GLM Solutions via Shrinkage*

Vali Asimit[1]
Professor of Actuarial Analytics

Oana Avramescu[2]
Senior Manager, Global Insurance Solutions Leaders

Ziwei Chen[1,*]
PhD Candidate

Diego Rivas[2]
Senior Product Manager

Claudio Senatore[2]
Senior Global Solution Leader, Insurance Solutions

Abstract

Generalised Linear Models are a core tool for modelling non-normal data and are routinely fitted via the Iteratively Reweighted Least Squares algorithm. While this algorithm is computationally attractive, its performance can be hindered by convergence issues, sensitivity to starting values, and substantial estimation error. This paper develops a practical enhancement of the Iteratively Reweighted Least Squares algorithm by replacing the standard least squares step with Stein-type shrinkage estimators, which form the mathematical foundation of the well-known actuarial concept of credibility premiums. These estimators reduce the theoretical mean squared error by introducing a controlled bias that significantly reduces variance, without requiring cross-validation or increasing computational complexity. As a result, the proposed approach provides a scalable and efficient alternative to commonly used penalised generalised linear modelling. In addition, we propose an optimisation-based strategy for selecting starting values, which improves the stability and convergence of both standard and shrinkage-based Iteratively Reweighted Least Squares implementations. The paper is not primarily methodological; instead, it focuses on the practical deployment of generalised linear modelling. Extensive numerical experiments, covering a wide range of settings relevant to practitioners, together with real-data applications, demonstrate consistent improvements in accuracy, stability, and computational efficiency over the industry standard benchmarks.

Keywords: Generalised Linear Model, Shrinkage Estimation, Iteratively Reweighted Least Squares.

JEL classification: C13, C25, C53

*Ziwei Chen is corresponding author. (1) Bayes Business School, City St George's, University of London, London, UK. (2) SAS. Email addresses: asimit@citystgeorges.ac.uk, Oana.Avramescu@sas.com, ziwei.chen.3@citystgeorges.ac.uk, Diego.Rivas@sas.com, Claudio.Senatore@sas.com.

1 Motivation and Contributions

Generalised linear models (GLMs) extend traditional linear regression methods to settings where the dependent variable follows a non-Gaussian distribution, allowing for a flexible relationship between the mean of the response and a linear predictor via a *link function (LF)* (Nelder and Wedderburn, 1972; McCullagh et al., 1989; Wüthrich and Merz, 2023). Owing to their interpretability and adaptability, GLMs have become a fundamental tool across a wide range of disciplines, including economics, epidemiology, engineering, and the social sciences. In actuarial science and insurance, GLMs play a central role in modelling claim frequencies and severities, non-life insurance pricing, and risk assessment, where common specifications include Poisson and negative binomial models for claim counts and Gamma models for claim sizes, while Tweedie models are often used for total claim amounts (Denuit et al., 2007; Frees, 2014). Key actuarial applications include *ratemaking*, where GLMs estimate expected losses to set premiums; *reserving*, where GLMs provide a flexible stochastic framework for modelling claim development over time at a granular, cell-level scale; and *risk classification*, where GLMs help identify and segment policyholders by underlying risk. This combination of flexibility, interpretability, and well-established statistical guarantees has made GLMs a standard methodology in both actuarial research and practice (Debón et al., 2008; Peters et al., 2009; Mouatassim and Ezzahid, 2012; Delong et al., 2021).

The most common approach to fitting GLMs is the *Iteratively Reweighted Least Squares (IRLS)* algorithm, which repeatedly updates the weights and solves a *Weighted Least Squares (WLS)* problem until convergence. Despite its widespread use, IRLS has several limitations. *First*, it may fail to converge in certain situations (Marschner, 2011). *Second*, the accuracy of WLS estimates is affected by instability in the precision matrix (the inverse of the covariance matrix), arising from bias in empirical eigenvalue estimators (Muirhead, 1987; Bai et al., 2010; Asimit et al., 2026). This explains the non-zero asymptotic *Mean Square Error (MSE)* of the *Ordinary Least Squares (OLS)* estimator in high-dimensional settings where both the sample size and the number of covariates grow large, in modern insurance datasets, where the number of rating factors and their interactions can be large relative to the available exposure, this asymptotic degradation of OLS is practically relevant, not a theoretical artefact[1] (El Karoui et al., 2013; El Karoui, 2013; Donoho and Montanari, 2016; Asimit et al., 2026). In practical terms, this instability manifests as excessive sensitivity of the estimated rating factors to small perturbations in the data—a well-known concern in ratemaking, where coefficient stability across model iterations is an important operational requirement. *Third*, IRLS is sensitive to starting values, and poor initial guesses can result in many iterations, suboptimal solutions, or even complete convergence failure (Green, 1984; Marschner, 2011).

The motivation of this paper is to address key limitations of GLM estimation discussed above. *Primarily*, we focus on improving OLS/WLS estimation using the shrinkage methods introduced in Asimit et al. (2026), which have been shown to outperform OLS in real data applications. As these estimators are repeatedly applied within IRLS procedures, they are expected to enhance the overall performance of the standard IRLS solution. As a secondary contribution, we propose an optimisation approach to generate improved starting values for IRLS. While this is

[1]Classical statistics states that with a fixed number of covariates and large sample size, the OLS estimator's MSE is negligible. In the big data era, this no longer holds, as both the number of covariates and the sample size can be large.

an important aspect of GLM implementation, we keep the computational details to a minimum and provide them in the online appendix, as they are primarily relevant for users implementing GLMs from scratch. This paper is not methodological in nature; rather, it focuses on practical GLM applications. In particular, we aim to demonstrate the benefits of shrinkage estimators in GLM modelling through extensive empirical evidence, providing guidance for both applied researchers and practitioners, and making the methodology directly deployable in standard actuarial workflows without additional cross-validation overhead.

Our contribution is threefold. *First*, we enhance the IRLS algorithm by replacing the standard linear regression estimator with the shrinkage estimators proposed in Asimit et al. (2026). Both simulation and real data analyses demonstrate that the resulting shrinkage-based GLM solutions outperform standard and penalised GLM benchmarks. *Second*, unlike penalised GLM approaches, the proposed shrinkage methods do not require cross-validation, making them computationally more efficient while achieving comparable or improved accuracy. All estimators are implemented in our **R** CRAN package `savvyGLM`[2], which is freely available for use and experimentation. *Third*, we propose an optimisation-based approach for selecting starting values, improving convergence for both standard IRLS and the shrinkage-based methods. Before concluding this section and outlining the structure of the paper, we clarify the notion of shrinkage used in this work. In common actuarial usage, *shrinkage* typically refers to regularisation by penalised methods such as Ridge or LASSO, which constrain model parameters toward zero to prevent overfitting. The notion of shrinkage in this paper is fundamentally different: following Stein (1956) and James et al. (1961), we shrink the *covariance structure* of the estimator—not the coefficients themselves—to achieve a theoretically optimal reduction in mean squared error, in closed form, so with shrinkage parameters derived analytically rather than through computationally intensive tuning procedures such as cross-validation. Practitioners familiar with credibility theory will recognise the underlying intuition: a weighted blend between an individual estimate and a stable prior target can outperform the individual estimate alone, even at the cost of a small controlled bias (Bühlmann, 1967).

The most common general-purpose GLM **R** CRAN packages are (i) `glm2` (Marschner, 2011), which relies on IRLS, and (ii) `glmnet` (Friedman et al., 2010), the industry-standard package designed to handle large datasets, including settings with high-dimensional or sparse covariates where variable selection is important. The latter implements penalised models, widely used in statistics and machine learning, which typically require computationally intensive cross-validation procedures. Such penalisation approaches are often referred to as shrinkage methods, since they shrink model parameters towards zero. However, this should not be confused with the notion of shrinkage considered in this paper, which is rooted in Stein's paradigm (Stein, 1956, 1960; James et al., 1961). Stein-type shrinkage estimators introduce bias in a controlled manner to reduce estimation error (in terms of theoretical MSE), achieving an optimal bias–variance trade-off in closed form without the need for cross-validation. While both penalised methods and Stein-type shrinkage estimators aim to improve estimation accuracy by reducing the overall variability of the GLM output, the latter provide computationally efficient and scalable alternatives to standard penalised approaches, making them strong competitors to current industry-standard solutions provided by the `glmnet` **R** CRAN package.

While the IRLS algorithm (detailed in Section SM.2.4) is the main solver for non-penalised GLMs, other algorithms are also available. Newton's Method (Section SM.2.2) and the Fisher

[2]Available at: https://cran.r-project.org/web/packages/savvyGLM/index.html

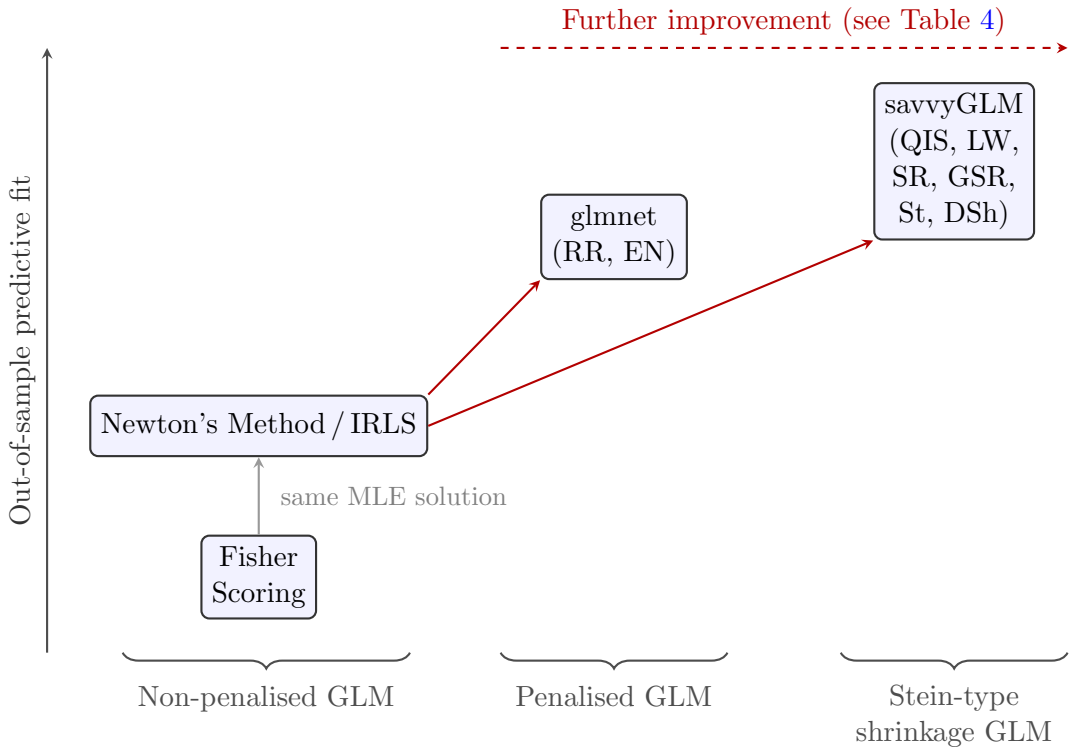


Figure 1: Diagram illustrating the main empirical findings (Tables 4 and SM.2.1). Vertical position reflects relative out-of-sample predictive fit on FrenchMTPL2. The dashed arrow indicates the direction of improvement demonstrated empirically in the numerical section.

Scoring Method (Section SM.2.3) provide alternative approaches for estimating non-penalised GLMs. Figure 1 summarises our conclusions based on implementations on reasonably sized real datasets. We *first* find that IRLS and Newton's Method exhibit similar performance and both clearly outperform the Fisher Scoring Method. *Second*, IRLS can be further improved through penalisation, and the `glmnet` **R** CRAN package provides the best off-the-shelf implementation. *Third*, we find that this can be further improved using the Stein-type shrinkage GLM introduced in this paper, for which our newly developed `savvyGLM` **R** CRAN package is available for free use.

The remainder of the paper is organised as follows. Section 2 introduces the Stein-type shrinkage estimators for multiple linear regression and clarifies their fundamental differences from well-known penalised regression approaches, to avoid potential confusion. Section 3 develops the proposed method for selecting starting values. Section 4 presents numerical results comparing standard non-penalised and penalised GLM solutions with the proposed Stein-type approach on both simulated and real datasets. Section 5 concludes the paper. Additional background material and further numerical results are provided in the Supplementary Material.

2 Overview of Shrinkage Linear Regression Estimators

Because IRLS reduces GLM estimation to a sequence of WLS problems, any improvement to the WLS step propagates directly to the GLM output; this is the key insight motivating the

estimators introduced in this section. This section introduces the Stein-type linear regression shrinkage estimators used in each linear regression step within the IRLS algorithm, detailed in Section SM.2.4, and compares them with well-known penalised linear regression estimators. The four Stein-type shrinkage estimators—*(simple) Slab Regression (SR)*, *Generalised Slab Regression (GSR)*, *Stein Estimator (St)*, and *Diagonal Shrinkage (DSh)*—have been studied in (Asimit et al., 2026) and are motivated by Stein's paradox: when estimating a Gaussian mean vector of dimension $p \geq 3$, the MLE is inadmissible—meaning that a shrinkage estimator exists which achieves strictly lower mean squared error regardless of the true parameter values. This global dominance result, as counter-intuitive as it is consequential, motivates the estimation paradigm adopted throughout this paper. Before presenting the advantages of Stein-type shrinkage in Section 2.2, we first discuss classical penalised estimators in Section 2.1, focusing on the role of eigenvalues in reducing estimation error in linear regression, a perspective that, to the best of our knowledge, has not been explored in the machine learning and statistical literature.

2.1 Penalised Linear Regression

Penalised linear regression is a widely studied technique in multivariate analysis for producing accurate and/or parsimonious prediction models, and is motivated by the so-called Tikhonov penalisation (Tikhonov, 1963). Specifically, for a response vector $\mathbf{y} \in \mathbb{R}^n$, covariates matrix $\mathbf{X} \in \mathbb{R}^{n\times(p+1)}$corresponding to p covariates, and a penalty function $g : \mathbb{R}^{p+1} \to \mathbb{R}_+$, where the first column is a vector of ones, the estimator is defined as:

$$\widehat{\boldsymbol{\beta}} := \underset{\boldsymbol{\beta}\in\mathbb{R}^{p+1}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + g(\boldsymbol{\beta}), \tag{1}$$

where $\|\cdot\|_p$ denotes the usual p-norm. For example, $\lambda\|\boldsymbol{\beta}\|_1$ in *LASSO* encourages sparsity, while $\lambda\|\boldsymbol{\beta}\|_2^2$ in *Ridge regression (RR)* reduces overfitting; the *elastic net (EN)* combines both to achieve a sparse and accurate model (Zou and Hastie, 2005). Such penalisation techniques are widely used in machine learning and statistical applications, and their effectiveness is well established. A practical challenge of these methods is the increased computational cost due to cross-validation, which becomes burdensome for large datasets. In contrast, Stein-type shrinkage estimators do not require cross-validation, as their tuning parameters are computed in closed form. It should be noted that Stein-type shrinkage estimators are designed to reduce estimation error through bias-variance trade-off, rather than to enforce sparsity or directly prevent overfitting.

A natural question is *why RR penalisation is effective*, given the abundant empirical evidence in the machine learning and statistical literature. From an optimisation perspective, ridge penalisation imposes a constraint on the parameter vector $\boldsymbol{\beta}$, preventing extreme solutions that often arise with OLS, the unconstrained version of (1). A theoretical explanation is provided by *Random Matrix Theory* (Bai et al., 2010), which shows that empirical eigenvalue estimators are significantly biased when both n and p are large. In particular, when the population eigenvalues are bounded within an interval, the extreme sample eigenvalues converge to the endpoints of this interval. As a result, the largest (smallest) sample eigenvalues are biased upwards (downwards); see, for example, (Ledoit and Wolf, 2004; Asimit et al., 2026) or Figure 2. This figure illustrates the extent of estimation bias in the empirical eigenvalues. When the goal is accurate covariance matrix estimation, the largest eigenvalues are the primary source of error. In contrast, when estimating the inverse covariance matrix, the smallest eigenvalues become the dominant

source of estimation error. These points will become clearer in the next paragraph, where we provide theoretical background to support a broader insight. Figure 3 extends the evidence from Figure 2, showing that the overall empirical eigenvalue estimation error (measured by the L_2 distance) depends on both the sample size n and the concentration ratio p/n. In particular, note that $p = 50$ in both the second boxplot of Figure 3a and the first boxplot of Figure 3b; however, the latter exhibits approximately twice the estimation error of the former.

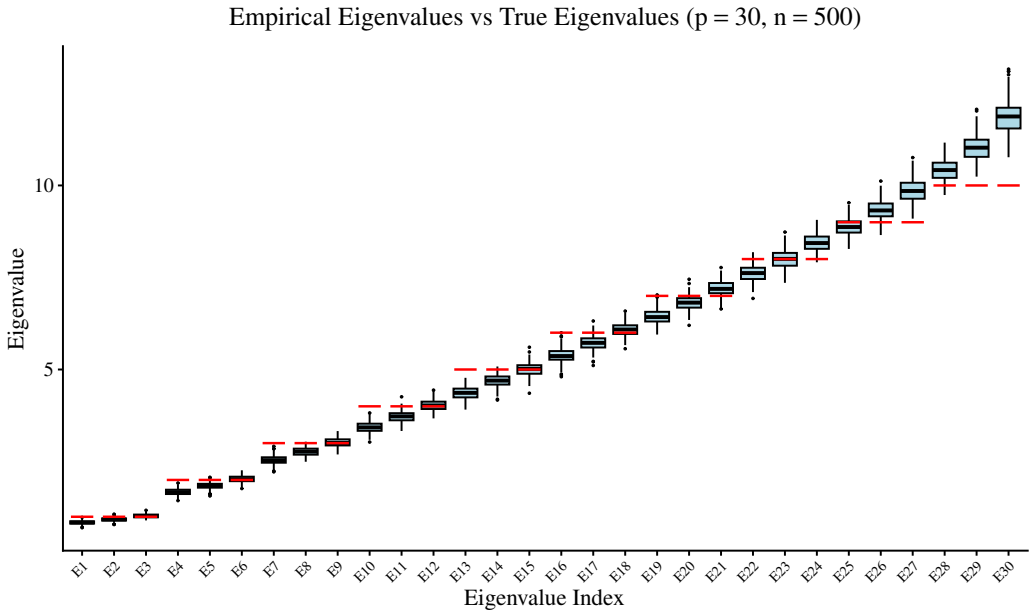


Figure 2: Boxplots of the empirical eigenvalues for a covariance matrix of dimension $p = 30$, based on samples of size $n = 500$. Each boxplot is based on $N = 250$ replications; in each replication, $n = 500$ independent 30-dimensional normally distributed observations are generated with independent components, zero mean and fixed variances $1, 1, 1, 2, 2, 2, \ldots, 10, 10, 10$, and the resulting samples are used to compute the empirical covariance matrices. The red horizontal segments indicate the population eigenvalues that are $1, 1, 1, 2, 2, 2, \ldots, 10, 10, 10$ in this case.

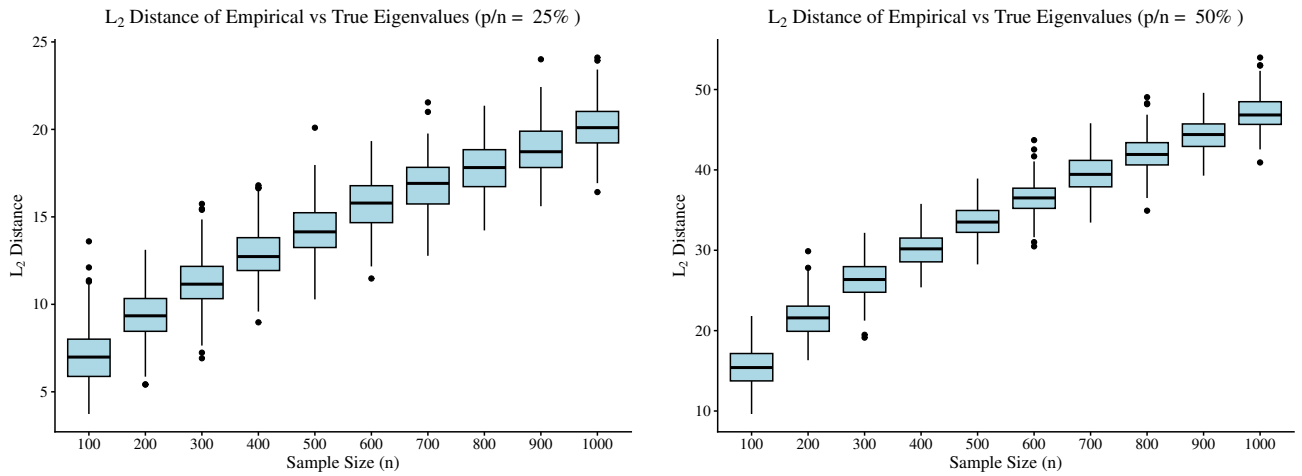
Figures 2 and 3 illustrate the limitations of OLS and explain why ridge regression (RR) provides an effective remedy. To see this, let $\mathbf{\Sigma} := \mathbf{X}^\top \mathbf{X}$, assumed to be full rank (so that $\mathbf{\Sigma}^{-1}$ exists). Recall that the OLS and RR estimators are given by

$$\widehat{\boldsymbol{\beta}}^{\mathrm{OLS}} = \mathbf{\Sigma}^{-1}\mathbf{X}^\top \mathbf{y} \quad \text{and} \quad \widehat{\boldsymbol{\beta}}^{\mathrm{RR}}(\lambda) = (\mathbf{\Sigma} + \lambda \mathbf{I})^{-1}\mathbf{X}^\top \mathbf{y}. \tag{2}$$

Essentially, the RR estimator modifies the OLS estimator by replacing $\mathbf{\Sigma}$ with $\mathbf{\Sigma}^{RR} := \mathbf{\Sigma} + \lambda \mathbf{I}$; as a result, RR effectively inflates the eigenvalues of $\mathbf{\Sigma}$ by the tuning parameter λ. This introduces an upward bias that is particularly beneficial for small eigenvalues. That is, when some true eigenvalues of $\mathbf{\Sigma}$ are small, their empirical counterparts tend to be even smaller, leading to instability in the precision matrix $\mathbf{\Sigma}^{-1}$, whose eigenvalues are the reciprocals of those of $\mathbf{\Sigma}$.[3]

We conclude by noting that $\widehat{\boldsymbol{\beta}}^{RR}(0) = \widehat{\boldsymbol{\beta}}^{OLS}$ and $\widehat{\boldsymbol{\beta}}^{RR}(\infty) = \mathbf{0}$, implying that RR acts as

[3]Since $\mathbf{\Sigma}$ is symmetric, it admits the eigen-decomposition $\mathbf{\Sigma} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top = \sum_k \lambda_k \mathbf{u}_k \mathbf{u}_k^\top$, where the columns of $\mathbf{U}$ are the eigenvectors $\mathbf{u}_k$ and $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues λ_k. Eigenvectors represent directions preserved by $\mathbf{\Sigma}$, while eigenvalues determine the scaling along these directions. As $\|\mathbf{u}_k\|_2 = 1$ for all k, accurate estimation of eigenvalues is the key challenge in covariance estimation.



(a) Fixed ratio $p/n = 25\%$ (b) Fixed ratio $p/n = 50\%$

Figure 3: Boxplots of the Euclidean (L_2) distance between empirical and "true" (covariance matrix) eigenvalues for various sample sizes n, with a fixed concentration ratio p/n. The left and right panels correspond to $p/n = 25\%$ and 50%, respectively. Each boxplot is based on $N = 250$ replications; in each replication, n independent p-dimensional normally distributed observations are generated with independent components, zero mean and variances drawn from a uniform distribution $U(0, 10)$, and the resulting samples are used to compute the empirical covariance matrices.

a shrinkage estimator toward the origin. Although RR effectively inflates the eigenvalues of $\mathbf{\Sigma}$ through the tuning parameter $\lambda > 0$, this parameter is typically selected via cross-validation, which can be computationally demanding. In contrast, the next section shows that Stein-type shrinkage estimators achieve a similar effect in a computationally more efficient manner, as further supported by the numerical results.

2.2 Stein-type Shrinkage Linear Regression

We have explained earlier that controlling the eigenvalues of $\mathbf{\Sigma}$ is beneficial in reducing estimation error in linear regression, and that repeated use of such improved estimators in place of OLS/WLS within IRLS is expected to improve the accuracy of GLM estimation. This motivates the use of covariance shrinkage estimators based on Random Matrix Theory, which was initiated by the seminal work of (Ledoit and Wolf, 2004), and followed by many extensions; see, for example, (Ledoit and Wolf, 2012; Bodnar et al., 2016; Ledoit and Wolf, 2022; Bodnar and Parolya, 2026). The main idea behind these estimators is their ability to optimally adjust the empirical eigenvalues by reducing a theoretical distance between the eigenvalues of the proposed estimators and the true eigenvalues. This provides a conceptually different approach to covariance estimation in very high dimensions—where both n and p are large, but also with a non-negligible concentration ratio p/n—while remaining computationally efficient; further background on general-purpose shrinkage estimation is provided in Section SM.1.

Among many covariance shrinkage estimators, we consider two choices: (i) the *Ledoit–Wolf (LW)* estimator defined in (Ledoit and Wolf, 2004), and (ii) the *Quadratic-Inverse Shrinkage (QIS)* estimator defined in (Ledoit and Wolf, 2022). These two estimators preserve the empirical eigenvectors of $\mathbf{\Sigma}$ and optimally modify its eigenvalues. The LW estimator pulls up/down small/large empirical eigenvalues in a transparent way; however, empirical evidence suggests

that the magnitude of this adjustment is relatively small, indicating that LW is most effective in non-extreme settings where small empirical eigenvalues are not close to zero, a regime in which RR may perform better. We also note that LW is computationally fast and stable, and does not require cross-validation as RR does. The QIS estimator shares these properties, but the modification of the empirical eigenvalues is less explicit; however, by design it can substantially adjust small eigenvalues, which suggests improved estimation of $\boldsymbol{\Sigma}^{-1}$ compared to LW.

One would therefore expect that the LW and QIS estimators of $\boldsymbol{\Sigma}$, denoted by $\boldsymbol{\Sigma}^{LW}$ and $\boldsymbol{\Sigma}^{QIS}$, respectively, could improve the OLS estimator through the plug-in linear regression estimators

$$\widehat{\boldsymbol{\beta}}^{\mathrm{LW}} = \left(\boldsymbol{\Sigma}^{LW}\right)^{-1}\mathbf{X}^{\top}\mathbf{y} \quad \text{and} \quad \widehat{\boldsymbol{\beta}}^{\mathrm{QIS}} = \left(\boldsymbol{\Sigma}^{QIS}\right)^{-1}\mathbf{X}^{\top}\mathbf{y}. \tag{3}$$

The LW and QIS estimators are general-purpose covariance shrinkage estimators, in the sense that they are not specifically tailored to improving linear regression estimation (OLS/WLS). This motivates the development of shrinkage estimators specific to linear regression in (Asimit et al., 2026), which we briefly discuss next.

