LINEARPFN: AMORTIZED VARIABLE SELECTION FOR LINEAR MODELS WITH INTERACTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Spike-and-slab regression is a standard Bayesian formulation of variable selection: it returns a posterior distribution over which candidate effects are active rather than a single selected subset, so that every candidate effect carries an inclusion probability. Its cost grows exponentially with the number of candidate effects, so the posterior can be enumerated exactly only when the number of predictors is small. Beyond that reach, the posterior has to be approximated, typically by Markov chain Monte Carlo over the model space, which requires a fresh run for every dataset and, within a fixed budget of steps, may fail to converge. We present LINEARPFN, a prior-data fitted transformer network that amortizes spike-and-slab inference for linear models with main effects and pairwise interactions. The network is pretrained once on synthetic datasets, drawn from an explicitly specified prior, and a single forward pass over a new dataset returns posterior inclusion probabilities, posterior-mean coefficients and posterior predictive distributions with no per-dataset fitting. The prior is conjugate by design, so that the posterior for each fixed set of active effects has a closed form, and wherever the exact posterior is still computable by enumeration we verify the network's outputs against it. On real predictor matrices from published social-science datasets, with outcomes drawn from the prior so that the true active set is known, LINEARPFN attains a higher per-dataset selection AUC and a higher F_1 under the median probability model rule than five classical baselines. The lead holds when the coefficients, the interactions or the noise depart from the prior.

1 INTRODUCTION

A linear model explains an outcome y as a weighted sum of predictors, with one coefficient per predictor that gives the change in y per unit of that predictor. Despite the rise of nonlinear statistical modeling, quantitative claims in the sciences are still frequently stated in this form (Cohen et al., 2013): the coefficients are interpretable, their statistical properties are well understood, and each one can be tested directly against the theory that motivated it. Linear models are often extended with pairwise interactions, which state that the effect of one predictor on y depends on the level of a second predictor (Aiken et al., 1991).

Fitting a linear-plus-interaction model requires a decision about which predictors and which interactions to include. A standard Bayesian formulation for variable selection is spike-and-slab regression (Mitchell & Beauchamp, 1988; George & McCulloch, 1993). Its prior either sets each coefficient exactly to zero (the spike) or draws it from a Gaussian (the slab). Under this prior, each model is defined by its active set, the effects whose coefficients are nonzero and thus drawn from the slab, and the posterior is a distribution over these models. The posterior probability that an effect is active is its posterior inclusion probability (PIP). With p predictors the candidate effects are an intercept, p main effects and $\binom{p}{2}$ pairwise interactions, $d = 1 + p + \binom{p}{2}$ in total. With the intercept always included there are 2^{d-1} models. At $p = 10$ this is $2^{55} \approx 3.6 \times 10^{16}$ models. The exact posterior is a sum over all models, so in practice it can be computed only at small p. Beyond that reach, practice relies on Markov chain Monte Carlo (MCMC) over the model space (Madigan et al., 1995), adaptive sampling (Clyde et al., 2011) or variational approximations (Wang et al., 2020).

Prior-data fitted networks (PFNs) (Müller et al., 2021) approximate a posterior by a different route. A transformer is trained on datasets sampled from a prior. Each dataset is split into context rows,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

