\gamma}_q^{\text{prop}}, \left\{\boldsymbol{\gamma}_{q'}^\bullet\right\}_{q' \neq q}; \mathbf{y}^{*(b)}\right)}{\mathcal{L}\left(\boldsymbol{\beta}^{(b)}, \boldsymbol{\gamma}_q^{(b-1)}, \left\{\boldsymbol{\gamma}_{q'}^\bullet\right\}_{q' \neq q}; \mathbf{y}^{*(b)}\right)} \times \frac{\widetilde{\mathcal{L}}_a\left(\boldsymbol{\beta}^{(b)}, \boldsymbol{\gamma}_q^{(b-1)}, \left\{\boldsymbol{\gamma}_{q'}^\bullet\right\}_{q' \neq q}; \mathbf{y}^{*(b)}\right)}{\widetilde{\mathcal{L}}_a\left(\boldsymbol{\beta}^{(b)}, \boldsymbol{\gamma}_q^{\text{prop}}, \left\{\boldsymbol{\gamma}_{q'}^\bullet\right\}_{q' \neq q}; \mathbf{y}^{*(b)}\right)}; \tag{21}$$

where $\boldsymbol{\gamma}_{q'}^\bullet = \boldsymbol{\gamma}_{q'}^{(b)}$ if $q' < q$ and $\boldsymbol{\gamma}_{q'}^\bullet = \boldsymbol{\gamma}_{q'}^{(b-1)}$ if $q' > q$.

- *Step 5.* As IAMS.

For the sake of comparison, in the next section we will also consider the algorithm obtained by simply adding the MH step to the original IAMS algorithm, i.e. approximating the NLG distribution with the original mixture g_ε instead of $\widetilde{g}_\varepsilon$. We label it as MH-IAMS algorithm and it follows the steps of the RIAMS algorithm but replacing $\widetilde{g}_\varepsilon$ and $\widetilde{\mathcal{L}}_a$ with g_ε and $\mathcal{L}_a$.

4.2 The Automatic algorithm

It is important to emphasize that the IAMS algorithm with the mixture approximation (6) is generally the best choice for sampling Poisson LGMs because of its computational efficiency, provided that the accuracy of the approximation does not compromise the convergence of the algorithm. Notably, the adjusted mixture $\widetilde{g}_\varepsilon$ is more computationally demanding than g_ε because of the increased number of components. Moreover, adding a rejection step increases computational cost, as it requires evaluating the acceptance probability, which in turn involves computing both the exact and approximated likelihoods.

Based on these considerations, our ultimate goal is to implement an automatic algorithm that applies RIAMS or MH-IAMS only when the approximation could cause IAMS to fail to converge. It is worth recalling that approximation issues arise when one or more residuals fall in the tails of the NLG distribution. When a residual lies in the left tail, the MH

step alone can address the issue; however, if it falls in the right tail, the mixture must also be adjusted. To determine whether a residual lies in the distribution tails, we use the previously introduced thresholds $\xi_U(y)$ for the upper tail and $\xi_L(y)$ for the lower tail. Both thresholds are defined as the points where the absolute value of the log difference between the NLG and mixture densities exceeds 1.

To put these considerations into practice, we introduce a preliminary step consisting of a training period to assess the need to strengthen the IAMS algorithm. The preliminary step of the Automatic algorithm is structured as follows:

- *Step 0a.* Perform T_1 iterations with the IAMS algorithm.
- *Step 0b.* Perform T_2 iterations and compute, as in the RIAMS algorithm, the proportions defined in (18). In addition, we compute, for each residual, the proportion of iterations exceeding the lower bound:

$$\kappa_{ij}^L = \frac{1}{T_2} \sum_{b=T_1+1}^{T_1+T_2} \mathbf{1}\left(\varepsilon_{ij}^{(b)} < \xi_L(y_i)\right), \qquad \forall (i, j). \tag{22}$$

  This corresponds to checking whether any residuals fall into a region where the mixture approximation (6) is inaccurate. At the end of the T_2 iterations, the effectively used mixtures are selected as in (19) and the algorithm for subsequent iterations is chosen based on the following rule:

  - IAMS if $\kappa_{ij}^L < p_L$ and $\kappa_{ij}^U < p_U$, $\forall (i, j)$;
  - MH-IAMS if $\exists (i, j)$ s.t. $\kappa_{ij}^L > p_L$, and $\kappa_{ij}^U < p_U$, $\forall (i, j)$;
  - RIAMS if $\exists (i, j)$ s.t. $\kappa_{ij}^U > p_U$. Note that, in this case, the adjusted mixture $\widetilde{g}_\varepsilon$ is adopted only on the residuals that are identified as problematic.