First, we introduce the SR and GSR estimators from (Asimit et al., 2026). The SR estimator considered in this paper imposes a slab constraint—in geometric lingo, a slab is the region between two parallel hyperplanes—as follows:

$$\widehat{\boldsymbol{\beta}}^{SR}(\mu;\mathbf{u}) := \underset{\boldsymbol{\beta}\in\mathbb{R}^{p+1}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2 + \mu\left(\mathbf{1}^T\boldsymbol{\beta}\right)^2, \tag{4}$$

where $\mu \geq 0$ controls the level of shrinkage and is data-driven, with a closed-form expression available. The GSR estimator considered in this paper generalises (4) by imposing constraints along multiple directions, and is defined as

$$\widehat{\boldsymbol{\beta}}^{GSR}(\boldsymbol{\mu}) := \underset{\boldsymbol{\beta}\in\mathbb{R}^{p+1}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2 + \sum_{k=0}^{p}\mu_k\left(\mathbf{u}_k^T\boldsymbol{\beta}\right)^2, \tag{5}$$

where each $\mu_k \geq 0$ controls shrinkage in the direction of $\mathbf{u}_k$, with $\mathbf{u}_k$'s denoting the eigenvectors of $\boldsymbol{\Sigma}$. By construction, both SR and GSR are penalised regression estimators with carefully chosen penalties such that the tuning parameters—μ for SR and μ_k's for GSR—can be calibrated without cross-validation, while remaining computationally efficient and stable. Interestingly, similarly to RR, GSR acts as an eigenvalue modifier, inducing a systematic upward adjustment of the spectrum, albeit at a different level for each eigenvalue of $\boldsymbol{\Sigma}$. This feature is potentially beneficial for improving estimation accuracy in GLM modelling, as we investigate empirically in later sections.

Second, we introduce the St and DSh estimators from (Asimit et al., 2026), which optimally combine the information embedded in the OLS estimator as follows:

$$\widehat{\boldsymbol{\beta}}^{\mathrm{St}} = a\,\widehat{\boldsymbol{\beta}}^{\mathrm{OLS}} \quad \text{and} \quad \widehat{\boldsymbol{\beta}}^{\mathrm{DSh}} = \operatorname{diag}(\mathbf{b})\,\widehat{\boldsymbol{\beta}}^{\mathrm{OLS}},$$

where the shrinkage parameters $a \in (0,1]$ (for St) and $\mathbf{b} \in (\mathbf{0},\mathbf{1}]$ (for DSh) are chosen to minimise the theoretical MSE and admit closed-form expressions, thus avoiding cross-validation. Note that this form of combining estimators is standard in shrinkage estimation; for further details, we refer the reader to Section SM.1. Equivalently, the St and DSh estimators can be viewed as

OLS estimators with modified covariance matrices:

$$\widehat{\boldsymbol{\beta}}^{\mathrm{St}} = \left(\boldsymbol{\Sigma}^{St}\right)^{-1}\mathbf{X}^{\top}\mathbf{y} \quad \text{and} \quad \widehat{\boldsymbol{\beta}}^{\mathrm{DSh}} = \left(\boldsymbol{\Sigma}^{DSh}\right)^{-1}\mathbf{X}^{\top}\mathbf{y}, \tag{6}$$

where $\boldsymbol{\Sigma}^{St} := a^{-1}\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}^{DSh} := \boldsymbol{\Sigma}\,\mathrm{diag}(\mathbf{b}^{-1})$, with $\mathbf{b}^{-1}$ denoting the componentwise reciprocal of $\mathbf{b}$, i.e., the k^{th} component of $\mathbf{b}^{-1}$ is b_k^{-1}. Notably, $\boldsymbol{\Sigma}^{St}$ preserves the eigenvectors of $\boldsymbol{\Sigma}$ while inflating all eigenvalues by the factor $a^{-1} \geq 1$. In contrast, DSh does not share this property. Hence, St uniformly inflates all eigenvalues (without shifting them as in RR and GSR), and is therefore expected to perform well in non-extreme settings where small empirical eigenvalues are not too close to zero, a regime in which RR may be more effective.

Similarly to LW and QIS, all Stein-type shrinkage estimators (SR, GSR, St, and DSh) are computationally efficient and stable, and do not require cross-validation, unlike RR and EN defined earlier. This makes them natural candidates for replacing the OLS/WLS steps within IRLS to improve the accuracy of GLM estimation, a point further supported by the numerical results presented later. Our IRLS implementations in this paper, for both simulated and real data, impose `maxit` = 250, meaning that the algorithm runs at most 250 weighted linear regressions to GLM estimation. Thus, when using `glm2`, the procedure relies on standard OLS/WLS-type computations at each iteration, up to 250 times. In contrast, our *savvyGLM* package follows the same IRLS structure but replaces the OLS/WLS computations with the expressions in (3) for LW and QIS, and (4)–(6) for the Stein-type shrinkage estimators.

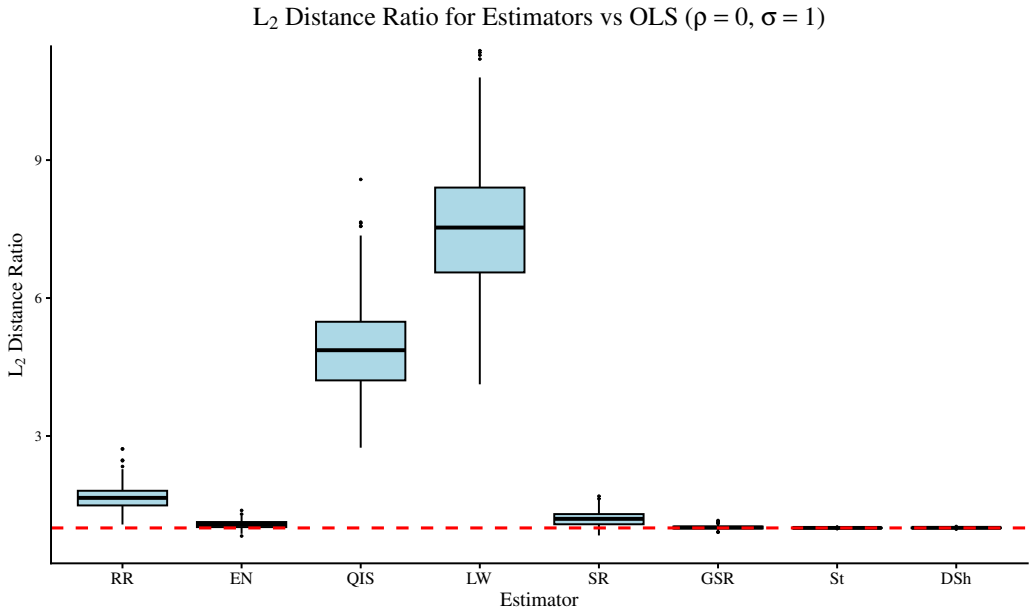
Figures 4–6 provide comparisons of the performance of (i) the penalised methods from the **R**'s `glmnet` package, namely RR and EN, (ii) the six regression approaches (LW, QIS, SR, GSR, St, and DSh), against the classical non-penalised OLS method. Such comparisons can be viewed as repeated GLM fitting under normally distributed data, replicated $N = 250$ times to obtain reliable performance comparisons. For example, when comparing RR to OLS, we compute the L_2 ratio over $N = 250$ replications; for a single replication it is defined as follows:

$$Ratio^{RR} = \frac{\sum_{k=0}^{p}\left(\widehat{\beta}_k^{\mathrm{RR}} - \beta_k\right)^2}{\sum_{k=0}^{p}\left(\widehat{\beta}_k^{\mathrm{OLS}} - \beta_k\right)^2}, \tag{7}$$

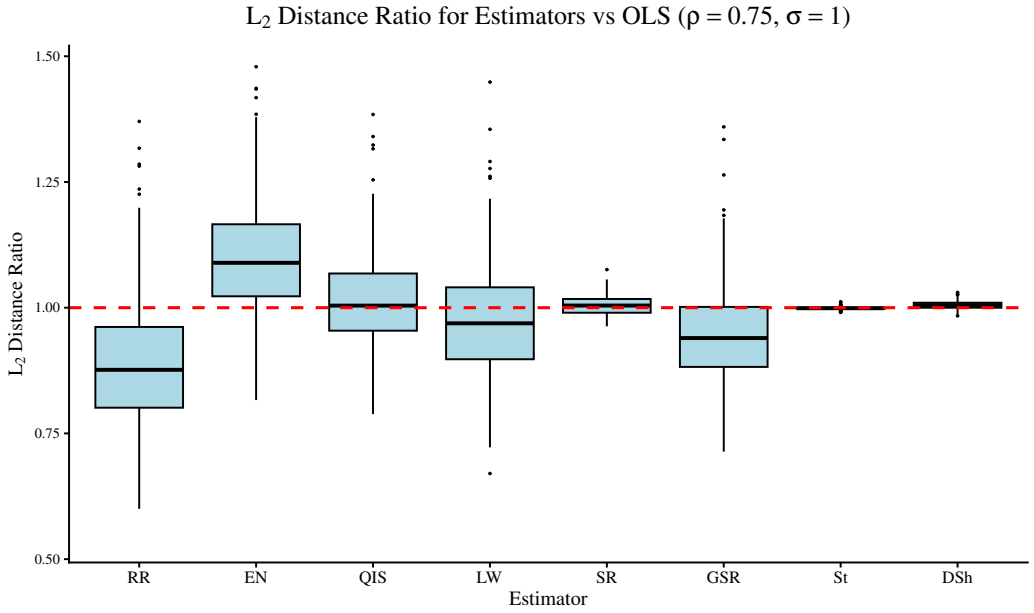
where β_k denotes the true parameter values in this example, which are assumed to be equal to 1 in Figures 4–6. These figures show several interesting results.

We first interpret the plots in Figure 4. When the covariates are independent, the estimation error in the inverse covariance matrix is negligible; consequently, the general-purpose covariance shrinkage estimators QIS and LW are not effective, both because the true covariance matrix is well-conditioned (i.e., has a low condition number) and because we are not in a high-dimensional regime (i.e., the concentration ratio p/n is small) where such estimators typically provide improvements. Moreover, RR is not expected to perform well in Figure 4a since the covariance structure is already well-conditioned, whereas EN performs surprisingly well. We also observe that the four Stein-type shrinkage estimators behave very similarly to OLS, with L_2 ratios close to 1, indicating that these estimators correctly identify that little or no shrinkage is needed in this well-behaved setting.

When the covariates are significantly correlated, as in Figure 4b, the covariance matrix becomes mildly ill-conditioned, which explains why RR performs best, achieving approximately a 13% improvement over OLS. In this setting, QIS and LW reasonably improve relative to the

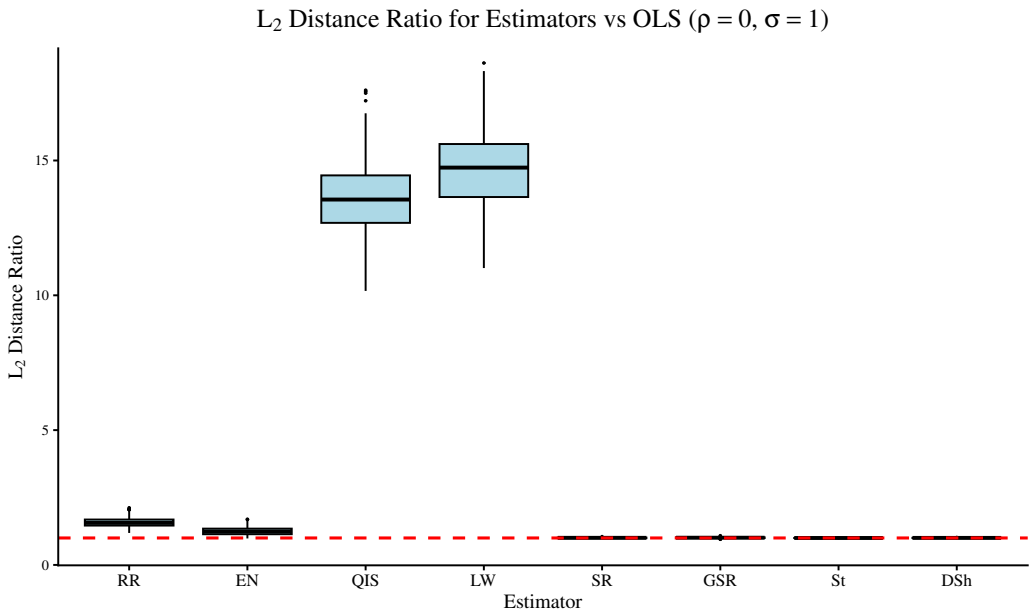


(a) Independent $p = 30$ Covariates ($\rho = 0$)

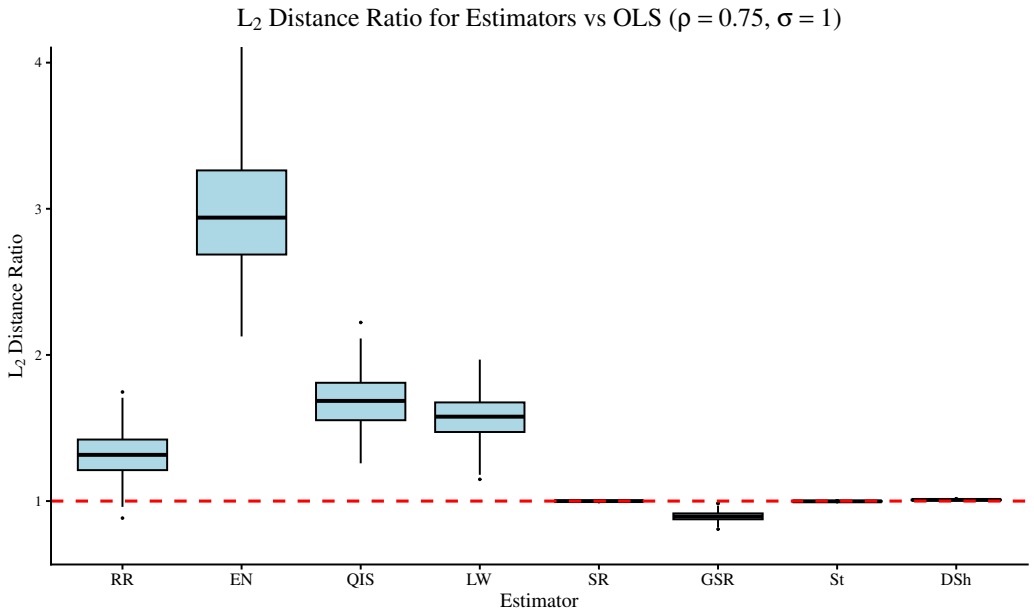


(b) Dependent $p = 30$ Covariates ($\rho = 0.75$)

Figure 4: Boxplots of the L_2 distance ratios defined in (7) are displayed, where each boxplot is based on $N = 250$ replications; values below the red dashed line therefore indicate that the corresponding estimator achieves higher parameter estimation accuracy than OLS. In each replication, a sample of size $n = 500$ is generated with $p = 30$ covariates drawn from a multivariate normal distribution with zero mean and fixed variances $1, 1, 1, 2, 2, 2, \ldots, 10, 10, 10$. The top panel (a) assumes independent covariates, while the bottom panel (b) imposes the correlation structure described in *Step 1* of the data generation process in Section SM.3.1, with $\rho = 0.75$. The linear model is generated as $y = \boldsymbol{\beta}^\top \mathbf{x} + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$.

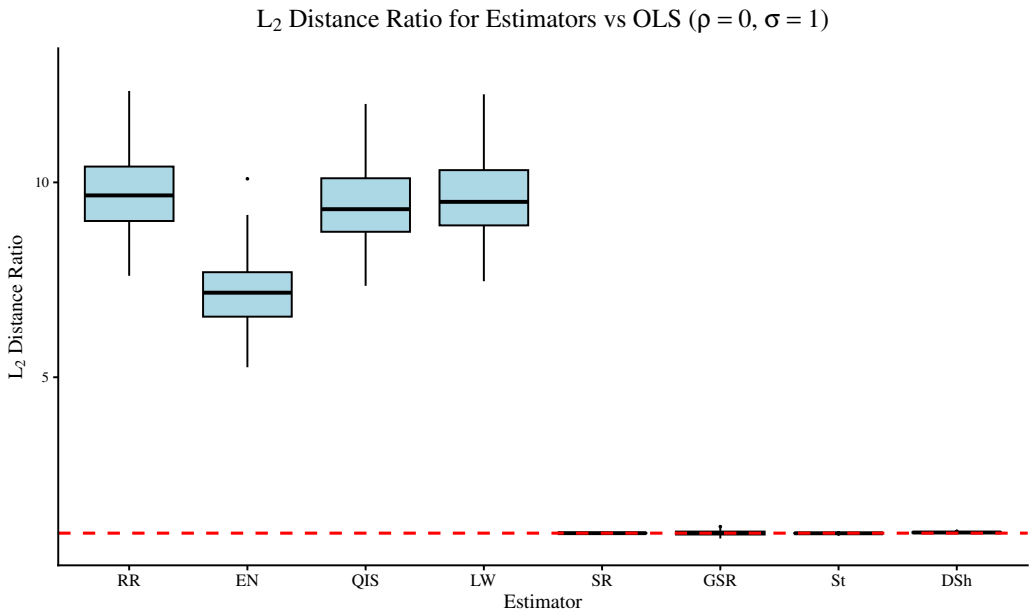


(a) Independent $p = 150$ Covariates ($\rho = 0$)

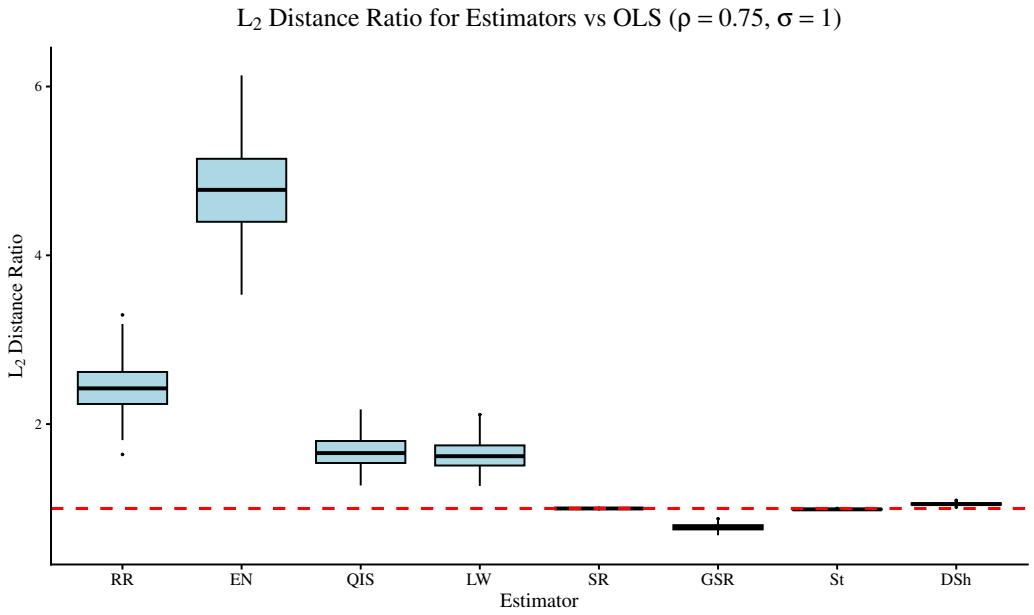


(b) Dependent $p = 150$ Covariates ($\rho = 0.75$)

Figure 5: Boxplots of the L_2 distance ratios defined in (7) are displayed, where each boxplot is based on $N = 250$ replications; values below the red dashed line therefore indicate that the corresponding estimator achieves higher parameter estimation accuracy than OLS. In each replication, a sample of size $n = 500$ is generated with $p = 150$ covariates drawn from a multivariate normal distribution with zero mean and fixed variances ranging from 1 to 10, each repeated 15 times, instead of 3 times as in Figure 4. The top panel (a) assumes independent covariates, while the bottom panel (b) imposes the correlation structure described in *Step 1* of the data generation process in Section SM.3.1, with $\rho = 0.75$. The linear model is generated as $y = \boldsymbol{\beta}^\top \mathbf{x} + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$.



(a) Independent $p = 450$ Covariates ($\rho = 0$)



(b) Dependent $p = 450$ Covariates ($\rho = 0.75$)

Figure 6: Boxplots of the L_2 distance ratios defined in (7) are displayed, where each boxplot is based on $N = 250$ replications; values below the red dashed line therefore indicate that the corresponding estimator achieves higher parameter estimation accuracy than OLS. In each replication, a sample of size $n = 500$ is generated with $p = 450$ covariates drawn from a multivariate normal distribution with zero mean and fixed variances ranging from 1 to 10, each repeated 45 times, instead of 3 times as in Figure 4. The top panel (a) assumes independent covariates, while the bottom panel (b) imposes the correlation structure described in *Step 1* of the data generation process in Section SM.3.1, with $\rho = 0.75$. The linear model is generated as $y = \boldsymbol{\beta}^\top \mathbf{x} + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$.

independent case, but the gains remain limited, as we are not in a high-dimensional regime. EN, by contrast, shows deteriorated performance, which may be attributed to the sensitivity of cross-validation in this context. The four Stein-type shrinkage estimators continue to perform no worse than OLS; however, GSR stands out with a notable improvement of around 5% over OLS and visibly lower variability than RR. This strong performance is driven by the way GSR differentially adjusts the empirical eigenvalues, rather than applying a uniform shift as in RR.

We next interpret the plots in Figures 5 and 6, which share similar settings to Figure 4, but with higher concentration ratios p/n, increasing from 6% to 30% (Figure 5) and 90% (Figure 6). In the case of independent covariates, OLS, SR, GSR, St, and DSh exhibit similar performance, while RR, EN, QIS, and LW perform noticeably worse than OLS. A similar pattern holds under dependent covariates, highlighting the difficulty of high-dimensional settings (when p/n are not too close to 0) for standard penalised methods (RR and EN) and general-purpose shrinkage estimators (QIS and LW). In contrast, SR and St show marginal improvements over OLS in Figures 5b and 6b, while DSh deteriorates slightly in the most high-dimensional case (Figure 6b). Notably, GSR shows a stellar performance in high-dimensional settings, improving over OLS by approximately 11.5% and 22% in Figures 5b and 6b, respectively.

Overall, Figures 4–6 illustrate the potential benefits of accurately solving linear regression subproblems for improving GLM estimation. In particular, even small gains in linear regression accuracy at each IRLS iteration may accumulate and have a butterfly effect on the overall GLM performance, especially in large-scale settings. Although the examples in this section focus on the standard multiple linear model, they provide useful insight into the potential of our proposed approach for GLMs; further evidence based on simulated and real data beyond the Gaussian setting is presented later. The numerical results presented in Section 4 suggest that estimator performance is sensitive to the choice of link function. As a practical guideline, GSR tends to perform best under the logit link, St under the Poisson square-root link, and Ridge regression retains an advantage under the log link—which is the standard specification for both frequency and severity models in P&C ratemaking.

3 Starting Values and Convergence Issues in IRLS

The performance and convergence of the IRLS algorithm for fitting GLMs depend on the choice of starting values. Poor initialisation may lead to slow convergence or divergence, which wastes computational resources on large datasets and generates an unnecessary carbon footprint. Although the default starting values provided by standard statistical software (such as **R**, **MATLAB**, and **Python**) perform well for many standard problems, they can lead to severe convergence failures under certain conditions. In this section, we provide a high-level discussion on the root causes of these failures. We aim to show practitioners why starting values matter and explain how we address these issues in our analysis.

We begin by noting that this section does not address penalised GLMs, as the purpose of this paper is to improve the IRLS algorithm for standard GLM deployment. The three main software implementations handle convergence issues in different ways. The **R** package `glm2` employs step halving to ensure that each IRLS iteration improves the objective function. **MATLAB** uses a step size control mechanism to enhance convergence at an additional computational cost. In contrast, the **Python** implementation in `statsmodels` does not incorporate these safeguards by default. Even with these modifications, good starting values remain essential. The default

initialisations used in these implementations are heuristic. They typically substitute the raw response y directly into the LF, adding a small constant to avoid undefined values like $\log(0)$. Because these workarounds are similar across platforms, focusing on the `glm2` baseline provides a representative view of standard software behaviour.

Lack of convergence in the IRLS algorithm has been documented by the authors of the `glm2` package (Marschner, 2011), who highlight specific GLMs where standard IRLS fails. In our models, convergence failures are tied to the *sqrt* LF. This LF creates a mathematical boundary condition requiring the internal linear predictor to remain positive ($\eta_i > 0$) at all times; the notations used in this section are detailed in Supplementary Material SM.2. The failure usually occurs during the very first IRLS iteration. Standard software initialises the algorithm by applying a simple transformation to the response variable to define the first working response and weights. The algorithm then performs a WLS regression to find the first set of estimated coefficients ($\widehat{\boldsymbol{\beta}}^{(1)}$), which are used to calculate the updated linear predictor ($\eta^{(1)} = \mathbf{X}\widehat{\boldsymbol{\beta}}^{(1)}$). If the initial working weights produce a poorly scaled first step, this coefficient vector can produce a negative updated linear predictor ($\eta_i^{(1)} \leq 0$). Because this occurs on the first iteration, the algorithm has no previous valid coefficients to use for step halving. As a result, the validity check fails and the algorithm halts.

This mechanical breakdown explains why small response values cause IRLS to fail when using the *sqrt* LF. Small values skew the initial heuristic weights, causing the first WLS step to drift into a negative region. In contrast, for Poisson and Gamma GLMs with a *log* LF, the inverse LF guarantees positive predictions regardless of the linear predictor's sign. Because the *log* LF accepts any real number for the linear predictor, it avoids this boundary violation, making convergence failures rare even with extremely small values of y.

To ensure the algorithm passes this first iteration when using the *sqrt* LF, we propose an optimisation method for starting values. As detailed in Supplementary Material SM.5.2, by solving a constrained convex quadratic programming instance before the IRLS loop, we enforce the $\mathbf{x}_i^\top \boldsymbol{\beta} > 0$ constraint. Unlike generic transformations that rely on the response variable, this method determines a set of initial coefficients by evaluating the actual covariate structure and the response. This set of coefficients yields a positive linear predictor and minimises the distance to the objective. Initiating the IRLS algorithm with these boundary-compliant coefficients reduces the probability of the first update step exceeding the negative region.

For practical deployment, our recommendations are straightforward. When analysing empirical insurance portfolios, feature preprocessing creates a well-conditioned design matrix. This stability allows the default initial WLS calculation to yield positive linear predictors without violating the domain of the *sqrt* LF. Relying on these default initialisations is necessary to maintain a reasonable computational workload. Solving constrained quadratic programming problems to generate custom starting values for hundreds of thousands of observations creates computational bottlenecks and excessive memory demands. Therefore, for large empirical datasets, default initialisations are both mathematically sound and efficient. Conversely, when fitting models to volatile synthetic data, introducing multicollinearity creates an ill-conditioned parameter space. In these theoretical cases, the default initialisation calculates invalid predictors and breaks down immediately. Under such structural instability, practitioners should use our `savvyGLM` package to generate boundary-compliant, data-driven starting values that guarantee robust convergence. For complete technical details regarding the default algorithmic behaviours, the exact formulation of our convex optimisation step, and convergence performance tables, please refer to

Supplementary Material SM.5.

4 Numerical Experiments

To evaluate the performance and practical utility of the proposed regularised IRLS estimators, this section presents a two-part numerical analysis. First, Section 4.1 summarises an extensive simulation study designed to test the estimation accuracy and algorithmic convergence of the estimators under controlled conditions across various distributions and LFs. Second, Section 4.2 transitions from synthetic data to an empirical application using the well-known French Motor Third-Party Liability (`freMTPL2`) dataset. This real-world analysis assesses the models' *out-of-sample (OOS)* predictive accuracy, risk-ranking capabilities, model calibration, and computational efficiency in standard actuarial conditions.

4.1 Simulated Data Analysis

A simulation study, detailed in Supplementary Material SM.3, evaluates the estimation accuracy (measured by the relative mean L_2 error) and computational efficiency (measured by the number of iterations until convergence) of various GLM estimators. We compare the standard `glm2` baseline against several regularised alternatives. As detailed in Section 2, these include our four proposed shrinkage estimators (SR, GSR, St, and DSh) and two established covariance shrinkage estimators, LW (Ledoit and Wolf, 2004) and QIS (Ledoit and Wolf, 2022). To implement LW and QIS within our modified IRLS framework, we treat the weighted information matrix as a sample covariance matrix and apply shrinkage prior to inversion. This regularises the WLS step and prevents the algorithm from failing in highly correlated environments. Because extreme multicollinearity in these simulated scenarios creates an ill-conditioned parameter space, default initialisation methods fail or diverge. To address this, we use our optimisation method detailed in Supplementary Material SM.5.2 to generate robust starting values for all IRLS algorithms. This constrained quadratic programming step is computationally feasible here because the simulated datasets are medium-sized. For all IRLS algorithms, we set a maximum limit of 250 iterations and a convergence tolerance of 10^{-6}. Finally, we benchmark against standard regularised models, specifically RR and EN. RR relies on an L_2 penalty, while EN incorporates both L_1 and L_2 penalties. Both of these benchmarks are fitted via coordinate descent rather than IRLS.

