Shrinkage GLM Modelling

Vali Asimit[1], Ziwei Chen[12], Yuantao Xie[3], Yue Zhang[13]

Abstract

Generalised Linear Models are widely used for analysing multivariate data with non-normal responses for which the Iteratively Reweighted Least Squares algorithm is the main device to solve small and medium scaled problems. The popularity of this algorithm is due to the fact that is reduced to solving a series of weighted least square instances that are computationally less expensive than the general purpose optimisation algorithms that could solve the underlying maximum likelihood problem. Deploying the Iteratively Reweighted Least Squares algorithm may be affected by convergence issues, sensitivity to starting values, and more importantly, by significant parameter estimation error. A recent paper has shown how effective shrinkage estimation could be to improve the estimation error of the ordinary least square estimator without increasing the computational cost. The efficiency of these novel shrinkage estimators is explained by their design to reduce the theoretical Mean Square Error that is achieved by introducing a small bias with a sizable reduction in the new estimators' variance. We show in this paper that the Iteratively Reweighted Least Squares algorithm could significantly benefit from replacing in each iteration the least square estimators with our shrinkage estimators. In addition, we introduce an optimisation method to obtain more reliable starting values, further enhancing convergence. Simulation studies and real-data applications demonstrate that our proposed methods improve convergence speed, stability, and overall performance compared to standard Generalised Linear Model non-penalised implementations.

Keywords: Generalised Linear Model, Shrinkage Estimation, Iteratively Reweighted Least Square.
JEL classification: C10; C87; C63.

1. Introduction

1.1. Motivation

Generalised linear models (GLMs) extend traditional linear regression methods to multivariate data that have non-Gaussian distributed dependent variables. The most common way to fit GLMs is the *Iteratively Reweighted Least Squares (IRLS)* algorithm, which iteratively updates

[1]Bayes Business School, City St George's, University of London, UK. Email addresses: asimit@asimit@citystgeorges.ac.uk (Vali Asimit), Ziwei.Chen.3@citystgeorges.ac.uk (Ziwei Chen), Yue.Zhang.10@city.ac.uk (Yue Zhang).
[2]Corresponding author.
[3]University of International Business and Economics, Beijing, China. Email addresses: xieyuantao@uibe.edu.cn (Yuantao Xie), 18811308639@163.com (Yue Zhang)

the weights and solves a *weighted least squares (WLS)* problem at each step until convergence. However, IRLS has various limitations. *First*, it can fail to converge in some cases (Marschner, 2011). *Second*, WLS estimation error is affected by the instability of covariance matrix inversion which is due to i) poor empirical eigenvalues estimation (Ledoit and Wolf, 2004; Asimit et al., 2025b) and ii) non-zero asymptotic *Mean Square Error (MSE)* of the *Ordinary Least Square (OLS)* estimator in the Kolmogorov setting where both the sample size and number of covariates get large (El Karoui et al., 2013; El Karoui, 2013; Donoho and Montanari, 2016; Asimit et al., 2025b). *Third*, IRLS is sensitive to starting values, and poor initial guesses may require many iterations, suboptimal solutions, or complete convergence failure (Green, 1984; Marschner, 2011).

The motivation of this paper is to address two issues countered in GLM estimation that are explained above. *Firstly,* the OLS/WLS estimator can be improved by using the shrinkage estimators coined in Asimit et al. (2025b), and therefore, it is expected that enhancing the estimation error in each iteration of IRLS would improve the GLM estimation error; in fact, this has been observed in a very small real data analysis in Asimit et al. (2025b) where those shrinkage estimators are shown to outperform OLS. *Secondly,* we propose an optimisation method to find better starting values for IRLS deployments, which reduce IRLS's sensitivity to initial estimates and improve its overall performance.

1.2. Literature Review

GLMs provide a way to analyse data when the response variable does not follow a normal distribution. They are often used in medicine (Boyle et al., 1997; Field and Wilcox, 2017; Kapre et al., 2020), biostatistics (Xia et al., 2013; Sohn and Li, 2018), actuarial science (Debón et al., 2008; Peters et al., 2009; Mouatassim and Ezzahid, 2012; Delong et al., 2021) and so on. The standard GLM assumes that the response variable follows a distribution from the exponential family for which its mean value is linked to a linear predictor (a linear combination of the independent variables) through a functional known as *link function (LF)*; the presence of the linear predictor explains the GLM terminology. A common approach for estimating the model parameters is *Maximum Likelihood Estimation (MLE)*, typically computed by Newton's Method or Fisher Scoring; both are carried out via the *Iteratively Reweighted Least Squares (IRLS)* algorithm (Nelder and Wedderburn, 1972). However, IRLS requires a matrix inversion at each step, which can be computationally unstable and/or expensive, especially for large or high-dimensional data; note that Quasi-Newton methods such as *Broyden-Fletcher-Goldfarb-Shanno (BFGS)*, *Davidon-Fletcher-Powell (DFP)*, and *Limited-memory BFGS (L-BFGS)* approximate the Hessian to reduce the computational cost. Other optimisation approaches have been proposed, including *Interior-Point Methods* for large-scale L_1-regularised logistic regression (Koh et al., 2007), *Smooth-Threshold Generalised Estimating Equations (SGEE)* for correlated longitudinal data (Li et al., 2013), and *self-concordant* optimisation techniques for some power GLMs equipped with power LFs (Asimit et al., 2025a). Marra and Radice (2017) extended GLMs by proposing bivariate copula additive models for jointly modelling multiple continuous responses, estimating parameters via penalised likelihood using a trust-region algorithm implemented in the

R package `GJRM`[1]. IRLS is also used with great success in various other problems, such as *Non-linear Regression* (Wood, 2017), *Heteroscedastic Linear Models* (Hooper, 1993), *Generalised Symmetric Linear Models* (Villegas et al., 2013), *Vector Generalised Linear/Additive Models* (Yee and Stephenson, 2007), and *Conway-Maxwell-Poisson regression* (Chatla and Shmueli, 2018).

Beyond standard GLMs, IRLS has also been used for estimation in cases where key model assumptions are relaxed. For example, some contingency tables do not conform to the usual GLM structure, and a composite LF is used to relate each observation to multiple linear predictors (Thompson and Baker, 1981). When there is limited prior support for a specific LF, parametric (Scallan et al., 1984) or P-spline methods (Muggeo and Ferrara, 2008) can be used to estimate the link flexibly, relaxing the assumption of a known LF. IRLS has been applied in settings where the response is not scalar, such as symbolic polygonal data (do Nascimento et al., 2024). Additionally, IRLS is used for estimation with missing data, such as in the E- and M-steps of the EM algorithm for GLMs, where parameters are iteratively updated using standard IRLS-based routines (Ibrahim et al., 1999).

IRLS has also been employed in various areas, including deviance-based model selection (Sakate and Kashid, 2014), subsampling for Bayesian methods (Lachmann et al., 2022), and variable selection using iterative reweighting and thresholding (Fan and Li, 2001). However, our main focus is on improving the IRLS estimation procedure itself. We propose a modified IRLS algorithm that replaces the usual WLS step with some of the shrinkage estimators discussed in Asimit et al. (2025b), namely *(simple) Slab Regression (SR)*, *Generalised Slab Regression (GSR)*, *Stein Estimator (St)*, and *Diagonal Shrinkage (DSh)*; such modification enhances the stability and efficiency of GLM estimation. These shrinkage estimators introduce a small bias with the advantage of significantly reducing the variance of the resulting estimator, which in turn improves the overall performance measured via MSE.

The shrinkage estimators from this paper are inspired by Stein's paradox, which showed that the MSE of the MLE mean vector estimator could be reduced through shrinkage at least in the Gaussian case (Stein, 1956, 1960; James et al., 1961). This puzzling result is concluded in a Bayesian setting and the most known shrinkage estimator is the James-Stein estimator (James et al., 1961) and its adaptations to unknown or diagonal covariance matrices (Baranchik, 1970; Stein, 1981). Later advances introduced shrinkage estimators for high-dimensional data where the number of covariates exceeds the sample size (Chételat and Wells, 2012), but the shrinkage estimation that we have referred so far is only available under Gaussian assumptions. A somehow non-parametric approach has recently risen where linear shrinkage estimators are available under a specific distributional structure (Wang et al., 2014; Bodnar et al., 2019), but such developments are a great step ahead towards distribution-free estimators.

We should clarify that shrinkage may be understood in many ways. The Stein-type shrinkage estimators that we have discussed so far are designed to improve a high-dimensional estimator by combining the information across the constituents of the parameter vector that needs to be

[1] `GJRM` is available at https://cran.r-project.org/web/packages/GJRM/index.html.

estimated. Specifically, parametric and non-parametric assumptions are considered to end up with a linear shrinkage estimator $(1-\rho)\hat{\boldsymbol{\theta}}+\rho\boldsymbol{\theta}_{target}$ for $\boldsymbol{\theta}$, where $\hat{\boldsymbol{\theta}}$ is the most common estimator (e.g., MLE for mean vector shrinkage estimation) and $\boldsymbol{\theta}_{target}$ is a parsimonious estimator that could be also deterministic. Note that the same idea could be deployed for covariance matrix shrinkage estimation (Ledoit and Wolf, 2004; Schäfer and Strimmer, 2005; Ledoit and Wolf, 2012) and precision matrix shrinkage estimation (the inverse of the covariance matrix) (Bodnar et al., 2016). While shrinking the covariance matrix may improve the OLS estimation, shrinking the OLS estimator is a better choice to reduce the estimation error in multiple linear regression (Asimit et al., 2025b). Penalised estimators for multiple linear regression, GLM and many other machine learning models rely on penalisation to overcome overfitting and poor out-of-sample performance, and/or achieve parsimonious models; numerous examples are possible, and here are well-known penalised methods related to our research topic: *Tikhonov penalisation* (Tikhonov, 1963; Hoerl and Kennard, 1970), *Basic pursuit* (Chen and Donoho, 1994), *LASSO* (Tibshirani, 1996), *Elastic-Net* (Zou and Hastie, 2005), *Generalised LASSO* (She, 2009; Tibshirani and Taylor, 2011), etc. While most of them shrink the OLS estimator around zero, i.e., $\boldsymbol{\theta}_{target}=\mathbf{0}$, when these penalised methods are applied to least square estimation, penalised regression is sought to be shrinkage estimators, but one could see now the conceptual differences between the class of penalised regression estimators and Stein type shrinkage estimators. In summary, this paper incorporates fully non-parametric Stein-type shrinkage estimators into the IRLS algorithm to enhance the GLM estimation without increasing the computational time.

1.3. Our Contributions

We propose new methods for solving GLM that aim to reduce the estimation error and enhance convergence by introducing data-driven starting values. Our contribution is three-fold. *First*, we enhance the IRLS-based algorithm by replacing the standard WLS estimator with shrinkage estimators (SR, GSR, St, and DSh). Our simulation results show that St-based IRLS consistently lowers the estimation error as compared to the usual IRLS, while GSR and DSh can also offer improvements under various settings via synthetic and real data. *Second*, the proposed shrinkage solutions often converge in fewer iterations, making them more computationally efficient. For Poisson and Gamma GLMs with *log* and *sqrt* LFs, the modified IRLS algorithms converge faster than the standard IRLS. For *Logistic regression (LR)*, they perform at a similar speed while maintaining efficiency. *Third*, we introduce an optimisation-based approach for choosing starting values, improving convergence for both the standard IRLS and the shrinkage-based methods. Our simulations indicate that this approach leads to higher convergence rates, especially for GLMs with *sqrt* LF when deployed in **R**.

The paper is organised as follows. Section 2 reviews existing methods for GLM fitting, focusing on Newton's method, Fisher Scoring and IRLS. Section 3 introduces the shrinkage estimators considered in this paper. In Section 4, we discuss our approach for selecting starting values and illustrate it with a small simulation. Section 5 presents a larger simulation study comparing our shrinkage-based approaches with standard GLMs for LR, Poisson, and Gamma models, while Section 6 provides some real data analyses. Finally, Section 7 summarises our main findings.

2. Background

In this section, we present an overview of GLMs, their formulation, and the standard numerical methods used for parameter estimation. We begin by describing the exponential dispersion model, which underlies GLMs, and then introduce the most common estimation approach MLE. We then discuss three widely used solvers for GLMs in Sections 2.1–2.3, emphasising their main features and typical usage scenarios.

A univariate GLM setting assumes that the response variable Y, defined on $\mathcal{Y} \subseteq \Re$, is explained by covariates/features $\mathbf{X}$ defined on $\mathcal{X} \subseteq \Re^p$. Let $\{P_{\theta,\phi} : \theta \in \Theta \subseteq \Re, \phi \in \Phi \subseteq \Re\}$ be the parametric set of distributions for Y, which is assumed to be an *exponential dispersion model* in canonical form with *canonical* parameter θ if its probability density/mass function is

$$f_Y(y;\theta,\phi) = \exp\left\{\frac{\theta y - b(\theta)}{a(\phi)} + c(y,\phi)\right\}. \tag{2.1}$$

Here, $a(\cdot)$, $b(\cdot)$ and $c(\cdot,\cdot)$ are real-valued functions defined on Φ, Θ and $\mathcal{Y} \times \Phi$, respectively, and ϕ is the dispersion parameter. Under standard conditions, the mean and variance of Y are

$$\mathrm{E}\big[Y\big] = b'(\theta) \quad \text{and} \quad \mathrm{Var}\big[Y\big] = a(\phi)b''(\theta).$$

The estimation procedure assumes an independent sample $Y_1, \ldots, Y_n$ such that Y_i is distributed as in (2.1) with its own parameter θ_i and ϕ; the conditional mean is linked through a linear predictor $\eta_i = \mathbf{x}_i^T\boldsymbol{\beta}$ via a real-valued function h:

$$\mathrm{E}[Y_i|\mathbf{X}_i = \mathbf{x}_i] = h\big(\mathbf{x}_i^T\boldsymbol{\beta}\big) \quad \text{for any } i = 1, \ldots, n, \tag{2.2}$$

where $\mathbf{x}_i$ is a d-dimensional vector of realised features/covariates. The inverse function of h (provided it exists) is the LF and is denoted by $g = h^{-1}$. The ϕ could vary and common assumption is that $a(\phi_i) = a(\phi)/w_i$, where $w_i > 0$ are given weights (or $w_i = 1$ if weights are not provided).

The most common estimation method for GLMs is MLE, and the log-likelihood function for an independent sample of size n is

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} \frac{\theta_i y_i - b(\theta_i)}{a(\phi)} + c(y_i,\phi), \quad \text{where } \theta_i = \big(b'\big)^{-1} \circ h\left(\mathbf{x}_i^\top\boldsymbol{\beta}\right),$$

Here, the symbol $\circ$ denotes function composition. Maximising the above is equivalent to minimising the following objective function

$$\mathcal{C}(\boldsymbol{\beta}) = -\sum_{i=1}^{n} w_i\big(\theta_i y_i - b(\theta_i)\big). \tag{2.3}$$

Although GLMs are often described in terms of the exponential family and the LF g, the estimation procedure depends directly on the function h. A common choice is the *canonical* LF

defined by

$$h(\eta) = b'(\eta), \qquad \eta \in \Re. \tag{2.4}$$

The technical conditions for the existence and uniqueness of the MLE estimate are well-known – e.g., see (Wedderburn, 1976; Mäkeläinen et al., 1981) – and require a strictly concave log-likelihood function and some boundary conditions. These conditions are satisfied by the instance in (2.3) if functions a, b and h satisfy certain regularity conditions. These constraints formalise the concept of *proper GLM* coined in Asimit et al. (2025a). The MLE solutions could be on the boundary of the parameter space, which makes the estimation quite problematic, but we exclude such extreme cases from our analysis; this is observed in the LR when there exists a hyperplane that perfectly separates the two classes, which is also known as *complete separation*, case in which there is a continuum of points on the boundary where the absolute maximum is attained (Albert and Anderson, 1984).