  The choice depends on two thresholds p_L and p_U that we recommend setting to small values. In the `SamplerPoisson` package, the default value is $p_L = p_U = 0.05$.

5 Applications

In this section, we present three applications to demonstrate the usefulness of the algorithms proposed in Section 4. The first application reprises the toy example introduced in Section 3, demonstrating the failure of the IAMS algorithm under increasing levels of model misspecification. This example involves a basic fixed-effect Poisson regression model. The other two applications are based on real datasets: one analyzes squirrel behavior and forest attributes across multiple plots in Scotland's Abernethy Forest (using a model with splines), while the other examines a monthly time series of death counts in the Italian province of Piacenza from 2013 to 2022. In both cases, Poisson mixed models formulated as LGMs are employed. The R code for estimation and result summaries is provided as supplementary material.

5.1 Toy example: reprise

In this section, we show results obtained by using our proposed algorithm in an extension of the toy example introduced in Section 3. In particular, four datasets are generated from a Poisson distribution with parameter

$$\log(\lambda_i) = 0.1 + x_{1i} + c x_{2i}, \qquad i = 1, \ldots, 30,$$

where both covariates are drawn from a standard Gaussian distribution and $c \in \{0, 0.4, 0.8, 1.2\}$. Subsequently, models are estimated by omitting the covariate x_2 to simulate increasing levels of model misspecification. We then compare the performance of the IAMS algorithm with that of the algorithms described in Section 4. Specifically, the comparison between the RIAMS and MH-IAMS algorithms highlights cases where the adjusted mixture approximation enhances convergence. Finally, the Automatic algorithm is evaluated to assess its ability to adaptively switch to a more robust method when necessary. For each algorithm we run a chain of $B = 110,000$ iterations, discarding the first 10,000 iterations as burn-in. For the Automatic algorithm, we set $T_1 = 500$, $T_2 = 250$ and $p_L = p_U = 0.05$.

Figure 5 presents the posterior distribution of β_0 and β_1 for the different values of the omitted variable coefficient c, across the MCMC algorithms under study. As shown in the plots, all algorithms converge to the target distributions when $c = 0$ and $c = 0.4$. However, for $c = 0.8$ and $c = 1.2$, the posterior distributions obtained using the IAMS algorithm deviate from the Stan benchmark for both the intercept β_0 and the regression coefficient β_1. Furthermore, the MH-IAMS algorithm also exhibits discrepancies from the target posterior when $c = 1.2$, suggesting that the mixture adjustment in the right tail, introduced in Section 4.1, can improve convergence. In contrast, the RIAMS sampler remains in agreement with the target posterior across all values of c, providing evidence in favor of its robustness in the presence of extreme residuals generated by the omission of a relevant covariate.

These results are closely linked to the acceptance rates observed for the compared algorithms, which are reported in Table 1 as a function of c. First, we note that the acceptance rate of the RIAMS algorithm is higher than that of the MH-IAMS algorithm for $c = 0.8$ (0.87 vs 0.72). Furthermore, when $c = 1.2$, the acceptance rate of MH-IAMS drops significantly to 0.01, leading to poor mixing, whereas the RIAMS algorithm continues to perform satisfactorily. Regarding the Automatic sampler, it can be seen that the IAMS algorithm is selected when $c = 0$ and $c = 0.4$, which is appropriate, as the IAMS algorithm converges in these cases, and meth-