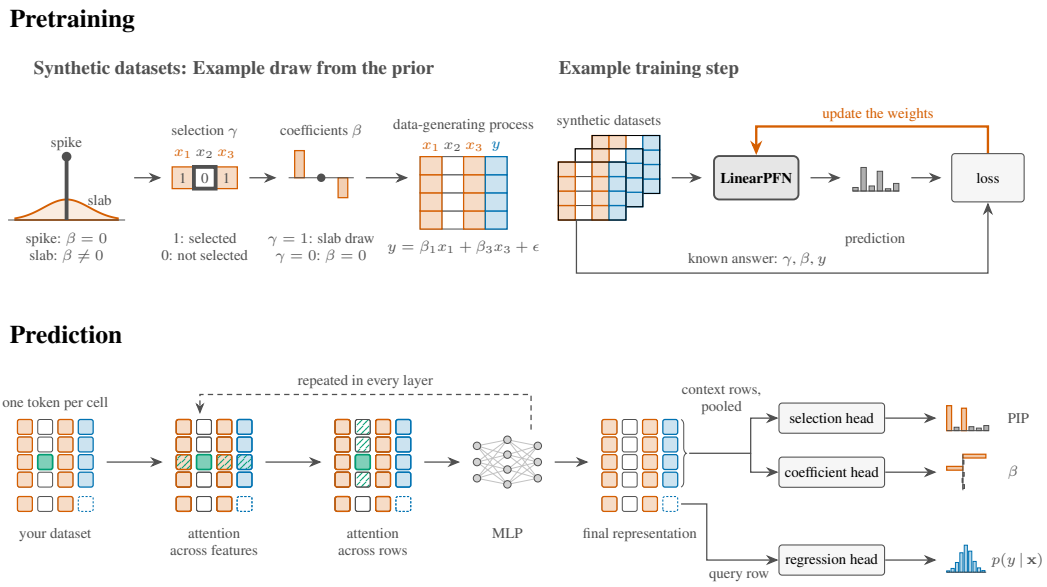

Figure 1: Pretraining and prediction of LINEARPFN. Top: each prior draw fixes an active set γ and spike-or-slab coefficients β for one synthetic dataset, and the loss compares the network's output with them. Bottom: in every layer, each cell (green) first attends to the other cells in its row and then to the other cells in its column (striped), and an MLP then updates each cell on its own. One forward pass over a new dataset returns inclusion probabilities and posterior-mean coefficients for every main effect and interaction, and a posterior predictive distribution for each query row.

which the network sees with their outcomes, and query rows, whose outcomes it predicts. The loss is evaluated on the query rows, so that the in-context predictions approximate the posterior predictive distribution under that prior. After training, no per-dataset fitting takes place. TABPFN (Hollmann et al., 2025; 2022) applies this idea to general tabular prediction with a broad prior over mostly nonlinear structural causal models. We instead restrict the prior to the model class that the variable-selection tradition in statistics already uses, linear main effects and pairwise interactions under a spike-and-slab prior (Mitchell & Beauchamp, 1988). The prior is conjugate, meaning that the posterior under each candidate model has a closed form. Therefore, for a small number of predictors the exact posterior can be computed by enumerating the models, and the outputs of the network are validated against it for up to 5 predictors.

LINEARPFN (Figure 1) is a transformer, based on NANOTABPFN (Pfefferle et al., 2025), pretrained on synthetic datasets drawn from an explicitly specified spike-and-slab prior. In one deterministic forward pass, a regression head returns the posterior predictive distribution, a selection head, trained with binary cross-entropy (BCE) against the simulated active sets in the manner of amortized causal discovery (Lorch et al., 2022), returns a PIP for every main effect and pairwise interaction, and a coefficient head returns posterior-mean coefficients. Our contributions are as follows:

1. **The first PFN for Bayesian variable selection.** To our knowledge, no PFN has returned inclusion probabilities over a regression model space.

2. **Validation against the exact posterior.** Because the posterior under each candidate model has a closed form, the exact posterior can be enumerated wherever the number of candidate effects allows it, and every output of the network is scored against it on six criteria fixed before training.

3. **A prior calibrated to social-science data.** The pretraining prior is built to resemble individual-level data as collected in the social sciences. Effect sizes are calibrated so that the standardized effect of an active main effect matches meta-analytic norms for published correlations in psychology.

4. **Benchmark evaluation.** Practical performance is measured against five baselines on 10,000 datasets that combine real predictor matrices from social-science data with outcomes drawn from the prior. Three variants each violate one assumption of the outcome model. On published tables with their actual outcomes, LINEARPFN is compared with an MCMC sampler over the model space that targets the same posterior.

2 RELATED WORK

Amortized Bayesian inference. Müller et al. (2021) demonstrated that PFNs approximate the posterior predictive distribution under a prior for Bayesian linear regression. TABPFN (Hollmann et al., 2022; 2025) scaled the idea to general tabular prediction with a broad prior over mostly nonlinear structural causal models. LINEARPFN is forked from NANOTABPFN (Pfefferle et al., 2025), a lightweight implementation of TABPFN. Neural posterior estimation (NPE) targets the parameters instead, usually with a normalizing flow over their joint posterior (Radev et al., 2022; Kipnis et al., 2025). LINEARPFN combines both: its regression head is trained as in a PFN, and its selection and coefficient heads are supervised by the simulated active set and coefficients, as in NPE.

Variable selection. Frequentist variable selection returns one subset of predictors, Bayesian selection a distribution over subsets. The lasso (Tibshirani, 1996) selects through an ℓ_1 penalty that sets some coefficients exactly to zero, and stability selection (Meinshausen & Bühlmann, 2010) keeps the effects the lasso selects often under resampling. Applied to pairwise interactions, the lasso can keep an interaction and drop its parent main effects. GLINTERNET (Lim & Hastie, 2026) and HIERNET (Bien & Tibshirani, 2020) enforce heredity inside the penalty, through overlapping groups and a convex constraint, respectively. Under a spike-and-slab prior (Mitchell & Beauchamp, 1988; George & McCulloch, 1993), every candidate effect carries an inclusion probability, and coefficients are averaged over models. Samplers over the model space (Madigan et al., 1995; Clyde et al., 2011) need a fitting run per dataset, and SUSIE (Wang et al., 2020; Zou et al., 2022) approximates the posterior variationally under an assumed maximum number of effects. Alrawkan et al. (2026) train a network on simulated linear models to classify predictors as active from ordinary least squares (OLS) summary statistics, without inclusion probabilities.