Minimising (2.3) could be done by either using an off-the-shelf solver designed for global (or convex) optimisation problems if the functional $\mathcal{C}$ is not convex (or convex). Convex instances are available for some specific GLMs, and a subset of such convex sets consists of *self-concordant* instances that have an efficient implementation (Asimit et al., 2025a). The vast majority of GLMs are non-convex and Newton's Method is the standard solver, where the main goal is to solve the non-linear system $\nabla\mathcal{C}(\boldsymbol{\beta}) = \mathbf{0}$; a second-order Taylor expansion leads to an iterative procedure for finding an approximation to the MLE of GLM parameter $\boldsymbol{\beta}$

$$\widehat{\boldsymbol{\beta}}^{(t+1)} = \widehat{\boldsymbol{\beta}}^{(t)} - H_{\mathcal{C}}^{-1}\big(\widehat{\boldsymbol{\beta}}^{(t)}\big)\nabla\mathcal{C}\big(\widehat{\boldsymbol{\beta}}^{(t)}\big) \quad \text{for all integers } t \geq 0. \tag{2.5}$$

Computational challenges may arise due to many reasons, one of which is when the global minimum is not an interior point case in which Newton's Method fails to converge. Also, inverting the Hessian matrix can be difficult when the problem size is large, and an alternative is to use the Fisher information matrix (the expected value of the Hessian), which leads to the Fisher Scoring Method. Both methods (Newton and Fisher Scoring) use a second-order Taylor approximation and, in some cases, yield equivalent results, which is not guaranteed unless some conditions are satisfied; for details, see Section 2.1 and 2.2. An interesting fact is that for GLMs with non-canonical LFs, Newton's Method usually converges faster than the Fisher Scoring Method (Wood, 2011). An alternative solver is via the IRLS algorithm that approximates a stationary point of (2.3) through a for-loop procedure where a WLS instance (with given weights) is run in every loop which is computationally very efficient; for more details, see Nelder and Wedderburn (1972); McCullagh et al. (1989) or Section 2.3. This explains why IRLS is the standard GLM solver given its computational advantage. However, an IRLS variant that employs step-halving may further improve convergence, which is implemented in the **R** package `glm2`[2].

The next sections present some technical details about various GLM solvers that we have men-

[2]The `glm2` package is available at https://cran.r-project.org/web/packages/glm2/index.html.

tioned above to supplement the high-level information presented so far.

2.1. Newton's Method

We begin with Newton's Method, a standard iterative procedure for solving the non-linear system $\nabla\mathcal{C}(\boldsymbol{\beta}) = \mathbf{0}$. Starting from an initial guess $\boldsymbol{\beta}^{(0)}$, the parameter estimate is updated as in (2.5). This approach is straightforward when both the gradient and the Hessian of the cost function $\mathcal{C}(\boldsymbol{\beta})$ are available. However, if the number of parameters p is large or the Hessian matrix is ill-conditioned, the method may lose accuracy or fail to converge. Equations (2.6) and (2.7) show the gradient and Hessian computations, and the *canonical* LF in (2.4) simplifies them considerably.

The gradient of $\mathcal{C}(\boldsymbol{\beta})$ with respect to β_j is given by

$$\frac{\partial\mathcal{C}}{\partial\beta_j} = \sum_{i=1}^{n} w_i\big(b'(\theta_i) - y_i\big)\frac{d\theta_i}{d\beta_j} = \sum_{i=1}^{n} w_i\big(b'(\theta_i) - y_i\big)\frac{h'(\eta_i)}{b''(\theta_i)}\,x_{ij}, \tag{2.6}$$

where $\eta_i = \mathbf{x}_i^\top\boldsymbol{\beta}$ for all $1 \le i \le n$ and $\theta_i = (b')^{-1} \circ h(\eta_i)$. Similarly, the (j,k) entry of the Hessian matrix is

$$\big(H_{\mathcal{C}}(\boldsymbol{\beta})\big)_{jk} = \sum_{i=1}^{n} w_i\left\{\frac{\big(h'(\eta_i)\big)^2}{b''(\theta_i)} + \big(b'(\theta_i) - y_i\big)\left(\frac{h''(\eta_i)}{b''(\theta_i)} - \frac{\big(h'(\eta_i)\big)^2\, b'''(\theta_i)}{\big(b''(\theta_i)\big)^3}\right)\right\}x_{ij}x_{ik}. \tag{2.7}$$

When the *canonical* LF defined in (2.4) is chosen, the (2.6) and (2.7) then simplify to

$$\frac{\partial\mathcal{C}}{\partial\beta_j} = \sum_{i=1}^{n} w_i\big(b'(\theta_i) - y_i\big)x_{ij} \quad \text{and} \quad \big(H_{\mathcal{C}}(\boldsymbol{\beta})\big)_{jk} = \sum_{i=1}^{n} w_i\, b''(\theta_i)x_{ij}x_{ik}.$$

In matrix form of (2.5), one can write $H_{\mathcal{C}}(\boldsymbol{\beta}^{(t)}) = \mathbf{X}^\top\mathbf{W}^*\mathbf{X}$ and $\nabla\mathcal{C}(\boldsymbol{\beta}^{(t)}) = \mathbf{X}^\top\mathbf{z}^*$, where $\mathbf{X} \in \Re^{n\times(p+1)}$ includes a column of ones for the intercept, and the diagonal matrix $\mathbf{W}^*$ and vector $\mathbf{z}^*$ depend on $\eta_i = \mathbf{x}_i^\top\boldsymbol{\beta}^{(t)}$ and $\theta_i = (b')^{-1} \circ h(\mathbf{x}_i^\top\boldsymbol{\beta}^{(t)})$.

Newton's Method is attractive because it can converge quickly under suitable conditions, but careful attention must be paid to Hessian inversion and the choice of initial values. In the next section, we describe the Fisher Scoring Method, which replaces the observed Hessian with the Fisher information matrix in the update step.

2.2. Fisher Scoring Method

Fisher Scoring is another iterative procedure for estimating GLMs. Unlike Newton's Method, which uses the observed Hessian in (2.7), Fisher Scoring replaces the Hessian with its expected value, known as the Fisher information matrix.

When we modify (2.7) by replacing y_i with its expectation $b'(\theta_i)$, the Fisher information matrix for each (j,k) entry becomes

$$\big(H_{\mathcal{C}}(\boldsymbol{\beta})\big)_{jk} = \sum_{i=1}^{n} w_i\frac{\big(h'(\eta_i)\big)^2}{b''(\theta_i)}\,x_{ij}x_{ik}.$$

This adjustment simplifies computation and reduces processing time. If the LF is canonical, this matrix coincides with the Hessian in Newton's Method, causing the two methods to be identical. However, with non-canonical LFs, Newton's Method may converge more rapidly because it uses the observed Hessian (Wood, 2011).

In practice, Fisher Scoring can be simpler when a closed-form expression for the Fisher information matrix is readily available, but it may become less efficient in complex or high-dimensional settings. As a result, the choice between Fisher Scoring and Newton's Method often depends on the model structure, the LF, and computational considerations. We next introduce the IRLS algorithm, a procedure that frames the estimation process as a sequence of WLS problems.

2.3. IRLS Implementation

As anticipated, IRLS is an effective GLM solver, which was introduced in Nelder and Wedderburn (1972) and has been the standard technique for GLM implementations. A detailed explanation of IRLS is given in Wood (2017) and we only provide a brief overview.

At iteration $t \geq 1$, let the linear predictor be $\eta_i^{(t)} = \mathbf{x}_i^\top \boldsymbol{\beta}^{(t)}$ and the mean response $\mu_i^{(t)} = h(\eta_i^{(t)})$ for each i. The algorithm typically starts with $\mu_i^{(0)} = y_i$ and $\eta_i^{(0)} = h^{-1}(\mu_i^{(0)})$, although adjustments may be required during implementation; see Section 4 for further discussion. At each step, IRLS solves the WLS problem

$$\widehat{\boldsymbol{\beta}}^{(t+1)} := \arg\min_{\boldsymbol{\beta}} \big(\mathbf{z}^{(t)} - \mathbf{X}\boldsymbol{\beta}\big)^\top \mathbf{W}^{(t)} \big(\mathbf{z}^{(t)} - \mathbf{X}\boldsymbol{\beta}\big),$$

where the weight matrix $\mathbf{W}^{(t)}$ and the pseudo-response $\mathbf{z}^{(t)}$ are defined by

$$\mathbf{W}^{(t)} = \operatorname{diag}\left(\frac{\big(h'\big(\eta_i^{(t)}\big)\big)^2}{V\big(\mu_i^{(t)}\big)}\right), \quad \mathbf{z}_i^{(t)} = \eta_i^{(t)} + \frac{y_i - \mu_i^{(t)}}{h'\left(\eta_i^{(t)}\right)}, \tag{2.8}$$

with $V\big(\mu_i^{(t)}\big)$ being the fixed variance function associated with the chosen distribution. Details on how to derive (2.8) can be found in Appendix B.

IRLS is implemented in statistical software such as **R**'s `glm2`, **Matlab**'s `fitglm`, and **Python**'s `statsmodels.GLM`. Each package includes specific convergence criteria, stability features, or stopping rules; see Appendix B. Moreover, our study focuses on the case $n > p$. When $n \leq p$, the design matrix does not have full rank, and our OLS-based methods including IRLS, become problematic. In those cases, penalised approaches such as *LASSO* or *Elastic-Net* are more appropriate. Since our shrinkage estimators build on the OLS formulation, we require $n > p$.

Having now described the three main approaches to GLM estimation, we turn to our proposed shrinkage estimators, which incorporate these standard frameworks while aiming to reduce MSE and improve estimation stability without enforcing sparsity.

3. Overview of Shrinkage Linear Regression Estimators

This section introduces four shrinkage estimators that replace the WLS step in the IRLS algorithm. These estimators, GSR, St, and DSh, have been studied in Asimit et al. (2025b) and are motivated by Stein's paradox (Stein, 1956, 1960; James et al., 1961). Stein's work shows that an unbiased estimator for a mean vector can often be improved by shrinking it toward a simpler target. All four estimators reduce the MSE compared to the OLS estimator and they are conceptually different than biased estimators obtained via penalisation. While our shrinkage estimators have a final form similar to a class of penalised regression estimators, the main difference between these two choices is that our shrinkage estimators do not require any form of cross-validation – like all penalised regressions require – and those shrinkage parameters are estimated by minimising the theoretical MSE of the shrinkage parameter vector. This improves the estimation error of the regression parameters, and in turn, out-of-sample performance for the dependent variable is enhanced.

In penalised regression, model estimation is obtained by minimising

$$\widehat{\boldsymbol{\beta}} := \underset{\boldsymbol{\beta}\in\Re^{p+1}}{\arg\min}\, \frac{1}{2}||\mathbf{y}-\mathbf{X}\boldsymbol{\beta}||_2^2 + g(\boldsymbol{\beta}), \tag{3.1}$$

where a penalty function $g : \Re^p \to \Re_+$ – e.g., $\lambda\|\boldsymbol{\beta}\|_1$ in the LASSO or $\lambda\|\boldsymbol{\beta}\|_2^2$ in Ridge regression – encourages model's sparsity and/or reduces overfitting. Note that our shrinkage estimators are designed to improve the estimation error, and not to attain a sparse model.

The shrinkage estimators are grouped into two classes. The *first* class, known as slab regressions, are discussed in Section 3.1 where shrinkage is achieved by adding a quadratic term to the loss function. This is the only shrinkage regression estimator that resembles to penalised regression estimators, but SR penalisation parameters are estimated rather than using cross-validation. Interestingly, the SR estimator is a special case of the *Generalised LASSO* estimator introduced in Tibshirani and Taylor (2011) which relies on cross-validation. The GSR estimator is seemingly similar to SR, but GSR significantly adjusts the eigenvalues of the covariance matrix which is done in a controlled manner (by minimising the MSE of the shrinkage estimator). Moreover, SR changes the eigenvalues and eigenvectors of the covariates covariance matrix, while GSR preserves the original eigenvectors and adjusts the eigenvalues like ridge regression (Hoerl and Kennard, 1970) that again relies on cross-validation. This could explain why our numerical experiments on synthetic and real data tend to recommend GSR more often than SR.

The *second* class is presented in Section 3.2, and is defined through multiplicative shrinkage. That is, the OLS estimator is scaled by a data-derived diagonal matrix, $\widehat{\boldsymbol{\beta}}(\mathbf{D}) = \mathbf{D}\,\widehat{\boldsymbol{\beta}}^{OLS}$ with $\mathbf{D} \in \Re^{(p+1)\times(p+1)}$. Note that Hocking et al. (1976) used the same definition, but their shrinkage model assumed standardised data, which is restrictive and inefficient since reducing the MSE on transformed data does not have the same effect on the original data; such restriction is removed in Asimit et al. (2025b) where the multiplicative shrinkage – described in Section 3.2 and used in this paper – were first introduced. Two shrinkage estimators are discussed in Section 3.2, namely, St and DSh; both estimators share the same property as GSR by preserving the original eigenvectors and adjusting the eigenvalues of the the covariates covariance matrix.

3.1. Slab Regression (SR and GSR)

The SR estimator imposes a quadratic constraint on a selected linear combination of the parameters, and it is defined as follows:

$$\widehat{\boldsymbol{\beta}}^{SR}(\mu;\mathbf{u}) := \underset{\boldsymbol{\beta}\in\Re^{p+1}}{\arg\min}\,\frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2+\mu\left(\mathbf{u}^T\boldsymbol{\beta}\right)^2,$$

where $\mu \geq 0$ controls the shrinkage and $\mathbf{u} \in \Re^{p+1}$ specifies the direction (for example, $\mathbf{u} = \mathbf{1}$). The GSR estimator generalises this idea by allowing shrinkage along multiple directions and is given by

$$\widehat{\boldsymbol{\beta}}^{GSR}(\boldsymbol{\mu}) := \underset{\boldsymbol{\beta}\in\Re^{p+1}}{\arg\min}\,\frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2+\sum_{l\in\mathcal{L}}\mu_l\left(\mathbf{u}_l^T\boldsymbol{\beta}\right)^2,$$

where each $\mu_l \geq 0$ controls the shrinkage in the direction given by $\mathbf{u}_l$, the eigenvectors of $\Sigma = \mathbf{X}^T\mathbf{X}$.