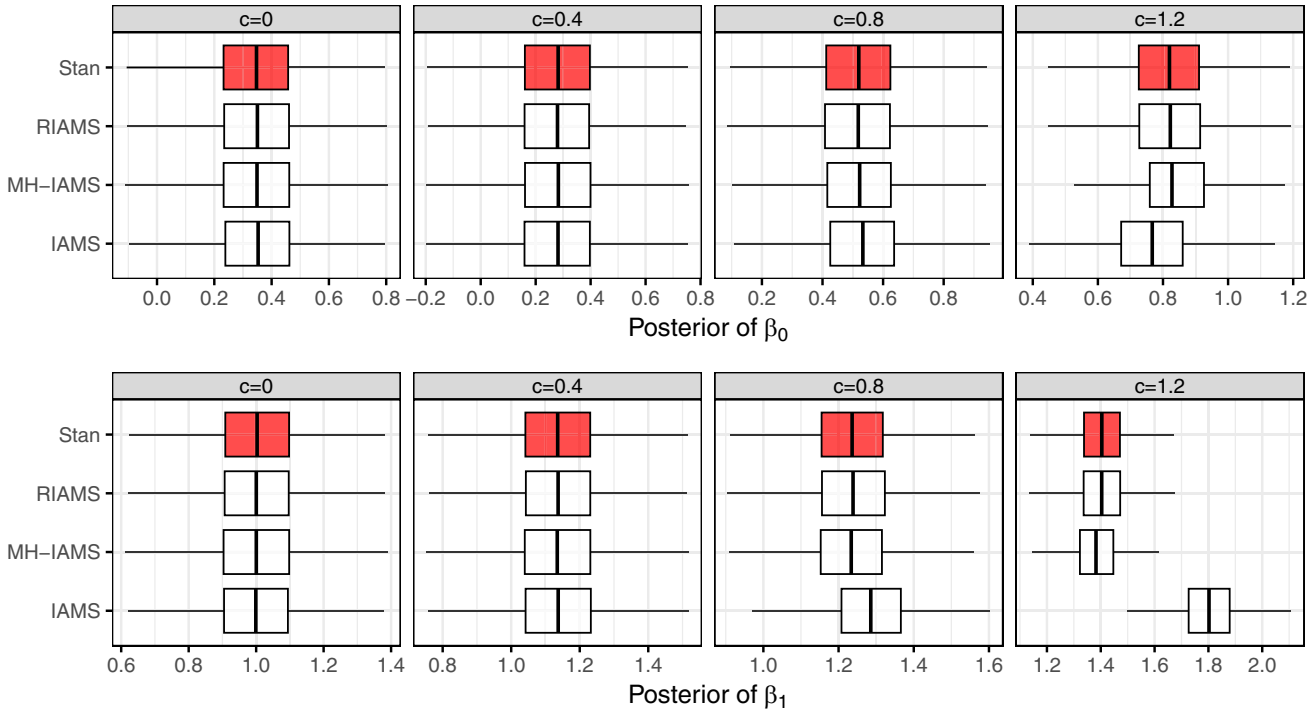

Fig. 5 Posterior distributions of model parameters. Each panel shows the posterior distribution of one parameter for a given c value (reported in the title) obtained with the considered algorithms

Table 1 Acceptance rates for each algorithm as a function of the underlying coefficient b

Algorithm	$c = 0$	$c = 0.4$	$c = 0.8$	$c = 1.2$
MH-IAMS	1.00	1.00	0.72	0.02
RIAMS	1.00	1.00	0.87	0.74
Automatic	IAMS	IAMS	0.87	0.75

Table 2 Relative computational times with respect to IAMS with 50 replications, 10,000 MCMC iterations each. Total elapsed times for Gibbs samplers ranges from 31s to 36s

Algorithm	$c = 0$	$c = 0.4$	$c = 0.8$	$c = 1.2$
MH-IAMS	2.43	2.40	2.46	2.01
RIAMS	2.49	2.38	2.62	2.38
Automatic	1.00	1.00	2.65	2.01

ods involving an MH step achieve an acceptance rate of 1. Conversely, for $c = 0.8$ and $c = 1.2$, the Automatic sampler correctly switches to the RIAMS algorithm, as the IAMS algorithm fails to converge due to large residuals.

The ability of the Automatic sampler to select the RIAMS algorithm only when necessary offers a clear advantage when considering computational time. Table 2 reports the ratio of the computational time of each algorithm to that of the IAMS algorithm. The results show that algorithms incorporating the MH step require more than twice the computational time of the IAMS algorithm. The key advantage of the Automatic sampler is that it only incurs the additional computational cost when the IAMS algorithm is at risk of failing to converge due to inaccuracies in the mixture approximation.

We emphasize that the primary goal of this example is to illustrate how the IAMS algorithm performs as the Poisson model becomes increasingly challenging to fit due to the presence of extreme residuals. Nonetheless, we take this opportunity to highlight that, when the target is a fixed-effects Poisson regression model, both `Stan` and the sampling algorithm proposed by D'Angelo and Canale (2023), implemented in the `bpr` package, offer appealing alternatives. `Stan` is attractive for its ease of use, as Poisson models can be readily implemented through user-friendly interfaces such as `rstanarm`. In contrast, the algorithm by D'Angelo and Canale (2023) offers notable computational efficiency, performing approximately 2.6 times faster than the IAMS algorithm with respect to effective sample size per second. For this reason, it represents a promising candidate for future extension to LGMs.