To provide a clear overview of these experiments, Table 1 aggregates the accuracy rankings to show how frequently each estimator achieved the best or second-best performance across all 140 simulated scenarios, where we can notice that the proposed shrinkage-based IRLS methods achieve the lowest error in over half of the tested scenarios, though their performance superiority is highly dependent on the chosen LF. Specifically, the proposed GSR estimator achieves the highest accuracy for Logistic Regression, while the St estimator provides the best overall performance for Poisson GLM under the *sqrt* LF. However, alternative methods excel in other specific settings: the QIS estimator is highly effective for Gamma GLM with a *sqrt* LF, and RR dominates when a *log* LF is applied. Furthermore, regarding computational efficiency, the proposed St and DSh estimators frequently exhibit faster convergence than the standard `glm2` for Poisson and Gamma models, with the St estimator dominating efficiency under the *log* LF. Conversely, for Logistic Regression, the LW estimator provides the most distinct computational advantage by requiring the fewest iterations to converge.

Table 1: **Comprehensive Summary of Simulation Study**

	logit Link Function								*sqrt* Link Function						*log* Link Function						Grand Total	
	Ex. Rare		Rare		Balanced		Tot. *logit*		Poisson		Gamma		Tot. *sqrt*		Poisson		Gamma		Tot. *log*		(140 Scenarios)	
Model	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd
RR	0	0	0	0	0	0	0	0	1	5	0	1	1	6	14	2	15	1	**29**	3	**30**	9
EN	4	1	3	3	0	3	7	7	3	2	1	2	4	4	0	10	0	11	0	**21**	11	32
QIS	1	1	2	0	3	2	6	3	1	2	9	7	10	**9**	0	0	0	0	0	0	16	12
LW	2	1	1	1	0	0	3	2	0	2	6	7	6	**9**	0	1	0	1	0	2	9	13
SR	2	5	1	2	5	1	8	8	1	0	2	0	3	0	1	0	2	0	3	0	14	8
GSR	8	7	10	3	7	8	**25**	**18**	1	7	0	1	1	8	2	4	0	4	2	8	28	**34**
St	3	4	3	6	1	0	7	10	13	2	2	2	**15**	4	3	3	3	3	6	6	28	20
DSh	0	1	0	5	4	6	4	12	0	0	0	0	0	0	0	0	0	0	0	0	4	12
Total	20	20	20	20	20	20	60	60	20	20	20	20	40	40	20	20	20	20	40	40	140	140

Notes. This table summarises the performance of all estimators across 140 independent scenarios, aggregating the results reported in Table SM.3.2, Table SM.3.3, and Table SM.3.4. "1st" and "2nd" denote the number of times an estimator achieved the lowest and second-lowest relative Mean L_2 error, respectively. Subtotals are provided for each LF (*logit*, *sqrt*, and *log*). For the subtotal and Grand Total columns, **bold red** text indicates the best performance (highest count), and underlined text indicates the second-best performance (second-highest count). Method superiority is highly dependent on the LF: GSR is the strongest performer for the *logit* LF, St dominates the Poisson *sqrt* LF, QIS is highly effective for the Gamma *sqrt* LF, and RR clearly dominates the *log* LF scenarios.

4.2 Real Data Analysis

To assess how well the shrinkage IRLS estimators perform in practice, this section applies the models to the `freMTPL2` dataset, a standard benchmark for insurance pricing. Unlike the simulation study, we use the default starting values provided by `glm2` for this empirical analysis. This choice is based on both mathematical stability and computational efficiency. First, our feature preprocessing ensures the empirical design matrix is well-conditioned, allowing the default initialisation to remain stable within the IRLS iterations. Second, generating starting values via constrained quadratic programming for hundreds of thousands of observations imposes an unnecessary computational burden. Although this initialisation method differs, we maintain the same convergence criteria, setting a maximum limit of 250 iterations and a tolerance of 10^{-6} for all IRLS algorithms.

In this section, we benchmark the same set of models evaluated in Section 4.1. We compare the shrinkage estimators (St, DSh, SR, GSR, LW, and QIS), implemented via our new **R** CRAN package *savvyGLM*, against the non-penalised baseline (`glm2`) and the regularised benchmarks (RR and EN). The analysis is structured as follows: Section 4.2.1 describes the dataset, feature preprocessing, and outlier handling. Section 4.2.2 explains the resampling framework and the OOS performance metrics used for comparison. Sections 4.2.3 and 4.2.4 present the severity modelling results using Gamma and Tweedie GLMs with a *log* LF. These sections examine how the shrinkage estimators affect predictive precision, risk discrimination, model calibration, and computational efficiency. Additional analyses, including severity models with a *sqrt* LF and claim frequency models using Poisson GLMs, are provided in Supplementary Materials SM.4.1–SM.4.3.

4.2.1 Dataset and Preprocessing

The `freMTPL2` dataset is widely used in the actuarial literature and is available via the `CASdatasets` package in **R**. The data comprises two parts, including a frequency table (`freMTPL2freq`) containing policy characteristics and claim counts, and a severity table (`freMTPL2sev`) detailing individual claim amounts.

We combine the frequency and severity tables using the unique policy identifier (`IDpol`) and aggregate the individual claim amounts to determine the total claim amount per policy. The analysis is limited to policies with both a positive claim count and a positive total claim amount. Across all analyses, the target response variable for the GLM is the average severity per policy, defined as $Y = \text{ClaimAmount}/\text{ClaimNb}$, and the raw claim count (`ClaimNb`) is utilised as the statistical weight for each observation to properly scale the variance. To evaluate the estimators across different sample sizes and varying degrees of data density, we conduct separate analysis on three specific data subsets based on claim frequency: i) policies with exactly one claim ($n = 23,571$), ii) policies with exactly two claims ($n = 1,298$), and iii) a combined set of all policies with at least one claim ($n = 24,944$).

Table 2 summarises the variables used in the analysis, their definitions, and the applied preprocessing steps. To address the extreme positive skewness and severe outliers typical of insurance covariates, we apply bounding and logarithmic transformations to select continuous predictors. Specifically, population `Density` is lower bounded at one and log transformed; `VehPower` is restricted to the interval $[1, 9]$ before log transformation; and the `BonusMalus` score is capped at a maximum of 150 before being log transformed.

Other continuous variables, such as driver age and vehicle age, exhibit nonlinear relationships with insurance risk; we then apply natural cubic splines with three degrees of freedom to both `DrivAge` and `VehAge`. To prevent data leakage during OOS evaluations, the interior spline knots are determined based only on the empirical quantiles of the training data distribution. These knot locations are subsequently projected onto the testing data, ensuring that the OOS spline features are evaluated strictly within the boundaries established by the training set. The categorical `AgeBand` is converted to a continuous numeric feature using the midpoint of its intervals.

For spatial and vehicle type categorical variables (`VehBrand`, `Area`, and `Region`), high cardinality presents a structural challenge. Applying standard one-hot encoding to these features inflates the dimensionality of the design matrix with highly sparse dummy variables, which exacerbates rank deficiency issues during the matrix inversion step of the IRLS algorithms. To avoid this dimensionality issue while preserving the predictive signal, we employ smoothed target encoding. To prevent target leakage, we apply a *Leave One Out (LOO)* approach within the training set. Otherwise, the encoded category means would contain the claim severity of the observations themselves, thereby integrating the target outcome into the predictive features and causing the model to memorise the data unnaturally. Specifically, the encoded value v_i for observation i belonging to category c is calculated as $v_i = \lambda\mu_c^{(-i)} + (1-\lambda)\mu_{\text{global}}$, where $\mu_c^{(-i)}$ is the average severity of category c after excluding observation i, and μ_{global} is the overall average severity of the training data. The credibility weight λ is defined as $\lambda = N_c/(N_c + k)$, where N_c is the number of claims in category c after excluding the current observation, and $k = 20$ is the smoothing parameter we selected. This allocation ensures that categories with very few claims rely more on the global mean, thus avoiding extreme estimates caused by insufficient sample size. For the test set, observations are mapped to the mean of each category in the training data, whilst any unseen categories are assigned the global training mean by default. The

Table 2: **Variable Dictionary and Preprocessing Summary**

Variable	Type	Description	Preprocessing / Transformation
`ClaimAmount`	Numeric	Portion of claim the insurance policy pays.	Raw target data.
`ClaimNb`	Integer	Number of claims during the exposure period.	Used strictly as GLM weights; excluded from predictor features.
`Severity`	Numeric	Average severity per policy.	Target Variable (Y): ClaimAmount/ClaimNb.
`Exposure`	Numeric	Exposure duration for the policy, in years.	Excluded from predictor features to enforce strict *a priori* rating.
`VehPower`	Integer	Vehicle power (ordered categorical values).	Bounded to $[1, 9]$, log-transformed, and standardised.
`VehAge`	Integer	Vehicle age, in years.	Natural cubic splines ($df = 3$), projected from train to test, standardised.
`DrivAge`	Integer	Driver age, in years.	Natural cubic splines ($df = 3$), projected from train to test, standardised.
`BonusMalus`	Integer	Bonus/malus score (< 100 bonus, > 100 malus).	Capped at maximum 150, log-transformed, and standardised.
`Density`	Numeric	Population density (inhabitants per km^2).	Lower bounded at 1, log-transformed, and standardised.
`AgeBand`	Categorical	Age band interval of the driver.	Converted to interval midpoint (numeric) and standardised.
`VehBrand`	Categorical	Vehicle brand category.	LOO Target Encoded (credibility-weighted mean severity) and standardised.
`Area`	Categorical	Area class of the city/community (A to F).	LOO Target Encoded (credibility-weighted mean severity) and standardised.
`Region`	Categorical	Policy region in France.	LOO Target Encoded (credibility-weighted mean severity) and standardised.
`VehGas`	Categorical	Fuel type: Diesel or Regular.	Dummy encoded.

Notes. This table details the variables extracted from the `freMTPL2` datasets. All spline basis generation, LOO target encoding, and standardisation steps are performed dynamically within each cross-validation fold, using parameters derived exclusively from the training set to mathematically guarantee no data leakage into the OOS testing set.

low-cardinality variable `VehGas` is treated with dummy coding, as it has only two categories.

Lastly, to ensure that the shrinkage penalties are applied equitably across all model parameters, all continuous features, derived spline basis columns, and target-encoded variables are standardised to have a zero mean and unit variance using scaling parameters derived solely from the training data.

4.2.2 Performance Measures and Evaluation Procedure

We evaluate the OOS performance of the estimators using repeated K-fold cross-validation. Specifically, we perform 100 repeats of 5-fold cross-validation, resulting in 500 independent OOS evaluations, indexed by $m \in \{1, \ldots, 500\}$. Averaging the performance across 500 replications gives a reliable assessment of predictive ability. In each replication m, 80% of the observations are used for training and 20% for testing. We fit the models on the training set using a Gamma or Tweedie distribution with *log* and *sqrt* LFs. For all IRLS-based estimators, the fitting process is limited to a maximum of 250 iterations and a convergence tolerance of 10^{-6}. In contrast, RR and EN are fitted using the **R**'s `glmnet` package with its default iteration limits and tolerances.

This distinction is necessary because coordinate descent checks convergence differently, using a nested combination of inner and outer loops instead of the single WLS method to update the coefficients. Predictions are then computed on the test set.

Because insurance claim sizes are always right-skewed, standard distance-based error metrics can be heavily distorted by a few extreme outliers. To limit the impact of these outliers, we calculate the Winsorised *Root Mean Squared Error (RMSE)* and the Winsorised *Mean Absolute Error (MAE)*. Let n_{test} be the number of observations in the test set, $y_{i,m}$ be the actual observed severity for observation i in replication m, and $\hat{y}_{i,m}$ be the predicted severity. We define a capping threshold $c_{q,m}$ as the q-th empirical quantile of the actual responses $y_{i,m}$, where $q \in \{0.95, 0.99\}$. The actual and predicted values are truncated at this threshold, yielding $\tilde{y}_{i,m} = \min(y_{i,m}, c_{q,m})$ and $\tilde{\hat{y}}_{i,m} = \min(\hat{y}_{i,m}, c_{q,m})$. The Winsorised metrics for replication m are calculated as:

$$\text{RMSE}^{q}_{W,m} = \sqrt{\frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} (\tilde{y}_{i,m} - \tilde{\hat{y}}_{i,m})^2}, \quad \text{MAE}^{q}_{W,m} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} |\tilde{y}_{i,m} - \tilde{\hat{y}}_{i,m}|.$$

To evaluate the overall goodness-of-fit of the models based on their assumed error distributions, we report the weighted OOS Deviance for each replication m. Let $w_{i,m}$ denote the exposure weight for the i-th test observation. For the Gamma severity models, the deviance is calculated as:

$$\text{Deviance}^{\text{Gamma}}_{m} = 2 \sum_{i=1}^{n_{\text{test}}} w_{i,m} \left(\frac{y_{i,m} - \hat{y}_{i,m}}{\hat{y}_{i,m}} - \ln\left(\frac{y_{i,m}}{\hat{y}_{i,m}} \right) \right).$$

For the Tweedie models, where the variance function depends on an index power parameter $p \in (1, 2)$, the deviance is calculated as:

$$\text{Deviance}^{\text{Tweedie}}_{m} = 2 \sum_{i=1}^{n_{\text{test}}} w_{i,m} \left(\frac{y_{i,m}^{2-p}}{(1-p)(2-p)} - \frac{y_{i,m} \hat{y}_{i,m}^{1-p}}{1-p} + \frac{\hat{y}_{i,m}^{2-p}}{2-p} \right).$$

To measure overall portfolio calibration, we calculate the *Actual-to-Expected (A/E)* deviation, denoted as $(\text{A/E})_m - 1$ and expressed as a percentage. This metric compares the total weighted actual claims to the total weighted predicted claims in replication m, subtracting 1 to centre the metric at zero:

$$(\text{A/E})_m - 1 = \left(\frac{\sum_{i=1}^{n_{\text{test}}} w_{i,m} y_{i,m}}{\sum_{i=1}^{n_{\text{test}}} w_{i,m} \hat{y}_{i,m}} - 1 \right) \times 100\%.$$

A value of exactly 0.00% indicates perfect calibration. A positive value means the model underpredicts total claims, while a negative value means it over-predicts. Model performance is evaluated by the absolute distance to zero.

To assess how well the model ranks policyholders by risk, we compute the exposure-weighted Gini index, reporting both its raw and normalised values. For each replication m, let the indices $(1), (2), \ldots, (n_{\text{test}})$ represent the test observations sorted such that the predicted values are in descending order, $\hat{y}_{(1),m} \geq \hat{y}_{(2),m} \geq \cdots \geq \hat{y}_{(n_{\text{test}}),m}$. Let $w_{(i),m}$ and $y_{(i),m}$ denote the exposure weight and actual claim associated with the i-th sorted prediction. We define the cumulative proportion of exposure weights up to rank j as $W_{j,m} = \sum_{i=1}^{j} w_{(i),m} / \sum_{i=1}^{n_{\text{test}}} w_{i,m}$, and the corresponding cumulative proportion of actual claims as $L_{j,m} = \sum_{i=1}^{j} w_{(i),m} y_{(i),m} / \sum_{i=1}^{n_{\text{test}}} w_{i,m} y_{i,m}$. Using the vector cross-multiplication method, the raw Gini index for replication m is computed

from the Lorenz curve coordinates as:

$$\text{Gini}_m^{\text{Raw}} = 2 \sum_{j=1}^{n_{\text{test}}-1} \frac{(L_{j,m}W_{j+1,m} - L_{j+1,m}W_{j,m})}{2}.$$

A higher positive raw Gini index indicates better risk differentiation. To facilitate standardised comparisons across varying data samples in different replications, we also report the Normalised Gini index. This metric is obtained by dividing the model's raw Gini index by the theoretical Gini index, which is calculated by sorting the observations in descending order according to their actual claims, $y_{i,m}$:

$$\text{Gini}_m^{\text{Norm}} = \frac{\text{Gini}_m^{\text{Raw}}(\text{sorted by } \hat{y})}{\text{Gini}_m^{\text{Raw}}(\text{sorted by } y)}.$$

A normalised Gini index closer to 1 (or 100%) indicates a model that approaches perfect risk differentiation.

We also use Aggregated Double Lift Charts to compare the differences between the predictions of each regularised estimator and the standard `glm2` benchmark. These charts are constructed following a three-step process: i) For each test observation i within replication m, we calculate the logarithmic ratio of the predicted values $R_{i,m} = \ln(\hat{y}_{k,i,m}/\hat{y}_{\text{glm2},i,m})$; ii) we rank the observations according to this ratio and divide them into ten buckets of equal volume. To this end, we calculate the cumulative sum of the exposure weights $w_{i,m}$ for the sorted observations, such that each decile bucket represents exactly 10% of the total exposure weight; and iii) for each bucket b and replication m, we aggregate the raw totals of actual claims, predicted claims, and exposure weights. We then define the four principal sums for each interval as:

$$S_{\text{Act},b,m} = \sum_{i \in b} w_{i,m} y_{i,m}, \quad S_{\text{Base},b,m} = \sum_{i \in b} w_{i,m} \hat{y}_{\text{glm2},i,m},$$
$$S_{\text{Mod},b,m} = \sum_{i \in b} w_{i,m} \hat{y}_{k,i,m}, \quad W_{b,m} = \sum_{i \in b} w_{i,m}.$$

The final average coordinates for each bucket b are computed by summing these values over all 500 replications and dividing by the total aggregated exposure weight:

$$\bar{y}_b = \frac{\sum_{m=1}^{500} S_{\text{Act},b,m}}{\sum_{m=1}^{500} W_{b,m}}, \quad \bar{\hat{y}}_{\text{Base},b} = \frac{\sum_{m=1}^{500} S_{\text{Base},b,m}}{\sum_{m=1}^{500} W_{b,m}}, \quad \bar{\hat{y}}_{\text{Mod},b} = \frac{\sum_{m=1}^{500} S_{\text{Mod},b,m}}{\sum_{m=1}^{500} W_{b,m}}.$$

The resulting plot allows for a direct visual evaluation of model performance in the extreme buckets where the estimators disagree most. A model demonstrates better predictive power if its prediction line tracks the actual claims line more closely than the baseline does in these regions of high divergence. If the actual claims align with the regularised predictions in the first and tenth buckets, it indicates that the estimator has captured risk signals that the standard baseline failed to distinguish.

Finally, we evaluate computational efficiency by recording the *Computation Time (CT)* in seconds and the *Number of Iterations (NoIt)* required for convergence. To easily compare models across the 500 replications, the metrics (RMSE, MAE, Deviance, CT, and NoIt) are reported as relative ratios against the `glm2` baseline. Let $M_{k,m}$ be the value of a metric for model k

in replication m, and $M_{\text{glm2},m}$ be the baseline value. The average relative ratio for model k is calculated as:

$$\text{Ratio}_k = \frac{1}{500}\sum_{m=1}^{500}\frac{M_{k,m}}{M_{\text{glm2},m}}.$$

A ratio below 1.000 indicates that the regularised model performs better than the `glm2` baseline (e.g., lower error or faster computation). The Raw Gini index, Normalised Gini index, and A/E - 1 are not scaled as ratios; instead, they are reported as the simple average of their respective values across all 500 replications, as their absolute scales are essential for actuarial interpretation.

4.2.3 Empirical Results on Average Severity per Policy – Gamma GLM

This section evaluates the practical performance of the estimators when modelling average severity per policy. We first focus on the Gamma GLM equipped with the *log* LF as it is a more common actuarial choice for insurance pricing. Table 3 and Figure 7 show the OOS performance for the Gamma GLM across three frequency panels. We also examine the performance of these estimators with the *sqrt* LF; detailed results for this alternative specification are included in Supplementary Material SM.4.1.

The empirical results reveal a clear tradeoff between precision and calibration. Our proposed St estimator consistently has the lowest 95th percentile distance metrics. For instance, in Panel C, it reduces the RMSE to 0.958 and the MAE to 0.938. However, this precision comes at the cost of portfolio miscalibration, resulting in an A/E deviation of 7.47%. The standard regularised benchmarks, RR and EN, effectively lower 99th percentile tail errors across most panels, and RR achieves the highest raw and normalised Gini values in Panels A and C. This shows their ability to manage severe large loss issues and rank risks effectively. However, they also have notable calibration problems; for example, RR produces much larger A/E deviations of 5.77% and 5.08% in those panels. Additionally, both RR and EN require intensive cross-validation, resulting in computation times that are often 20 to 250 times longer than the baseline `glm2`.

It is also important to note the general decline in calibration in Panel B. Across all estimators, including the baseline `glm2`, the A/E deviations are significantly higher, ranging from 5% to over 20%. This instability could be due to the much smaller sample size of exactly two claims ($n = 1298$) compared to the single and aggregate claim panels. This makes the portfolio highly sensitive to individual large loss volatility, leading to poorer calibration. For industry practitioners, our proposed SR and GSR estimators offer the most practical value. They require no cross-validation, which saves computation time while maintaining reasonable risk discrimination and providing better portfolio calibration. The SR estimator achieves the best overall calibration in Panel A with an A/E deviation of -0.12% and a competitive 1.15% in Panel C.

These numerical findings are supported by the Aggregated Double Lift charts in Figure 7. By examining the prediction disagreements, we see clear behavioural differences at the portfolio extremes. In bucket 1, which isolates policyholders where the RR and EN predict significantly lower average severity per policy than the standard `glm2`, they aggressively overcorrect. Across the single claim and all claims panels, the predicted values for RR and EN are much lower than the number of claims actually observed in bucket 1, while the baseline `glm2` stays much closer to the true risk level. This shows that both RR and EN penalise these risks too heavily, leading to severe underpricing for the safest drivers. Similarly, in bucket 10, which captures risks where the RR and EN predict the highest average severity per policy relative to the baseline `glm2`,

Table 3: **OOS for Gamma GLM (*log* LF) on freMTPL2 Average Severity per Policy**

	Winsorised RMSE		Winsorised MAE			Gini				
Model	**95%**	**99%**	**95%**	**99%**	**Deviance**	**Raw**	**Norm**	**A/E - 1**	**CT**	**NoIt**
Panel A) Exactly 1 Claim ($n = 23,571$)										
glm2	1.000	1.000	1.000	1.000	1.000	0.115	0.172	1.35%	**1.000**	1.000
RR	1.056	**0.893**	1.057	0.990	1.047	**0.119**	**0.178**	5.77%	20.324	48.614
EN	1.004	0.979	1.008	0.996	1.037	0.111	0.167	2.93%	20.872	127.220
QIS	0.998	1.000	0.997	0.998	1.000	0.114	0.171	1.54%	1.163	0.979
LW	1.002	0.999	1.003	1.002	1.000	0.116	0.174	1.93%	1.703	**0.580**
SR	0.997	1.003	0.996	0.999	**1.000**	0.114	0.170	**-0.12%**	3.053	0.914
GSR	0.999	0.999	0.999	0.998	1.002	0.113	0.170	1.81%	2.342	0.869
St	**0.967**	0.995	**0.947**	**0.959**	1.001	0.115	0.172	5.66%	2.961	0.892
DSh	1.006	1.000	1.017	1.013	1.001	0.116	0.174	0.35%	2.504	0.876
Panel B) Exactly 2 Claims ($n = 1,298$)										
glm2	1.000	1.000	1.000	1.000	1.000	**0.179**	**0.271**	11.95%	**1.000**	1.000
RR	0.961	**0.961**	0.999	0.980	1.040	0.132	0.204	14.58%	256.312	69.366
EN	0.979	0.991	1.001	0.997	1.050	0.128	0.201	12.99%	152.309	135.528
QIS	1.027	1.009	1.041	1.036	**0.985**	0.177	0.267	5.99%	3.695	0.633
LW	1.054	1.030	1.064	1.061	1.014	0.170	0.254	**5.24%**	3.676	0.758
SR	1.002	1.006	0.999	1.002	1.004	0.179	0.270	12.08%	5.485	0.543
GSR	0.991	0.992	0.995	0.993	1.003	0.173	0.263	12.90%	5.278	0.620
St	**0.956**	0.971	**0.943**	**0.943**	1.011	0.179	0.271	20.11%	6.984	**0.496**
DSh	0.980	0.989	0.979	0.977	1.022	0.161	0.241	15.40%	5.234	0.584
Panel C) All Claims $\geq$ 1 ($n = 24,944$)										
glm2	1.000	1.000	1.000	1.000	1.000	0.105	0.157	2.18%	1.000	1.000
RR	1.046	**0.892**	1.046	0.982	1.026	**0.115**	**0.171**	5.08%	18.610	38.992
EN	1.011	0.968	1.013	0.994	1.026	0.109	0.163	3.28%	20.125	109.230
QIS	1.000	1.000	1.000	1.000	1.000	0.106	0.158	2.23%	**0.952**	1.011
LW	1.004	0.994	1.004	1.000	0.999	0.106	0.159	2.84%	1.534	**0.590**
SR	0.997	1.004	0.997	0.999	1.004	0.106	0.158	**1.15%**	3.003	0.946
GSR	1.002	0.998	1.000	0.999	1.000	0.106	0.158	2.56%	2.135	0.781
St	**0.958**	0.989	**0.938**	**0.950**	1.003	0.105	0.157	7.47%	3.277	1.030
DSh	0.999	0.999	1.003	1.002	**0.998**	0.109	0.163	1.99%	2.131	0.831

Notes. This table reports the OOS performance ratios on average severity per policy for regularised estimators relative to the standard `glm2` benchmark on the freMTPL2 dataset. Results are averaged over 500 replications utilising a Gamma distribution with a *log* LF. An A/E deviation closer to 0.00% indicates superior portfolio calibration. For distance metrics like RMSE, MAE, Deviance, CT, NoIt, lower is better. For the raw Gini and normalised Gini index, higher is better. The best performing estimator in each column is highlighted in **bold red**, and the second best is underlined. Note that in all 500 replications, only QIS failed once during the convergence of panel C; all other models converged.