3.2. Multiplicative Shrinkage (St and DSh)

In the multiplicative approach, the OLS estimator is directly scaled through a diagonal matrix $\mathbf{D}$. The St estimator applies a single global shrinkage parameter, i.e., $\mathbf{D} = a\mathbf{I}_{p+1}$, and the shrinkage estimator for a is as follows:

$$\widehat{\boldsymbol{\beta}}^{\mathrm{St}} = \widehat{a^*}\,\widehat{\boldsymbol{\beta}}^{\mathrm{OLS}}, \quad \text{where} \quad \widehat{a^*} := \frac{\left(\widehat{\boldsymbol{\beta}}^{\mathrm{OLS}}\right)^T\widehat{\boldsymbol{\beta}}^{\mathrm{OLS}}}{\left(\widehat{\boldsymbol{\beta}}^{\mathrm{OLS}}\right)^T\widehat{\boldsymbol{\beta}}^{\mathrm{OLS}}+\widehat{\mathrm{MSE}\left(\widehat{\boldsymbol{\beta}}^{\mathrm{OLS}}\right)}} \in [0,1).$$

The DSh estimator extends this idea by applying coefficient-specific shrinkage factors, i.e., $\mathbf{D} = \mathrm{diag}(\mathbf{b})$, and the shrinkage estimator for $\mathbf{b}$ is as follows:

$$\widehat{\boldsymbol{\beta}}^{\mathrm{DSh}} = \mathrm{diag}(\widehat{\mathbf{b}^*})\,\widehat{\boldsymbol{\beta}}^{\mathrm{OLS}}, \quad \text{with} \quad \widehat{b_k^*} = \frac{(\widehat{\beta}_k^{\mathrm{OLS}})^2}{(\widehat{\beta}_k^{\mathrm{OLS}})^2+\widehat{\sigma^2}\,\sigma_k} \in [0,1),$$

where $\widehat{\sigma^2}$ is the estimated error variance and σ_k is the k^{th} diagonal entry of $\Sigma = \mathbf{X}^T\mathbf{X}$.

4. IRLS Starting Values

The performance and convergence of the IRLS algorithm in fitting GLMs depend strongly on the choice of starting values. Poor starting values may lead to many iterations or even a failure to converge, and even when the starting values are near the true parameters, the algorithm can be unstable unless they are very close to the global optimal solution. These issues are more common in models with nonlinear or complex LFs. In this section, we describe practical choices for starting values by reviewing the default behaviour in the `glm2` package and presenting a convex optimisation method that is solely used to compute improved starting values for the IRLS algorithm. Convex optimisation methods provide reliable solutions for estimating parameters; unlike IRLS, they do not rely on iterative updates but yield starting values that lead to more

stable IRLS performance; `CVXR` in **R**, `CVXPY` in **Python**, or `CVX` in **MATLAB** implement these approaches.[3]

The current default starting values used in `glm2` are discussed in Section 4.1, while our proposed starting values are provided in Section 4.2.

4.1. Starting Values in `glm2`

The default starting values in `glm2` for the mean response, $\mu_i^{(0)}$, are often set as $\mu_i^{(0)} = y_i$, which a natural choice. However, this can cause numerical problems depending on the underlying distribution and its associated LF. We now provide some examples to illustrate our point.

Assume now a LR with a *logit* LF, where the log-likelihood is given by

$$l(\eta; y) = \sum_{i=1}^{n} \left[y_i \log(h(\eta_i)) + (1 - y_i) \log(1 - h(\eta_i)) \right].$$

Although it is theoretically sound to set $\mu_i^{(0)} = h\big(\eta_i^{(0)}\big)$, numerical issues occur if $\mu_i^{(0)}$ is near 0 or 1 due to the log term in the above. This issue is overcome in practice by making some adjustments; e.g., $\mu_i^{(0)} = \frac{y_i + 0.5}{2}$ and $\eta_i^{(0)} = \log\left(\mu_i^{(0)} / \big(1 - \mu_i^{(0)}\big)\right)$ are chosen in `glm2`.

Assume now a *Poisson regression (PoR)* with a generic LF h, where the log-likelihood is given by

$$l(\eta; y) = \sum_{i=1}^{n} \left[-h(\eta_i) + y_i \log(h(\eta_i)) - \log(y_i!) \right].$$

Here, $\mu_i^{(0)} = h\big(\eta_i^{(0)}\big)$ is a common starting point, but this may be problematic when $\mu_i^{(0)} = 0$. For example, *log* LF choices ($h(\eta) = e^{\eta}$) in `glm2` are adjusted by taking $\mu_i^{(0)} = y_i + 0.1$ and $\eta_i^{(0)} = \log(y_i + 0.1)$. Similarly, *sqrt* LF choices ($h(\eta) = \eta^2$) are adjusted in `glm2` by taking $\mu_i^{(0)} = y_i + 0.1$ and $\eta_i^{(0)} = \sqrt{y_i + 0.1}$.

Assume now a *Gamma regression (GaR)* with a generic LF h, where the log-likelihood is given by

$$l(\eta; y) = \sum_{i=1}^{n} \left[-\frac{1}{\phi} \left(\frac{y_i}{h(\eta_i)} + \log(h(\eta_i)) \right) + \frac{1 - \phi}{\phi} \log(y_i) \right] - n \log\left(\phi^{\frac{1}{\phi}} \Gamma\left(\frac{1}{\phi} \right) \right).$$

Once again, the typical starting value, $\mu_i^{(0)} = h\big(\eta_i^{(0)}\big)$, may be far from being ideal when y_i or $\mu_i^{(0)}$ is near zero. The starting values are not adjusted in `glm2` for Gamma regression, and this could be a problem for a user that has examples with small values for the dependent variable. This is not uncommon in practice, and one example is insurance claims data – such as medical insurance – where very small claims are possible.

Table 1 summarises the adjusted default starting values and the validation checks implemented in the `glm2` package for different models and LFs. Note that if y_i is very small, then $\log(y_i)$ or $\sqrt{y_i}$ may still be unstable for the Gamma distribution. During the IRLS iterations, the updated predictor $\eta_i^{(t)} = \mathbf{x}_i^{\top} \boldsymbol{\beta}^{(t)}$ can become non-positive, causing problems that fail validity checks for

[3] Available at: https://stanford.edu/~boyd/software.html

$\eta_i^{(t)}$. Even after adjusting the default starting values in `glm2`, instabilities and early failures may frequently occur.

Table 1: Internal Starting Values and Validations in `glm2` for Various Distributions and LFs

Model	LF	Adjusted Initial Mean $\mu_i^{(0)}$	Initial Predictor $\eta_i^{(0)}$	Valid $\eta_i^{(t)}$?
LR	*logit*	$\frac{y_i+0.5}{2}$	$\log\left(\frac{\mu_i^{(0)}}{1-\mu_i^{(0)}}\right)$	TRUE
PoR	*sqrt*	$y_i+0.1$	$\sqrt{y_i+0.1}$	if $\eta_i^{(t)}>0$
PoR	*log*	$y_i+0.1$	$\log(y_i+0.1)$	TRUE
GaR	*sqrt*	y_i	$\sqrt{y_i}$	if $\eta_i^{(t)}>0$
GaR	*log*	y_i	$\log(y_i)$	TRUE

Notes: This table shows the starting values and checks used by the `glm2` package for different models (LR, PoR, and GaR) and LFs. The adjusted initial mean $\mu_i^{(0)}$ and predictor $\eta_i^{(0)}$ are chosen to ensure numerical stability. The final column indicates whether the predictor $\eta_i^{(t)}$ remains valid during iterations. For simple LFs like *logit* and *log*, the predictor stays valid throughout. For the *sqrt* LF, the predictor must be positive; if it becomes non-positive or non-finite, the model may become unstable or fail to converge.

4.2. Optimisation-Based Starting Values

We have outlined in Section 4.1 the importance and possible pitfalls of starting values for deploying IRLS solutions, and we also discussed some bespoke solutions made in `glm2`. We now provide a novel optimisation-based method to define starting values that are data-driven and LF-driven, which adapts the well-known bespoke solutions available in the existing packages. Our method minimises the difference between a transformed version of the initial mean response $(g^*(\mu_i^{(0)}))$ and the linear predictor $(\mathbf{x}_i^\top\boldsymbol{\beta}^{(0)})$; g^* is defined to ensure that the initial linear predictor is valid and that any necessary constraints are met, as detailed in Table 1. The starting value is the solution of the instance given as (4.1)

$$\begin{cases} \min\limits_{\boldsymbol{\beta}^{(0)}} & \sum\limits_{i=1}^{n}\left(g^*\left(\mu_i^{(0)}\right)-\mathbf{x}_i^\top\boldsymbol{\beta}^{(0)}\right)^2 \\ \text{s.t.} & \mathbf{x}_i^\top\boldsymbol{\beta}^{(0)}\geq\epsilon, \quad \text{for all } 1\leq i\leq n \text{ if required by the chosen LF,} \end{cases} \tag{4.1}$$

where $\mu_i^{(0)}=h\left(\eta_i^{(0)}\right)$ and $\epsilon>0$ is a parameter – ϵ is usually small and the default value in our implementations is 10^{-6} – that ensures $\eta_i^{(0)}=\mathbf{x}_i^\top\boldsymbol{\beta}^{(0)}\geq 0$ as indicated in Table 1. The choice of g^* is essential in (4.1) and one may make different setting; for example, $g^*(\mu)=\sqrt{\mu}$ and $g^*(\mu)=\log(\mu+0.1)$ are natural choices for the *sqrt* LF and *log* LF, respectively. The inequality constraints in (4.1) may be discarded in some cases as anticipated in Table 1; e.g., *logit* or *log* do not require extra restrictions.

In a nutshell, the starting value solution in (4.1) relies on a scalable convex quadratic instance that aims to reduce the distance between $\mathbf{x}_i^\top\boldsymbol{\beta}^{(0)}$ and $g^*\left(\mu_i^{(0)}\right)$. Our solution could address different issues such as i) lack of convergence and ii) excessive iterations. In order to test such claims we compare the convergence performance of Poisson and Gamma GLMs with *sqrt* LF by using our proposed starting values and the default starting values in **Matlab**, **Python**, and **R**. Table 2 reports the number of convergence failures – lower numbers indicate better convergence performance – based on each software implementation (reported outside the brackets) and our starting values (reported inside the brackets). For example, when we look at Poisson GLM

for **R**'s implementation in `glm2`, we compare the convergence failures of `glm2` with its starting solution to `glm2` with our starting solution in (4.1), so that we have like-for-like analysis. Values tabulated in Table 2 are based on $N = 100$ samples of size $n = 500$ based on the first *Data Generation Process (DGP1)* that is provided in Appendix C.

Table 2: Number of failures to convergence with *sqrt* LF DGP1

p	1% n			10% n		
ρ	**-0.5**	**0**	**0.5**	**-0.5**	**0**	**0.5**
Panel A: Poisson Distribution						
μ=**0**						
R	0 (0)	0 (0)	**20 (0)**	**1 (0)**	0 (0)	0 (0)
Matlab	39 (39)	68 (68)	86 (86)	13 (13)	15 (15)	11 (11)
Python	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
μ=**3**						
R	**9 (0)**	**67 (0)**	**100 (0)**	**57 (0)**	**100 (1)**	**100 (0)**
Matlab	75 (75)	96 (96)	100 (100)	96 (96)	100 (100)	100 (100)
Python	0 (0)	0 (0)	**1 (0)**	**1 (0)**	0 (0)	0 (0)
μ=**5**						
R	**96 (0)**	**100 (1)**	**100 (1)**	**100 (0)**	**100 (0)**	**100 (0)**
Matlab	99 (99)	100 (100)	100 (100)	100 (100)	100 (100)	100 (100)
Python	0 (0)	**4 (0)**	**5 (0)**	0 (0)	0 (0)	**1 (0)**
Panel B: Gamma Distribution						
μ=**0**						
R	**100 (3)**	**100 (2)**	**100 (1)**	**100 (0)**	**100 (2)**	**100 (0)**
Matlab	**100 (99)**	100 (100)	100 (100)	**68 (67)**	92 (92)	100 (100)
Python	100 (100)	100 (100)	**98 (80)**	**98 (96)**	**98 (96)**	**99 (90)**
μ=**3**						
R	**100 (1)**	**100 (0)**	**100 (3)**	**100 (0)**	**100 (0)**	**100 (0)**
Matlab	100 (100)	**100 (99)**	100 (100)	61 (65)	**82 (81)**	100 (100)
Python	**98 (92)**	**100 (38)**	**99 (0)**	**97 (42)**	**100 (0)**	**99 (0)**
μ=**5**						
R	**100 (1)**	**100 (1)**	**100 (1)**	**100 (0)**	**100 (0)**	**94 (0)**
Matlab	100 (100)	100 (100)	100 (100)	**65 (64)**	82 (83)	99 (100)
Python	**99 (1)**	**99 (0)**	**98 (0)**	**97 (0)**	**99 (0)**	**93 (0)**

Notes: This table shows the number of convergence failures for `glm2` over $N = 100$ replications, each with $n = 500$ observations. The numbers outside parentheses are for the default starting values within **R**, **Matlab** and **Python**, and the numbers inside parentheses are for our optimisation-based adjusted starting values. Bold numbers indicate cases where our starting values produced fewer failures than the default. A convergence failure means that the algorithm does not reach a valid solution within a maximum of $10,000$ iterations. The results are reported for different mean parameters μ and predictor correlations ρ at sample proportions of 1% and 10% of the total sample size. Panel A shows results for the Poisson distribution, while Panel B shows results for the Gamma distribution.

As shown in Table 2, our optimisation-based starting value solution could significantly improve the convergence for Poisson and Gamma GLMs, especially for **R**'s `glm2`; similar improvements are observed in **MATLAB**'s `fitglm` and **Python**'s `statsmodels.GLM`. These findings indicate that our method produces stable and feasible starting values that are always "no worse" than the existing bespoke solutions. Similar results are shown in Appendix D for second *Data Generation Process (DGP2)* that is described in Appendix C.

5. Simulation study

This section presents a series of simulation experiments designed to evaluate the performance of several GLM estimation methods. In our study, we compare the standard IRLS method – as given in **R**'s package `glm2` – with our IRLS method for which the WLS step is replaced by the four previously-mentioned shrinkage approaches (SR, GSR, St, and DSh). Note that all four shrinkage-based GLMs proposed in this paper are implemented in our new **R** package, *savvyGLM*[4]. In Section 5.1, we describe the design of these experiments and the performance metrics used. Then our evaluation focuses on two key aspects: i) the reduction in the L_2 error between the "true" and estimated regression coefficients in Section 5.2, and ii) computational efficiency as measured by the number of iterations required for convergence in Section 5.3.

5.1. Simulation Design and Performance Metrics

Synthetic datasets of size $n = 500$ are generated under various configurations. The simulations vary two main factors: the correlation coefficient ρ among predictors, with values -0.75, -0.5, 0, 0.5, and 0.75; and the predictor-to-sample size ratio, with p/n set at 1%, 10%, 25%, and 50%. For each combination, 250 independent replications were performed. The GLM models considered include LR for both balanced and imbalanced datasets, as well as Poisson and Gamma regression models, each implemented with both *sqrt* and *log* LFs. Further details on the IRLS algorithm and the DGP1 are provided in Appendix B and Appendix C, respectively.

The estimation accuracy is quantified by the *Mean L_2 Error (ML_2)*, defined as the average Euclidean distance between the estimated and true regression coefficient vectors over the N replications:

$$ML_2(\text{model}) = \frac{1}{N}\sum_{k=1}^{N} L_2\big(\hat{\boldsymbol{\beta}}_k^{\text{model}}\big), \quad \text{where} \quad L_2\big(\hat{\boldsymbol{\beta}}_k^{\text{model}}\big) = \sqrt{\sum_{j=1}^{p}\Big(\hat{\beta}_{k,j}^{\text{model}} - \beta_{k,j}^{\text{true}}\Big)^2}.$$

Here, $\beta_{k,j}^{\text{true}}$ is the true j^{th} regression coefficient for the k^{th} dataset, and $\hat{\beta}_{k,j}^{\text{model}}$ is the corresponding estimated coefficient. To facilitate comparison, we compute the *Relative Mean L_2 Error (RML_2)* of each model relative to the benchmark (`glm.fit2` IRLS implementation) as

$$RML_2 = \frac{ML_2\big(\text{benchmark}\big) - ML_2\big(\text{model}\big)}{ML_2\big(\text{benchmark}\big)}.$$

A positive RML_2 indicates an improvement over the benchmark. Table 3 summarises the true response distributions and predictor structures used in the simulations under different LFs.