5.2 Nuts data

In this section, we deal with an application of Poisson regression to the `nuts` dataset available in the R-package `COUNT` discussed in Zuur et al. (2013). The dataset comprises obser-

vations on $n = 52$ plots providing information about squirrel behavior and attributes related to to Scotland's Abernethy Forest. This application is not meant to provide a sound statistical analysis of these data but to illustrate the merit of the algorithms proposed in Section 4. The response variable is the number of cones stripped by squirrels. The linear predictor is specified as the sum of two linear effects, i.e., the standardized mean tree height ($\mathbf{x}_{\text{height}}$) and standardized canopy closure per plot ($\mathbf{x}_{\text{canopy}}$), plus a smooth effect of the standardized number of trees per plot ($\mathbf{x}_{\text{trees}}$).

This specification delivers an LGM where the likelihood is

$$y_i|\lambda_i \sim \mathcal{P}(\lambda_i), \qquad i = 1, \ldots, 52,$$

and the vector of linear predictors is specified as

$$\log(\boldsymbol{\lambda}) = \beta_0 + \beta_1 \mathbf{x}_{\text{height}} + \beta_2 \mathbf{x}_{\text{canopy}} + f(\mathbf{x}_{\text{trees}}).$$

The term $f(\mathbf{x}_{\text{trees}})$ indicates the assumption of a smooth effect of $\mathbf{x}_{\text{trees}}$, which we implement using the popular Bayesian P-splines (Lang and Brezger 2004) as:

$$f(\mathbf{x}_{\text{trees}}) = \mathbf{Z}\boldsymbol{\delta}, \quad \boldsymbol{\delta}|\omega^2 \sim \mathcal{N}_m\left(\mathbf{0}, \omega^2 \mathbf{K}^-\right),$$

where $\mathbf{Z} \in \mathbb{R}^{n \times m}$ is constructed from cubic B-spline basis functions evaluated at m knots, while $\boldsymbol{\delta}$ is the vector of coefficients for which we assume a second-order random walk (RW2) prior, determined by the rank-deficient precision matrix $\mathbf{K}$. To allow model identifiability, model components with an RW2 prior are subject to the linear constraint $\mathbf{A}\boldsymbol{\delta} = \mathbf{0}$, where $\mathbf{A}$ is a $2 \times m$-dimensional matrix with the first row being a vector of ones and the second row being the sequence of the first m integers: this corresponds to implementing the linear constraints $\sum_{k=1}^{m} \delta_k = 0$ and $\sum_{k=1}^{m} k\delta_k = 0$. Accordingly, the linear term should be reintroduced into the linear predictor. While this setting is standard within samplers tailored for LGMs, it is not straightforward to implement when estimating the model with `Stan`. To estimate the model with `Stan` and verify that the RIAMS algorithm is able to target the correct posterior, we adopt the generic mixed model parameterization described in Scheipl et al. (2012). The nonlinear term can be re-expressed as

$$f(\mathbf{x}_{\text{trees}}) = \beta_3 \mathbf{x}_{\text{trees}} + \mathbf{Z}_{\text{o}}\boldsymbol{\gamma}, \quad \boldsymbol{\gamma}|\sigma^2 \sim \mathcal{N}_{m'}\left(\mathbf{0}, \sigma^2 \mathbf{I}_{m'}\right);$$

where $\mathbf{Z}_{\text{o}} \in \mathbb{R}^{n \times m'}$ is a matrix of $m' < m$ orthonormal basis vectors, obtained via the spectral decomposition of the covariance matrix of $\mathbf{Z}\boldsymbol{\delta}$, i.e., $\mathbf{Z}\mathbf{K}^-\mathbf{Z}^\top$, which has rank m'. The main drawback of this approach is that $\mathbf{Z}_{\text{o}}$ becomes a dense matrix, even when $\mathbf{Z}$ and $\mathbf{K}$ are sparse. This loss of sparsity can lead to significant inefficiencies, especially with large datasets, thereby underscoring the limitations of `Stan` for models that involve sparse matrices. The model hierarchy is completed by specifying priors for fixed-effect coefficients $\beta_k \sim \mathcal{N}(0, 1000)$, $k = 0, 1, 2, 3$ and for the scale parameter σ^2 for which we chose $\sigma^2 \sim \text{Gamma}(1, 0.001)$.

In what follows, we compare results obtained by implementing the MCMC sampler in Stan, which again serves as a benchmark, with results obtained by the IAMS, MH-IAMS, and Automatic samplers. In this application, the Automatic sampler selects the RIAMS algorithm. Preliminary checks indicated that, among the 99 latent variables introduced by the first augmentation scheme, κ_{ij}^L and κ_{ij}^U defined in Equations (18) and (22) exceeded the threshold 0.05 in 9 and 3 cases, respectively. Figure 6 shows the posterior distributions of each model parameter for the compared algorithms. It can be noticed that posterior distributions obtained with the IAMS algorithm fail to overlap with Stan posteriors across all model parameters, highlighting a lack of convergence. By contrast, the Automatic (RIAMS) and MH-IAMS algorithms show good agreement with Stan posteriors. These results suggest that a rejection step is necessary to achieve convergence in this application.