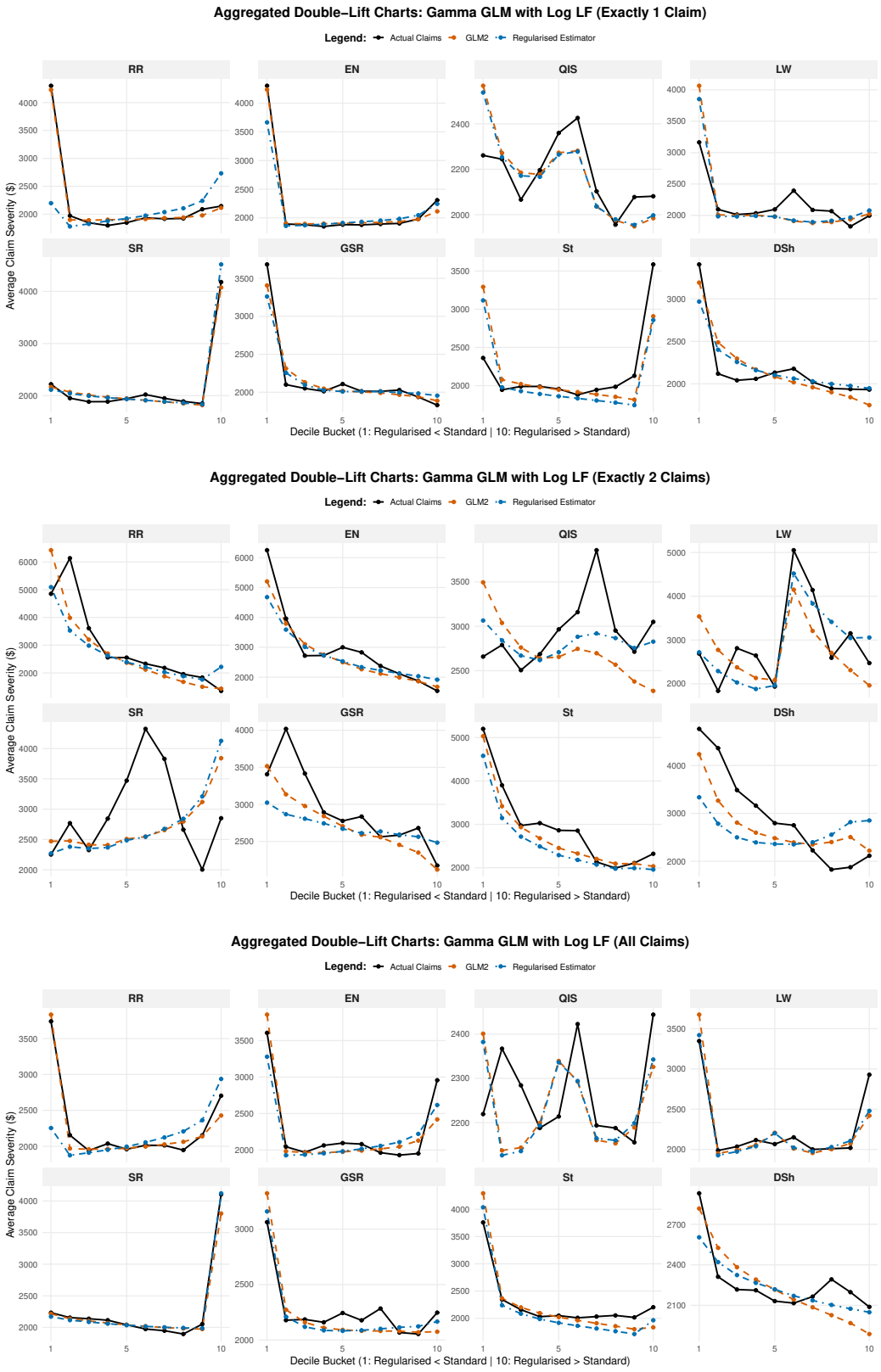


Figure 7: Aggregated Double Lift Charts comparing regularised estimators against the standard Gamma GLM using a *log* LF on freMTPL2 average severity per policy.

their advantage is completely lost. RR overpredicts the actual claims in this upper tail in both panels, while the standard `glm2` is more accurate. This indicates that both RR and EN penalties distort pricing at both ends of the portfolio distribution.

In contrast, our proposed SR and GSR estimators show greater reliability. Although the `glm2` baseline tracks the actual claims reasonably well in the single claim and all claims panels, the SR and GSR predictions nearly match the trends with the true observations across ten decile buckets. For example, in the GSR plot for the all claims panel, the baseline `glm2` noticeably overprices the safest risks in bucket 1 and underprices the riskiest profiles in bucket 10. The GSR estimator corrects these errors well, pulling the predictions much closer to the actual observed claims at both ends of the portfolio. They respond to the underlying risk profile without introducing the extreme boundary distortions seen in RR and EN, demonstrating that they refine the baseline model safely without compromising overall pricing accuracy.

4.2.4 Empirical Results on Average Severity per Policy – Tweedie GLM

This section expands our evaluation of average severity per policy by applying the Tweedie GLM. Because the Tweedie distribution offers a more flexible compound structure than the strict Gamma assumption, our primary focus is to determine if this flexibility improves overall modelling performance and portfolio calibration. Table 4 and Figure 8 show the results for the Tweedie GLM using the *log* LF. The corresponding results for the Tweedie GLM using the *sqrt* LF are available in Supplementary Material SM.4.2.

The overall performance ranking among the estimators stays the same as in the Gamma results. The St estimator still leads in the 95th percentile distance metrics. The RR and EN estimators capture tail errors but show declines in calibration. Across all panels, the Tweedie models produce lower raw and normalised Gini indices, indicating a slight drop in pure risk discrimination. When we look at the A/E deviations, the Tweedie assumption improves portfolio calibration for the majority of the models. By comparing Table 4 to the Gamma results, 7 out of the 9 models show an improved A/E ratio in both Panel B and Panel C. Furthermore, the absolute best calibration score in Panel B improves from 5.24% under the Gamma assumption to 1.31% here. However, in Panel A, the absolute best calibration score becomes slightly worse. For example, the proposed SR estimator changes from 0.12% to 1.09%.

In addition to the mixed calibration results, the Tweedie GLM comes with a large computational burden. The fitting process requires an internal grid search of the profile log-likelihood function to find the best variance power index ρ. This increases computation time for all estimates. As shown in the final column of Table 4, the optimal ρ value approaches 1.700 across almost all models and panels. This finding suggests that the empirical average severity per policy data lean toward a Gamma distribution structure. Since the optimal distribution is close to Gamma, it helps explain why the relative performance of different estimators matches the Gamma GLM results relatively well. For industry practitioners, the additional computational cost of the grid search may therefore outweigh the modest gains in calibration observed in some panels.

This similarity is further supported by the Aggregated Double Lift charts in Figure 8. Using the Tweedie distribution does not resolve the boundary distortion issues caused by RR and EN. As seen in the Gamma GLM, the RR and EN estimates overcorrect at the portfolio extremes. In bucket 1, the predicted values from RR are much lower than the actual observed claims, leading to underestimation of the safest risk profiles. In bucket 10, RR overestimates actual claims,

Table 4: **OOS for Tweedie GLM (*log* LF) on freMTPL2 Average Severity per Policy**

	Winsorised RMSE		Winsorised MAE			Gini					
Model	**95%**	**99%**	**95%**	**99%**	**Deviance**	**Raw**	**Norm**	**A/E - 1**	**CT**	**NoIt**	**Power**
Panel A) Exactly 1 Claim ($n = 23,571$)											
glm2	1.000	1.000	1.000	1.000	1.000	0.110	0.165	1.53%	**1.000**	1.000	1.700
RR	1.067	**0.908**	1.062	1.002	1.076	**0.115**	**0.173**	4.87%	67.158	161.986	1.697
EN	1.001	0.994	1.004	0.999	1.077	0.107	0.161	2.05%	76.070	580.813	1.695
QIS	1.000	1.000	0.999	1.000	1.000	0.110	0.164	1.54%	2.687	0.928	1.700
LW	1.003	0.999	1.003	1.002	1.000	0.111	0.166	1.98%	6.846	0.793	1.700
SR	0.997	1.002	0.996	0.998	**0.998**	0.110	0.163	1.09%	8.625	1.099	1.700
GSR	1.001	0.999	0.999	0.999	1.002	0.111	0.166	1.73%	7.773	0.881	1.700
St	**0.970**	0.996	**0.953**	**0.964**	1.001	0.111	0.166	4.64%	8.426	**0.737**	1.700
DSh	1.004	1.000	1.016	1.012	1.000	0.113	0.170	**0.50%**	5.644	0.943	1.700
Panel B) Exactly 2 Claims ($n = 1,298$)											
glm2	1.000	1.000	1.000	1.000	**1.000**	0.173	0.262	10.09%	**1.000**	1.000	1.700
RR	0.971	0.962	0.996	0.977	1.059	0.136	0.211	12.54%	149.101	84.753	1.696
EN	0.987	0.995	0.997	0.996	1.081	0.134	0.213	10.94%	162.743	284.314	1.696
QIS	1.018	1.017	1.020	1.024	1.016	**0.175**	**0.264**	8.24%	10.874	0.686	1.699
LW	1.068	1.054	1.072	1.080	1.047	0.172	0.256	**1.31%**	7.929	0.633	1.696
SR	1.002	1.005	0.999	1.001	1.006	0.172	0.259	10.18%	11.425	0.587	1.700
GSR	0.997	0.995	0.998	0.997	1.003	0.170	0.259	10.42%	11.750	0.577	1.700
St	**0.946**	**0.960**	**0.929**	**0.928**	1.006	0.173	0.262	20.49%	11.786	0.520	1.700
DSh	0.984	0.993	0.987	0.986	1.016	0.159	0.239	12.41%	10.997	**0.468**	1.700
Panel C) All Claims ≥ 1 ($n = 24,944$)											
glm2	1.000	1.000	1.000	1.000	1.000	0.106	0.158	1.94%	**1.000**	1.000	1.700
RR	1.054	**0.905**	1.048	0.991	1.064	**0.114**	**0.171**	4.22%	86.333	133.468	1.696
EN	1.003	0.986	1.005	0.997	1.070	0.108	0.161	2.44%	97.406	552.184	1.695
QIS	1.001	1.000	1.001	1.001	1.000	0.106	0.158	1.81%	3.587	0.968	1.700
LW	1.004	0.998	1.003	1.001	0.999	0.107	0.160	2.20%	10.535	0.755	1.700
SR	0.997	1.003	0.997	0.999	1.001	0.107	0.159	**1.64%**	12.517	1.086	1.700
GSR	1.001	0.999	0.999	0.999	**0.999**	0.108	0.161	2.16%	11.157	0.871	1.700
St	**0.970**	0.995	**0.955**	**0.966**	1.002	0.106	0.159	5.21%	12.073	**0.746**	1.700
DSh	0.993	0.998	0.995	0.996	1.000	0.109	0.163	2.24%	9.168	0.904	1.700

Notes. This table reports the OOS performance ratios on average severity per policy for regularised estimators relative to the standard `glm2` benchmark on the `freMTPL2` dataset. Results are averaged over 500 replications utilising a Tweedie distribution with a *log* LF. An A/E deviation closer to 0.00% indicates superior portfolio calibration. For distance metrics like RMSE, MAE, Deviance, CT, NoIt, lower is better. For the raw Gini and normalised Gini index, higher is better. The optimal variance power index ρ selected via in sample profile log likelihood is reported in the final column. The best performing estimator in each column is highlighted in **bold red**, and the second best is underlined. All models converged across all panels.

Figure 8: Aggregated Double Lift Charts comparing regularised estimators against the Tweedie GLM using a *log* LF on freMTPL2 average severity per policy.

losing the advantage over the standard `glm2` that tracks true risk much better at the upper tail. Compared to the Gamma results, the GSR estimator begins to deviate from actual claims under the Tweedie assumption. In contrast, the SR estimator maintains its reliability: it stays aligned with actual observed claim values across all ten decile buckets for both the single claim and all claims panels.

Taken together, the results across Tables 3 and 4 reveal a systematic and practically meaningful pattern: the Gamma GLM consistently achieves higher Gini indices across all panels, indicating superior risk discrimination, while the Tweedie GLM yields smaller A/E deviations for the majority of estimators in Panels B and C, indicating better portfolio calibration. Neither distributional assumption therefore dominates on the full set of diagnostic criteria relevant to pricing practice.

This finding reflects a tension that is familiar to practising pricing actuaries. The frequency/severity framework itself already combines two separately fitted GLMs — a Poisson model for claim counts and a Gamma model for claim sizes — whose predictions are multiplied to obtain the pure premium. Within the severity component, the choice between Gamma and Tweedie is rarely resolved on purely statistical grounds: the Tweedie distribution is a compound Poisson-Gamma and its variance power index ρ must be estimated from the data, making it a strictly richer but also more computationally demanding specification. In practice, teams commonly evaluate both fitted models against lift charts and calibration diagnostics, with the final selection typically guided by actuarial judgement. The blending algorithm proposed in Supplementary Material SM.6 offers a complementary approach: rather than discarding one model, it combines the predictions of both via a single data-calibrated weight $\delta \in [0, 1]$, providing a transparent and auditable procedure.

5 Conclusions

The first two contributions of this paper introduce a class of shrinkage-enhanced GLM estimation methods. We develop modified IRLS algorithms that embed non-parametric shrinkage estimators directly into the iterative fitting procedure. The resulting framework delivers more accurate parameter estimates while maintaining the computational efficiency of standard IRLS implementations, as illustrated through extensive numerical evidence.

The third contribution proposes an optimisation-based strategy for selecting high-quality starting values in GLM estimation. This provides a practical and robust alternative to ad hoc initialisation, improving stability and convergence behaviour in routine GLM deployments, as further supported by extensive numerical experiments.

6 Data availability statement

The primary data source for the empirical analysis is the French Motor Third-Party Liability (MTPL) dataset. This dataset is freely accessible and can be downloaded via the `CASdatasets` package in **R**.

7 Funding statement

This work received no specific grant from any funding agency, commercial, or not-for-profit sectors.

8 Competing interests

On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

Asimit, V., Cidota, M. A., Chen, Z., and Asimit, J. (2026). Slab and shrinkage linear regression estimation. https://openaccess.city.ac.uk/id/eprint/35005/.

Bai, Z., Silverstein, J. W., et al. (2010). *Spectral analysis of large dimensional random matrices*, volume 20. Springer.

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. and Parolya, N. (2026). Reviving pseudo-inverses: Asymptotic properties of large dimensional moore–penrose and ridge-type inverses with applications. *The Annals of Statistics*, 54(2):1053–1079.

Bühlmann, H. (1967). Experience rating and credibility. *ASTIN Bulletin*, 4(3):199–207.

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.

Denuit, M., Maréchal, X., Pitrebois, S., and Walhin, J.-F. (2007). *Actuarial modelling of claim counts: Risk classification, credibility and bonus-malus systems*. John Wiley & Sons.

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.

Frees, E. W. (2014). *Regression Modeling with Actuarial and Financial Applications*. Cambridge University Press.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

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.

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.

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.

Ledoit, O. and Wolf, M. (2022). Quadratic shrinkage for large covariance matrices. *Bernoulli*, 28(3):1519–1547.

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.

Muirhead, R. J. (1987). Developments in eigenvalue estimation. In *Advances in Multivariate Statistical Analysis: Pillai Memorial Volume*, pages 277–288. Springer.

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.

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.

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.

Wüthrich, M. V. and Merz, M. (2023). *Statistical foundations of actuarial learning and its applications*. Springer.

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.

Supplementary Material to

GLM Solutions via Shrinkage

Vali Asimit, Oana Avramescu, Ziwei Chen,
Diego Rivas, Claudio Senatore

This supplementary material complements the main paper by providing additional technical details, theoretical background, and extended empirical results. Section SM.1 provides further background on shrinkage estimation. Section SM.2 details the mathematical mechanics of GLM parameter estimation, exploring *Maximum Likelihood Estimation (MLE)*, Newton's Method, Fisher Scoring, and the IRLS algorithm, and illustrates how these solvers behave differently using real insurance data. Section SM.3 presents an extensive simulation study evaluating the estimation accuracy and computational efficiency of the proposed shrinkage estimators against benchmarks. Section SM.4 provides additional empirical results for the `freMTPL2` data application, extending the severity and frequency modelling to include the *sqrt* LF for Gamma, Tweedie, and Poisson GLMs. Section SM.5 details the algorithmic causes of convergence failures, outlining software default initialisations and presenting our optimisation method for starting values along with supporting convergence tables. Finally, Section SM.6 proposes a fast algorithm for combining predictions from multiple GLMs, a recurring model selection challenge in non-life pricing practice.

SM.1 Further Background on Shrinkage Estimation

The shrinkage estimators proposed in this paper are motivated by Stein's paradox, which shows that the MSE of the MLE for a mean vector can be reduced by introducing shrinkage, at least in the multivariate normal setting (Stein, 1956, 1960; James et al., 1961). This puzzling result, often presented in a Bayesian context, underpins the well-known James–Stein estimator (James et al., 1961). From a practical actuarial perspective, the key insight is straightforward: when estimating multiple related quantities—such as expected losses across different risks—it is often better to combine information across risks rather than estimate each one independently based solely on its own data. In other words, some degree of pooling can improve overall estimation accuracy. While this idea has been studied formally in statistics for almost 70 years, it closely mirrors long-standing actuarial practice. In particular, (Whitney, 1918) introduced the idea of adjusting individual premiums toward a portfolio average to improve stability—an approach that reflects the same underlying principle of shrinkage. This concept was later formalised mathematically through credibility theory in (Bühlmann, 1967). For a broader discussion of the development of credibility theory in actuarial practice, see (Goulet, 1998).

A large volume of academic research has been put forward on Stein-type estimators under various assumptions. 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 non-parametric approach has recently risen where linear shrinkage estimators are available under some specific distributional assumptions (Wang et al., 2014; Bodnar et al., 2019), but such developments are a great step ahead towards distribution-free estimators.

The Stein-type shrinkage estimators discussed so far are designed to improve high-dimensional estimation by combining information across the components of the parameter vector, and should not be confused with the popular penalised estimators such as RR (Hoerl and Kennard, 1970), LASSO (Tibshirani, 1996), or EN (Zou and Hastie, 2005). In particular, a standard example is the linear shrinkage estimator for an unknown vector $\boldsymbol{\theta}$, given by $(1-\rho)\widehat{\boldsymbol{\theta}} + \rho\widehat{\boldsymbol{\theta}}_{\text{target}}$, where $\widehat{\boldsymbol{\theta}}$ is a conventional estimator, $\widehat{\boldsymbol{\theta}}_{\text{target}}$ is a structured or parsimonious target estimator, and ρ is the shrinkage intensity. In the credibility premium setting, $\boldsymbol{\theta}$ represents

the population mean vector (i.e., the true premiums for the p policies), $\widehat{\boldsymbol{\theta}}$ denotes the observed mean vector (i.e., the empirical premiums based on individual experience), and $\widehat{\boldsymbol{\theta}}_{\text{target}}$ is a vector whose entries correspond to a collective benchmark, such as the portfolio average or a market-wide premium level.

The same principle extends naturally beyond vector estimation to matrix-valued parameters. A prominent example is covariance matrix shrinkage estimation (Ledoit and Wolf, 2004; Schäfer and Strimmer, 2005; Ledoit and Wolf, 2012), as well as precision matrix (inverse covariance) shrinkage estimation (Bodnar et al., 2016). A comprehensive review of recent developments in shrinkage estimation can be found in (Bodnar et al., 2022).

SM.2 Technical Details about GLM

This section provides an overview of GLM modelling. Readers unfamiliar with the topic may consult the standard GLM literature (Wood, 2017; Wüthrich and Merz, 2023), although the exposition here is sufficient to understand the implementations used in this paper. Section SM.2.1 presents the background on GLMs and the MLE formulation. Subsequent sections detail the numerical methods used for parameter estimation. It is important for practitioners to understand these mechanics. Standard software relies on these algorithms to find the optimal coefficients. When a pricing model fails to converge, the cause is often the solver algorithm rather than the data. By understanding how Newton's Method (Section SM.2.2), the Fisher Scoring Method (Section SM.2.3), and the IRLS algorithm (Section SM.2.4) work, practitioners can diagnose software errors and build stable models. To demonstrate this practical value, Section SM.2.5 compares all three solvers using a Gamma GLM with a *log* LF on the same actual insurance data utilised in Section 4.2. This comparison demonstrates why standard IRLS requires step halving to handle volatile data and why Fisher Scoring fails.

SM.2.1 Model Description

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.

A univariate GLM setting assumes that the response variable Y, defined on $\mathcal{Y} \subseteq \mathbb{R}$, is explained by covariates/features $\mathbf{X}$ defined on $\mathcal{X} \subseteq \mathbb{R}^p$. Let $\{P_{\theta,\phi} : \theta \in \Theta \subseteq \mathbb{R}, \phi \in \Phi \subseteq \mathbb{R}\}$ 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

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

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. The mean and variance of Y are $\mathrm{E}\big[Y\big] = b'(\theta)$ and $\mathrm{Var}\big[Y\big] = a(\phi)b''(\theta)$, respectively. The estimation procedure assumes an independent sample $Y_1, \ldots, Y_n$ such that Y_i is distributed as in (SM.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:

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

where $\mathbf{x}_i$ is a d-dimensional vector of realised covariates/features. 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

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

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

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

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

$$h(\eta) = b'(\eta), \quad \eta \in \mathbb{R}. \tag{SM.2.4}$$

The technical conditions for the existence and uniqueness of the MLE estimate are well-known and require a strictly concave log-likelihood function and some boundary conditions (Wedderburn, 1976; Mäkeläinen et al., 1981). These conditions are satisfied by the instance in (SM.2.3) if functions a, b and h satisfy certain regularity conditions, which are formalised as *proper GLM* (Asimit et al., 2025). 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 (SM.2.3) could be done by 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., 2025). However, because the statistical structure of GLMs naturally lends itself to specialised iterative algorithms, custom solvers are typically preferred over general-purpose ones. A common approach for estimating the model parameters is MLE computed by Newton's Method or Fisher Scoring, or via the IRLS algorithm (Nelder and Wedderburn, 1972). Quasi-Newton methods such as *Broyden-Fletcher-Goldfarb-Shanno (BFGS)*, *Davidon-Fletcher-Powell (DFP)*, and *Limited-memory BFGS (L-BFGS)* can also be used to approximate the Hessian to reduce the computational cost in high-dimensional

settings.

SM.2.2 Newton's Method

We begin with Newton's Method, a standard iterative procedure for solving the non-linear system $\nabla\mathcal{C}(\boldsymbol{\beta}) = \mathbf{0}$. To find the MLE of the GLM parameter $\boldsymbol{\beta}$, the method uses a second-order Taylor expansion to iteratively update the parameter estimates. Let $t \geq 0$ denote the current iteration step. Starting from an initial guess $\widehat{\boldsymbol{\beta}}^{(0)}$, the estimate is updated by evaluating the gradient $\nabla\mathcal{C}$ and the Hessian matrix $H_{\mathcal{C}}$ at the current parameter vector $\widehat{\boldsymbol{\beta}}^{(t)}$:

$$\widehat{\boldsymbol{\beta}}^{(t+1)} = \widehat{\boldsymbol{\beta}}^{(t)} - H_{\mathcal{C}}^{-1}\big(\widehat{\boldsymbol{\beta}}^{(t)}\big)\nabla\mathcal{C}\big(\widehat{\boldsymbol{\beta}}^{(t)}\big). \tag{SM.2.5}$$

To apply this update rule, we need first to define the general equations for the gradient and Hessian for any given $\boldsymbol{\beta}$, which are then evaluated at $\widehat{\boldsymbol{\beta}}^{(t)}$ during each loop. Equations (SM.2.6) and (SM.2.7) show these general computations, and the canonical LF in (SM.2.4) simplifies them considerably.

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

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

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

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

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

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

By substituting these general formulas back into the iterative step from (SM.2.5), we can write the update in matrix form. One can write $H_{\mathcal{C}}(\widehat{\boldsymbol{\beta}}^{(t)}) = \mathbf{X}^\top\mathbf{W}^*\mathbf{X}$ and $\nabla\mathcal{C}(\widehat{\boldsymbol{\beta}}^{(t)}) = \mathbf{X}^\top\mathbf{z}^*$, where $\mathbf{X} \in \mathbb{R}^{n\times(p+1)}$ includes a column of ones for the intercept, and the diagonal matrix $\mathbf{W}^*$ and vector $\mathbf{z}^*$ depend on the iteration-specific values $\eta_i = \mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}^{(t)}$ and $\theta_i = (b')^{-1} \circ h(\mathbf{x}_i^\top\widehat{\boldsymbol{\beta}}^{(t)})$.

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. Computational challenges may arise if the global minimum is not an interior point, a case in which Newton's Method fails to converge. Additionally, inverting the Hessian matrix can be computationally expensive and numerically unstable when the problem size is large. Therefore, in the next section, we describe the Fisher Scoring Method, which replaces the observed Hessian with the Fisher information matrix in the update step to mitigate some of these inversion difficulties.

SM.2.3 Fisher Scoring Method

Fisher Scoring is another iterative procedure for estimating GLMs. Unlike Newton's Method, which uses the observed Hessian in (SM.2.7), Fisher Scoring replaces the Hessian with its expected value, known as the Fisher information matrix. That is, (SM.2.7) changes by replacing y_i with its expectation $b'(\theta_i)$, and in turn, the (j,k) entry of the Fisher information matrix 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.

It is imperative to note that Fisher Scoring, Newton's method, and IRLS coincide only when canonical LFs are used. However, GLMs are widely employed in general insurance pricing, which remains one of their primary applications in actuarial practice. In this context, premiums are typically required to follow multiplicative rating structures, ensuring transparency and interpretability—features that are both industry standards and regulatory expectations. Such structures are naturally obtained by adopting a log LF, which is the canonical LF for the Poisson GLM, but not for the Gamma GLM and, more generally, for commonly used Tweedie GLMs (i.e., Poisson–Gamma models; see (Delong et al., 2021) for further details). Consequently, while Fisher Scoring and Newton's method are foundational concepts in GLM theory, IRLS remains the standard algorithm for fitting non-penalised GLMs in practice.

SM.2.4 IRLS Procedure

As an alternative to direct Hessian inversion, the IRLS algorithm approximates a stationary point of (SM.2.3) through an iteration procedure where a WLS instance (with given weights) is run in every loop. This explains why IRLS is the standard GLM solver given its computational efficiency. Furthermore, IRLS variants that employ step-halving can further improve convergence which is implemented in the **R** CRAN package `glm2`. The starting point is maximising the log-likelihood in (SM.2.3), which implies finding the corresponding stationary points:

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

where ω_i are some exogenous weights if available, otherwise, all are assumed to be 1; note that $\mu_i = h(\eta_i)$ and $V(\mu_i)$ is the variance function. 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. These equations are equivalent to minimising a 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

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

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

$$z_i^{(k)} = \eta_i^{(k)} + \frac{y_i - \mu_i^{(k)}}{h'\left(\eta_i^{(k)}\right)}, \quad W_{ii}^{(k)} = \frac{\left(h'\left(\eta_i^{(k)}\right)\right)^2}{V\left(\mu_i^{(k)}\right)} \quad \text{for all } j \in \{1, \ldots, n\}, \tag{SM.2.8}$$

with $\mu_i^{(k)} = h(\eta_i^{(k)})$ and h' being the derivative of the inverse LF. In summary, the IRLS algorithm consists of two main components: *initialisation* and an *iterative procedure*.

Initialisation: set starting values $\mu_i^{(0)} = y_i$ and $\eta_i^{(0)} = h^{-1}(y_i)$, and adjust to ensure valid choices (e.g., $\mu_i^{(0)} > 0$ for the *log* LF). Compute $\mathbf{z}^{(0)}$ and $\mathbf{W}^{(0)}$ via (SM.2.8), and obtain the initial estimate

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

Iterative procedure: 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 (SM.2.8).

(b) Update the parameter estimates by solving the weighted least squares problem

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

(c) Stop when a convergence criterion is satisfied. Default criteria vary across implementations, and the three main IRLS implementations are given below:

- **R** (`glm.fit2`): relative change in deviance

$$\frac{\left|\text{Dev}^{(k+1)} - \text{Dev}^{(k)}\right|}{0.1 + \left|\text{Dev}^{(k+1)}\right|} < \tau, \quad \text{where } \tau \text{ (typically } 10^{-8}\text{) is the tolerance level;}$$

  step-halving is applied if the deviance increases.