5.2. Analysis of Estimation Accuracy

The estimation accuracy of the proposed shrinkage approaches (SR, GSR, St, and DSh) was evaluated against the benchmark `glm2` by comparing their L_2 errors under various statistical

[4]Available at: https://github.com/Ziwei-ChenChen/savvyGLM

Table 3: True response distribution and predictors used in GLM with different LFs.

Model	LF	True Response	Predictor Used in GLM
LR	*logit*	$Y_i \sim Binomial\left(1, 1/(1+e^{-\eta_i})\right)$	$h(\eta_i) = 1/(1+e^{-\eta_i})$
PoR	*sqrt*	$Y_i \sim Poisson(\eta_i^2)$	$h(\eta_i) = \eta_i^2$
PoR	*log*	$Y_i \sim Poisson(e^{\eta_i})$	$h(\eta_i) = e^{\eta_i}$
GaR	*sqrt*	$Y_i \sim Gamma(\eta_i^2, 1)$	$h(\eta_i) = \eta_i^2$
GaR	*log*	$Y_i \sim Gamma\left(e^{\eta_i}, 1\right)$	$h(\eta_i) = e^{\eta_i}$

Notes: This table shows the true response distributions and the predictors used in the GLM models with different LFs. The first column gives the model type (LR, PoR, and GaR). The second column lists the LF used. The third column shows the true response distribution used in the simulations, and the fourth column shows the predictor function used in the GLM. Data are generated according to the specified response distribution and the corresponding LF is applied for model fitting.

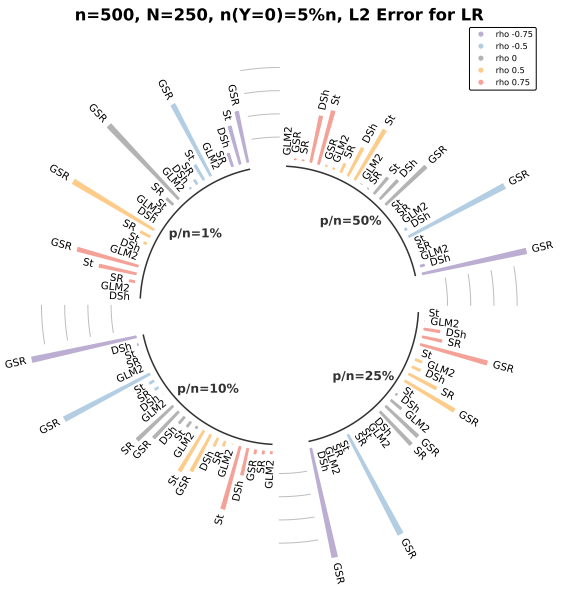
scenarios. Figures 1 and 2 show the frequency with which each model achieved the lowest L_2 error over 250 replications, while Tables 4–6 present the corresponding RML_2 values.

For LR, the results in Figure 1 and Table 4 indicate that the shrinkage-based models generally have lower L_2 errors than `glm2`, especially when the predictor-to-sample size ratio p/n is low (e.g., 1%). As p/n increases, the benefits become smaller and the performance of our methods becomes similar to that of `glm2`. Under balanced data, our methods handle strong negative correlations ($\rho < 0$) well, but for imbalanced data the results depend on ρ and tend to improve when $\rho > 0$. In some cases, GSR and SR achieve the lowest L_2 error most frequently, although the overall gain may be small or even negative when a few replications perform poorly.

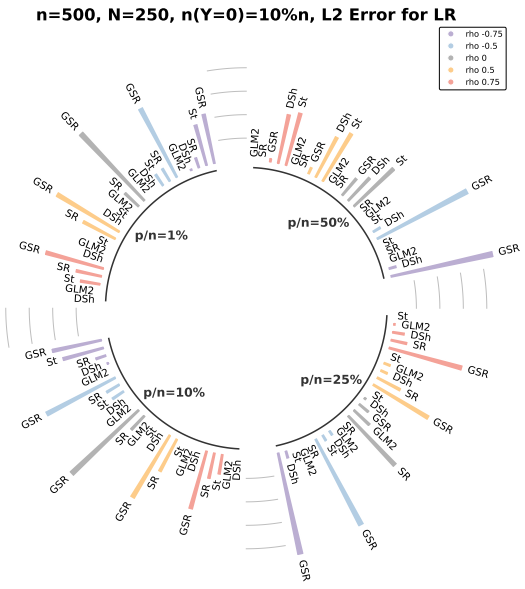
For Poisson and Gamma GLMs with the *sqrt* LF shown in Figures 2a, 2b and Table 5, our shrinkage-based models generally achieve lower L_2 errors than `glm2`. In these settings, the St or DSh models often show larger error reductions and are the best performers more frequently, especially at higher p/n ratios and when ρ is strongly negative. The SR usually performs similarly to `glm2`, while GSR shows moderate improvements by yielding positive RML_2 errors. Figures 2c, 2d and Table 6 present results using the *log* LF. In these cases, GSR or St consistently produce the highest improvements in relative mean L_2 error and the lowest L_2 errors, particularly at higher p/n ratios (e.g., 50%). Although DSh may perform less well when p/n is very small, it still shows positive improvements in most cases, while SR remains similar to `glm2`. Overall, the *log* LF appears to offer greater benefits than the *sqrt* LF, with these gains increasing as the correlation *rho* becomes more positive. Moreover, under the *sqrt* LF, the Gamma model benefits more from shrinkage than the Poisson model. In summary, our shrinkage-based GLMs generally outperform or are at least comparable to `glm2`, although the size of improvement depends on the distribution, LFs, and correlation structure. Among the proposed methods, the St stands out as the most consistently effective, especially at high p/n ratios. Table 7 summarises the overall trends in this section under different settings.

5.3. Analysis of Computational Efficiency

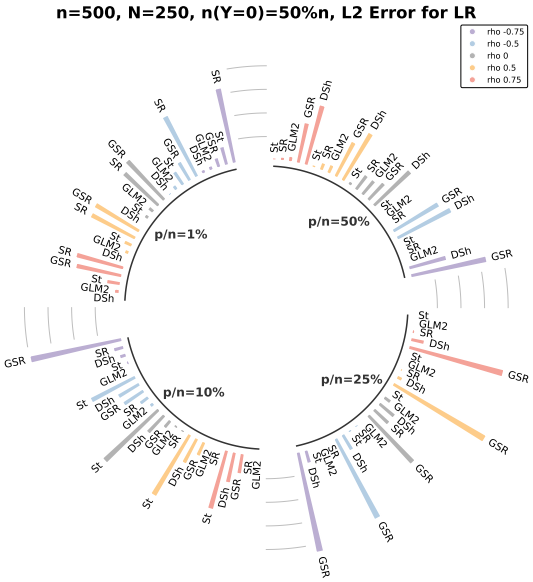
The computational efficiency of the models was evaluated by comparing the number of iterations required to achieve convergence. Figures 3 and 4 provide a visual representation of the iteration counts, which highlight how often each model reached convergence with the minimum number of iterations across 250 replications for each scenario.



(a) LR with Imbalanced data ($n(Y = 0) = 5\%n$)



(b) LR with Imbalanced data ($n(Y = 0) = 10\%n$)



(c) LR with Balanced data ($n(Y = 0) = 50\%n$)

Figure 1: Comparison of L_2 errors for the LR model. Top row: models with imbalanced data. Bottom row: models with balanced data. Longer bars indicate better performance.

For LR, Figure 3 indicates that the SR and GSR models perform similarly to the benchmark `glm2`, with no notable difference in iteration counts. In contrast, the DSh and St models generally require more iterations than `glm2`, regardless of whether the data are balanced or imbalanced. This suggests that for LR, our proposed methods do not provide a clear computational advantage and may even converge more slowly in some cases.

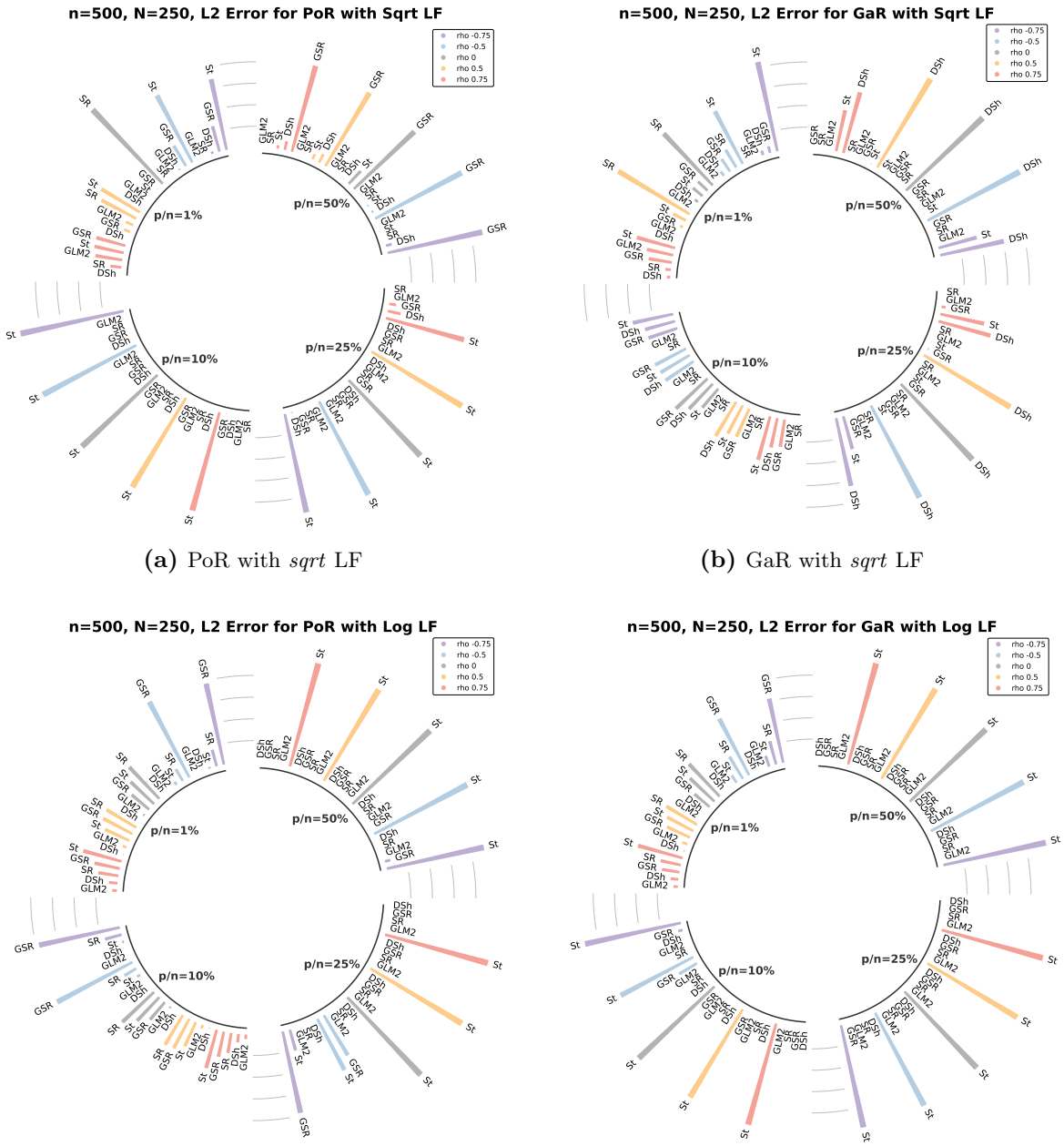
In contrast, for Poisson and Gamma GLMs with the *sqrt* LF, Figures 4a and 4b demonstrate that the St or DSh models generally converge more quickly than `glm2`, requiring fewer iterations across most tested p/n ratios. The SR also shows improved efficiency in scenarios with smaller p/n ratios (e.g., 1%) and positive ρ values although this advantage is less consistent. For

Table 4: Relative Mean L_2 Errors For LR

Extreme Rare Event Case: $\frac{n(Y=0)}{n(Y=1)} = \frac{5}{95}$												
	ρ = -0.75				ρ = -0.5				ρ = 0			
p/n	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**
SR	1.48%	0.08%	0.01%	0.00%	0.73%	0.14%	0.01%	0.00%	1.62%	**0.88%**	**0.04%**	0.00%
GSR	-0.74%	**2.15%**	**0.26%**	**0.07%**	**3.43%**	**1.72%**	**0.24%**	**0.08%**	**4.71%**	0.80%	0.02%	**0.04%**
St	0.97%	0.58%	0.01%	0.00%	2.47%	0.31%	-0.05%	0.01%	2.28%	0.30%	-0.05%	0.02%
DSh	**3.25%**	-1.41%	-0.09%	0.00%	1.78%	-1.32%	-0.15%	0.01%	1.47%	-0.92%	-0.21%	0.00%
	ρ = 0.5				ρ = 0.75							
p/n	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**				
SR	1.77%	0.03%	0.03%	0.00%	1.07%	-0.01%	0.02%	0.00%				
GSR	**5.10%**	0.26%	**0.09%**	-0.07%	**4.02%**	-1.05%	**0.26%**	-0.08%				
St	3.12%	**0.62%**	-0.09%	**0.04%**	3.65%	**0.14%**	-0.17%	**0.07%**				
DSh	1.71%	-0.15%	-0.26%	-0.01%	1.26%	-0.04%	-0.36%	0.05%				
Rare Event Case: $\frac{n(Y=0)}{n(Y=1)} = \frac{10}{90}$												
	ρ = -0.75				ρ = -0.5				ρ = 0			
p/n	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**
SR	0.96%	0.16%	0.01%	0.00%	1.08%	0.22%	0.01%	0.00%	3.02%	0.01%	**0.04%**	0.00%
GSR	**3.18%**	**2.89%**	**0.33%**	**0.09%**	**3.78%**	**2.20%**	**0.28%**	**0.09%**	**5.64%**	**0.03%**	-0.07%	0.01%
St	3.03%	0.73%	0.04%	0.01%	2.51%	0.30%	0.08%	0.01%	2.24%	-0.06%	-0.09%	**0.05%**
DSh	1.65%	-2.74%	-0.05%	0.01%	1.71%	-1.69%	-0.06%	0.01%	1.52%	-0.99%	-0.24%	0.01%
	ρ = 0.5				ρ = 0.75							
p/n	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**				
SR	3.98%	-0.18%	0.03%	0.00%	1.99%	-0.07%	0.02%	0.00%				
GSR	**4.81%**	-1.31%	**0.14%**	-0.04%	**3.90%**	-1.59%	**0.35%**	-0.07%				
St	2.60%	**0.05%**	-0.14%	**0.09%**	2.94%	-0.10%	-0.23%	**0.14%**				
DSh	1.43%	-0.08%	-0.26%	0.07%	0.91%	-0.13%	-0.40%	0.12%				
Balanced: $\frac{n(Y=0)}{n(Y=1)} = \frac{50}{50}$												
	ρ = -0.75				ρ = -0.5				ρ = 0			
p/n	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**
SR	**7.55%**	0.67%	0.02%	0.00%	**6.77%**	0.53%	0.03%	0.00%	4.16%	-0.21%	0.03%	0.00%
GSR	-1.64%	**4.16%**	**0.50%**	**0.10%**	-0.90%	0.46%	**0.40%**	0.08%	**4.78%**	-1.06%	**0.57%**	0.02%
St	2.55%	0.08%	0.01%	-0.01%	2.46%	**1.18%**	-0.06%	0.00%	1.36%	**0.31%**	0.31%	-0.01%
DSh	0.56%	-2.83%	0.19%	0.08%	0.92%	-0.84%	0.09%	**0.09%**	-1.16%	0.11%	0.42%	**0.03%**
	ρ = 0.5				ρ = 0.75							
p/n	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**				
SR	**3.74%**	0.02%	0.00%	0.00%	**2.27%**	-0.01%	0.00%	0.00%				
GSR	3.12%	-0.90%	**1.60%**	**0.12%**	2.15%	-0.74%	**3.38%**	0.21%				
St	0.62%	**0.26%**	0.49%	0.03%	0.46%	**0.14%**	1.09%	0.06%				
DSh	-2.17%	0.14%	0.78%	**0.12%**	-2.64%	0.04%	1.91%	**0.29%**				

Notes: This table shows the relative Mean L_2 errors for the LR using the *logit* LF. The errors are given as percentages relative to the benchmark `glm2`. Negative values indicate worse performance than `glm2`, while positive values indicate better performance. Bold numbers mark the best performance for each setting. Results are reported for different predictor-to-sample size ratios p/n and correlation values ρ. All models used the same starting values, which were computed using an optimisation-based procedure provided in Section 4.2. Note that all models converged in all 250 replications with $n = 500$.

the *log* LF, Figure 4c and 4d show that both Poisson and Gamma GLMs exhibit noticeable improvements in convergence efficiency with the St estimator, which consistently requires fewer iterations than `glm2` across all p/n ratios. The DSh and GSR models also display competitive



(a) PoR with *sqrt* LF

(b) GaR with *sqrt* LF

(c) PoR with *log* LF

(d) GaR with *log* LF

Figure 2: Comparison of L_2 errors for PoR and GaR using two LFs. Top row: models with the *sqrt* LF. Bottom row: models with the *log* LF. Longer bars indicate better performance.

performance in several scenarios, which further supports their efficiency under this setting. However, the SR aligns closely with `glm2` in terms of iteration counts, indicating no significant computational gains in this case. Notably, the St demonstrates particular strength for the Gamma GLM with the *log* LF, which achieves reduced iteration counts in most scenarios. Overall, these results suggest that the St and DSh models consistently converge faster than `glm2` for Poisson and Gamma GLMs, while the SR and GSR models offer mixed improvements that depend on the predictor-to-sample ratio and the chosen LF.