Table 3 compares the performance of the MH-IAMS and Automatic algorithms, reporting the effective sample size and the effective sample size per second for each, based on $B = 100,000$ MCMC iterations after a burn-in of 10,000 iterations. The results show that the RIAMS algorithm, correctly chosen by the Automatic procedure, achieves significantly higher effective sample sizes, both in absolute terms and relative to elapsed time. This improvement is closely linked to the adjustment of the mixture in the right tail for problematic residuals, which leads to a significant increase in acceptance rates under the RIAMS algorithm. As shown in Table 4, the acceptance rate for $\boldsymbol{\beta}$ increases substantially (from 0.23 to 0.62), while $\boldsymbol{\gamma}$ sees a moderate rise (from 0.58 to 0.76). This directly translates into a reduction in chain autocorrelation. Finally, regarding total computational time, the IAMS algorithm is significantly faster (13 seconds for 110,000 iterations), whereas the additional rejection step in the MH-IAMS and Automatic samplers increases computational time to 47 and 56 seconds, respectively.

5.3 Time series of counts

In this section, we analyze monthly all-cause mortality among individuals aged 85 and over in the Italian province of Piacenza, covering the period from 2013 to 2022. The model accounts for both year-specific and seasonal (monthly) variations through the inclusion of structured random effects. Observed counts y_{mt} are modeled as:

$$y_{mt}|\theta_{mt} \sim \mathcal{P}\left(\theta_{mt} E_{mt}\right), \quad m = 1, \ldots, 12; \quad t = 1, \ldots, 10$$