- **MATLAB** (`fitglm`): change in coefficients

$$\max_i \left|\beta_i^{(k+1)} - \beta_i^{(k)}\right| \leq \tau \max\left(\sqrt{\epsilon}, \max_i \left|\beta_i^{(k)}\right|\right), \text{ where } \tau \approx 10^{-6} \text{ and } \epsilon \approx 2.2 \times 10^{-16}.$$

- **Python** (`statsmodels.GLM`): absolute change in deviance

$$\left|\text{Dev}^{(k+1)} - \text{Dev}^{(k)}\right| \leq \tau, \quad \text{with } \tau \text{ typically set to } 10^{-8}.$$

SM.2.5 Algorithmic Comparison for Gamma GLM with *log* LF

For a Gamma distribution with mean μ_i and variance $V(\mu_i) = \mu_i^2$, the canonical parameter is $\theta_i = -1/\mu_i$, and the cumulant function is $b(\theta_i) = -\log(-\theta_i) = \log(\mu_i)$. Under a *log* LF, we have $\eta_i = \mathbf{x}_i^\top \boldsymbol{\beta}$ and $\mu_i = h(\eta_i) = \exp(\eta_i)$, which implies $\theta_i = -\exp(-\eta_i)$. Substituting θ_i and $b(\theta_i)$ into the general objective function from (SM.2.3) yields the negative log-likelihood (ignoring constants independent of $\boldsymbol{\beta}$) for the Gamma GLM:

$$\mathcal{C}(\boldsymbol{\beta}) = -\sum_{i=1}^{n} w_i \left(-\frac{y_i}{\mu_i} - \log \mu_i \right) = \sum_{i=1}^{n} w_i \big(y_i \exp\big(-\mathbf{x}_i^\top \boldsymbol{\beta}\big) + \mathbf{x}_i^\top \boldsymbol{\beta} \big),$$

where w_i represents the weight. To find the MLE for $\boldsymbol{\beta}$, we need to minimise this objective function. Then substituting into the general gradient formula (SM.2.6) yields:

$$\frac{\partial \mathcal{C}(\boldsymbol{\beta})}{\partial \beta_j} = \sum_{i=1}^{n} w_i (\mu_i - y_i) \frac{\mu_i}{\mu_i^2} x_{ij} = \sum_{i=1}^{n} w_i \left(1 - \frac{y_i}{\mu_i} \right) x_{ij}.$$

In matrix notation, let $\mathbf{w} = [w_1, \ldots, w_n]^\top$ be the vector of prior weights. The gradient vector is given by:

$$\nabla \mathcal{C}(\boldsymbol{\beta}) = \mathbf{X}^\top \left[\mathbf{w} \circ \left(\mathbf{1} - \frac{\mathbf{y}}{\boldsymbol{\mu}} \right) \right] = \mathbf{X}^\top \operatorname{diag}(\mathbf{w}) \left(\mathbf{1} - \frac{\mathbf{y}}{\boldsymbol{\mu}} \right),$$

Newton's Method updates the parameter vector using the exact *observed* curvature of the data. We substitute our derivatives into the general Hessian formula (SM.2.7) to find the second derivative of the objective function, $H_{\mathcal{C}}(\boldsymbol{\beta})$. Therefore, the (j, k) entry of the Hessian matrix simplifies to:

$$\big(H_{\mathcal{C}}(\boldsymbol{\beta}) \big)_{jk} = \sum_{i=1}^{n} w_i \left\{ 1 + (\mu_i - y_i) \left(-\frac{1}{\mu_i} \right) \right\} x_{ij} x_{ik} = \sum_{i=1}^{n} w_i \frac{y_i}{\mu_i} x_{ij} x_{ik}.$$

In matrix notation, this observed Hessian is given by:

$$H_{\mathcal{C}}(\boldsymbol{\beta}) = \mathbf{X}^\top \operatorname{diag} \left(\mathbf{w} \circ \frac{\mathbf{y}}{\boldsymbol{\mu}} \right) \mathbf{X}. \tag{SM.2.9}$$

This positive definite matrix and the gradient are substituted directly into Newton's Method update step (SM.2.5). The computation steps are explicitly detailed in Algorithm 1.

Algorithm 1 Newton's Method for Gamma GLM (*log* LF)

Input: Design matrix $\mathbf{X}$, claims $\mathbf{y}$, weight vector $\mathbf{w}$, max iterations K, tolerance ε.
Output: MLE coefficients $\widehat{\boldsymbol{\beta}}$
Initialisation: Set $\boldsymbol{\beta}^{(0)}$ via standard WLS initialization, $\text{Dev}_{\text{old}} \leftarrow \infty$.
for $k = 1, \ldots, K$ **do**
    Compute linear predictor $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta}^{(k-1)}$ and mean $\boldsymbol{\mu} = \exp(\boldsymbol{\eta})$.
    Compute gradient $\nabla\mathcal{C} = \mathbf{X}^\top \operatorname{diag}(\mathbf{w})\Big(\mathbf{1} - \frac{\mathbf{y}}{\boldsymbol{\mu}}\Big)$.
    Compute observed Hessian $H_{\mathcal{C}} = \mathbf{X}^\top \operatorname{diag}\Big(\mathbf{w} \circ \frac{\mathbf{y}}{\boldsymbol{\mu}}\Big)\mathbf{X}$.
    Update coefficients: $\boldsymbol{\beta}^{(k)} = \boldsymbol{\beta}^{(k-1)} - H_{\mathcal{C}}^{-1}\nabla\mathcal{C}$.
    Compute Gamma deviance: $\text{Dev}_{\text{new}} = 2\sum_{i=1}^{n} w_i\Big(\frac{y_i}{\mu_i} - \log\frac{y_i}{\mu_i} - 1\Big)$.
    if $|\text{Dev}_{\text{old}} - \text{Dev}_{\text{new}}|/(0.1 + |\text{Dev}_{\text{new}}|) < \varepsilon$ **then break**
    $\text{Dev}_{\text{old}} \leftarrow \text{Dev}_{\text{new}}$
end for
return $\boldsymbol{\beta}^{(k)}$

Fisher Scoring smooths the parameter updates by replacing the observed Hessian with the Fisher Information matrix, which is the expected value of $H_{\mathcal{C}}(\boldsymbol{\beta})$. Since $\mathrm{E}[\mathbf{y}] = \boldsymbol{\mu}$, taking the expectation of (SM.2.9) simplifies the matrix to:

$$H_{\mathcal{C}}^{FS}(\boldsymbol{\beta}) = \mathbb{E}\big[H_{\mathcal{C}}(\boldsymbol{\beta})\big] = \mathbf{X}^\top \operatorname{diag}\left(\mathbf{w} \circ \frac{\mathbb{E}[\mathbf{y}]}{\boldsymbol{\mu}}\right)\mathbf{X} = \mathbf{X}^\top \operatorname{diag}(\mathbf{w})\mathbf{X}.$$

For the Gamma distribution with a *log* LF, $H_{\mathcal{C}}^{FS}$ is entirely independent of both $\boldsymbol{\beta}$ and $\mathbf{y}$. On one hand, this provides a computational advantage that the matrix $\mathbf{X}^\top \operatorname{diag}(\mathbf{w})\mathbf{X}$ is constant and can be inverted once before the iterative loop begins, greatly increasing the processing speed per iteration. On the other hand, the critical disadvantage is a significant loss of information. The computation steps are explicitly detailed in Algorithm 2.

Algorithm 2 Fisher Scoring for Gamma GLM (*log* LF)

Input: Design matrix $\mathbf{X}$, claims $\mathbf{y}$, weight vector $\mathbf{w}$, max iterations K, tolerance ε.
Output: MLE coefficients $\widehat{\boldsymbol{\beta}}$
Initialisation: Set $\boldsymbol{\beta}^{(0)}$ via standard WLS initialization, $\text{Dev}_{\text{old}} \leftarrow \infty$.
Precompute Expected Hessian: $H_{\mathcal{C}}^{FS} = \mathbf{X}^\top \operatorname{diag}(\mathbf{w})\mathbf{X}$ and its inverse $(H_{\mathcal{C}}^{FS})^{-1}$.
for $k = 1, \ldots, K$ **do**
    Compute linear predictor $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta}^{(k-1)}$ and mean $\boldsymbol{\mu} = \exp(\boldsymbol{\eta})$.
    Compute gradient $\nabla\mathcal{C} = \mathbf{X}^\top \operatorname{diag}(\mathbf{w})\Big(\mathbf{1} - \frac{\mathbf{y}}{\boldsymbol{\mu}}\Big)$.
    Update coefficients: $\boldsymbol{\beta}^{(k)} = \boldsymbol{\beta}^{(k-1)} - (H_{\mathcal{C}}^{FS})^{-1}\nabla\mathcal{C}$.
    Compute Gamma deviance: $\text{Dev}_{\text{new}} = 2\sum_{i=1}^{n} w_i\Big(\frac{y_i}{\mu_i} - \log\frac{y_i}{\mu_i} - 1\Big)$.
    if $|\text{Dev}_{\text{old}} - \text{Dev}_{\text{new}}|/(0.1 + |\text{Dev}_{\text{new}}|) < \varepsilon$ **then break**
    $\text{Dev}_{\text{old}} \leftarrow \text{Dev}_{\text{new}}$
end for
return $\boldsymbol{\beta}^{(k)}$

To show the differences between these frameworks, we implement Algorithms 1 and 2 in **R** and compare them to the standard IRLS implementation (**R**'s `glm.fit2`). We assess their performance on the `freMTPL2` dataset on average severity per policy using the 5-fold cross-validation resampling framework outlined in Section 4.2.2. All three solvers fit a Gamma GLM with a *log* LF using the same default starting values. Table SM.2.1 shows the OOS performance ratios compared to the standard IRLS.

Table SM.2.1: **OOS Performance Ratios for GLM Solvers (Gamma GLM, *log* LF) on freMTPL2 Severity**

	Winsorised RMSE		Winsorised MAE			Gini				
Solver	**95%**	**99%**	**95%**	**99%**	**Deviance**	**Raw**	**Norm**	**A/E - 1**	**CT**	**NoIt**
Panel A) Exactly 1 Claim ($n = 23,571$)										
IRLS (glm2)	**1.000**	1.000	**1.000**	**1.000**	**1.000**	0.115	0.172	1.35%	1.000	1.000
Fisher Scoring	1.029	1.074	1.026	1.103	1.020	**0.115**	**0.173**	**-1.34%**	0.423	1.247
Newton's Method	**1.000**	**1.000**	1.000	**1.000**	**1.000**	0.115	0.172	1.36%	**0.249**	**0.292**
Panel B) Exactly 2 Claims ($n = 1,298$)										
IRLS (glm2)	1.000	1.000	1.000	1.000	1.000	0.179	0.271	11.95%	1.000	1.000
Fisher Scoring	1.079	1.304	1.094	1.398	2.482E+06	0.173	0.261	**1.64%**	0.290	1.749
Newton's Method	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**0.179**	**0.271**	11.96%	**0.250**	**0.335**
Panel C) All Claims $\geq$ **1** ($n = 24,944$)										
IRLS (glm2)	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**0.105**	0.157	**2.18%**	1.000	1.000
Fisher Scoring	1.437	2.296	1.396	2.759	4.863E+05	0.102	0.151	-43.39%	0.420	6.510
Newton's Method	1.000	**1.000**	1.000	1.000	**1.000**	**0.105**	**0.157**	**2.17%**	**0.245**	**0.276**

Notes. This table reports the OOS performance ratios for the fundamental optimisation solvers relative to the standard `glm2` (IRLS) benchmark on the `freMTPL2` dataset. Results are averaged over 500 replications utilising a Gamma distribution with a *log* LF. An A/E deviation closer to 0.00% indicates superior portfolio calibration. For distance metrics like RMSE, MAE, Deviance, CT, NoIt, lower is better. For the raw Gini and normalised Gini index, higher is better. The best performing solver in each column is highlighted in **bold red**, and the second best is underlined. Regarding convergence across the 500 replications: Newton's Method successfully converged in every instance across all panels. The standard IRLS benchmark experienced only 1 failure in Panel C. In contrast, Fisher Scoring failed to converge 12 times in Panel A, 53 times in Panel B, and 136 times in Panel C.

Newton's Method and the standard IRLS converge to the same optimal coefficients. They produce identical predictive performance, with RMSE, MAE, and Deviance ratios of 1.000. However, their internal mechanics differ. Newton's Method uses the observed data curvature. It needs fewer iterations (NoIt) and less computational time (CT). In contrast, Fisher Scoring relies on an expected curvature and fails to converge in many instances. Because this expected curvature ignores the actual observed response, the Fisher scoring method is inaccurate when calculating the step size for highly skewed data. As noted in the table, Fisher Scoring fails to converge 12 times in Panel A, 53 times in Panel B, and 136 times in Panel C. When it does converge, it produces high OOS Deviance errors in Panel B and Panel C. The standard IRLS algorithm operates differently. At each iteration, it updates the parameters by solving a WLS problem. It matches the stable predictions of Newton's Method as it uses an internal step-halving mechanism to catch and correct divergent steps. This shows that Newton's Method is efficient and stable on its own, while the standard IRLS depends on step-halving to handle volatile data.

SM.3 Details about the Simulated Data Analysis

This section details the simulation experiments used to evaluate the GLM estimation methods. We compare the standard IRLS baseline (**R**'s `glm2`) against our modified IRLS framework, which incorporates the four proposed shrinkage estimators (SR, GSR, St, and DSh) along with the LW (Ledoit and Wolf, 2004) and QIS (Ledoit and Wolf, 2022) estimators. We also benchmark against standard regularised GLMs (RR and EN), fitted via coordinate descent using the **R** package `glmnet`. The modified IRLS estimators are implemented in our new **R** CRAN package *savvyGLM*[SM.3.4]. All IRLS algorithms (via `glm2` and *savvyGLM*) use the optimisation method detailed in Section SM.5.2 to generate starting values, alongside a maximum limit of 250 iterations and a convergence tolerance of 10^{-6}. In contrast, RR and EN use the default iteration limits and tolerances of `glmnet`, as they rely on nested loops rather than a single sequence of WLS updates. In Section SM.3.2, 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 SM.3.3, and ii) computational efficiency as measured by the number of iterations required for convergence in Section SM.3.4.

SM.3.1 Data Generation Process

This section describes how data are generated in this paper. We use two variants, namely, *the first data generation process (DGP1)* and *the second data generation process (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, including *Logistic Regression (LR)*, *Poisson Regression (PoR)*, and *Gamma Regression (GaR)*, along with various LF choices. Specifically, we only consider the canonical LF for LR, which is the *logit* LF, i.e., $g(\mu) = \log\big(\mu/(1-\mu)\big)$; its inverse LF is $h(\eta) = \big(1+e^{-\eta}\big)^{-1}$. Further, we include *log* LF $g(\mu) = \log(\mu)$ and *sqrt* 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., 2025).

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.

[SM.3.4] Available at: https://cran.r-project.org/web/packages/savvyGLM/index.html

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 $Y_i \sim \mathit{Binomial}(1, (1+e^{-\eta})^{-1})$. 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 $Y_i \sim \mathit{Gamma}(\mu_i, 1)$ where $\mu_i = \eta_i^2$ and $\mu_i = e^{\eta_i}$ for *sqrt* LF and *log* LF, respectively.

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

SM.3.2 Parametrisations 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 Sections SM.2.4 and SM.3.1, 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) 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 SM.3.1 summarises the true response distributions and predictor structures used in this paper.

Table SM.3.1: Default Starting Values and Validations across GLM Implementations

Model	LF	True Response	Predictor Used in GLM
LR	*logit*	$Y_i \sim Binomial\big(1, 1/(1+e^{-\eta_i})\big)$	$h(\eta_i) = 1/(1+e^{-\eta_i})$
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*	$Y_i \sim Gamma(e^{\eta_i}, 1)$	$h(\eta_i) = e^{\eta_i}$

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.

SM.3.3 Analysis of Estimation Accuracy

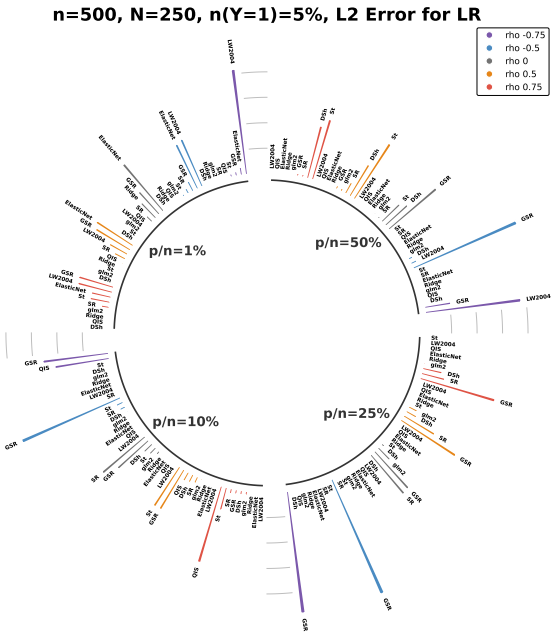
The estimation accuracy of the shrinkage-based IRLS approaches was evaluated against the benchmark `glm2` estimator, as well as standard regularised models, specifically RR and the EN. Within our modified IRLS framework, we investigate our four proposed novel shrinkage estimators (SR, GSR, St, and DSh) alongside two established shrinkage estimators from the literature (LW and QIS). We provide a comprehensive discussion of their comparative performance by analysing the relative mean L_2 errors (RML_2) under various statistical scenarios. The detailed results for the *logit*, *sqrt*, and *log* LFs are reported in Tables SM.3.2–SM.3.4. To facilitate the identification of the main trends across all 140 simulated scenarios, the aggregated "first" and "second-best" performances are summarised in Table 1. Furthermore, Figures SM.3.1 and SM.3.2 visually illustrate the claim frequency with which each model achieved the lowest L_2 error over 250 replications.

Overall, the simulation results demonstrate that the shrinkage estimators embedded within our modified IRLS algorithm effectively stabilise parameter estimates and generally outperform `glm2` across a wide range of settings. As shown in the Grand Total column of Table 1, our four proposed methods (SR, GSR, St, and DSh) achieve the lowest L_2 error in 52.8% (74 out of 140) of all tested scenarios. However, the performance superiority of the estimators is highly dependent on the chosen LF and the underlying data distribution.

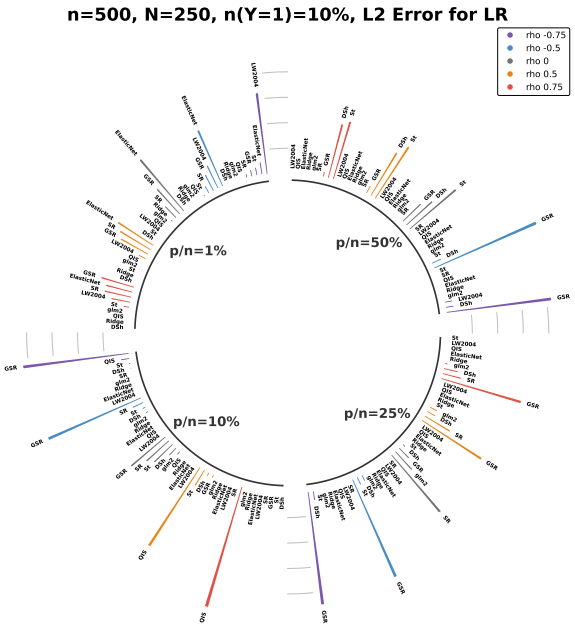
For the LR model using the *logit* LF (Table SM.3.2), the proposed GSR estimator appears as the dominant method, achieving the lowest error in 25 out of 60 scenarios. The benefits of shrinkage are pronounced when the predictor-to-sample size ratio p/n is low and when dealing with strong negative correlations ($\rho < 0$) in balanced datasets. As p/n increases, the performance gains decrease, and the shrinkage estimators behave more similarly to `glm2`. For the Poisson and Gamma GLMs, the choice of LF dictates the optimal estimator. Under the *sqrt* LF (Table SM.3.3), the St estimator consistently provides the largest error reductions for the Poisson distribution, especially at higher p/n ratios. Conversely, for the Gamma distribution under the *sqrt* LF, the QIS estimator performs robustly, slightly outperforming our proposed methods. When the *log* LF is applied to the Poisson and Gamma models (Table SM.3.4), the behaviour of the estimators changes significantly. In these settings, the standard RR dominates,

achieving the best performance in 29 out of 40 scenarios. While our proposed IRLS-based estimators (particularly GSR and St) still provide positive improvements over `glm2`, the standard RR and EN estimators handle the *log* link structure more effectively in both distributions.

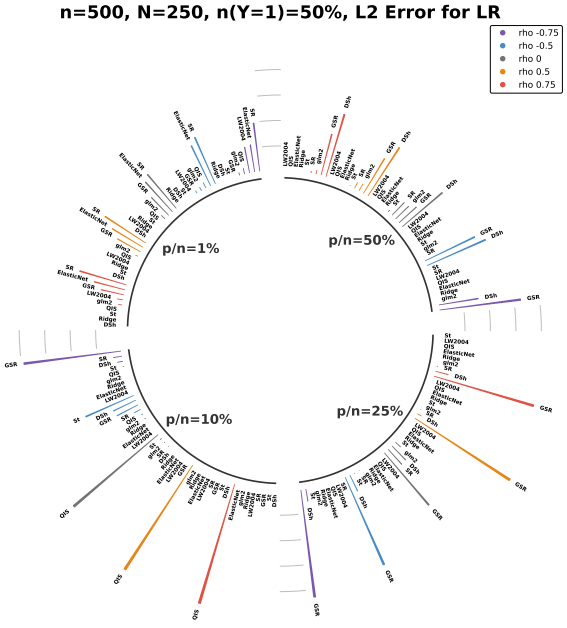
In summary, replacing the WLS step with shrinkage estimators generally yields lower or comparable estimation errors relative to `glm2`. Our modified IRLS shrinkage framework is highly effective for *logit* and *sqrt* LFs, while standard regularisation techniques like RR remain preferable when a *log* LF is specified.



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



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



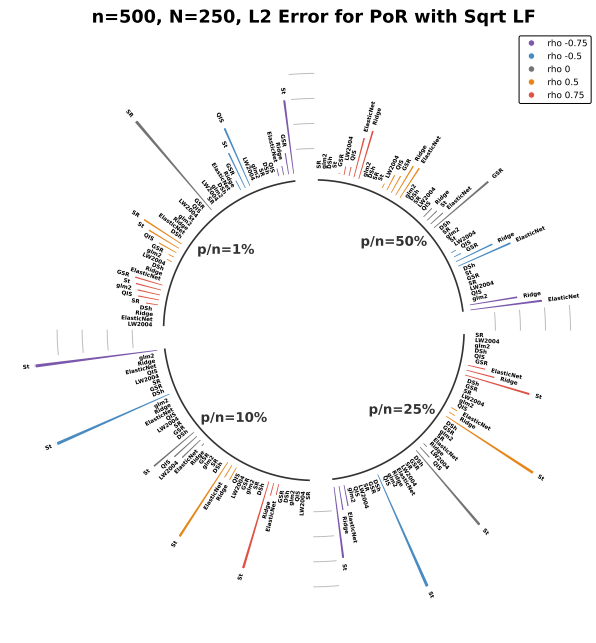
(c) LR with Balanced data ($n(Y=0)=50n$)

Figure SM.3.1: 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.

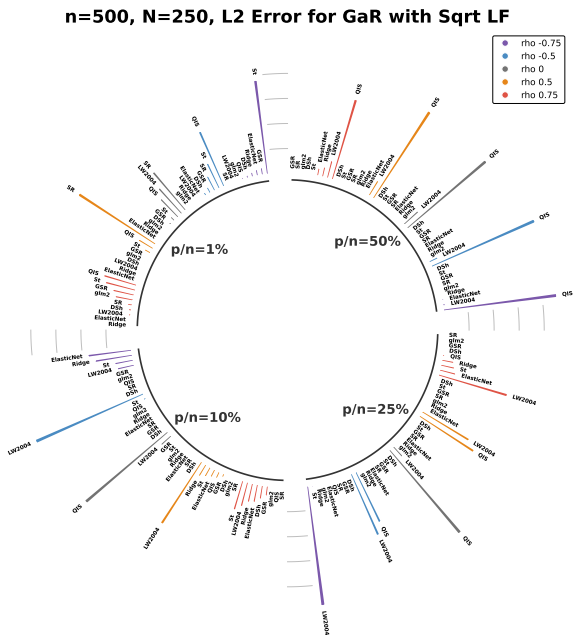
Table SM.3.2: **Relative Mean L_2 Errors For LR**

ρ	-0.75				-0.5				0				0.5				0.75			
p/n	1%	10%	25%	50%	1%	10%	25%	50%	1%	10%	25%	50%	1%	10%	25%	50%	1%	10%	25%	50%
Panel A) Extreme Rare Event Case: $\frac{n(Y=0)}{n(Y=1)} = \frac{5}{95}$																				
RR	18.19	-13.33	-1.91	-0.33	8.84	-41.41	-3.76	-0.66	2.68	-123.77	-7.70	-1.29	-5.93	-117.47	-14.68	-2.33	-17.72	-18.70	-23.47	-3.58
EN	25.47	-11.57	-1.86	-0.34	**18.37**	-39.73	-3.79	-0.68	**12.50**	-121.27	-7.77	-1.31	**7.81**	-111.06	-14.71	-2.33	**4.92**	-10.96	-23.48	-3.58
QIS	3.21	1.05	-0.50	-0.02	2.57	-28.69	-2.94	-0.41	2.33	-99.31	-7.17	-1.07	2.47	-81.08	-13.67	-2.12	1.67	**6.68**	-21.57	-3.30
LW	**30.91**	-12.11	-1.46	**0.54**	17.84	-41.37	-3.53	-0.23	-11.16	-126.94	-7.48	-1.12	-0.87	-120.71	-14.25	-2.19	1.75	-19.10	-22.85	-3.44
SR	0.51	0.08	0.01	0.00	0.86	0.14	0.01	0.00	1.79	**0.86**	**0.04**	0.00	1.90	0.05	0.03	0.00	1.11	0.00	0.02	0.00
GSR	3.17	**2.14**	**0.26**	0.07	3.48	**1.73**	**0.24**	**0.08**	4.75	0.81	0.02	**0.04**	4.90	0.35	**0.09**	-0.07	3.86	-1.22	**0.27**	-0.08
St	3.00	0.58	0.01	0.00	2.55	0.30	-0.05	0.01	2.26	0.29	-0.05	0.02	2.97	**0.51**	-0.10	**0.04**	3.59	-0.25	-0.17	**0.07**
DSh	1.78	-1.42	-0.09	0.00	1.60	-1.36	-0.15	0.01	1.31	-1.00	-0.21	0.01	1.43	-0.31	-0.26	-0.01	1.01	-0.40	-0.36	0.05
Panel B) Rare Event Case: $\frac{n(Y=0)}{n(Y=1)} = \frac{10}{90}$																				
RR	-16.66	-27.21	-3.17	-0.53	-22.28	-79.15	-5.96	-1.01	-26.29	-178.50	-11.81	-1.97	-37.06	-18.07	-22.66	-3.58	-55.20	42.03	-37.83	-5.52
EN	16.62	-23.85	-3.02	-0.54	**13.73**	-74.95	-5.93	-1.03	**9.86**	-168.54	-11.87	-2.00	**5.42**	-8.94	-22.85	-3.59	3.46	49.04	-38.02	-5.52
QIS	3.51	-9.17	-1.80	-0.27	2.81	-60.03	-5.30	-0.83	2.45	-127.42	-11.75	-1.83	2.64	**17.27**	-22.46	-3.39	1.52	**67.63**	-37.11	-5.22
LW	**21.03**	-25.63	-2.87	-0.19	5.90	-79.05	-5.91	-0.85	-37.87	-183.23	-11.97	-1.87	-15.39	-19.84	-22.91	-3.45	-6.97	42.12	-38.13	-5.36
SR	0.94	0.16	0.01	0.00	1.12	0.23	0.01	0.00	3.25	0.07	**0.04**	0.00	4.15	-0.12	0.03	0.00	2.09	-0.05	0.02	0.00
GSR	3.13	**2.89**	**0.33**	**0.09**	3.65	**2.22**	**0.28**	**0.09**	5.64	**0.35**	-0.07	0.01	4.67	-0.92	**0.14**	-0.04	**3.74**	-1.35	**0.36**	-0.07
St	3.03	0.75	0.04	0.01	2.40	0.34	0.08	0.01	2.18	0.15	-0.09	**0.05**	2.50	0.25	-0.14	**0.09**	2.83	0.05	-0.23	**0.14**
DSh	1.28	-2.75	-0.05	0.01	1.43	-1.71	-0.06	0.01	1.38	-0.82	-0.24	0.01	1.27	0.10	-0.26	0.07	0.69	0.00	-0.40	0.12
Panel C) Balanced: $\frac{n(Y=0)}{n(Y=1)} = \frac{50}{50}$																				
RR	-260.77	-64.39	-6.79	-1.07	-348.56	-129.85	-12.44	-1.91	-405.86	-1.36	-24.99	-3.58	-333.09	59.62	-53.90	-6.44	-286.27	77.09	-107.18	-10.15
EN	0.00	-56.65	-6.33	-1.04	-7.91	-117.25	-12.11	-1.91	-12.66	8.54	-24.53	-3.60	-4.91	66.47	-52.49	-6.47	-2.25	82.74	-103.28	-10.19
QIS	3.84	-36.22	-5.41	-0.86	1.86	-92.03	-11.97	-1.84	0.18	**35.97**	-25.49	-3.61	0.36	**85.48**	-54.61	-6.52	0.09	**94.60**	-107.54	-10.24
LW	-9.08	-61.51	-6.49	-0.96	-109.22	-128.93	-12.47	-1.89	-325.98	-3.10	-25.58	-3.63	-129.35	59.24	-54.73	-6.55	-46.01	77.40	-107.71	-10.30
SR	**8.46**	0.65	0.02	0.00	**7.63**	0.50	0.03	0.00	**4.74**	-0.28	0.03	0.00	**4.29**	0.00	0.00	0.00	**2.47**	0.01	0.00	0.00
GSR	-4.11	**4.05**	**0.50**	**0.10**	-3.34	0.66	**0.40**	0.08	4.05	-1.20	**0.56**	0.02	3.09	-0.97	**1.57**	0.12	1.86	-0.68	**3.29**	0.21
St	1.64	0.06	0.01	-0.01	1.55	**1.51**	-0.06	0.00	0.64	0.05	0.30	-0.01	0.39	0.23	0.46	0.03	0.30	0.14	1.01	0.06
DSh	-0.75	-2.90	0.19	0.08	-0.41	-0.50	0.09	**0.09**	-2.26	-0.10	0.41	**0.03**	-2.71	0.08	0.75	**0.12**	-3.01	0.03	1.81	**0.29**

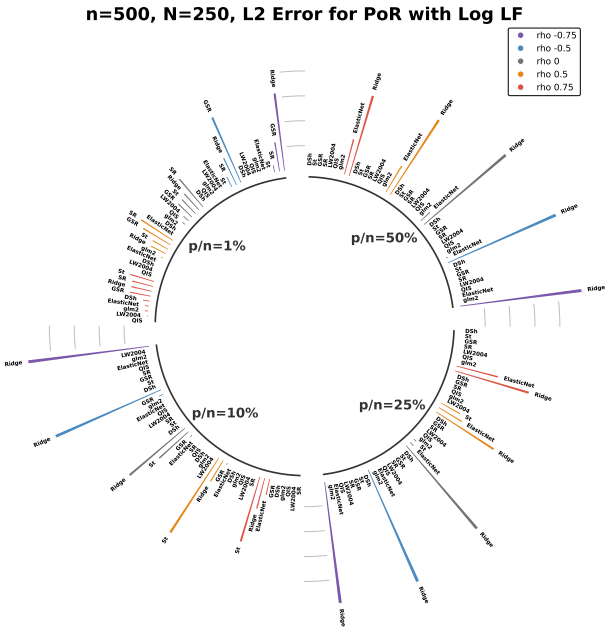
Notes. This table reports the relative Mean L_2 errors for the LR model using the *logit* LF, expressed as percentage improvements relative to the benchmark `glm2` estimator. A positive value indicates a percentage reduction in the L_2 error (better performance), while a negative value indicates a percentage increase (worse performance) compared to `glm2`. **Bold red** numbers mark the best performance, and underlined numbers indicate the second-best performance for each setting. Results are evaluated across varying predictor-to-sample size ratios p/n and correlation structures ρ. All models were initialised with the same starting values, computed via the optimisation-based procedure detailed in Section SM.5.2. Note that not all models converged in all 250 replications when $n = 500$; see Section SM.5 for convergence details.