Table 5: Relative Mean L_2 Errors For PoR and GaR with *sqrt* LF

Possion Distribution												
	ρ = -0.75				ρ = -0.5				ρ = 0			
p/n	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**
SR	-4.69%	-0.29%	0.19%	0.26%	-3.11%	-0.40%	0.07%	0.18%	**4.20%**	0.02%	0.19%	0.11%
GSR	0.45%	3.91%	9.05%	**13.79%**	0.23%	2.33%	5.92%	**9.93%**	0.22%	1.67%	3.70%	**5.96%**
St	**0.72%**	**13.46%**	**18.35%**	6.46%	**0.44%**	**5.84%**	**13.32%**	6.84%	0.21%	**2.41%**	**6.16%**	5.26%
DSh	0.03%	3.08%	8.24%	10.77%	-0.30%	1.17%	3.32%	8.04%	-0.43%	0.41%	1.14%	4.54%
	ρ = 0.5				ρ = 0.75							
p/n	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**				
SR	-0.32%	0.33%	0.09%	0.00%	-2.33%	0.17%	0.02%	0.01%				
GSR	-0.12%	0.77%	2.06%	**3.75%**	-0.09%	0.49%	1.49%	**3.18%**				
St	**0.11%**	**1.97%**	**4.60%**	2.55%	**0.01%**	**2.12%**	**3.10%**	1.62%				
DSh	-0.34%	0.41%	1.13%	3.08%	-0.36%	0.42%	1.41%	2.57%				
Gamma Distribution												
	ρ = -0.75				ρ = -0.5				ρ = 0			
p/n	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**
SR	-0.17%	-0.03%	-0.04%	0.36%	0.14%	-0.01%	-0.11%	-0.24%	**0.66%**	0.04%	0.14%	0.15%
GSR	0.62%	8.47%	13.97%	16.21%	0.41%	4.39%	6.17%	9.78%	0.27%	**2.22%**	2.26%	4.57%
St	**1.14%**	8.00%	16.70%	22.71%	**0.56%**	3.80%	5.07%	8.73%	0.27%	1.70%	2.75%	4.96%
DSh	0.31%	**8.60%**	**22.44%**	**34.52%**	0.13%	**4.43%**	**13.02%**	**24.36%**	0.01%	2.15%	**7.21%**	**14.22%**
	ρ = 0.5				ρ = 0.75							
p/n	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**				
SR	**1.34%**	0.01%	-0.05%	-0.14%	-1.59%	0.02%	0.07%	0.10%				
GSR	-0.02%	1.56%	2.70%	4.91%	-0.17%	1.49%	3.32%	6.33%				
St	0.16%	1.36%	2.04%	4.12%	**0.01%**	**1.55%**	4.39%	8.02%				
DSh	-0.25%	**1.58%**	**5.41%**	**11.75%**	-0.42%	1.42%	**5.37%**	**12.01%**				

Notes: This table shows the relative Mean L_2 errors for the PoR and GaR using the *sqrt* LF. The errors are given as percentages relative to the benchmark `glm2`. Negative values indicate worse performance than `glm2`, while positive values indicate better performance. Bold numbers mark the best performance for each setting. Results are reported for different predictor-to-sample size ratios p/n and correlation values ρ. All models used the same starting values, which were computed using an optimisation-based procedure provided in Section 4.2. Note that not all models converged in all $N = 250$ replications with $n = 500$; see Section Appendix D for more details on convergence.

6. Real Data Analysis

In this section, we assess the out-of-sample performance of our shrinkage-based approaches (SR, GSR, St, and DSh) compared to the benchmark `glm2`. The evaluation is carried out on two types of datasets: i) the Crabs and Heart datasets from the `glm2` package and ii) a U.S. flood insurance dataset. In both studies, we perform $N = 110$ replications. In each replication, the data are randomly split into 70% training and 30% testing sets. Prediction performance is measured by the MSE and reported as MSE ratio $= \text{MSE}_{\texttt{glm2}}/\text{MSE}_{\texttt{model}}$. A ratio greater than one indicates that our shrinkage-based GLMs yield a lower MSE than `glm2`. For both studies, after discarding the five highest and five lowest ratios, the MSE ratio and the count of replications where the ratio exceeds one are computed from the remaining 100 replications. The performance is reported for both *log* and the *sqrt* LFs. Preprocessing details for the Crabs and Heart datasets can be found in the documentation of the `glm2` package, while the preprocessing of the flood insurance dataset is described in Appendix E.

Table 6: Relative Mean L_2 Errors For PoR and GaR with *log* LF

Poisson Distribution

	ρ = -0.75				ρ = -0.5				ρ = 0			
p/n	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**
SR	19.39%	2.16%	1.08%	0.56%	14.01%	2.19%	0.78%	0.55%	**8.63%**	1.28%	0.64%	0.34%
GSR	**33.17%**	**31.17%**	**31.00%**	30.24%	**27.28%**	**27.82%**	29.37%	29.35%	2.33%	10.26%	20.96%	26.89%
St	14.53%	12.91%	24.81%	**37.89%**	9.00%	18.93%	**30.22%**	**40.47%**	3.92%	**16.67%**	**30.31%**	**41.27%**
DSh	1.67%	8.34%	19.86%	27.81%	-8.08%	8.85%	21.77%	27.79%	-12.42%	5.58%	21.24%	28.10%

	ρ = 0.5				ρ = 0.75			
p/n	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**
SR	3.96%	0.22%	0.12%	0.07%	1.35%	0.02%	0.03%	0.02%
GSR	5.27%	16.23%	24.25%	28.66%	7.53%	21.87%	27.81%	29.98%
St	**5.74%**	**22.87%**	**35.37%**	**43.66%**	**11.02%**	**31.38%**	**40.98%**	**45.71%**
DSh	-8.30%	12.01%	24.63%	29.07%	1.39%	20.28%	27.99%	30.69%

Gamma Distribution

	ρ = -0.75				ρ = -0.5				ρ = 0			
p/n	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**
SR	20.70%	2.24%	0.98%	0.51%	17.33%	1.73%	0.92%	0.53%	**10.92%**	1.21%	0.53%	0.20%
GSR	**29.99%**	32.63%	34.92%	39.55%	**23.02%**	30.52%	33.36%	37.18%	3.50%	16.63%	26.31%	33.35%
St	18.42%	**48.64%**	**58.29%**	**65.34%**	11.15%	**35.69%**	**47.62%**	**56.74%**	6.33%	**24.90%**	**37.64%**	**47.79%**
DSh	5.99%	36.99%	44.72%	50.54%	-5.60%	23.24%	34.70%	41.90%	-10.17%	12.27%	26.51%	34.51%

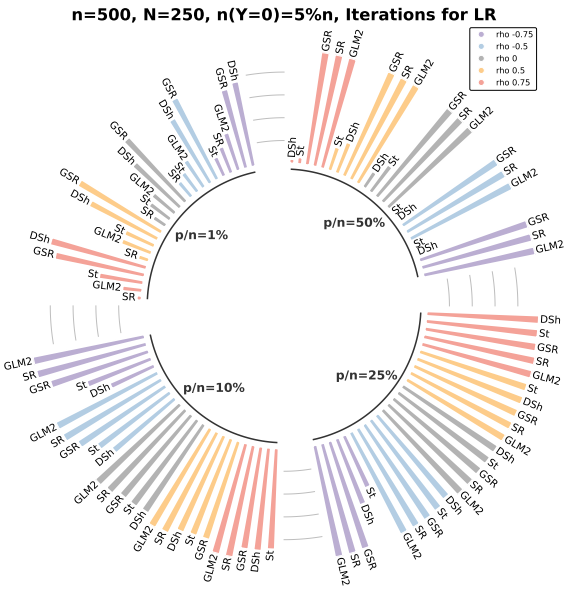
	ρ = 0.5				ρ = 0.75			
p/n	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**
SR	**4.65%**	0.19%	0.09%	0.04%	1.69%	0.03%	0.00%	0.01%
GSR	3.13%	18.53%	27.37%	32.90%	7.03%	23.52%	30.22%	34.09%
St	3.38%	**26.61%**	**39.42%**	**47.68%**	**10.60%**	**34.25%**	**44.11%**	**49.79%**
DSh	-10.54%	14.30%	27.31%	33.29%	1.57%	21.82%	30.01%	33.81%

Notes: This table shows the relative Mean L_2 errors for the PoR and GaR using the *log* LF. The errors are given as percentages relative to the benchmark `glm2`. Negative values indicate worse performance than `glm2`, while positive values indicate better performance. Bold numbers mark the best performance for each setting. Results are reported for different predictor-to-sample size ratios p/n and correlation values ρ. All models used the same starting values, which were computed using an optimisation-based procedure provided in Section 4.2. The true regression parameters (detailed in Appendix C) ensure that the expected response remains within a reasonable range, preventing the generation of overly large Y values that could cause IRLS failures. Note that all models converged in all $N = 250$ replications with $n = 500$.

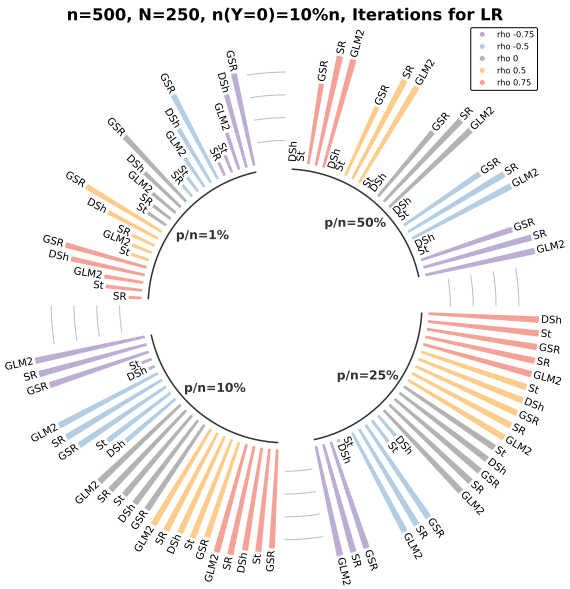
Table 7: Summary of the Analysis of Estimation Accuracy

Model and LF	**Key Findings**	**Overall Best**
LR with *logit* LF	SR, GSR, and St perform at least as well as `glm2`	GSR
PoR with *sqrt* LF	GSR, St, and DSh outperform `glm2`	St
GaR with *sqrt* LF	GSR, St, and DSh beat `glm2`	DSh
PoR with *log* LF	All shrinkage methods improve on `glm2`	St
GaR with *log LF*	All estimators outperform `glm2`	St

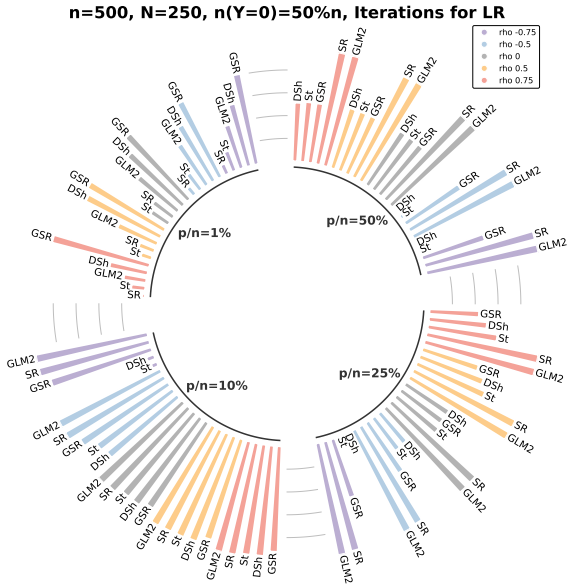
Notes: The results summarise the main trends observed in the simulation studies about estimation accuracy. Overall, the proposed shrinkage GLM methods either outperform or match the benchmark `glm2`. The improvement depends on the predictor-to-sample size ratio p/n and the correlation parameter ρ.