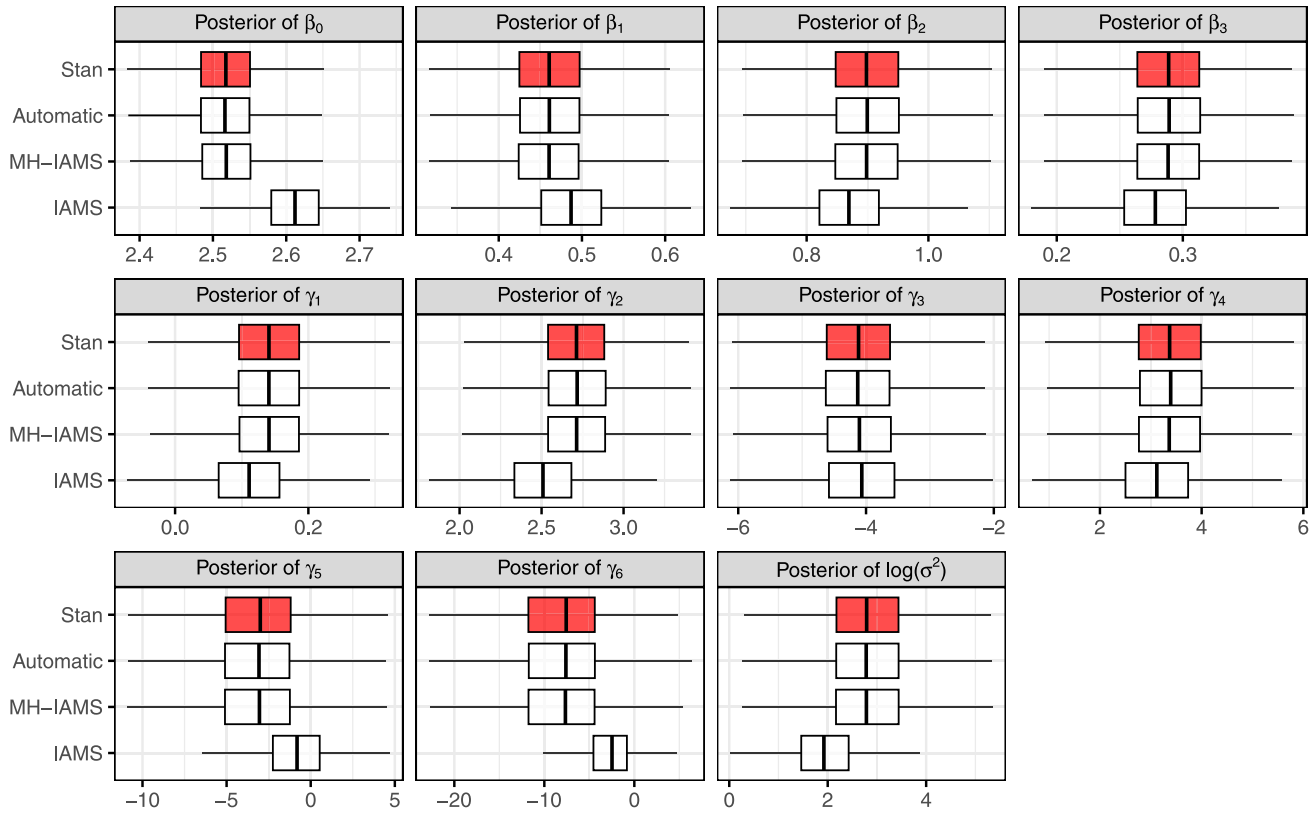

Fig. 6 Posterior distributions of model parameters for the example discussed in Section 5.2. Each panel shows the posterior distributions of one parameter obtained under the considered algorithms

Table 3 Effective sample size (n_{eff}) and effective sample size per second (n_{eff}/sec) for the MH-IAMS and RIAMS algorithms, $B = 100,000$ MCMC iterations

Parameter	MH-IAMS		RIAMS	
	n_{eff}	n_{eff}/sec	n_{eff}	n_{eff}/sec
β_0	1429	30	4283	77
β_1	2365	49	7208	129
β_2	1731	36	4474	80
β_3	2943	61	8839	159
γ_1	6529	136	8789	158
γ_2	7580	158	13604	244
γ_3	4613	96	6793	122
γ_4	8899	186	9197	165
γ_5	6339	132	9122	164
γ_6	3798	79	5464	98
σ^2	6964	145	10327	185

$$\log(\theta_{mt}) = \beta_0 + \phi_m + \psi_t$$

where E_{mt} refers to expected counts and θ_{mt} represents the relative risk at year t and month m. Expected counts are computed via internal standardization under the assumption that age-specific mortality rate is constant over the whole study period, entering the model as an offset. The log-linear predictor is expressed as the sum of an intercept term β_0, a monthly random effect ϕ_m, and a yearly random effect ψ_t. A random walk prior of order 1 (RW1) is specified for the year random effects vector $\boldsymbol{\psi} = (\psi_1, \ldots, \psi_{10})^\top$, while a circular RW1 model is specified for the month random effects vector $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_{12})^\top$, to appropriately account for seasonal periodicity. The precision matrices associated with the year and month effects are denoted by $\mathbf{K}_\psi$ and $\mathbf{K}_\phi$, respectively. For details on the structure of these matrices, see Rue and Held (2005). In summary, the random effects prior distributions are:

Table 4 Acceptance rates and elapsed time of the algorithms compared in Section 5.2

	IAMS	MH-IAMS	RIAMS
Acceptance rate for $\boldsymbol{\beta}$	-	0.23	0.62
Acceptance rate for $\boldsymbol{\gamma}$	-	0.58	0.76
Elapsed time (for $B = 100,000$)	13s	47s	56s

$$\boldsymbol{\phi}|\sigma_\phi^2 \sim \mathcal{N}_{12}\left(\mathbf{0}, \sigma_\phi^2 \mathbf{K}_\phi^-\right) \quad \text{and} \quad \boldsymbol{\psi}|\sigma_\psi^2 \sim \mathcal{N}_{10}\left(\mathbf{0}, \sigma_\psi^2 \mathbf{K}_\psi^-\right).$$