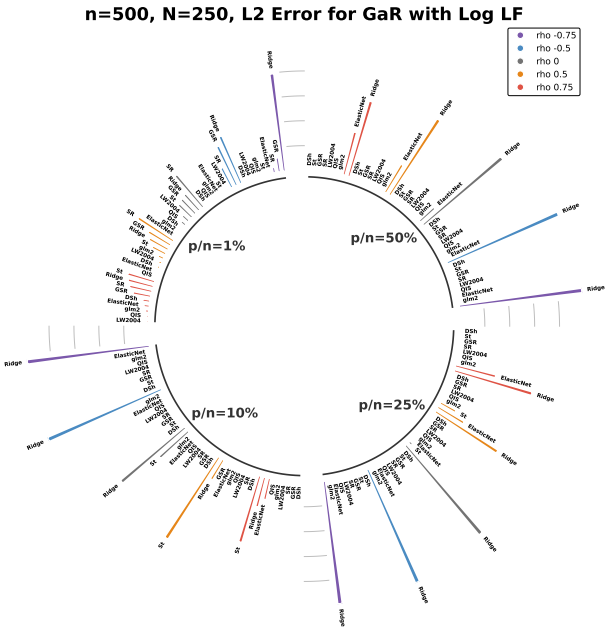
(a) PoR with *sqrt* LF



(b) GaR with *sqrt* LF



(c) PoR with *log* LF



(d) GaR with *log* LF

Figure SM.3.2: 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.

Table SM.3.3: **Relative Mean L_2 Errors For PoR and GaR with *sqrt* LF**

ρ	-0.75				-0.5				0				0.5				0.75			
p/n	1%	10%	25%	50%	1%	10%	25%	50%	1%	10%	25%	50%	1%	10%	25%	50%	1%	10%	25%	50%
Panel A) Poisson Distribution																				
RR	0.00	5.44	13.48	28.64	-0.68	1.48	4.64	12.78	-1.69	0.22	0.60	4.17	-2.49	0.01	1.18	4.60	-4.09	-0.66	1.88	**5.80**
EN	0.07	5.43	13.49	**28.75**	-0.62	1.47	4.64	**12.88**	-1.41	0.24	0.61	4.24	-1.83	0.02	1.17	**4.62**	-2.30	-0.66	1.88	5.79
QIS	-1.85	2.29	9.67	17.25	**0.54**	1.21	6.29	10.16	-0.99	-0.46	4.92	4.39	0.05	0.37	2.04	3.79	-0.28	0.16	1.61	3.92
LW	0.08	2.43	10.34	17.07	0.04	1.49	6.57	9.92	-0.89	0.20	4.82	4.29	-0.21	0.48	2.14	3.73	-0.50	0.22	1.71	3.82
SR	-4.69	-0.29	0.19	0.26	-3.11	-0.40	0.07	0.13	**4.16**	0.02	0.19	0.10	-0.33	0.33	0.09	0.00	-2.33	0.17	0.02	0.01
GSR	0.45	3.90	9.04	13.78	0.23	2.32	5.91	9.91	0.22	1.67	3.70	**5.94**	-0.12	0.76	2.04	3.74	-0.09	0.48	1.47	3.16
St	**0.72**	**13.46**	**18.39**	6.44	0.44	**5.84**	**13.31**	6.81	0.21	**2.40**	**6.16**	5.24	**0.11**	**1.96**	**4.59**	2.55	**0.01**	**2.11**	**3.10**	1.63
DSh	0.02	3.07	8.23	10.75	-0.31	1.17	3.31	8.00	-0.42	0.40	1.13	4.52	-0.34	0.40	1.11	3.07	-0.36	0.40	1.38	2.55
Panel B) Gamma Distribution																				
RR	-0.41	11.88	30.30	48.81	-1.39	5.04	16.25	30.79	-3.38	1.95	7.70	16.62	-7.96	1.55	6.29	14.42	-12.44	1.31	6.60	15.43
EN	0.40	**12.19**	30.59	49.11	-0.20	5.41	16.62	31.08	-1.00	2.27	7.99	16.87	-2.16	1.68	6.47	14.44	-4.20	1.35	6.69	15.53
QIS	-0.99	8.34	30.81	**54.04**	**0.72**	6.62	24.37	**41.34**	0.38	**6.09**	**16.56**	**26.00**	0.20	2.30	**9.43**	**18.86**	-0.08	1.35	6.96	**16.80**
LW	0.15	11.34	**35.63**	46.65	0.14	**8.04**	**24.59**	36.26	0.54	4.23	14.78	23.40	-0.08	**2.79**	9.43	16.95	-0.18	**1.92**	**7.82**	15.06
SR	-0.17	-0.03	-0.03	0.33	0.14	-0.01	-0.09	-0.27	**0.66**	0.04	0.15	0.15	**1.33**	0.02	-0.04	-0.12	-1.59	0.03	0.09	0.09
GSR	0.62	8.41	14.02	16.34	0.41	4.39	6.11	9.63	0.26	2.23	2.21	4.33	-0.02	1.58	2.57	4.88	-0.17	1.51	3.38	6.34
St	**1.14**	8.04	16.66	22.82	0.56	3.81	5.08	8.70	0.28	1.70	2.74	4.96	0.16	1.36	2.00	4.11	**0.01**	1.54	4.42	8.13
DSh	0.31	8.51	22.37	34.36	0.13	4.42	12.95	24.24	0.02	2.15	7.15	14.16	-0.25	1.59	5.35	11.70	-0.43	1.44	5.29	11.89

Notes. This table reports the relative Mean L_2 errors for the PoR and GaR models using the *sqrt* LF, expressed as percentage improvements relative to the benchmark `glm2` estimator. A positive value indicates a percentage reduction in the L_2 error (better performance), while a negative value indicates a percentage increase (worse performance) compared to `glm2`. Bold red numbers mark the best performance, and underlined numbers indicate the second-best performance for each setting. Results are evaluated across varying predictor-to-sample size ratios p/n and correlation structures ρ. All models were initialised with the same starting values, computed via the optimisation-based procedure detailed in Section SM.5.2. Note that not all models converged in all $N = 250$ replications when $n = 500$; see Section SM.5 for convergence details.

Table SM.3.4: **Relative Mean L_2 Errors For PoR and GaR with *log* LF**

ρ	-0.75				-0.5				0				0.5				0.75			
p/n	1%	10%	25%	50%	1%	10%	25%	50%	1%	10%	25%	50%	1%	10%	25%	50%	1%	10%	25%	50%
Panel A) Poisson Distribution																				
RR	**41.49**	**62.28**	**75.02**	**86.87**	21.66	**43.86**	**61.13**	**78.93**	-2.38	**17.11**	**40.71**	**67.18**	-21.48	6.82	**41.47**	**71.00**	-26.89	27.91	**58.99**	**79.79**
EN	10.60	32.59	60.39	82.06	2.64	21.52	50.89	76.25	-3.94	6.95	35.31	65.62	-29.16	-0.37	40.48	70.68	-35.30	27.22	58.89	79.69
QIS	1.43	16.91	38.81	67.73	1.48	15.98	35.64	63.08	-2.85	-0.23	15.05	45.76	1.30	11.51	29.92	56.39	1.29	14.27	33.96	61.22
LW	6.45	41.09	56.75	72.57	2.66	16.51	21.40	41.80	-7.51	-91.16	-80.72	-37.72	2.24	13.24	32.12	51.69	3.74	16.91	38.60	66.40
SR	17.41	2.11	1.06	0.56	12.67	2.15	0.76	0.55	**8.41**	1.29	0.64	0.34	4.13	0.21	0.12	0.07	1.49	0.02	0.03	0.02
GSR	35.30	31.68	31.30	30.53	**27.82**	28.25	29.59	29.69	0.56	10.08	21.03	27.13	**4.26**	16.11	24.36	28.91	5.01	21.96	28.00	30.30
St	11.05	12.41	24.68	37.93	6.93	18.71	30.19	40.65	2.80	16.38	30.25	41.44	4.21	**22.68**	35.44	43.89	**8.30**	**31.39**	41.19	46.04
DSh	-5.17	7.74	19.85	27.90	-12.21	8.43	21.79	27.97	-14.46	5.12	21.23	28.24	-11.79	11.77	24.73	29.22	-3.82	20.24	28.19	30.91
Panel B) Gamma Distribution																				
RR	**47.29**	**76.56**	**85.17**	**91.93**	**24.38**	**57.24**	**71.42**	**82.96**	-0.98	**26.99**	**47.74**	**68.67**	-35.08	9.83	**45.47**	**70.59**	-21.47	31.32	**61.38**	**79.12**
EN	13.80	51.06	71.41	86.60	2.95	33.32	60.50	79.65	-4.46	14.42	42.24	67.05	-42.68	4.37	44.93	70.38	-28.49	31.10	61.22	79.05
QIS	1.69	15.07	36.60	61.08	1.42	11.14	28.95	46.97	-2.16	-1.10	8.66	28.02	0.86	13.56	31.38	51.50	1.67	15.57	35.11	57.37
LW	3.80	16.50	38.64	64.23	2.49	13.12	31.03	48.34	-2.07	-3.52	5.95	27.47	1.50	15.38	32.88	52.02	3.65	17.56	37.61	59.49
SR	18.78	2.20	0.98	0.50	15.67	1.71	0.91	0.52	**10.65**	1.21	0.52	0.20	**4.79**	0.19	0.09	0.04	1.93	0.03	0.00	0.01
GSR	32.34	33.11	35.13	39.56	23.62	30.84	33.49	37.24	2.26	16.48	26.32	33.41	0.57	18.44	27.40	32.91	4.29	23.55	30.29	34.20
St	16.45	48.45	58.22	65.21	8.97	35.53	47.58	56.72	4.90	24.80	37.60	47.82	-0.71	**26.41**	39.45	47.73	**7.27**	**34.22**	44.24	49.96
DSh	1.97	36.93	44.77	50.42	-9.60	23.05	34.72	41.90	-12.65	12.05	26.48	34.54	-16.42	14.05	27.35	33.31	-3.42	21.82	30.13	33.92

Notes. This table reports the relative Mean L_2 errors for the PoR and GaR models using the *log* LF, expressed as percentage improvements relative to the benchmark `glm2` estimator. A positive value indicates a percentage reduction in the L_2 error (better performance), while a negative value indicates a percentage increase (worse performance) compared to `glm2`. Bold red numbers mark the best performance, and underlined numbers indicate the second-best performance for each setting. Results are evaluated across varying predictor-to-sample size ratios p/n and correlation structures ρ. All models were initialised with the same starting values, computed via the optimisation-based procedure detailed in Section SM.5.2. The true regression parameters (detailed in SM.3.1) 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 when $n = 500$.

SM.3.4 Analysis of Computational Efficiency

The computational efficiency of the IRLS-based models was evaluated by comparing the number of iterations required to achieve convergence. *Note that the standard regularised models, specifically RR and the EN, solve the optimisation problem using a coordinate descent algorithm via the* ***R****'s* `glmnet` *package; thus, their iteration steps are fundamentally different from IRLS and are excluded from this direct comparison.* Figures SM.3.3 and SM.3.4 provide a visual representation of the iteration counts, highlighting how often each model reached convergence with the minimum number of iterations across 250 replications for each scenario.

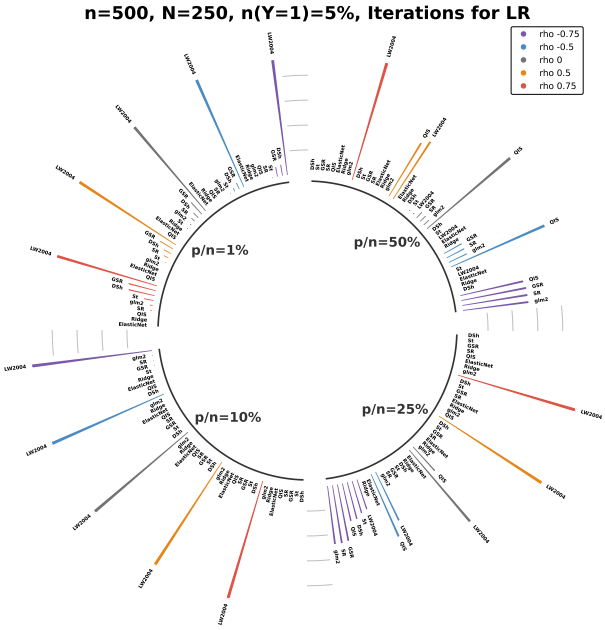
For LR, Figure SM.3.3 illustrates a significant divergence in iteration counts between the various shrinkage estimators. Notably, the LW estimator demonstrates the highest computational efficiency; regardless of whether the data is balanced or unbalanced, it consistently converges with the fewest iterations across all predictor-to-sample ratios (p/n). The QIS estimator also shows improved efficiency at the highest p/n ratio (50%). In contrast, although our proposed GSR model reduces the number of iterations to some extent at lower p/n ratios (e.g., 1%), as the dimensionality increases, the performance of the proposed methods (SR, GSR, St and DSh) is generally comparable to that of the baseline method `glm2`, or even requires more iterations. This suggests that for LR, the LW estimator provides a clear computational advantage, whereas our proposed methods do not reliably accelerate convergence.

In contrast, for Poisson and Gamma GLMs with the *sqrt* LF, Figures SM.3.4a and SM.3.4b demonstrate that the shrinkage-based IRLS estimators generally converge much faster than `glm2`. For the Poisson distribution, the St and DSh estimators we propose generally require the fewest iterations, followed by the QIS and LW methods. For the Gamma distribution, the proposed DSh estimator is far ahead in computational efficiency, outperforming both the QIS method and `glm2`. In these scenarios, the SR and GSR estimators also show improved efficiency over the benchmark, though less consistently than St and DSh. For the *log* LF, Figures SM.3.4c and SM.3.4d show that the proposed St estimator overwhelmingly dominates in convergence efficiency for both the Poisson and Gamma GLMs, consistently requiring the fewest iterations across almost all p/n ratios. For the Poisson model, the LW estimator also performs highly competitively, while our DSh and GSR methods show marked convergence improvements over `glm2` across both distributions. In this setting, the SR model is the only shrinkage estimator that aligns closely with `glm2`, indicating negligible computational gains.

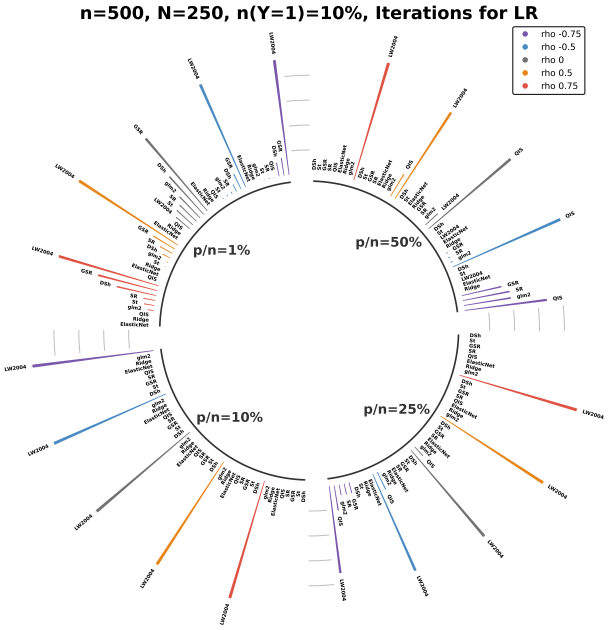
Overall, these results suggest that the proposed St and DSh models consistently and significantly converge faster than `glm2` for Poisson and Gamma GLMs. However, for the LR model with the *logit* LF, the established LW estimator stands out as the most computationally efficient approach, while our proposed shrinkage estimators offer mixed iteration improvements that depend heavily on the predictor-to-sample ratio and the chosen LF.

SM.4 Further Details about the Real Data Analysis

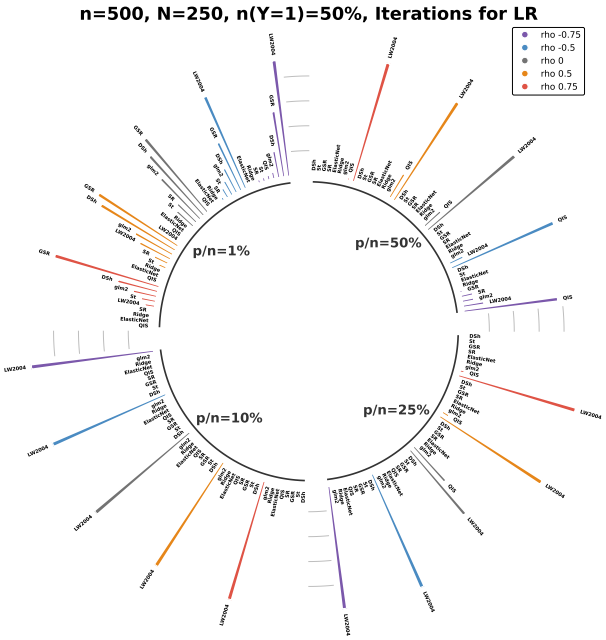
This section presents additional empirical results for the real data application discussed in Section 4.2 by evaluating the proposed shrinkage estimators across three extended modelling scenarios. In Section SM.4.1, we detail the OOS performance for modelling average severity per policy using the Gamma GLM with the *sqrt* LF. Section SM.4.2 expands this average severity analysis by applying the same *sqrt* LF to the Tweedie GLM. Finally, Section SM.4.3 transitions



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

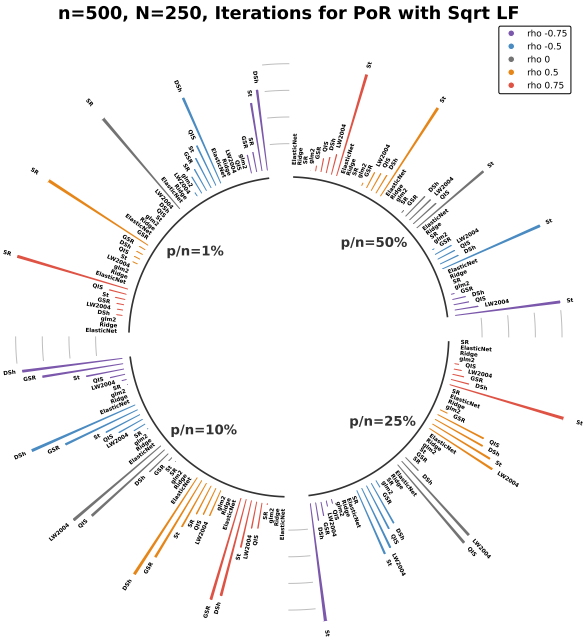


(b) LR with Imbalanced data ($n(Y = 0) = 10n$)

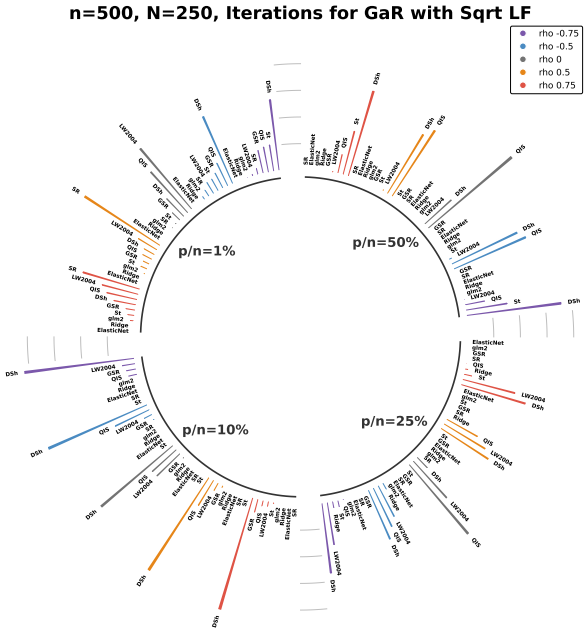


(c) LR with Balanced data ($n(Y = 0) = 50n$)

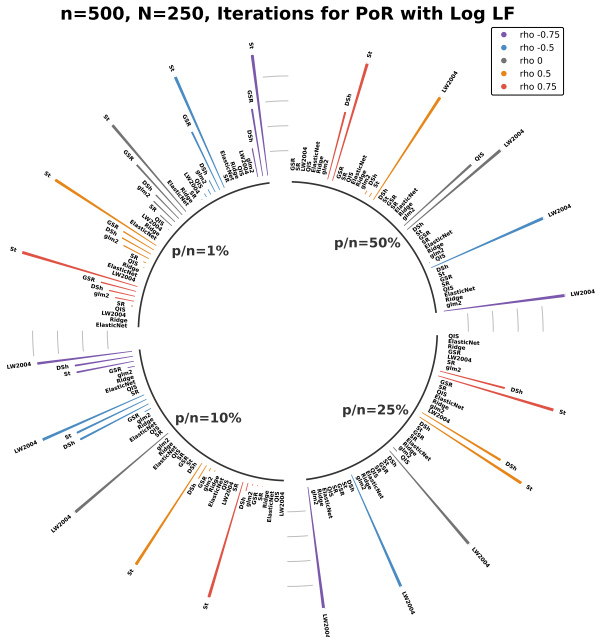
Figure SM.3.3: Comparison of iterations for the LR model. Top row: models with imbalanced data. Bottom row: models with balanced data. Longer bars indicate better performance.



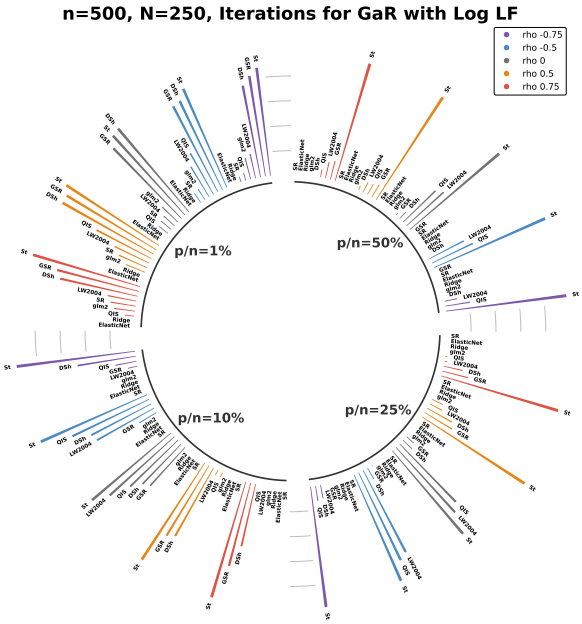
(a) PoR with *sqrt* LF



(b) GaR with *sqrt* LF



(c) PoR with *log* LF



(d) GaR with *log* LF

Figure SM.3.4: 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.

to modelling average frequency, where we evaluate the estimators under the Poisson GLM using both the *log* LF and the *sqrt* LF.

SM.4.1 Average Severity per Policy – Gamma GLM

This section discusses the OOS performance of the Gamma GLM using the *sqrt* LF. Table SM.4.1 and Figure SM.4.1 summarise the numerical and visual results. Similar to the *log* LF results, we see a difference between accuracy and calibration, along with significant instability. Our proposed St achieves the lowest 95th percentile distance metrics, reducing the RMSE to 0.957 in Panel A and 0.958 in Panel C. However, using the *sqrt* LF leads to severe deviance explosions for several models. The SR, St, and LW estimators, along with the EN competitor, show extreme deviance values, yielding ratios in the hundreds of millions or billions compared to the baseline `glm2` in Panels A and C. Miscalibration also rises, with the EN and St estimators reporting A/E deviations of 5.96% and 7.09% in Panel A. In Panel B, calibration worsens for all estimators, including the baseline `glm2`, with A/E deviations ranging from 12.82% for SR to 39.50% for St. Despite the severe convergence issues for some models, estimators like DSh, QIS, and GSR remain stable under the *sqrt* LF. The DSh estimator maintains strong calibration, achieving an A/E deviation of 0.03% in Panel A and 1.04% in Panel C, along with stable deviance ratios of 0.991 and 0.972.