(a) LR with Imbalanced data ($n(Y = 0) = 5\%n$)



(b) LR with Imbalanced data ($n(Y = 0) = 10\%n$)

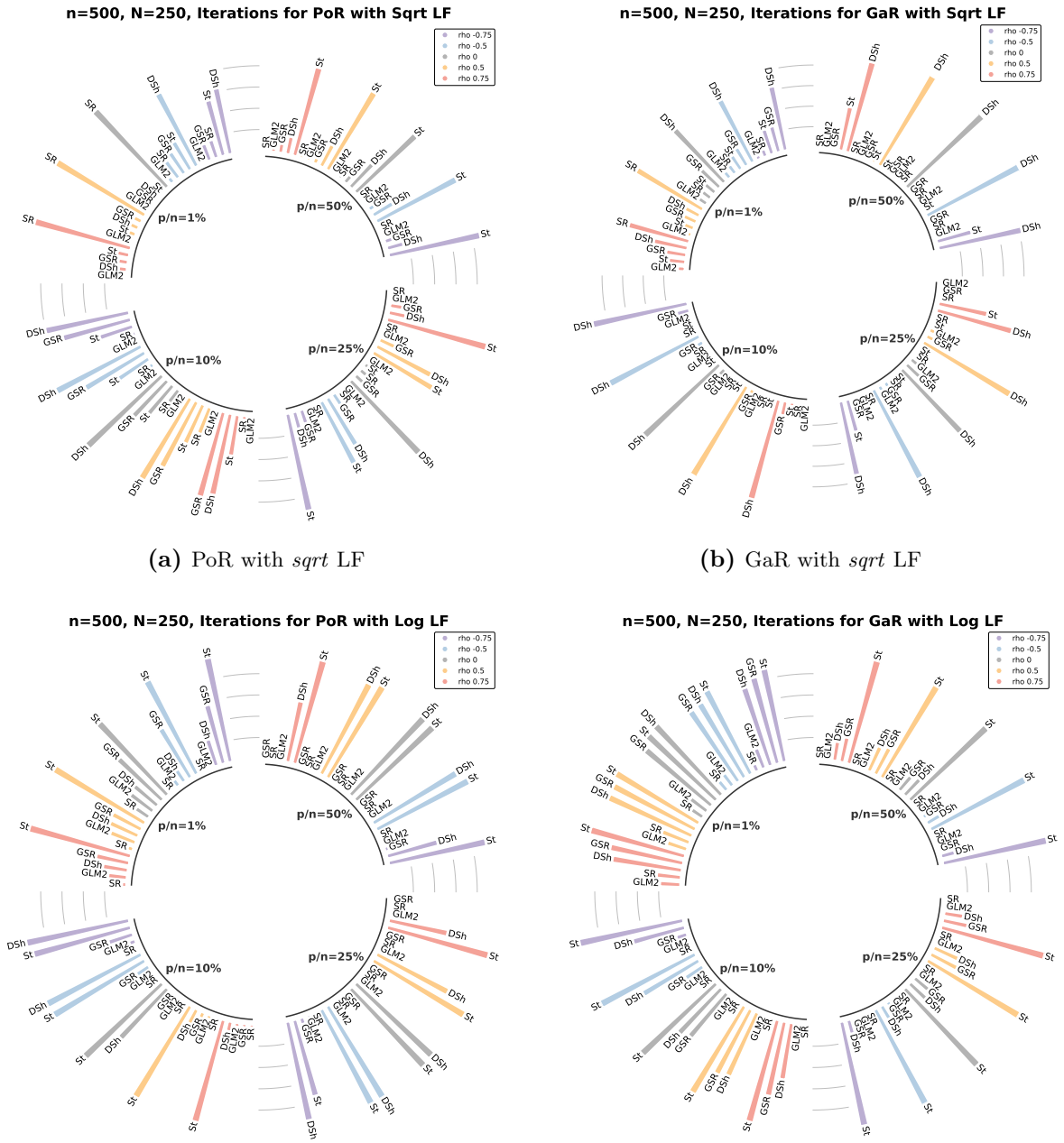


(c) LR with Balanced data ($n(Y = 0) = 50\%n$)

Figure 3: Comparison of iterations for the LR model. Top row: models with imbalanced data. Bottom row: models with balanced data. Longer bars indicate better performance.

6.1. Crabs and Heart Datasets from `glm2`

Table 8 compares the performance of Poisson GLMs on the Crabs and Heart datasets for both LFs. Panels A and B show the model counts and MSE ratio statistics, respectively, under the *log* LF. In Panel A, the row labelled **best** reports how many times each model achieved the lowest MSE, while the row labelled **win glm2** shows how many replications each shrinkage model outperformed `glm2`. For example, on the Crabs dataset, GSR achieved the lowest MSE in 28 replications, and on the Heart dataset, DSh topped the chart in 52 replications. By contrast, `glm2` was best in only 10 replications for both datasets. Panel B shows the corresponding MSE ratios. The DSh model achieves an average ratio of 1.0444 on the Heart dataset, while GSR



(a) PoR with *sqrt* LF

(b) GaR with *sqrt* LF

(c) PoR with *log* LF

(d) GaR with *log* LF

Figure 4: Comparison of iterations for PoR and GaR using two LFs. Top row: models with the *sqrt* LF. Bottom row: models with the *log* LF. Longer bars indicate better performance.

attains an average ratio of 1.013 on the Crabs dataset. Both ratios exceed one, which indicates an improvement over `glm2`.

Panels C and D show results for the *sqrt* LF. In Panel C, DSh leads in 35 replications on the Crabs dataset compared to 26 for `glm2`, while SR leads in 27 replications on the Heart dataset compared to 20 for `glm2`. Panel D shows the MSE ratio statistics, where DSh reaches an average ratio of 1.0064 on the Crabs dataset, and St achieves 1.0211 on the Heart dataset. Overall, these findings suggest that our shrinkage-based GLMs provide modest but consistent improvements over `glm2`, depending on both the dataset and the chosen LF.

Table 8: Poisson GLM for Crabs and Heart data

Poisson GLM with *log* LF										
Panel A: Model Counts										
	Crabs dataset					**Heart dataset**				
	glm2	**SR**	**GSR**	**St**	**DSh**	**glm2**	**SR**	**GSR**	**St**	**DSh**
best	10	14	**35**	23	28	10	20	16	12	**52**
win glm2	-	57	**74**	38	56	-	40	39	54	**65**
Panel B: MSE Ratios										
Average	-	1.0030	**1.0131**	0.9791	1.0128	-	0.9810	0.9887	1.0002	**1.0444**
25% quantile	-	0.9974	0.9982	0.9088	0.9632	-	0.8959	0.9298	0.9310	0.9334
50% quantile	-	1.0010	**1.0112**	0.9573	1.0089	-	0.9686	0.9800	1.0173	**1.0768**
75% quantile	-	1.0080	1.0285	1.0520	**1.0577**	-	1.0774	1.0331	1.0550	**1.1949**
Poisson GLM with *sqrt* LF										
Panel C: Model Counts										
	Crabs dataset					**Heart dataset**				
	glm2	**SR**	**GSR**	**St**	**DSh**	**glm2**	**SR**	**GSR**	**St**	**DSh**
best	26	17	15	17	**35**	20	**27**	15	24	23
win glm2	-	22	36	32	**48**	-	**58**	36	39	44
Panel D: MSE Ratios										
Average	-	0.9663	0.9901	0.9704	**1.0064**	-	1.0086	0.9818	**1.0211**	0.9775
25% quantile	-	0.9293	0.9782	0.9031	0.9662	-	0.9954	0.9421	0.9100	0.8136
50% quantile	-	0.9660	0.9915	0.9550	0.9971	-	**1.0039**	0.9900	0.9563	0.9355
75% quantile	-	0.9979	1.0064	1.0343	**1.0483**	-	1.0224	1.0237	1.1613	**1.1617**

Notes: This table shows the performance of Poisson GLM on the Crabs and Heart datasets using both *log* and *sqrt* LFs. In Panels A and C, the row labelled **best** gives the number of replications where each model achieved the lowest MSE, and the row labelled **win glm2** shows the number of replications where the model beat the benchmark `glm2`, which are computed from the middle 100 replications after removing the five highest and five lowest ratios based on $N = 110$ replications. Panels B and D present the mean and the 25%, 50%, and 75% quantiles of the MSE ratio. Bold numbers mark the best performance compared to `glm2`; `glm2` is the reference model and is marked with a dash. All models used the same starting values from the optimisation-based procedure provided in Section 4.2, with a maximum of 250 iterations and a tolerance of 10^{-6}.

6.2. U.S. Flood Insurance Dataset

This section examines the performance of Gamma GLMs on a U.S. flood insurance dataset obtained from "OpenFEMA"[5]. The dataset comprises claims from the *National Flood Insurance Programme (NFIP)* across 50 U.S. states. For our analysis, we focus on claims from *Florida (FL)*, *Texas (TX)*, and *Louisiana (LA)*, states with the highest number of flood-related claims. The dependent variable, *ratioCoverage*, is defined as the ratio of *amountPaidOnBuildingClaim* to *totalBuildingInsuranceCoverage* and is capped at the 99^{th} percentile to reduce the effect of extreme values. The dataset includes 15 covariates that describe financial, building, and geographic features; details can be found in Appendix E. Data from 2014 to 2023 are used separately for out-of-sample performance analysis. For each year and state, we calculate the *average MSE ratio (AMR)* and the *count (CNT)* of repetitions where the ratio exceeds one.

Tables 9 and 10 summarise the performance of Gamma GLMs using the *sqrt* and *log* LFs, respectively. For the *sqrt* LF, the overall count summary in Panel A shows that the DSh con-

[5]The dataset is available at https://www.fema.gov/openfema-data-page/fima-nfip-redacted-claims-v2.

sistently achieves the highest win counts compared to `glm2` across most of the 100 replications. In many cases, the DSh is identified as the best overall model. The SR, GSR, and St models also show modest improvements over `glm2`; in some years and states, the average MSE ratios for SR and GSR models are close to or above one, indicating competitive performance even though their overall win counts are generally lower than those of DSh. In contrast, for *log* OF, the performance difference is more obvious. The overall results indicate that the St leads with the highest win counts and best overall performance in most settings. The SR and GSR models also provide moderate improvements that vary by state and year, yet they still outperform `glm2` on average. Besides, the DSh performs worse under *log* LF than under *sqrt* LF, with fewer but still respectable wins compared to the St. Overall, these results indicate that the St is the most effective method when using *log* LF, while the DSh is most competitive with *sqrt* LF.

Extreme flooding events recorded by the *National Centre for Environmental Information (NCEI)*[6] provide additional context for the results presented. Several major disasters impacted the study regions during the analysis period (2014-2023). In Texas, extreme rainfall and flooding events in 2015, 2016, 2017, and 2019 caused billions of dollars in losses. The 2016 Louisiana flood, a historic event, destroyed over 50,000 homes, while Hurricanes *Laura* and *Delta* in 2021 brought widespread damage to homes across Texas and Louisiana. In Florida, Hurricane *Ian* in 2022 caused significant damage, and historical rainfall with flash flooding occurred in 2023. These events emphasise the importance of robust models in predicting flood-related losses.

For the *log* LF, at least one of our four shrinkage-based GLMs outperforms the benchmark `glm2` in every year and state. In particular, the St consistently delivers superior predictions during extreme events. For the *sqrt* LF, the performance varies by event; while our shrinkage models generally provide predictions comparable to or better than `glm2` in most years, the AMR falls below one for the 2019 extreme events in Texas and Louisiana. Nevertheless, the DSh remains a robust choice under other extreme conditions. Overall, Tables 9 and 10 indicate that the proposed shrinkage-based GLMs, especially the St and DSh models, achieve higher estimation accuracy than the standard `glm2` approach. Overall, the analysis of the real data shows that our non-parametric Stein-type shrinkage GLMs consistently produce lower MSEs than the benchmark GLM method across different LFs and states. These results indicate that the proposed methods may provide more reliable predictions for this flood insurance application.

7. Conclusions

The first two contributions of this paper are related to the introduction of novel shrinkage solutions to GLM modelling. We have proposed new IRLS implementations for GLM modelling by integrating several non-parametric shrinkage estimators into the IRLS algorithm. Our solutions are shown to reduce the estimation error without increasing the computational time. The third contribution of this paper is that we have introduced an optimisation-based approach to find enhanced starting values for GLM deployments. Such a solution helps overcome the common issues with the traditional IRLS method, such as its sensitivity to starting values and possible

[6] Available at: https://www.ncei.noaa.gov/access/billions/

Table 9: Gamma GLM with *sqrt* LF for Flood data

Year	n	AMR				CNT			
		SR	GSR	St	DSh	SR	GSR	St	DSh
Florida (FL)									
2014	2,352	0.9996	0.9951	0.8930	**1.0626**	22	0	0	**72**
2015	1,080	1.0014	1.0058	0.9704	**1.0456**	64	51	14	**81**
2016	5,044	0.9986	0.9985	0.9932	**1.0026**	0	1	0	**59**
2017	16,216	0.9998	0.9997	0.9950	**1.0043**	17	12	0	**90**
2018	2,864	1.0199	0.9988	1.0067	**1.4336**	78	58	79	**100**
2019	492	0.9999	0.9867	0.9306	**1.0526**	47	8	0	**82**
2020	6,322	0.9996	0.9991	0.9918	0.9848	**24**	6	0	0
2021	384	1.0251	1.1188	1.1889	**2.4489**	92	96	**100**	**100**
2022	31,061	0.9980	0.9998	0.9998	**1.0009**	3	30	26	**68**
2023	7,754	0.9955	1.0098	1.0187	**1.1426**	9	55	**100**	**100**
Texas (TX)									
2014	654	1.0003	0.9893	0.7971	**1.1199**	56	1	0	**86**
2015	8,972	0.9995	0.9990	0.9614	**1.0044**	25	0	0	**54**
2016	10,508	0.9992	**1.0008**	0.9977	0.9842	7	**77**	0	5
2017	61,211	0.9996	0.9999	0.9993	**1.0023**	33	43	17	**78**
2018	2,091	0.9886	0.9978	0.9953	0.9839	0	**29**	0	22
2019	8,998	0.9995	0.9998	0.9970	0.9866	10	**39**	0	0
2020	1,191	1.0070	0.9990	0.9965	**1.0561**	32	31	22	**49**
2021	1,237	0.9983	**1.0037**	1.0001	1.0036	10	**72**	48	57
2022	346	1.0037	1.1391	1.0293	**1.4331**	64	**100**	82	82
2023	285	1.0020	1.1167	2.0324	**3.1536**	71	**100**	**100**	**100**
Louisiana (LA)									
2014	486	0.9972	0.9830	0.9178	**1.1090**	0	0	0	**75**
2015	488	0.9939	0.9817	0.9017	**1.0403**	0	9	0	**62**
2016	26,704	0.9997	**1.0001**	0.9982	1.0001	13	**67**	0	57
2017	1,741	1.0014	1.0061	0.9961	**1.0094**	70	**77**	20	43
2018	346	1.0049	1.0551	1.0112	**1.0871**	60	**79**	57	62
2019	2,031	0.9983	0.9997	0.9931	0.9781	6	**46**	0	11
2020	2,737	**1.0004**	0.9988	0.9913	0.9824	**58**	23	0	24
2021	11,615	0.9998	**1.0001**	0.9977	0.9989	37	**55**	0	36
2022	123	1.0293	**1.0541**	1.0305	1.0231	**78**	53	67	46
2023	109	1.0000	0.9628	0.6973	0.9648	**48**	22	0	40

Panel A: Count Summary (FL, TX, LA)

		glm2	SR	GSR	St	DSh
2014-2023	**win glm2**	-	12	12	8	**22**
2014-2023	**best**	5	1	5	0	**19**

Notes: This table shows the AMR and the CNT of replications with an MSE ratio greater than one, comparing Gamma GLM with the *sqrt* LF between the benchmark `glm2` and our proposed models (SR, GSR, St, and DSh). For Florida 2015, out of 110 replications, all models converged 108 times; in other settings, all models converged in all 110 replications. After excluding the five highest and five lowest ratios, the AMR and CNT are calculated over the remaining 100 replications. Bold numbers in columns 3–6 indicate the model with the largest improvement in AMR for each setting, and bold numbers in columns 7–10 show the model with the highest CNT. Panel A at the bottom summarises the overall count of replications in which each model outperformed `glm2` and identifies the best overall model. All models used the same starting values from the optimisation-based procedure provided in Section 4.2, with a maximum of 250 iterations and a tolerance of 10^{-6}.

slow convergence.