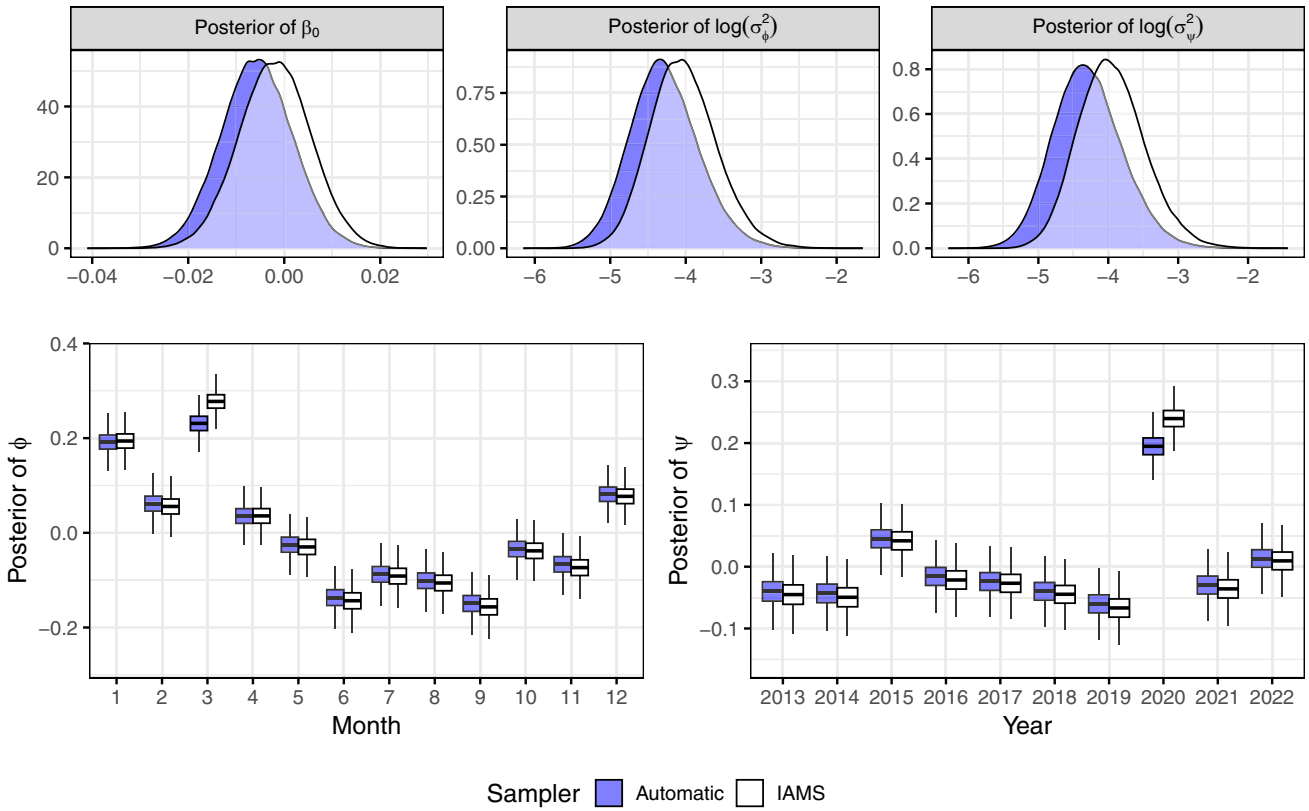

Fig. 7 Posterior distributions of model parameters for the example discussed in Section 5.3. The plots in the top row shows the posterior of the basic parameters, whereas the bottom plots shows the seasonal and the temporal coefficients

Model specification is completed by assigning a Gamma $(1, 0.001)$ prior to both scale parameters σ^2_ψ and σ^2_ϕ. To ensure model identifiability, we impose a sum-to-zero constraint on both random effect vectors.

Model estimation is carried out using the IAMS, MH-IAMS, and Automatic algorithms. During the training phase, the Automatic sampler selects the RIAMS algorithm. Figure 7 (top row) compares the posterior densities of the intercept term β_0 and the scale parameters σ^2_ψ and σ^2_ϕ between the IAMS and RIAMS algorithms, highlighting a miscalibration in the posterior distributions produced by the IAMS algorithm. The source of this miscalibration becomes evident in the bottom row of Figure 7, where posterior distributions of the random effects are reported. The largest discrepancies between the IAMS and RIAMS algorithms are observed in March for the monthly effects $\boldsymbol{\phi}$, and in 2020 for the yearly effects $\boldsymbol{\psi}$. These discrepancies are primarily driven by the impact of the COVID-19 pandemic, which caused a sharp deviation in the temporal pattern of mortality risk in the Piacenza province. This deviation results in extreme residuals that fall into regions where the original mixture approximation employed by IAMS lacks sufficient accuracy. As shown in Table 5, the acceptance rates of the MH-IAMS sampler are below 10% for both random effects vectors, and are substantially lower than the rates achieved by the RIAMS algorithm.

Table 5 Acceptance rates and elapsed time of the algorithms compared in Section 5.3

	IAMS	MH-IAMS	RIAMS
Acceptance rate for β_0	-	0.49	0.82
Acceptance rate for $\boldsymbol{\phi}$	-	0.09	0.76
Acceptance rate for $\boldsymbol{\psi}$	-	0.09	0.81
Elapsed time (for $B = 100,000$)	41s	185s	187s

6 Concluding remarks

The IAMS algorithm is a particularly appealing and computationally efficient method for sampling from the posterior distributions of Poisson LGMs. It enables the use of standard sampling algorithms originally designed for LGMs with Gaussian likelihoods, once auxiliary variables have been generated. However, this popular auxiliary mixture sampling approach relies on an approximation step, whose impact has yet to be fully explored in the literature.