3 METHODS

3.1 THE PRIOR

LINEARPFN is trained on synthetic datasets drawn from a prior over data-generating processes. A dataset has n rows and p predictors. For row i, the outcome follows a linear model with pairwise interactions, $y_i = \beta_0 + \sum_j \beta_j x_{ij} + \sum_{j<k} \beta_{jk}\, x_{ij} x_{ik} + \epsilon_i$, with the intercept β_0, the main-effect coefficients β_j and the interaction coefficients β_{jk}. We write e for any of the labels 0, j and jk, so e runs over all $d = 1 + p + \binom{p}{2}$ candidate effects, and collect the coefficients in $\beta \in \mathbb{R}^d$. Given which effects are active, the coefficients follow a spike-and-slab prior, and the model reads in matrix form

$$\beta_e \mid \gamma_e, c, \sigma^2 \;\sim\; (1-\gamma_e)\,\delta_0 + \gamma_e\,\mathcal{N}(0,\, v_e\,\sigma^2), \qquad \mathbf{y} = \mathbf{Z}\beta + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_n). \tag{1}$$

The expanded predictor matrix $\mathbf{Z} = [\mathbf{1},\ \mathbf{X},\ (\mathbf{x}_j \odot \mathbf{x}_k)_{j<k}] \in \mathbb{R}^{n \times d}$ has one column per candidate effect: a column of ones for the intercept, the p z-standardized columns $\mathbf{x}_j$ of the predictor matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ for the main effects, and their elementwise products $\mathbf{x}_j \odot \mathbf{x}_k$ for the interactions. The active set $\gamma \in \{0,1\}^d$ marks which effects are active, with the intercept always active. An inactive effect ($\gamma_e = 0$) has a coefficient of zero, the point mass δ_0 (spike). An active effect ($\gamma_e = 1$) has a Gaussian coefficient (slab) with variance $v_e\sigma^2$. The noise ϵ has variance σ^2. Three quantities are drawn once per dataset:

$$\gamma \sim \pi(\gamma \mid p), \qquad c \sim \mathrm{LogNormal}(\mu_c, s_c^2), \qquad \sigma^2 \sim \mathrm{InvGamma}(a_0, b_0). \tag{2}$$

The active-effects prior $\pi(\gamma \mid p)$ is a distribution over active sets, described below. The slab scale c (truncated to $[c_{\min}, c_{\max}]$) sets how large the effects are, and σ^2 sets the noise level. The slab variance is $v_e = c$ for main effects, $v_e = c\rho$ for interactions, and $v_e = \tau_0^2$ for the intercept. An active main effect therefore has variance $c\,\sigma^2$ and an active interaction $c\,\rho\,\sigma^2$, where ρ is the fixed

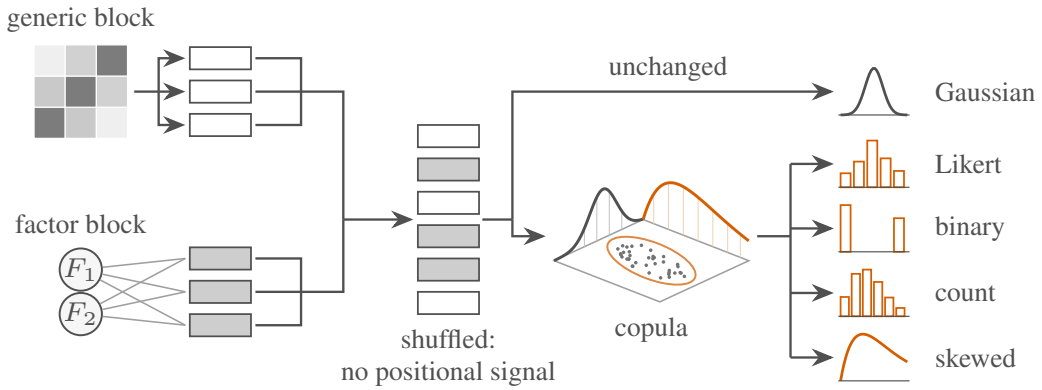

Figure 2: Predictor prior. Left: the generic block is generated jointly from a random correlation matrix R (top), and the factor block loads on latent factors (bottom). Middle: the two blocks are merged and the column order is shuffled. Right: a Gaussian copula maps each column to its marginal, either unchanged (Gaussian) or transformed to a Likert, binary, count or skewed family.

variance ratio of an interaction to a main effect. The intercept has variance $\tau_0^2\sigma^2$, with τ_0^2 fixed. Because every slab variance is proportional to σ^2, the slab scale c measures effect size in units of the noise variance. The remaining symbols are fixed hyperparameters, listed with the formal statement of the prior in Appendix B (Equation 6). A dataset is thus drawn in three steps: the predictor matrix $\mathbf{X}$ from the predictor prior, the active set γ from the active-effects prior, and the slab scale c and the noise variance σ^2 from the effect-size prior. Given these, β and $\mathbf{y}$ follow Equation 1. The prior was built to match real predictor matrices in the social sciences and to be conjugate, so that for small p the exact posterior is available for validation (Section 3.3).