Acknowledgements

This paper benefited from valuable feedback provided by participants at the 26th International Conference on Computational Statistics. The authors sincerely appreciate the constructive

Table 10: Gamma GLM with *log* LF for Flood data

Year	n	AMR				CNT			
		SR	GSR	St	DSh	SR	GSR	St	DSh
Florida (FL)									
2014	2,352	1.0003	1.0119	**1.0120**	0.7049	53	**100**	**100**	0
2015	1,080	0.9999	0.9927	**1.0818**	0.5939	42	43	**73**	5
2016	5,044	0.9991	1.0023	**1.2121**	0.6306	31	59	**90**	2
2017	16,216	1.0004	1.0011	**1.0498**	1.0125	69	49	**100**	81
2018	2,864	1.0038	1.0352	**1.9156**	1.3277	79	74	**100**	**100**
2019	492	1.0002	**1.1060**	0.7845	0.9167	48	**82**	23	37
2020	6,322	0.9994	1.0057	**1.1855**	0.8866	21	86	**100**	11
2021	384	1.0015	1.0318	3.9377	**11.8067**	53	57	**100**	98
2022	31,061	0.9999	1.0017	**1.2538**	1.0739	27	36	**100**	100
2023	7,754	1.0000	1.0130	1.1238	**1.1901**	52	**100**	**100**	**100**
Texas (TX)									
2014	654	1.0004	1.0208	1.0111	**1.0700**	60	**87**	62	52
2015	8,972	1.0000	**1.0041**	1.0038	0.8629	47	93	**100**	0
2016	10,508	1.0000	0.9982	**1.5168**	0.8082	58	36	**100**	0
2017	61,211	1.0002	0.9977	**1.0438**	0.9506	69	2	**100**	0
2018	2,091	1.0011	0.9999	**1.1147**	0.9720	57	52	**100**	43
2019	8,998	1.0001	1.0054	**1.0900**	0.9244	81	93	**100**	11
2020	1,191	1.0008	1.1014	**1.7522**	1.6257	54	91	**100**	**100**
2021	1,237	0.9996	1.0425	1.4041	**2.9765**	47	53	**100**	**100**
2022	346	1.0104	1.2986	8.9405	**12.7265**	81	91	**100**	**100**
2023	285	1.0041	2.2491	**57.4642**	35.8121	50	**100**	92	96
Louisiana (LA)									
2014	486	1.0008	0.9882	**1.0020**	0.9573	**73**	41	58	50
2015	488	1.0157	1.1516	**1.6037**	1.1029	**100**	88	**100**	55
2016	26,704	1.0003	0.9961	**1.0789**	0.9788	75	21	**100**	30
2017	1,741	1.0000	1.0185	**1.6185**	0.8984	47	65	**100**	18
2018	346	1.0026	1.1247	**1.6582**	1.3682	74	74	**84**	55
2019	2,031	1.0025	1.0318	**1.1489**	1.0099	75	82	**100**	34
2020	2,737	1.0002	0.9886	**1.2053**	0.9996	66	4	**100**	48
2021	11,615	1.0014	1.0078	**1.3132**	1.1159	99	92	**100**	**100**
2022	123	1.0091	1.2438	**1.8381**	1.7034	74	60	**84**	62
2023	109	1.0018	**1.1547**	0.8654	0.4590	**75**	69	32	5

Panel A: Count Summary (FL, TX, LA)

		glm2	SR	GSR	St	DSh
2014-2023	**win glm2**	-	25	23	**28**	15
2014-2023	**best**	0	0	3	**22**	5

Notes: This table shows the AMR and the CNT of replications with an MSE ratio greater than one, comparing Gamma GLM with the *log* LF between the benchmark `glm2` and our proposed models (SR, GSR, St, and DSh). Every model converged in all 110 replications across all settings. After excluding the five highest and five lowest ratios, the AMR and CNT are calculated over the remaining 100 replications. Bold numbers in columns 3–6 indicate the model with the largest improvement in AMR for each setting, and bold numbers in columns 7–10 show the model with the highest CNT. Panel A at the bottom summarises the overall count of replications in which each model outperformed `glm2` and identifies the best overall model. All models used the same starting values from the optimisation-based procedure provided in Section 4.2, with a maximum of 250 iterations and a tolerance of 10^{-6}.

comments and insightful discussions, which significantly contributed to refining this work.

The authors would like to express their sincere gratitude to Professor Rosalba Radice and Dr Dimitrina Dimitrova for their valuable guidance, thoughtful suggestions, and encouragement throughout the development of this paper.

References

Albert, A. and Anderson, J. A. (1984). On the existence of maximum likelihood estimates in Logistic regression models. *Biometrika*, 71(1):1–10.

Asimit, V., Badescu, A., Chen, Z., and Zhou, F. (2025a). Efficient and proper generalised linear models with power link functions. *Insurance: Mathematics and Economics*, 122(May):91–118.

Asimit, V., Cidota, M. A., Chen, Z., and Asimit, J. (2025b). Slab and shrinkage linear regression estimation. https://openaccess.city.ac.uk/id/eprint/35005/.

Baranchik, A. J. (1970). A family of minimax estimators of the mean of a multivariate normal distribution. *The Annals of Mathematical Statistics*, 41(2):642–645.

Bodnar, T., Gupta, A. K., and Parolya, N. (2016). Direct shrinkage estimation of large dimensional precision matrix. *Journal of Multivariate Analysis*, 146:223–236. Special Issue on Statistical Models and Methods for High or Infinite Dimensional Spaces.

Bodnar, T., Okhrin, O., and Parolya, N. (2019). Optimal shrinkage estimator for high-dimensional mean vector. *Journal of Multivariate Analysis*, 170:63–79. Special Issue on Functional Data Analysis and Related Topics.

Boyle, P., Flowerdew, R., and Williams, A. (1997). Evaluating the goodness of fit in models of sparse medical data: a simulation approach. *International journal of epidemiology*, 26(3):651–656.

Chatla, S. B. and Shmueli, G. (2018). Efficient estimation of Com–Poisson regression and a generalized additive model. *Computational Statistics & Data Analysis*, 121:71–88.

Chen, S. and Donoho, D. (1994). Basis pursuit. In *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 41–44. IEEE.

Chételat, D. and Wells, M. T. (2012). Improved multivariate normal mean estimation with unknown covariance when p is greater than n. *The Annals of Statistics*, 40(6):3137 – 3160.

Debón, A., Montes, F., and Puig, F. (2008). Modelling and forecasting mortality in Spain. *European Journal of Operational Research*, 189(3):624–637.

Delong, L., Lindholm, M., and Wüthrich, M. V. (2021). Making Tweedie's Compound Poisson model more accessible. *European Actuarial Journal*, 11(1):185–226.

do Nascimento, R. L., de Souza, R. M., and Cysneiros, F. J. d. A. (2024). Generalized linear models for symbolic polygonal data. *Knowledge-Based Systems*, 290:111569.

Donoho, D. and Montanari, A. (2016). High dimensional robust m-estimation: Asymptotic variance via approximate message passing. *Probability Theory and Related Fields*, 166:935–969.

El Karoui, N. (2013). Asymptotic behavior of unregularized and ridge-regularized high-dimensional robust regression estimators: rigorous results. arXiv preprint arXiv:1311.2445.

El Karoui, N., Bean, D., Bickel, P. J., Lim, C., and Yu, B. (2013). On robust regression with high-dimensional predictors. *Proceedings of the National Academy of Sciences*, 110(36):14557–14562.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360.

Field, A. P. and Wilcox, R. R. (2017). Robust statistical methods: A primer for clinical psychology and experimental psychopathology researchers. *Behaviour research and therapy*, 98:19–38.

Green, P. J. (1984). Iteratively Reweighted Least Squares for Maximum Likelihood Estimation, and Some Robust and Resistant Alternatives. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 46(2):149–170.

Hocking, R. R., Speed, F., and Lynn, M. (1976). A class of biased estimators in linear regression. *Technometrics*, pages 425–437.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Hooper, P. M. (1993). Iterative weighted least squares estimation in heteroscedastic linear models. *Journal of the American Statistical Association*, 88(421):179–184.

Ibrahim, J. G., Lipsitz, S. R., and Chen, M.-H. (1999). Missing covariates in generalized linear models when the missing data mechanism is non-ignorable. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(1):173–190.

James, W., Stein, C., et al. (1961). Estimation with quadratic loss. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 361–379. University of California Press.

Kapre, R., Zhou, J., Li, X., Beckett, L., and Louie, A. Y. (2020). A novel gamma GLM approach to MRI relaxometry comparisons. *Magnetic resonance in medicine*, 84(3):1592–1604.

Koh, K., Kim, S.-J., and Boyd, S. (2007). An interior-point method for large-scale l1-regularized Logistic regression. *Journal of Machine learning research*, 8(Jul):1519–1555.

Lachmann, J., Storvik, G., Frommlet, F., and Hubin, A. (2022). A subsampling approach for Bayesian model selection. *International Journal of Approximate Reasoning*, 151:33–63.

Ledoit, O. and Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411.

Ledoit, O. and Wolf, M. (2012). Nonlinear shrinkage estimation of large-dimensional covariance matrices. *The Annals of Statistics*, 40(2):1024 – 1060.

Li, G., Lian, H., Feng, S., and Zhu, L. (2013). Automatic variable selection for longitudinal generalized linear models. *Computational Statistics & Data Analysis*, 61:174–186.

Mäkeläinen, T., Schmidt, K., and Styan, G. (1981). On the existence and uniqueness of the maximum likelihood estimate of a vector-valued parameter in fixed-size samples. *Annals of Statistics*, 9(4):758–567.

Marra, G. and Radice, R. (2017). Bivariate copula additive models for location, scale and shape. *Computational Statistics & Data Analysis*, 112:99–113.

Marschner, I. (2011). glm2: Fitting generalized linear models with convergence problems. *The R Journal*, 3(2):12–15.

McCullagh, P., Nelder, J., and Wedderburn, R. (1989). *Generalized Linear Models*. Second ed., Chapman and Hall/CRC.

Mouatassim, Y. and Ezzahid, E. H. (2012). Poisson regression and zero-inflated Poisson regression: application to private health insurance data. *European actuarial journal*, 2(2):187–204.

Muggeo, V. M. and Ferrara, G. (2008). Fitting generalized linear models with unspecified link function: A p-spline approach. *Computational Statistics & Data Analysis*, 52(5):2529–2537.

Nelder, J. and Wedderburn, R. (1972). Generalized linear models. *Journal of the Royal Statistical Society: Series A*, 135(3):370–384.

Peters, G. W., Shevchenko, P. V., and Wüthrich, M. V. (2009). Model uncertainty in claims reserving within Tweedie's Compound Poisson models. *ASTIN Bulletin: The Journal of the IAA*, 39(1):1–33.

Sakate, D. and Kashid, D. (2014). A deviance-based criterion for model selection in GLM. *Statistics*, 48(1):34–48.

Scallan, A., Gilchrist, R., and Green, M. (1984). Fitting parametric link functions in generalised linear models. *Computational Statistics & Data Analysis*, 2(1):37–49.

Schäfer, J. and Strimmer, K. (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical applications in genetics and molecular biology*, 4(1):Article 32.

She, Y. (2009). Sparse regression with exact clustering. *Electronic Journal of Statistics*, 39(3):1360–1392.

Sohn, M. B. and Li, H. (2018). A GLM-based latent variable ordination method for microbiome samples. *Biometrics*, 74(2):448–457.

Stein, C. (1956). Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, volume 4, pages 197–207. University of California Press.

Stein, C. (1960). Multiple regression contributions to probability and statistics. *Essays in Honor of Harold Hotelling*, 103.

Stein, C. M. (1981). Estimation of the Mean of a Multivariate Normal Distribution. *The Annals of Statistics*, 9(6):1135 – 1151.

Thompson, R. and Baker, R. (1981). Composite link functions in generalized linear models. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 30(2):125–131.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288.

Tibshirani, R. J. and Taylor, J. (2011). The solution path of the generalized lasso. *The Annals of Statistics*, 39(3):1335–1371.

Tikhonov, A. N. (1963). On the solution of ill-posed problems and the method of regularization. In *Doklady akademii nauk*, volume 151, pages 501–504. Russian Academy of Sciences.

Villegas, C., Paula, G. A., Cysneiros, F. J. A., and Galea, M. (2013). Influence diagnostics in generalized symmetric linear models. *Computational Statistics & Data Analysis*, 59:161–170.

Wang, C., Tong, T., Cao, L., and Miao, B. (2014). Non-parametric shrinkage mean estimation for quadratic loss functions with unknown covariance matrices. *Journal of Multivariate Analysis*, 125:222–232.

Wedderburn, R. (1976). On the existence and uniqueness of the maximum likelihood estimates for certain generalized linear models. *Biometrika*, 63(1):27–32.

Wood, S. N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 73(1):3–36.

Wood, S. N. (2017). *Generalized additive models: an introduction with R*. Chapman and Hall/CRC.

Xia, F., Chen, J., Fung, W. K., and Li, H. (2013). A Logistic normal multinomial regression model for microbiome compositional data analysis. *Biometrics*, 69(4):1053–1063.

Yee, T. W. and Stephenson, A. G. (2007). Vector generalized linear and additive extreme value models. *Extremes*, 10:1–19.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2):301–320.

Appendix A. Summary of Supplementary Material

This supplementary material complements our paper by providing additional technical details about the topics discussed in the paper, a full description of the data generation process and how the real data have been pre-processed, but also further empirical evidence that supplements the evidence provided in the main paper. That is, Appendix B outlines the standard IRLS, Appendix C explains the data generation process used to produce the synthetic data used in this

paper, while Appendix D further evidence about the lack of convergence in GLM deployment Finally, Appendix E describes the flood insurance data, which are the real data used in our numerical experiments.

Appendix B. Fitting GLM with the IRLS Procedure

This section outlines the implementation of the IRLS algorithm for fitting GLMs. A succinct description is provided and the unfamiliar reader may wish to find further details in the standard GLM literature (Wood, 2017) though our description is sufficient to understand the GLM implementation we propose in this paper.

The starting point is maximising the log-likelihood in (2.3), which implies finding the corresponding stationary points:

$$\sum_{i=1}^{n} \frac{\omega_i(y_i - \mu_i)}{V(\mu_i)} \frac{\partial \mu_i}{\partial \beta_j} = 0, \quad \text{for all } j \in \{0, \dots, p\},$$

where ω_i are some exogenous weights if available, otherwise, all are assumed to be equal to 1; note that $\mu_i = h(\eta_i)$ and $\eta_i = \mathbf{x}_i^\top \boldsymbol{\beta}$, while $V(\mu_i)$ is the variance function. These equations are equivalent to minimising an WLS-like instance given by

$$\mathcal{S} = \sum_{i=1}^{n} \frac{(y_i - \mu_i)^2}{V(\mu_i)},$$

which is solved iteratively. At iteration $k \geq 0$, the following WLS problem is solved

$$\widehat{\boldsymbol{\beta}}^{(k+1)} = \arg\min_{\boldsymbol{\beta}} \left\| \sqrt{\mathbf{W}^{(k)}} \left(\mathbf{z}^{(k)} - \mathbf{X}\boldsymbol{\beta} \right) \right\|^2,$$

where pseudodata $\mathbf{z}^{(k)}$ and weight matrix $\mathbf{W}^{(k)}$ are

$$z_i^{(k)} = \eta_i^{(k)} + \frac{y_i - \mu_i^{(k)}}{h'\left(\eta_i^{(k)}\right)}, \quad W_{ii}^{(k)} = \frac{\left(h'\left(\eta_i^{(k)}\right)\right)^2}{V\left(\mu_i^{(k)}\right)} \quad \text{for all } j \in \{1, \dots, n\}, \tag{B.1}$$

with $\mu_i^{(k)} = h(\eta_i^{(k)})$ and h' being the derivative of the inverse LF.