The IAMS algorithm usually provides an accurate approximation of the posterior distribution in Poisson regression models. However, in certain applications where large residuals are observed relative to the augmented model, the algorithm may fail to converge to the target posterior distribution. This is primarily due to the rapid decay of the right tail in the adopted mixture approximation, as discussed in Section 3. It is important to note that the failure of the approximation can be particularly insidious, as there are no explicit warnings to alert users. A useful first step is to compare the true log-likelihood with the approximated one. This check generally causes limited computational burden, and any observed discrepancy may indicate potential issues with the approximation.

In this paper, we proposed a robust version of the IAMS algorithm, involving an acceptance step during the sampling process and an adjusted mixture approximation that provides a more accurate approximation of the right tail of the NLG distribution, favoring convergence to the posterior distribution even in the presence of large residuals. Since both the acceptance step and the employment of the adjusted mixture require additional computational effort with respect to the IAMS algorithm, a sampler that judges the need for such an algorithm during a training period has been designed and implemented in the R package `SamplerPoisson`.

Further research may help to develop more efficient algorithms for Poisson models with random effects, where computational cost can be a limiting factor. Some preliminary runs also suggest that, for fixed-effects Poisson regression models, the algorithm proposed by D'Angelo and Canale (2023) offers a computationally efficient alternative to IAMS. Its potential extension to latent Gaussian models appears promising but requires additional investigation.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11222-025-10781-w.

Author Contributions All the authors conceived of the presented idea. A.G. and F.G. developed the theory and performed the computations. All authors discussed the results. A.G. and F.G. wrote the main manuscript. All authors reviewed the manuscript.

Funding Open access funding provided by Alma Mater Studiorum - Università di Bologna within the CRUI-CARE Agreement.

Data Availability Data and code are provided as supplementary information files.

Declarations

Competing interests The authors declare no competing interests.

References

Albert, J.H., Chib, S.: Bayesian analysis of binary and polychotomous response data. J. Am. Stat. Assoc. **88**(422), 669–679 (1993)

D'Angelo, L., Canale, A.: Efficient posterior sampling for Bayesian Poisson regression. J. Comput. Graph. Stat. **32**(3), 917–926 (2023)

Dvorzak, M., Wagner, H.: Sparse Bayesian modelling of underreported count data. Stat. Model. **16**(1), 24–46 (2016)

Frühwirth-Schnatter, S., Frühwirth, R.: Auxiliary mixture sampling with applications to logistic models. Computational Statistics & Data Analysis **51**(7), 3509–3528 (2007)

Frühwirth-Schnatter, S., Frühwirth, R., Held, L., Rue, H.: Improved auxiliary mixture sampling for hierarchical models of non-Gaussian data. Stat. Comput. **19**, 479–492 (2009)

Frühwirth-Schnatter, S., Wagner, H.: Auxiliary mixture sampling for parameter-driven models of time series of counts with applications to state space modelling. Biometrika **93**(4), 827–841 (2006)

Knorr-Held, L., Rue, H.: On block updating in Markov random field models for disease mapping. Scand. J. Stat. **29**(4), 597–614 (2002)

Lang, S., Brezger, A.: Bayesian P-splines. J. Comput. Graph. Stat. **13**(1), 183–212 (2004)

Martin, A.D., Quinn, K.M., Park, J.H.: MCMCpack: Markov Chain Monte Carlo in R. J. Stat. Softw. **42**(9), 1–21 (2011)

Mengersen, K.L., Tweedie, R.L.: Rates of convergence of the Hastings and Metropolis algorithms. Ann. Stat. **24**(1), 101–121 (1996)

Polson, N.G., Scott, J.G., Windle, J.: Bayesian inference for logistic models using Pólya-Gamma latent variables. J. Am. Stat. Assoc. **108**(504), 1339–1349 (2013)

Robert, C.P., Casella, G.: Monte Carlo Statistical Methods, vol. 2. Springer, New York (1999)

Rue, H., Held, L.: Gaussian Markov Random Fields: Theory and Applications. Monographs on Statistics and Applied Probability, vol. 104. Chapman and Hall/CRC, New York (2005)

Scheipl, F., Fahrmeir, L., Kneib, T.: Spike-and-slab priors for function selection in structured additive regression models. J. Am. Stat. Assoc. **107**(500), 1518–1532 (2012)

Stan Development Team: RStan: the R interface to Stan. R package version 2.32.6 (2024). https://mc-stan.org/

Zuur, A.F., Hilbe, J., Ieno, E.N.: A Beginner's Guide to GLM and GLMM with R: a Frequentist and Bayesian Perspective for Ecologists. Highland Statistics, Newburgh (2013)