The Aggregated Double Lift charts in Figure SM.4.1 visually confirm this instability. Unlike the reliable behaviour seen under the *log* LF, the *sqrt* specification reveals serious predictive failures at the portfolio extremes. In the single claim and all claims panels, estimators like RR, EN, SR, and St do not capture the actual claims in the boundary deciles. For instance, the SR and RR predictions diverge sharply from the actual observations in bucket 1 and bucket 10. In contrast, the DSh and GSR estimators manage to avoid this extreme boundary distortion to some extent. They maintain a stable profile that follows the standard `glm2` and the actual claims across all ten buckets. This confirms that while the *sqrt* LF creates mathematical failure for most regularisation methods, DSh and GSR can possibly retain their structural stability.

SM.4.2 Average Severity per Policy – Tweedie GLM

In this section, we show the OOS performance of the Tweedie GLM using the *sqrt* LF. Table SM.4.2 and Figure SM.4.2 show the numerical and visual results. When we compare these results to the Gamma models with the *sqrt* LF, a major similarity appears. The Tweedie distribution does not fix the massive deviance explosions seen in the Gamma GLM. Although the internal profile log-likelihood grid search successfully finds an optimal variance power index ρ between 1.68 and 1.70, it fails to control the stability of the models. Estimators like LW, SR and St still produce huge Deviance ratios in Panel A and Panel C, reaching values on the scale of 10^4 and 10^5.

Because the deviance stability fails, the calibration errors remain significant. For instance, the St estimator yields the smallest tail errors at the 95th percentile, but it has a very large A/E deviation of 42.07% in Panel B. The RR and EN estimators reduce the 99th percentile tail error, but they come with a huge computational cost, often 40 to 100 times slower than the benchmark `glm2`. When we look at portfolio calibration, the Tweedie GLM shows mixed results. It actually improves the A/E ratio for the majority of the models. By comparing Table SM.4.2 to the Gamma results, 8 out of 9 models improve in Panel B, and 7 out of 9 models improve

Table SM.4.1: **OOS for Gamma GLM (*sqrt* LF) on freMTPL2 Average Severity per Policy**

	Winsorised RMSE		Winsorised MAE			Gini				
Model	**95%**	**99%**	**95%**	**99%**	**Deviance**	**Raw**	**Norm**	**A/E - 1**	**CT**	**NoIt**
Panel A) Exactly 1 Claim ($n = 23,571$)										
glm2	1.000	1.000	1.000	1.000	1.000	0.118	0.178	1.09%	**1.000**	1.000
RR	1.034	**0.873**	1.073	0.991	1.056	**0.122**	**0.185**	5.21%	36.689	44.886
EN	1.006	0.961	1.011	0.991	8.185E+08	0.115	0.173	5.96%	84.950	771.404
QIS	0.999	1.000	1.000	1.000	1.000	0.118	0.177	1.14%	1.272	1.031
LW	1.015	0.996	1.020	1.012	2.380E+09	0.118	0.177	3.05%	3.470	0.923
SR	1.000	1.000	1.000	1.000	4.190E+08	0.118	0.178	1.14%	2.059	1.019
GSR	0.998	0.998	0.995	0.995	**0.990**	0.122	0.184	2.14%	4.388	0.998
St	**0.957**	0.999	**0.925**	**0.944**	1.671E+09	0.116	0.175	7.09%	4.670	**0.719**
DSh	1.009	0.999	1.021	1.015	0.991	0.118	0.178	**0.03%**	4.410	0.846
Panel B) Exactly 2 Claims ($n = 1,298$)										
glm2	1.000	1.000	1.000	1.000	1.000	0.174	0.264	12.83%	**1.000**	1.000
RR	0.876	0.942	0.966	0.959	1.039	0.102	0.154	17.53%	166.802	30.323
EN	0.978	0.994	0.994	0.994	1.039	0.133	0.210	14.53%	242.468	137.208
QIS	0.978	0.988	0.978	0.979	**0.997**	**0.175**	**0.267**	15.24%	3.310	0.542
LW	0.977	0.984	0.990	0.987	1.001	0.162	0.245	13.92%	3.314	0.498
SR	1.001	1.001	1.000	1.001	1.001	0.174	0.264	**12.82%**	5.132	0.859
GSR	0.956	0.978	0.954	0.956	1.017	0.158	0.239	18.63%	4.826	0.503
St	**0.858**	**0.939**	**0.820**	**0.839**	1.048	0.172	0.262	39.50%	5.633	0.452
DSh	0.977	0.994	0.972	0.974	1.028	0.147	0.219	17.07%	4.600	**0.424**
Panel C) All Claims ≥ 1 ($n = 24,944$)										
glm2	1.000	1.000	1.000	1.000	1.000	0.105	0.156	1.51%	**1.000**	1.000
RR	1.012	**0.869**	1.047	0.973	1.008	**0.119**	**0.178**	5.32%	32.240	44.562
EN	1.010	0.938	1.015	0.984	1.017	0.111	0.165	5.09%	59.485	434.600
QIS	1.000	1.000	0.999	1.000	1.000	0.105	0.156	1.55%	1.019	0.996
LW	1.014	0.975	1.016	1.002	2.426E+09	0.107	0.160	3.42%	3.036	0.877
SR	1.000	1.000	1.000	1.000	1.000	0.105	0.157	1.56%	2.069	0.989
GSR	0.996	0.997	0.994	0.995	0.973	0.104	0.155	2.56%	3.700	0.932
St	**0.958**	0.992	**0.933**	**0.947**	1.766E+09	0.102	0.152	7.50%	4.029	**0.675**
DSh	1.005	1.000	1.010	1.007	**0.972**	0.109	0.163	**1.04%**	3.758	0.821

Notes. This table reports the OOS performance ratios for standard competitors and our proposed regularised estimators relative to the baseline `glm2` benchmark on the freMTPL2 dataset. Results are averaged over 500 replications utilising a Gamma distribution with a *sqrt* LF. An A/E deviation closer to 0.00% indicates superior portfolio calibration. For distance metrics like RMSE, MAE, Deviance, CT, NoIt, lower is better. For the raw Gini and normalised Gini index, higher is better. The best performing estimator in each column is highlighted in **bold red**, and the second best is underlined. Note that for panel A, glm2, QIS, SR, GSR, St, and DSh each failed to converge once; for panel C, glm2, St, SR, and QIS each failed to converge once; while for panel B, all models converged.

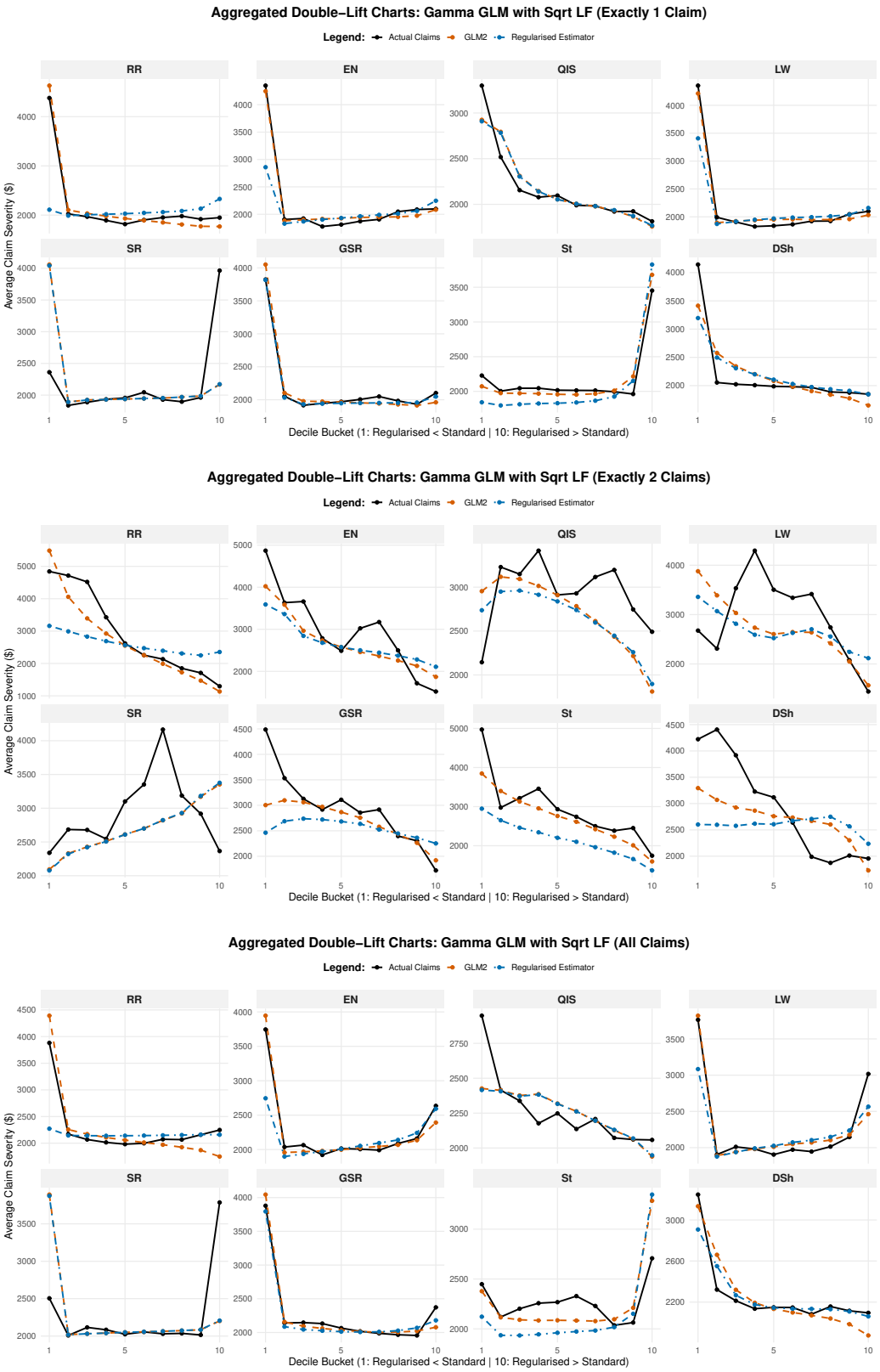


Figure SM.4.1: Aggregated Double Lift Charts comparing standard competitors and proposed regularised estimators against the baseline Gamma GLM using a *sqrt* LF on freMTPL2 average severity per policy.

in Panel C. Furthermore, the absolute best calibration score in Panel B improves from 12.82% under the Gamma assumption to 10.86% here. However, in Panel A and Panel C, the absolute best calibration scores become slightly worse. For example, the DSh estimator achieves the best A/E deviation in Panel A, but its score changes from 0.03% under the Gamma model to -0.11% under the Tweedie model.

The Aggregated Double Lift charts in Figure SM.4.2 visually confirm these problems. Just as seen in the *log* LF results, the RR and EN estimators overcorrect the safest risks in bucket 1 and the riskiest profiles in bucket 10. This pulls their predictions far away from the actual observed claims. Unstable estimators like St diverge noticeably across the middle deciles. Ultimately, because the *sqrt* LF fails to maintain predictive stability and deviance control under both Gamma and Tweedie distributions, it makes the resulting model outputs difficult to trust and implement in standard pricing practice.

Table SM.4.2: **OOS for Tweedie GLM (*sqrt* LF) on freMTPL2 Average Severity per Policy**

	Winsorised RMSE		Winsorised MAE			Gini					
Model	**95%**	**99%**	**95%**	**99%**	**Deviance**	**Raw**	**Norm**	**A/E - 1**	**CT**	**NoIt**	**Power**
Panel A) Exactly 1 Claim ($n = 23,571$)											
glm2	1.000	1.000	1.000	1.000	1.000	0.116	0.174	1.12%	**1.000**	1.000	1.683
RR	1.050	**0.877**	1.083	0.998	0.874	**0.121**	**0.182**	4.64%	40.586	39.533	1.695
EN	1.005	0.984	1.008	0.998	0.861	0.116	0.175	4.56%	104.348	1.123E+03	1.694
QIS	0.999	1.000	0.999	0.999	1.004	0.116	0.174	1.20%	1.319	1.006	1.682
LW	1.008	0.997	1.009	1.005	3.740E+05	0.116	0.174	3.54%	6.093	0.821	1.681
SR	1.000	1.000	1.000	1.000	0.808	0.117	0.175	1.21%	3.975	0.926	1.697
GSR	1.000	1.000	0.995	0.996	0.781	0.120	0.180	1.73%	7.132	0.829	1.700
St	**0.957**	1.004	**0.928**	**0.949**	1.216E+05	0.116	0.175	6.39%	7.049	0.834	1.682
DSh	1.007	1.000	1.019	1.014	**0.780**	0.117	0.176	**-0.11%**	6.694	**0.723**	1.700
Panel B) Exactly 2 Claims ($n = 1,298$)											
glm2	1.000	1.000	1.000	1.000	1.000	0.171	0.259	**10.86%**	**1.000**	1.000	1.700
RR	0.857	**0.921**	0.936	0.928	1.037	0.108	0.164	17.40%	59.415	30.237	1.696
EN	0.985	0.995	0.992	0.992	1.052	0.146	0.228	12.27%	90.829	170.344	1.697
QIS	0.985	0.990	0.986	0.986	**0.997**	0.170	0.259	12.36%	3.756	0.610	1.700
LW	0.988	0.988	0.998	0.994	1.003	0.160	0.242	11.22%	4.048	0.531	1.699
SR	1.000	1.000	0.999	1.000	1.000	**0.172**	**0.260**	10.98%	4.412	0.761	1.700
GSR	0.958	0.975	0.958	0.958	1.023	0.145	0.221	16.10%	4.699	0.492	1.699
St	**0.844**	0.926	**0.801**	**0.818**	1.047	0.169	0.257	42.07%	4.889	**0.441**	1.699
DSh	0.978	0.995	0.970	0.973	1.026	0.145	0.216	15.24%	4.594	0.454	1.700
Panel C) All Claims $\geq$ 1 ($n = 24,944$)											
glm2	1.000	1.000	1.000	1.000	1.000	0.105	0.157	1.37%	**1.000**	1.000	1.682
RR	1.024	**0.872**	1.054	0.978	**0.861**	**0.119**	**0.179**	4.73%	39.281	39.117	1.696
EN	1.006	0.974	1.009	0.995	0.884	0.110	0.164	3.95%	80.605	694.177	1.695
QIS	1.000	1.000	0.999	1.000	1.683E+04	0.105	0.157	1.41%	1.243	1.031	1.681
LW	1.010	0.992	1.010	1.004	1.760E+05	0.108	0.161	2.96%	5.416	0.783	1.686
SR	1.000	1.000	1.000	1.000	2.154E+05	0.106	0.158	1.45%	4.418	0.939	1.685
GSR	0.996	0.998	0.994	0.994	1.825E+04	0.103	0.153	2.19%	6.276	0.805	1.698
St	**0.948**	0.995	**0.915**	**0.936**	4.975E+04	0.104	0.156	7.57%	6.601	**0.745**	1.678
DSh	1.000	0.999	1.005	1.003	1.825E+04	0.109	0.163	**1.16%**	5.745	0.754	1.699

Notes. This table reports the OOS performance ratios for regularised estimators relative to the standard `glm2` benchmark on the `freMTPL2` dataset. Results are averaged over 500 replications utilising a Tweedie distribution with a *sqrt* LF. An A/E deviation closer to 0.00% indicates superior portfolio calibration. For distance metrics like RMSE, MAE, Deviance, CT, NoIt, lower is better. For the raw Gini and normalised Gini index, higher is better. The optimal variance power index ρ selected via in sample profile log likelihood is reported in the final column. The best performing estimator in each column is highlighted in **bold red**, and the second best is underlined. All models converged across all panels.

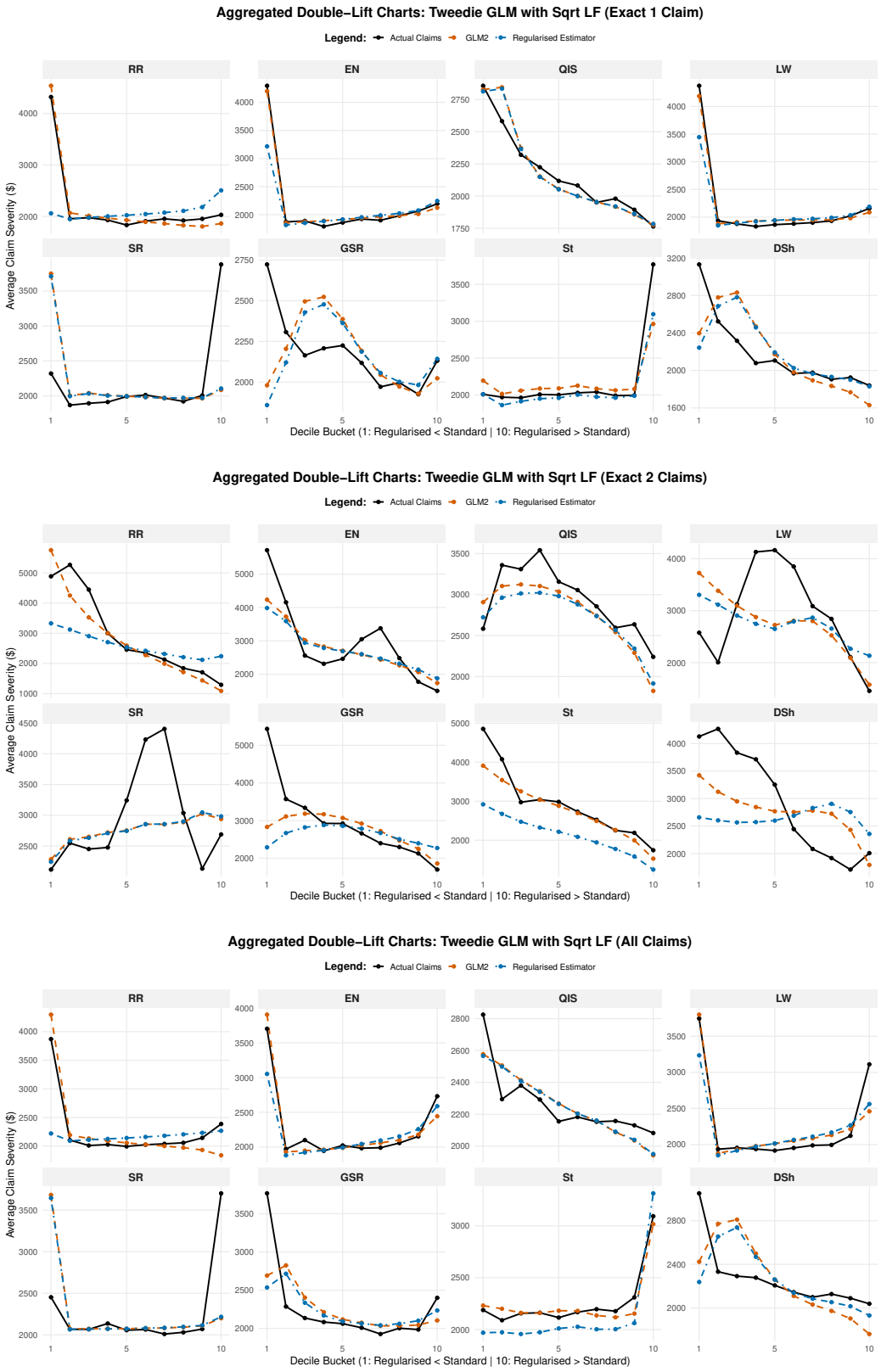


Figure SM.4.2: Aggregated Double Lift Charts comparing regularised estimators against the Tweedie GLM using a *sqrt* LF on freMTPL2 average claim per policy.

SM.4.3 Claim Frequency – Poisson GLM

This section presents the OOS performance when modelling claim frequency using a Poisson GLM. The dependent variable is the number of claims per policyholder. This value is discrete and bounded above. Because of this, the extreme outlier losses seen in severity modelling do not occur. We calculate distance metrics like RMSE and MAE on the raw predictions without winsorisation. Table SM.4.3 and Figure SM.4.3 show the numerical and visual results for the standard *log* LF and the non-canonical *sqrt* LF. We evaluate prediction error, risk discrimination, portfolio calibration and computational efficiency.

Table SM.4.3: **OOS for Poisson GLM on freMTPL2 Claim Frequency**

Model	RMSE	MAE	Deviance	Gini^{Raw}	$\text{Gini}^{\text{Norm}}$	A/E - 1	CT	NoIt
Panel A) *log* LF								
glm2	1.000	1.000	1.000	0.284	0.294	0.00%	**1.000**	**1.000**
RR	1.000	1.001	1.001	0.282	0.292	0.00%	14.738	47.611
EN	1.000	1.000	**1.000**	0.284	0.294	0.00%	28.971	156.344
QIS	1.000	1.000	1.000	0.284	0.294	0.01%	1.458	**1.000**
LW	1.000	1.000	**1.000**	0.284	0.294	**0.00%**	2.110	1.003
SR	1.000	1.000	1.000	0.284	0.294	-0.03%	8.047	1.744
GSR	1.000	1.000	1.000	0.284	0.294	-0.05%	6.149	1.053
St	1.000	1.013	1.000	**0.284**	**0.294**	-2.74%	8.271	**1.000**
DSh	**1.000**	**0.999**	1.000	0.283	0.293	0.22%	8.335	**1.000**
Panel B) *sqrt* LF								
glm2	1.000	1.000	1.000	0.288	0.298	17.90%	**1.000**	**1.000**
RR	0.999	1.073	0.995	0.287	0.298	0.18%	23.230	49.068
EN	**0.999**	1.073	**0.995**	**0.289**	**0.299**	**0.05%**	26.205	85.016
QIS	1.000	1.000	1.000	0.288	0.298	17.90%	1.798	1.009
LW	1.000	1.001	1.000	0.288	0.298	17.52%	2.731	1.048
SR	1.005	**0.900**	1.029	0.286	0.296	54.72%	4.260	1.089
GSR	1.000	1.031	0.997	0.288	0.299	9.93%	4.018	1.037
St	1.000	0.995	1.001	0.288	0.298	19.23%	4.182	1.047
DSh	1.000	1.043	0.997	0.287	0.298	7.24%	3.926	1.035

Notes. This table reports the OOS performance ratios on claim frequency for shrinkage estimators and standard competitors relative to the baseline `glm2` benchmark on the freMTPL2 dataset. Panel A is for *log* LF and panel B is for *sqrt* LF. Results are averaged over 500 replications utilising a Poisson distribution. An A/E deviation closer to 0.00% indicates superior portfolio calibration. For distance metrics like RMSE, MAE, Deviance, CT, NoIt, lower is better. For the raw Gini and normalised Gini index, higher is better. The best performing estimator in each column is highlighted in **bold red**, and the second best is underlined. Note that all models in panel A converged for 500 replications, whereas in panel B the EN model failed to converge 6 times and the SR model 8 times.

The results for Panel A show that using regularisation for this frequency portfolio does not help much under the standard *log* LF. For all regularised benchmarks and shrinkage estimators, the RMSE and Deviance ratio match the benchmark `glm2` closely. The raw and normalised Gini indices stay flat at roughly 0.284 and 0.294 for all models. Almost all estimators are perfectly calibrated with an A/E deviation of 0.00%. This result means that the standard GLM already accounts for almost all the expected frequency patterns in the dataset. There is little room for shrinkage methods to improve it. The Aggregated Double Lift charts in the top panel of Figure SM.4.3 confirm this. Across all ten decile buckets, the predictions of all regularised estimators stay on par with the baseline `glm2` predictions and the actual observed frequencies.

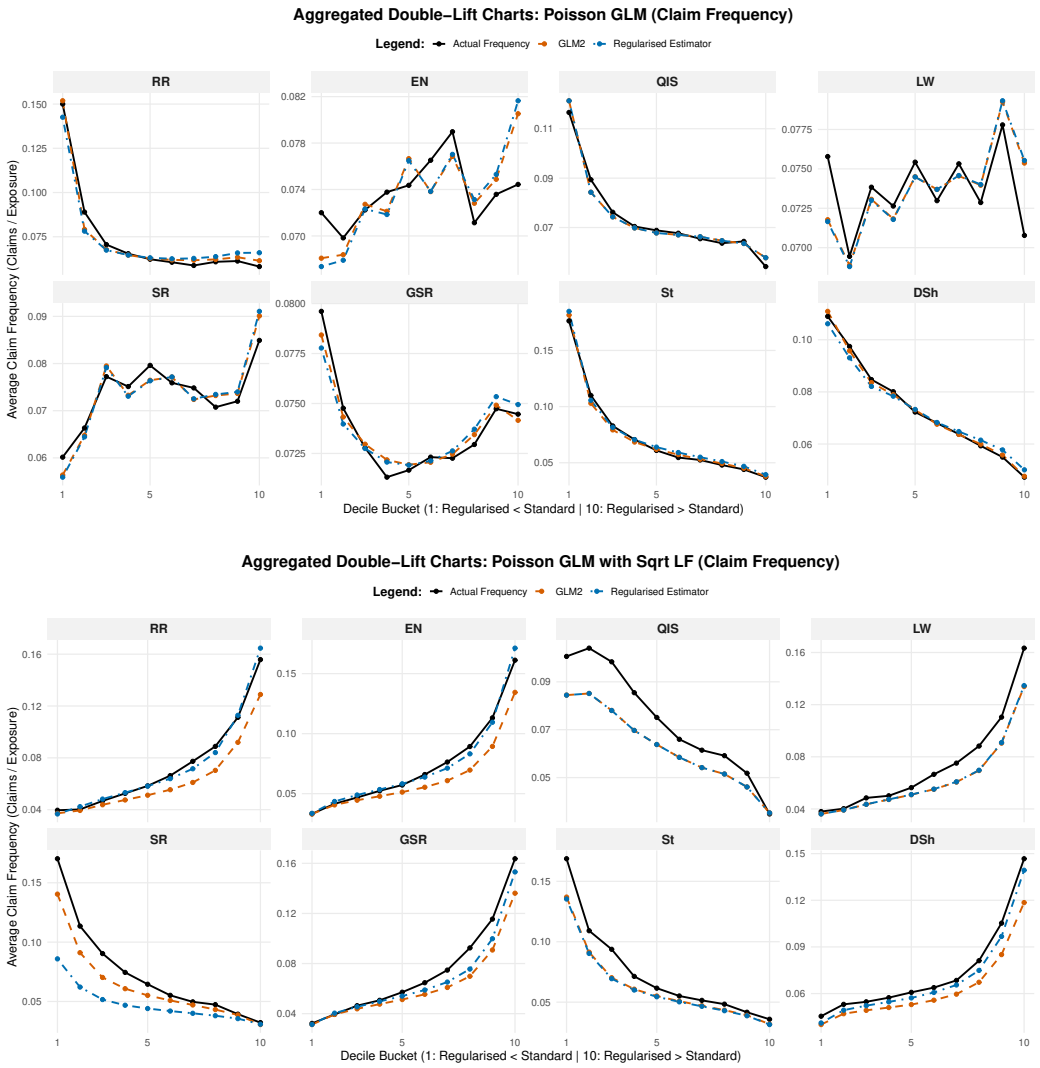


Figure SM.4.3: Aggregated Double Lift Charts comparing the standard competitors and our proposed regularised estimators against the baseline Poisson GLM using *log* and *sqrt* LFs on freMTPL2 claim frequency.

Panel B shows the results when forcing the models to use the non-canonical *sqrt* LF. This choice causes instability. Table SM.4.3 shows that the *log* LF models in Panel A converged successfully in all 500 repetitions. In contrast, the EN and SR models failed to converge in 6 and 8 instances under the *sqrt* LF. The RR and EN models maintain stable precision, risk discrimination, and calibration after they converge. However, using the non-canonical *sqrt* LF leads to severe predictive instability for the proposed estimators. Most notably, the proposed SR estimator fails to calibrate, showing an A/E bias of 54.72%. The Aggregated Double Lift charts in the bottom panel of Figure SM.4.3 show this massive miscalibration. The standard `glm2` and the RR models track the actual frequencies well. Unstable estimators like SR, QIS, and St diverge sharply. They understate the actual claim frequency in the lowest risk deciles (buckets 1 to 3) and fail to reflect the real risk profile.