In summary, the IRLS algorithm proceeds as follows:

1. **Initialisation**: Set starting values $\mu_i^{(0)} = y_i$ and $\eta_i^{(0)} = h^{-1}(y_i)$, adjusting if necessary to ensure valid choices (e.g., $\mu_i^{(0)} > 0$ for *log* LF). Compute $\mathbf{z}^{(0)}$ and $\mathbf{W}^{(0)}$ via (B.1), and solve the initial WLS

$$\widehat{\boldsymbol{\beta}}^{(0)} = \left(\mathbf{X}^\top \mathbf{W}^{(0)} \mathbf{X} \right)^{-1} \mathbf{X}^\top \mathbf{W}^{(0)} \mathbf{z}^{(0)}.$$

2. **Iteration**: For each iteration $k \geq 0$,
   (a) Update $\eta_i^{(k)} = \mathbf{x}_i^\top \widehat{\boldsymbol{\beta}}^{(k)}$ and $\mu_i^{(k)} = h(\eta_i^{(k)})$, and compute $\mathbf{z}^{(k)}$ and $\mathbf{W}^{(k)}$ via (B.1).

(b) Solve the WLS instance to update the parameters' estimates

$$\widehat{\boldsymbol{\beta}}^{(k+1)} = \left(\mathbf{X}^\top \mathbf{W}^{(k)} \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{W}^{(k)} \mathbf{z}^{(k)}.$$

(c) The iterative process terminates when the chosen convergence criterion is met. Note that the default stopping criteria differ between the implementations in **R**, **MATLAB**, and **Python**.

R (`glm.fit2`): The default convergence is based on the relative change in deviance:

$$\frac{\left|\text{Dev}^{(k+1)} - \text{Dev}^{(k)}\right|}{0.1 + \left|\text{Dev}^{(k+1)}\right|} < \tau,$$

where τ (default, e.g., 10^{-8}) is a small tolerance. Step-halving is applied if the deviance increases.

MATLAB (`fitglm`): The default convergence check monitors the change in regression coefficients:

$$\max_i \left|\beta_i^{(k+1)} - \beta_i^{(k)}\right| \leq \tau \times \max\left(\sqrt{\epsilon}, \max_i \left|\beta_i^{(k)}\right|\right),$$

where default τ is typically set to 10^{-6}, and $\epsilon \approx 2.2204 \times 10^{-16}$.

Python (`statsmodels.GLM`): The default convergence is determined by the absolute change in deviance:

$$\left|\text{Dev}^{(k+1)} - \text{Dev}^{(k)}\right| \leq \tau,$$

with default τ is usually chosen to be 10^{-8}.

Note that the variance functions $V(\mu)$ depends on the distributional assumptions; e.g., $V(\mu) = \mu(1-\mu)$ for LR, $V(\mu) = \mu$ for Poisson GLM, and $V(\mu) = \mu^2$ for Gamma GLM. The choice of LF determines h^{-1} and h', which are needed in deploying IRLS.

Appendix C. Data Generation Process

This section describes how data are generated in this paper. We use two variants, namely, DGP1 and DGP2. The DGP2 setting is the same as DGP1 except that all coefficients are made positive, i.e. the "true" model parameters are chosen as $\beta_j^{DGP2} = |\beta_j^{DGP1}|$. Therefore, we describe only DGP1, which we do for the three GLM models (LR, Poisson, and Gamma) and various LF choices. Specifically, we only consider the canonical LF for LR, which is the *logit* LF, i.e., $g(\mu) = \log\left(\mu/(1-\mu)\right)$; its inverse LF is $h(\eta) = \left(1 + e^{-\eta}\right)^{-1}$. Further, we include *log* LF $g(\mu) = \log(\mu)$ and *square root* LF $g(\mu) = \sqrt{\mu}$ for Poisson and Gamma GLMs; their corresponding inverse LFs are $h(\eta) = e^\eta$ and $h(\eta) = \eta^2$, respectively, which are defined on the entire real line as the linear predictor η. Note that the canonical LF for Poisson GLM is the *log* LF, which is considered in our paper, but we do not include the canonical LF for Gamma GLM which is improper; for details, see e.g. (Asimit et al., 2025a).

We now provide the details about DGP1 for each LR, Poisson, and Gamma GLMs and clarify the data generation corresponding to the chosen LF. Note that our implementations do not question whether we choose the "right" LF, and thus, LF selection is not the purpose of our analyses. Simply speaking, the "true" LF is used in the GLM deployment so that we evaluate the estimation error purely from the IRLS' perspective which is the fairest way. Establishing that our IRLS solver is "better" than the standard IRLS solver gives us confidence to use our solver for research questions such as LF selection, penalised GLM to reduce overfitting, optimal subset from the covariates' space, etc.

Step 1: Generate the covariate matrix $\mathbf{X} = \{X_{i,j}\}_{i=1,j=1}^{n\times p}$ from a multivariate normal distribution with mean zero, unit variances and structured correlation matrix Σ. The off-diagonal elements of Σ are defined such that $\text{Cov}(X_{a,j}, X_{b,j}) = \rho^{|a-b|}$, where $-1 < \rho < 1$ controls the strength of dependence. This means that Σ is a Toeplitz matrix.

Step 2: Define the regression coefficients β_j^{DGP1} for all $j \in \{0, \ldots, p\}$:

(a) For LR with *logit* LF, and Poisson/Gamma GLMs with *sqrt* LF, we use alternating signs and increasing magnitudes, i.e., $1, -1, 2, -2, \ldots$.

(b) For Poisson and Gamma GLMs with *log* LF, we use:

$$\beta_j^{DGP1} = (-1)^j \cdot 0.1 \cdot 0.95^{\lceil j/2 \rceil}, \quad \text{for all } j \in \{0, \ldots, p\},$$

to ensure the responses stay within reasonable ranges and avoid numerical issues during IRLS procedures. Using setting (a) for these GLMs would make the response's conditional mean to explode numerically, which would be unfeasible synthetic data for any GLM solver.

Step 3: For each $i \in \{0, \ldots, n\}$, compute the linear predictor $\eta_i = \beta_0^{DGP1} + \sum_{j=1}^{p} \beta_j^{DGP1} x_{i,j}$ and generate the response variable Y_i as follows:

(a) For LR, generate $Y_i \sim \textit{Binomial}(1, (1 + e^{-\eta})^{-1})$. We first generate a large sample (e.g., $5,000$ observations), and then select samples of 500 based on the desired proportions between the two states ($Y = 0$ and $Y = 1$) so that the response variable is balanced/imbalanced as desired.

(b) For Gamma GLM, generate $Y_i \sim \textit{Gamma}(\mu_i, 1)$ where $\mu_i = \eta_i^2$ and $\mu_i = e^{\eta_i}$ for *sqrt* LF and *log* LF, respectively.

(c) For Poisson GLM, generate $Y_i \sim \textit{Poisson}(\mu_i)$ where $\mu_i = \eta_i^2$ and $\mu_i = e^{\eta_i}$ for *sqrt* LF and *log* LF, respectively.

Appendix D. Simulation Convergence Failures in GLMs with *sqrt* LF

Table D.11 is complementary to Table 2 and reports the number of convergence failures for Poisson and Gamma GLMs with a *sqrt* LF and DGP2 setup in Appendix C. Results are

based on $N = 100$ replications with a sample size $n = 500$, under varying correlation values ρ and covariate-to-sample size ratios p/n. We compare three software implementations – **R**, **Matlab**, and **Python** – by using their default starting values (values outside parentheses) and the proposed optimisation-based starting values (values inside parentheses) as explained in Section 4.2.

We now discuss the results in Table 2. For Poisson GLM in Panel A, the convergence failures mainly occur when the mean parameter $\mu = 0$, particularly in **R** and **Matlab**. Using the optimisation-based starting values eliminates these failures in **R** and slightly improves performance in **Matlab**, while **Python** shows no failures in any setting. At higher mean values ($\mu = 3$ and 5), all methods achieve full convergence regardless of the starting values. For Gamma GLM in Panel B, the convergence issues are more common, especially when $\mu = 0$. The optimisation-based starting values substantially reduce failures in **R** and **Python** and lead to slight improvements in some **Matlab** cases. As for Poisson GLM, the cases in which $\mu = 3$ or 5 always converge for all solvers. Overall, the optimisation-based starting values lead to improved or equal convergence rates as compared to the default solvers in **R**, **Matlab**, and **Python**.

Table D.12 is complementary to Table 5 and reports the number of convergence failures across $N = 250$ replications with a sample size $n = 500$ for Poisson and Gamma GLMs fitted with a *sqrt* LF and DGP1 setup in Appendix C. All results use only our optimisation-based starting values and are presented across different correlation levels ρ and covariate-to-sample size ratios p/n. For Poisson GLM in Panel A, the convergence is consistently achieved by GSR, St, and DSh across all scenarios; in contrast, `glm2` and SR show occasional convergence failures, mostly in high-dimensional settings (i.e., $p/n = 50\%$) or under strong correlation. For Gamma GLM in Panel B, convergence is slightly more challenging, and we observe that solvers based on GSR and DSh perform well with only a few failures; SR and St methods show slightly more failures under some combinations of high p/n ratios and extreme ρ values, while the default `glm2` solver also experiences convergence issues under these same conditions. Overall, the optimisation-based starting values lead to slight improvement or equal convergence rates when applied to our shrinkage regression as compared to the default solver in **R**.

Appendix E. Description of Flood Data Insurance

This section describes the flood insurance data that have one dependent variable (Y) where $Y = X_1/X_2$ and 15 covariates (X_2 to X_{16}) that provide information on financial aspects, structural details, and the location of insured properties; further details are in Table E.13. Variable Y represents the proportion of insurance coverage used in claims and serves as a measure to assess the adequacy of flood insurance coverage. To address the influence of extreme values, Y was capped at its 99^{th} percentile.

Data have been preprocessed following insurance industry standard procedures and details can be found in (Asimit et al., 2025a), which include bounding numerical covariates, reducing the number of categories for some categorical variables, and the usual one-hot-encoding. The most important pre-processing aspects are as follows: i) examples with only positive values for X_2,

Table D.11: Number of failures to convergence with *sqrt* LF DGP2

p	**1% n**			**10% n**		
ρ	**-0.5**	**0**	**0.5**	**-0.5**	**0**	**0.5**
Panel A: Poisson Distribution						
μ**=0**						
R	**22 (0)**	**2 (0)**	0 (0)	**3 (0)**	0 (0)	**1 (0)**
Matlab	90 (90)	62 (62)	51 (51)	10 (10)	10 (10)	11 (11)
Python	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
μ**=3**						
R	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
Matlab	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
Python	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
μ**=5**						
R	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
Matlab	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
Python	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
Panel B: Gamma Distribution						
μ**=0**						
R	**100 (2)**	**100 (2)**	**100 (2)**	**100 (0)**	**100 (0)**	**100 (0)**
Matlab	**100 (99)**	100 (100)	100 (100)	99 (100)	96 (96)	**69 (68)**
Python	**99 (71)**	**100 (98)**	**100 (99)**	**99 (98)**	**98 (93)**	95 (98)
μ**=3**						
R	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
Matlab	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
Python	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
μ**=5**						
R	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
Matlab	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
Python	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)

Notes: This table shows the number of convergence failures for `glm2` over $N = 100$ replications, each with $n = 500$ observations. The numbers outside parentheses are for the default starting values within **R**, **Matlab** and **Python**, and the numbers inside parentheses are for our optimisation-based adjusted starting values described in Section 4.2. Bold numbers indicate cases where our starting values produced fewer failures than the default. A convergence failure means that the algorithm does not reach a valid solution within a maximum of $10,000$ iterations. The results are reported for different mean parameters μ and predictor correlations ρ at sample proportions of 1% and 10% of the total sample size. Panel A shows results for the Poisson distribution, while Panel B shows results for the Gamma distribution.

X_3, and X_4 are retained to avoid numerical issues during logarithmic transformation; ii) X_7, originally a character field for deductible codes, was converted to numeric values on descriptions provided in the *FEMA NFIP Claims* Dataset and restricted to be strictly positive; iii) X_6 is capped at its 99^{th} quantile to handle outliers and is lower bounded by 0 (including 0); iv) X_{15} and X_{16} are used together and grouped into clusters to represent the location of buildings; v) categorical variables such as X_{10}, X_{11} and X_{14} are reduced to two groups based on their meaning so that very low-frequency categories are avoided; vi) X_{12} is converted to building's age in years and we group these into three categories.

Table D.12: Convergence Performance For PoR and GaR with *sqrt* LF

Panel A: Poisson Distribution																				
	ρ = -0.75				ρ = -0.5				ρ = 0				ρ = 0.5				ρ = 0.75			
p/n	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**
GLM2	0	0	0	2	0	0	0	5	0	0	0	4	0	0	0	2	0	1	0	2
SR	0	0	0	2	0	0	0	5	0	0	0	5	0	0	0	3	0	0	0	3
GSR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
St	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DSh	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Panel B: Gamma Distribution																				
	ρ = -0.75				ρ = -0.5				ρ = 0				ρ = 0.5				ρ = 0.75			
p/n	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**	**1%**	**10%**	**25%**	**50%**
GLM2	3	1	1	1	1	0	2	1	3	2	0	1	4	1	2	0	1	2	0	1
SR	3	0	1	1	1	0	2	0	4	2	0	0	4	1	0	1	1	0	0	3
GSR	3	0	1	0	1	0	0	0	3	2	0	0	4	1	0	0	1	0	0	1
St	3	1	2	3	1	0	2	4	3	2	1	0	4	1	0	0	1	0	0	2
DSh	3	0	1	0	1	0	2	0	3	2	0	0	4	1	0	0	1	0	0	1

Notes: This table reports the number of convergence failures for the PoR and GaR using the *sqrt* LF across different correlation coefficients ρ and covariate-to-sample size ratios p/n. Each entry shows how many out of $N = 250$ replications with $n = 500$ failed to reach a valid solution within a maximum of 250 iterations, using the optimisation-based procedure described in Section 4.2 to adjust starting values. Panel A presents results for the Poisson L_2 error results; see Table 5.

Table E.13: Summary of Data Preprocessing for Flood Insurance Dataset

Variable	**Type**	**Regrouped**	**Bounded**	**One-hot Encoded**	**Resulting Columns**
Y – **ratioCoverage**	Numeric	No	Yes	No	N/A
X_1 – **amountPaidOnBuildingClaim**	Numeric	No	Yes	No	N/A
X_2 – **totalBuildingInsuranceCoverage**	Integer	No	Yes	No	N/A
X_3 – **buildingPropertyValue**	Numeric	No	Yes	No	N/A
X_4 – **buildingDamageAmount**	Integer	No	Yes	No	N/A
X_5 – **numberOfFloorsInTheInsuredBuilding**	Integer	No	No	No	N/A
X_6 – **waterDepth**	Integer	No	Yes	No	N/A
X_7 – **buildingDeductibleCode**	Character	No	Yes	No	N/A
X_8 – **elevatedBuildingIndicator**	Integer	No	No	Yes	2
X_9 – **postFIRMConstructionIndicator**	Character	No	No	Yes	2
X_{10} – **ratedFloodZone**	Character	Yes	No	Yes	2
X_{11} – **buildingDescriptionCode**	Integer	Yes	No	Yes	2
X_{12} – **originalConstructionDate**	Character	Yes	No	Yes	3
X_{13} – **replacementCostBasis**	Character	Yes	No	Yes	2
X_{14} – **causeOfDamage**	Character	Yes	No	Yes	2
X_{15} – **latitude**	Numeric	Yes	No	Yes	4
X_{16} – **longitude**	Numeric	Yes	No	Yes	4

Notes: This table summarises the dependent variable and features used in the US flood insurance dataset. The last column, *Resulting Columns*, indicates the number of new variables created after one-hot encoding or clustering; numeric variables that are not transformed "N/A" is shown.