These results confirm that the proposed estimators provide good precision and computational speed for modelling severity, but the standard GLM with a *log* LF best captures the frequency

patterns. Researchers should use extreme caution when applying shrinkage to frequency models with a *sqrt* LF, as this combination causes severe miscalibration and instability.

SM.5 Details About the Convergence Failure

This section provides technical details for the discussion in Section 3 of the main text. Section SM.5.1 outlines default initialisation behaviour across **R**, **MATLAB** and **Python**. Section SM.5.2 presents the optimisation method formulation for starting values. Section SM.5.3 reports convergence results in two parts. The first part compares the three software packages using both default and our optimisation-based starting values. The second part provides convergence failure records for the simulation studies described in Section SM.3.

SM.5.1 Starting Values in GLM Software Implementations

The default starting values in standard software packages (such as **R**'s `glm2`, **MATLAB**'s `fitglm`, and **Python**'s `statsmodels.GLM`) for the mean response, $\mu_i^{(0)}$, are often set as $\mu_i^{(0)} = y_i$, which is a natural choice. However, this can cause numerical problems depending on the underlying distribution and its associated LF. To prevent this, each software implementation applies its own adjustments. We now provide some examples to illustrate our point.

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

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

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 $\eta_i^{(0)} = \log\big(\mu_i^{(0)} / \big(1 - \mu_i^{(0)}\big)\big)$ are chosen across `glm2`, `fitglm`, and `statsmodels.GLM` for binary data.

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

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

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 ($h(\eta) = e^{\eta}$) and *sqrt* LF choices ($h(\eta) = \eta^2$) are adjusted by taking $\mu_i^{(0)} = y_i + 0.1$ in **R**'s `glm2`. In contrast, **MATLAB** uses a slightly larger constant shift of $\mu_i^{(0)} = y_i + 0.25$, while **Python** avoids static constants entirely by smoothing with the sample mean, taking $\mu_i^{(0)} = (y_i + \bar{y})/2$.

Assume a *Gamma regression (GaR)* with a generic LF, 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 **R** or **MATLAB** for Gamma regression ($\mu_i^{(0)} = y_i$), and this could be a problem for a user that has examples with small values for the dependent variable. **Python**, however, continues to use the sample mean adjustment $\mu_i^{(0)} = (y_i + \bar{y})/2$, providing more numerical safety. This is not uncommon in practice, and one example is insurance claims data – such as medical insurance – where very small claims are possible.

Table SM.5.1: Default Starting Values and Validations across GLM Implementations

Model	**LF**	**Adjusted Initial Mean $\mu_i^{(0)}$**			**Initial Predictor $\eta_i^{(0)}$**	**Valid $\eta_i^{(t)}$?**
		R (`glm2`)	**MATLAB** (`fitglm`)	**Python** (`statsmodels.GLM`)		
LR	*logit*	$\frac{y_i+0.5}{2}$	$\frac{y_i+0.5}{2}$	$\frac{y_i+0.5}{2}$	$\log\left(\frac{\mu_i^{(0)}}{1-\mu_i^{(0)}}\right)$	TRUE
PoR	*sqrt*	$y_i + 0.1$	$y_i + 0.25$	$\frac{y_i+\bar{y}}{2}$	$\sqrt{\mu_i^{(0)}}$	if $\eta_i^{(t)} > 0$
PoR	*log*	$y_i + 0.1$	$y_i + 0.25$	$\frac{y_i+\bar{y}}{2}$	$\log\left(\mu_i^{(0)}\right)$	TRUE
GaR	*sqrt*	y_i	$\max(y_i, \epsilon)$	$\frac{y_i+\bar{y}}{2}$	$\sqrt{\mu_i^{(0)}}$	if $\eta_i^{(t)} > 0$
GaR	*log*	y_i	$\max(y_i, \epsilon)$	$\frac{y_i+\bar{y}}{2}$	$\log\left(\mu_i^{(0)}\right)$	TRUE

Notes. This table summarises the default starting values ($\mu_i^{(0)}$) and predictor checks used by three major GLM software implementations: **R** (`glm2`), **MATLAB** (`fitglm`), and **Python** (`statsmodels.GLM`). The initial linear predictor is universally constructed via the inverse LF as $\eta_i^{(0)} = h^{-1}(\mu_i^{(0)})$. The term $\bar{y}$ represents the sample mean of the response variable. For MATLAB's binomial models, the formula shown assumes binary data (number of trials $N = 1$). For MATLAB's Gamma models, ϵ represents the machine epsilon (a safely small positive number) used to prevent boundary errors at exact zero. The final column indicates whether the linear predictor $\eta_i^{(t)}$ remains mathematically valid during iterations depending on the chosen LF. For the *sqrt* LF, if the predictor becomes non-positive, the mathematical mapping breaks and the model may become unstable or fail to converge.

Table SM.5.1 summarises the adjusted default starting values and the validation checks implemented across these three major software packages 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 when unadjusted (as in **R** and **MATLAB**). During the solver iterations, the updated predictor $\eta_i^{(t)} = \mathbf{x}_i^\top \boldsymbol{\beta}^{(t)}$ can become non-positive, causing problems that fail the mathematical validity checks for $\eta_i^{(t)}$ under a *sqrt* LF. Even after adjusting the default starting values at the initialisation stage, instabilities and early failures may frequently occur across all software implementations if the linear predictor drifts into an invalid mathematical domain during the iterative process.

SM.5.2 Optimisation-Based Starting Values Formulation

Section SM.5.1 outlines the importance of starting values for IRLS and the specific initialisation adjustments used across standard software implementations. We now provide a novel optimisation-based method to define starting values that are data-driven and LF-driven, which builds on these standard software adjustments. Our method minimises the difference between a transformed version of the initial mean response ($g^*(\mu_i^{(0)})$) and the linear predictor ($\mathbf{x}_i^\top \boldsymbol{\beta}^{(0)}$). Here, the function g^* applies the inverse LF to the software-adjusted initial mean while ensuring all mathematical constraints are met. The starting value is the solution of the instance given as (SM.5.1):

$$\begin{cases} \min\limits_{\boldsymbol{\beta}^{(0)}} & \sum\limits_{i=1}^{n} \left(g^*\left(\mu_i^{(0)}\right) - \mathbf{x}_i^\top \boldsymbol{\beta}^{(0)} \right)^2 \\ \text{s.t.} & \mathbf{x}_i^\top \boldsymbol{\beta}^{(0)} \geq \epsilon, \quad \text{for all } 1 \leq i \leq n \text{ if required by the chosen LF.} \end{cases} \tag{SM.5.1}$$

In this formulation, $\epsilon > 0$ is a small threshold (set to 10^{-6} by default) that ensures the predictor remains strictly positive when necessary, as indicated by the validity column in Table SM.5.1. The choice of g^* depends on the LF; for example, $g^*(\mu) = \sqrt{\mu}$ and $g^*(\mu) = \log(\mu)$ are the natural choices for the *sqrt* and *log* LFs. The inequality constraints in (SM.5.1) are only enforced when the LF requires a bounded domain (such as the *sqrt* LF). For unconstrained links like *logit* or *log*, these constraints are dropped.

In summary, the starting value solution in (SM.5.1) relies on a scalable convex quadratic instance. Convex optimisation finds the optimal starting coefficients without relying on iterative updates. This approach can prevent the algorithm from crashing or failing at the start, which often happens if the default initial guess is poor. By using this optimisation, we might ensure a solid starting point so that the following IRLS steps can find a reliable solution. These convex instances can be easily implemented using standard solver packages such as `CVXR` in **R**, `CVXPY` in **Python**, or `CVX` in **MATLAB**. However, solving this optimisation problem introduces an additional computational burden, especially when the dataset or the number of parameters is large. Therefore, this optimisation approach should be viewed as a robust fallback. It is the optimal choice given only when the default initialisation fails to provide a stable starting point.

SM.5.3 Convergence Failures in Simulations

To evaluate if the optimisation method reduces convergence failures, we compare Poisson and Gamma GLMs using the *sqrt* LF. We test the proposed starting values against the defaults in **R**'s `glm2`, **MATLAB**'s `fitglm`, and **Python**'s `statsmodels.GLM`. Table SM.5.2 reports the failure counts for $N = 100$ samples of size $n = 500$ under DGP1 described in Section SM.3.1. The values outside the parentheses represent the software defaults, and the values inside represent our starting values. Table SM.5.2 shows that the optimisation method improves convergence for all three packages. When the default solvers fail, our method reduces the failure count, achieving near-zero failures for **R** and **Python** in most cases. **Python** records fewer default failures than **R** and **MATLAB**, but our initialisation provides the lowest failure counts across all panels.

Table SM.5.3 provides complementary convergence results using DGP2 (Section SM.3.1). For the Poisson distribution in Panel A, convergence failures occur when the mean parameter $\mu = 0$ in **R** and **MATLAB**. Our initialisation method eliminates these failures in **R** and improves the convergence rate in **MATLAB**. For the Gamma distribution in Panel B, convergence is more difficult across all scenarios where $\mu = 0$. However, the optimisation method consistently reduces the failure counts for all three software solvers. These results indicate that the proposed initialisation method remains robust compared to standard software defaults under this DGP.

After establishing that the optimisation method improves the software solvers, we evaluate its performance when applied to our proposed shrinkage estimators in the simulation data analysis provided in Section SM.3. Complementing the main L_2 error results, Table SM.5.4 reports the convergence failures that happened for Poisson and Gamma models using the *sqrt* LF, as well as LR models using the *logit* LF. For the Poisson distribution in Panel A, most methods achieve convergence, though `glm2`, SR, and EN show occasional failures in high dimensions ($p/n = 50\%$). For the Gamma distribution in Panel B, convergence is more challenging. Estimators based on GSR and DSh perform well, converging in almost all cases, whereas RR and EN suffer from frequent convergence failures in these settings. Note that a complementary convergence table for the *log* LF results in Table SM.3.4 is omitted, as all Poisson and Gamma models successfully

Table SM.5.2: **Number of Convergence Failures with *sqrt* LF (DGP1)**

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

Notes. This table reports the number of convergence failures over $N = 100$ replications, each with $n = 500$ observations. The numbers outside the parentheses represent failures using the default starting values in **R**, **MATLAB**, and **Python**. The numbers inside the parentheses represent failures using our optimisation method for starting values. Values highlighted in **bold red** indicate cases where the proposed starting values successfully reduced the number of failures compared to the default. A convergence failure indicates that the algorithm did not reach a valid solution within a maximum of 10,000 iterations. Results are reported for different mean parameters μ and predictor correlations ρ at covariate to sample size ratios (p/n) of 1% and 10%. Panel A (left) shows results for the Poisson distribution, and Panel B (right) shows results for the Gamma distribution.

converged in those scenarios without any failures. For the LR models, Panels C and D report the convergence failures across the extreme rare event and rare event scenarios. In both cases, only the LW estimator fails to converge, specifically when the covariate ratio is high ($p/n = 50\%$) and the correlation is negative or zero. Aside from the LW estimator, the optimisation method ensures all other shrinkage estimators remain stable and converge under these challenging conditions. The balanced class scenario is omitted from the table, as all estimators successfully converged without any failures in that setting.

Table SM.5.3: **Number of Convergence Failures with *sqrt* LF (DGP2)**

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

Notes. This table reports the number of convergence failures over $N = 100$ replications, each with $n = 500$ observations. The numbers outside the parentheses represent failures using the default starting values in **R**, **MATLAB**, and **Python**. The numbers inside the parentheses represent failures using our optimisation method for starting values. Values highlighted in **bold red** indicate cases where the proposed starting values successfully reduced the number of failures compared to the default. A convergence failure indicates that the algorithm did not reach a valid solution within a maximum of 10,000 iterations. Results are reported for different mean parameters μ and predictor correlations ρ at covariate to sample size ratios (p/n) of 1% and 10%. Panel A (left) shows results for the Poisson distribution, and Panel B (right) shows results for the Gamma distribution.

Table SM.5.4: **All Convergence Failures For Simulation Study in SM.4**

ρ	-0.75				-0.5				0				0.5				0.75			
p/n	1%	10%	25%	50%	1%	10%	25%	50%	1%	10%	25%	50%	1%	10%	25%	50%	1%	10%	25%	50%
Panel A) Poisson Distribution (*sqrt* LF)																				
glm2	0	0	0	**2**	0	0	0	**4**	0	0	0	**4**	0	0	0	**1**	0	0	0	**1**
RR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
EN	0	0	0	**1**	0	0	**1**	**2**	0	0	0	**7**	0	0	0	**8**	0	0	0	**7**
QIS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LW	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SR	0	0	0	**1**	0	0	0	**2**	0	0	0	**5**	0	0	0	**2**	0	0	0	**2**
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 (*sqrt* LF)																				
glm2	**3**	**1**	**1**	**1**	**1**	0	**1**	**1**	**3**	**2**	0	**1**	**4**	**1**	0	0	**1**	**2**	0	**1**
RR	**106**	**25**	**2**	**1**	**129**	**24**	**6**	**2**	**104**	**41**	**10**	**6**	**99**	**50**	**13**	**3**	**81**	**55**	**19**	**14**
EN	**112**	**25**	**5**	**10**	**131**	**25**	**9**	**14**	**99**	**41**	**10**	**16**	**98**	**52**	**13**	**14**	**81**	**55**	**20**	**35**
QIS	**3**	0	**1**	0	**1**	0	0	0	**3**	**2**	0	0	**4**	**1**	0	0	**1**	0	0	**1**
LW	**3**	0	**1**	0	**1**	0	0	0	**4**	**2**	0	0	**4**	**1**	0	0	**2**	0	0	**1**
SR	**3**	0	**1**	**1**	**1**	0	**1**	0	**4**	**2**	0	0	**4**	**1**	0	**1**	**1**	0	0	**2**
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**	**2**	**1**	0	**2**	**4**	**3**	**2**	**1**	0	**4**	**1**	**2**	0	**1**	0	0	**2**
DSh	**3**	0	**1**	0	**1**	0	0	0	**3**	**2**	0	0	**4**	**1**	0	0	**1**	0	0	**1**
Panel C) LR Extreme Rare Event Case: $\frac{n(Y=0)}{n(Y=1)} = \frac{5}{95}$ (*logit* LF)																				
glm2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
EN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
QIS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LW	0	0	0	**53**	0	0	0	**176**	0	0	0	**2**	0	0	0	0	0	0	0	0
SR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
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 D) LR Rare Event Case: $\frac{n(Y=0)}{n(Y=1)} = \frac{10}{90}$ (*logit* LF)																				
glm2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
EN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
QIS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LW	0	0	0	**65**	0	0	0	**15**	0	0	0	0	0	0	0	0	0	0	0	0
SR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
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

Notes. This table reports the number of convergence failures that happened for the PoR and GaR using the *sqrt* LF (Panels A and B) and LR using the *logit* LF (Panels C, D) 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 proposed optimisation method to adjust starting values. Non-zero failure counts are highlighted in **bold red** to indicate where the models struggled to converge.

SM.6 Prediction-Space Shrinkage: Combining Two Severity GLMs

In standard actuarial practice, the choice between a Gamma GLM and a Tweedie GLM for severity modelling is rarely straightforward. Both distributions are plausible candidates for modelling positive continuous losses, and neither dominates uniformly across the portfolio. Rather than discarding one model in favour of the other, the algorithm proposed in this section provides a principled method for combining their predictions via a single shrinkage intensity parameter $\delta \in [0, 1]$, estimated directly from the data. The blending algorithm proposed here operates at the *prediction level*: the parameter δ compresses the divergence between the log-scale rating factors of two independently fitted GLMs toward a data-calibrated consensus. The underlying principle is the same — reducing estimation uncertainty by pulling toward a reference point — but the level at which it operates is different. SM6 therefore extends the paper's central idea from parameter space to prediction space, offering practitioners a principled alternative to ad hoc severity model selection without requiring any re-estimation of the individual GLMs. The approach is computationally inexpensive, requires no additional distributional assumptions beyond those already embedded in the individual GLMs, and produces a blended rating factor vector that is fully interpretable and auditable — properties that are essential in regulated pricing environments where model outputs must be explainable to both internal governance functions and external supervisory authorities. We propose a fast algorithm for combining predictions from multiple GLMs. Since *log* LFs are most common in actuarial applications, we restrict attention to this setting, although extensions to other LFs are possible. Furthermore, we focus on combining two GLMs; more general schemes could be used to aggregate a larger number of models. However, in practice we do not recommend extending the procedure to many models, as predictive performance (generalisation to new data) may deteriorate. Consequently, combining two models is typically sufficient in large-scale applications from both a practical and accuracy perspective.

Under the *log* link assumption, our goal is to linearly shrink the rating factors of two GLMs with predicted values $\hat{y}_i^{(1)}$ and $\hat{y}_i^{(2)}$ for $1 \leq i \leq n$. The implementation requires strictly positive predictions, i.e., $\hat{y}_i^{(1)}, \hat{y}_i^{(2)} > 0$ for all $1 \leq i \leq n$, which does not reduce generality and leads to a stable Newton-type algorithm for multiplicative GLM blending.

Therefore, we define a multiplicative convex combination:

$$\hat{y}_i(\delta) = (\hat{y}_i^{(1)})^{\delta} (\hat{y}_i^{(2)})^{1-\delta}, \quad \delta \in [0, 1] \quad \text{for all } 1 \leq i \leq n.$$

Let $a_i = \log \hat{y}_i^{(1)}$, $b_i = \log \hat{y}_i^{(2)}$, and $d_i = a_i - b_i$ for all $1 \leq i \leq n$. Then,

$$\eta_i(\delta) = b_i + \delta d_i, \qquad \hat{y}_i(\delta) = \mu_i(\delta) = \exp\big(\eta_i(\delta)\big) \quad \text{for all } 1 \leq i \leq n.$$

We estimate the rating factor shrinkage intensity δ by minimising the least squares objective:

$$f(\delta) = \sum_{i=1}^{n} (y_i - \mu_i(\delta))^2 = \|\mathbf{y} - \boldsymbol{\mu}(\delta)\|_2^2.$$

Define the vector of residuals $\mathbf{r}(\delta) = \mathbf{y} - \boldsymbol{\mu}(\delta)$, where $\boldsymbol{\mu}(\delta) = \exp(\mathbf{b} + \delta \mathbf{d})$. The vectorised

gradient and Hessian are given by

$$g(\delta) = -2\,\mathbf{d}^\top(\mathbf{r}(\delta) \odot \boldsymbol{\mu}(\delta)) \quad \text{and} \quad H(\delta) = 2\,\mathbf{d}^\top(\mathbf{d} \odot \boldsymbol{\mu}(\delta) \odot (2\boldsymbol{\mu}(\delta) - \mathbf{y})),$$

where $\odot$ denotes the Hadamard (elementwise) product of two vectors or matrices of the same dimension.

We now modify the numerical procedure to improve numerical stability. Define $\eta_i(\delta) = b_i + \delta d_i$, and let $c = \max_i \eta_i(\delta)$ and $w_i = \exp(\eta_i(\delta) - c)$; then, $\mu_i(\delta) = w_i e^c$. The Newton ratio can be expressed in numerically stable form as

$$\frac{g(\delta)}{H(\delta)} = \frac{\sum_{i=1}^n (y_i e^{-c} - w_i) w_i d_i}{\sum_{i=1}^n d_i^2 w_i (2w_i - y_i e^{-c})}.$$

We next present the constrained Newton algorithm in Algorithm 3, which solves $\min_{\delta\in[0,1]} f(\delta)$ via projected Newton iterations, i.e., the update is given by

$$\delta \leftarrow \xi\left(\delta - \frac{g(\delta)}{H(\delta)}\right), \quad \text{where } \xi(x) = \min\big(1, \max(0, x)\big).$$

Algorithm 3 Stable Newton–Projection for GLM Blending

1: **Input:** Observed claims y_i, and two GLM predictions $\hat{y}_i^{(1)}$ and $\hat{y}_i^{(2)}$.
2: **Output:** Optimal rating factor shrinkage $\delta^* \in [0, 1]$.
3: Precompute $a_i = \log \hat{y}_i^{(1)}$, $b_i = \log \hat{y}_i^{(2)}$, and $d_i = a_i - b_i$ for all $1 \le i \le n$.
4: Initialise $\delta \leftarrow 0.5$ if the modeller is agnostic about the importance of the two GLMs.
5: **repeat**
6: Compute linear predictors $\eta_i = b_i + \delta d_i$ for all $1 \le i \le n$.
7: Stabilise and compute scaled response: $c = \max_i \eta_i$, $w_i = \exp(\eta_i - c)$ and $\tilde{y}_i = y_i e^{-c}$.
8: Compute gradient and Hessian:

$$g = -2\sum_{i=1}^n (\tilde{y}_i - w_i)\, w_i d_i \quad \text{and} \quad H = 2\sum_{i=1}^n d_i^2 w_i (2w_i - \tilde{y}_i).$$

9: Compute Newton update and its projection

$$\delta_{\text{new}} \leftarrow \delta - \frac{g}{H} \quad \text{and} \quad \delta_{\text{new}} \leftarrow \min\big(1, \max(0, \delta_{\text{new}})\big).$$

10: Stop if $|\delta_{\text{new}} - \delta| < \varepsilon$, otherwise update $\delta \leftarrow \delta_{\text{new}}$.
11: **until** convergence
12: **return** δ^*

We apply the proposed blending algorithm to combine the predictions of the Gamma and Tweedie models on the `freMTPL2` dataset as in Section 4.2. Table SM.6.1 presents the OOS performance metrics for the individual models with the combined GLM. Across all three panels (exactly one claim, exactly two claims, and all claims), the ensemble method effectively balances the predictive strengths of the individual components. The estimated shrinkage intensity

Table SM.6.1: **OOS Comparison: Gamma, Tweedie, and Multiplicative GLM Combination (*log* LF) on freMTPL2 Average Severity per Policy**

	Gamma			Tweedie			Combined GLM			
Model	**Raw**	**Norm**	**A/E - 1**	**Raw**	**Norm**	**A/E - 1**	**Raw**	**Norm**	**A/E - 1**	δ
Panel A) Exactly 1 Claim ($n = 23,571$)										
GLM2	0.115	0.172	1.35%	0.110	0.165	1.53%	0.113	0.169	1.52%	0.188
RR	**0.119**	**0.178**	5.77%	**0.115**	**0.173**	4.87%	**0.119**	**0.179**	5.52%	0.812
EN	0.111	0.167	2.93%	0.107	0.161	2.05%	0.113	0.170	2.52%	0.753
QIS	0.114	0.171	1.54%	0.110	0.164	1.54%	0.113	0.170	1.59%	0.322
LW	0.116	0.174	1.93%	0.111	0.166	1.98%	0.114	0.170	2.17%	0.406
SR	0.114	0.170	**-0.12%**	0.110	0.163	1.09%	0.110	0.164	1.09%	0.046
GSR	0.113	0.170	1.81%	0.111	0.166	1.73%	0.114	0.171	1.85%	0.638
St	0.115	0.172	5.66%	0.111	0.166	4.64%	0.114	0.170	5.10%	0.446
DSh	0.116	0.174	0.35%	0.113	0.170	**0.50%**	0.116	0.174	**0.49%**	0.355
Panel B) Exactly 2 Claims ($n = 1,298$)										
GLM2	**0.179**	**0.271**	11.95%	0.173	0.262	10.09%	0.178	**0.270**	11.97%	0.800
RR	0.132	0.204	14.58%	0.136	0.211	12.54%	0.128	0.198	14.08%	0.844
EN	0.128	0.201	12.99%	0.134	0.213	10.94%	0.120	0.192	12.81%	0.839
QIS	0.177	0.267	5.99%	**0.175**	**0.264**	8.24%	**0.179**	0.270	10.72%	0.781
LW	0.170	0.254	**5.24%**	0.172	0.256	**1.31%**	0.170	0.254	**4.66%**	0.830
SR	0.179	0.270	12.08%	0.172	0.259	10.18%	0.177	0.268	11.92%	0.803
GSR	0.173	0.263	12.90%	0.170	0.259	10.42%	0.173	0.264	12.56%	0.793
St	0.179	0.271	20.11%	0.173	0.262	20.49%	0.178	0.269	20.93%	0.800
DSh	0.161	0.241	15.40%	0.159	0.239	12.41%	0.161	0.241	15.28%	0.797
Panel C) All Claims ≥ 1 ($n = 24,944$)										
GLM2	0.105	0.157	2.18%	0.106	0.158	1.94%	0.106	0.158	2.17%	0.745
RR	**0.115**	**0.171**	5.08%	**0.114**	**0.171**	4.22%	**0.116**	**0.173**	4.92%	0.799
EN	0.109	0.163	3.28%	0.108	0.161	2.44%	0.111	0.166	3.40%	0.781
QIS	0.106	0.158	2.23%	0.106	0.158	1.81%	0.106	0.158	1.86%	0.759
LW	0.106	0.159	2.84%	0.107	0.160	2.20%	0.107	0.159	2.86%	0.764
SR	0.106	0.158	**1.15%**	0.107	0.159	**1.64%**	0.107	0.160	**1.67%**	0.090
GSR	0.106	0.158	2.56%	0.108	0.161	2.16%	0.106	0.159	2.55%	0.783
St	0.105	0.157	7.47%	0.106	0.159	5.21%	0.106	0.158	7.10%	0.719
DSh	0.109	0.163	1.99%	0.109	0.163	2.24%	0.110	0.164	2.06%	0.771

Notes. This table compares out-of-sample (OOS) performance in terms of Raw and Normalised Gini, as well as the A/E criterion, across the Gamma GLM (also reported in Table 3), the Tweedie GLM (also reported in Table 4), and their ensemble model, which combines the Gamma and Tweedie GLMs as specified in Algorithm 3, under a log link on the freMTPL2 dataset. Results are averaged over 500 replications. An $A/E - 1$ value closer to 0.00% indicates better calibration, while higher Gini values indicate better discriminatory performance. The best and second-best performers are highlighted separately within each model class and are shown in **red** and underlined, respectively.

δ weights the Gamma and Tweedie inputs for each estimator. For instance, the baseline `glm2` strongly favours the Tweedie model ($\delta = 0.188$) in Panel A, while the RR and EN place over 75% of the weight on the Gamma model predictions.

Comparing the performance criteria reveals a trade-off when combining information from the Tweedie and Gamma models. The combined GLM either matches or slightly improves upon the calibration (A/E) and discrimination (Gini) of the independent models. In Panel C, the combined models achieve A/E deviation comparable to the Tweedie GLM while maintaining the slightly better risk ranking capabilities of the Gamma GLM. The proposed shrinkage estimators, particularly SR and DSh, show reasonable calibration across the blended predictions. The SR estimator achieves the A/E deviation (1.67%) in Panel C while DSh maintains a tight calibration profile (0.49%) for the subset of policies with exactly one claim (Panel A).

References in Supplementary Material

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. (2025). Efficient and proper generalised linear models with power link functions. *Insurance: Mathematics and Economics*, 122(May):91–118.

Bodnar, O., Bodnar, T., and Parolya, N. (2022). Recent advances in shrinkage-based high-dimensional inference. *Journal of Multivariate Analysis*, 188:104826.

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.

Bühlmann, H. (1967). Experience rating and credibility. *ASTIN Bulletin*, 4(3):199–207.

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.

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

Goulet, V. (1998). Principles and application of credibility theory. *Journal of Actuarial Practice*, 6.

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

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.

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.

Ledoit, O. and Wolf, M. (2022). Quadratic shrinkage for large covariance matrices. *Bernoulli*, 28(3):1519–1547.

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.

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

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.

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.

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

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.

Whitney, A. W. (1918). The theory of experience rating. *Proceedings of the Casualty Actuarial Society*, 4:275–293.

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.

Wüthrich, M. V. and Merz, M. (2023). *Statistical foundations of actuarial learning and its applications*. Springer.

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.