Predictor prior. The number of rows n is drawn uniformly between 20 and 2,000, and, independently, the number of predictors p (i.e., features) uniformly between 2 and 30. Figure 2 shows how the predictor matrix $\mathbf{X}$ is constructed, which mimics individual-level data of the kind collected in the social sciences: each of the n rows is one independently sampled case (e.g., person, household, city, country) and each of the p columns one measured characteristic. A per-dataset share f of the columns forms the factor block, and the remaining columns form the generic block. The share follows a Beta distribution with a point mass at $f = 0$, so the factor block is empty in 19.7% of the datasets, and over all datasets the share of factor columns has median 0.25. The factor block stands in for the multi-item scales through which the social sciences measure latent constructs (e.g., IQ, social capital) (Borsboom et al., 2003). The generic block represents columns that are treated as manifest variables but still correlate with one another. Its columns are jointly Gaussian with a random correlation matrix (Davies & Higham, 2000), and each also receives a small loading on the factors. The column order is then shuffled, so position carries no information. A Gaussian copula (Cario & Nelson, 1997) maps each column to a marginal drawn from Gaussian, Likert, binary, count and skewed families. These distributions are the typical scales found in experimental and observational social-science data (Likert, 1932; Micceri, 1989; Coxe et al., 2009; Cain et al., 2017). Finally, every column is z-standardized. Appendix B.1 states every step in further detail.

Active-effects prior. Every dataset follows strong heredity: an interaction is eligible only when both of its parent main effects, those of its two predictors, are active. The active-effects prior mixes two ways of drawing γ (Figure 5). Under the *count prior*, the number k_{main} of active main effects is drawn from a zero-inflated truncated geometric distribution, whose point mass at zero is the share of datasets with no active effect, and k_{main} main effects are picked uniformly among the p candidates. The number of active interactions is drawn from the same family, and that many pairs are picked uniformly among the eligible ones. The number of active effects therefore does not grow with the number of predictors (see Ley & Steel, 2009). Under the *rate prior*, one inclusion rate for main effects and one for interactions are drawn per dataset, and every candidate main effect, then every eligible pair, is included with that probability, so more candidates yield more active effects. Each

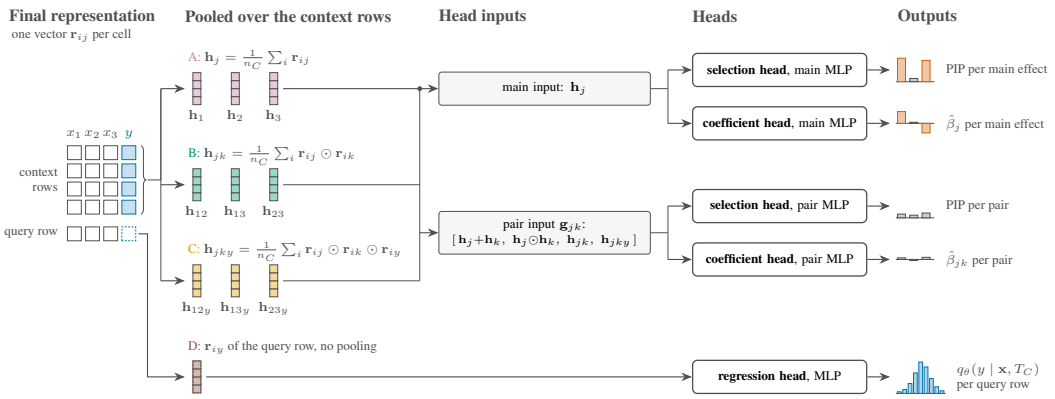

Figure 3: The three read-out heads. Left: the final representation, one vector $\mathbf{r}_{ij}$ per cell in row i and column j. Middle: the pooled vectors of Equation 3. Path A averages each predictor column over the context rows into $\mathbf{h}_j$. Paths B and C are the *combine-then-mean summaries*, which multiply the cells of two predictor columns within each row, in C also with the outcome cell, and then average into $\mathbf{h}_{jk}$ and $\mathbf{h}_{jky}$. Path D is the outcome cell $\mathbf{r}_{iy}$ of a query row, not pooled. Right: the selection and coefficient heads each use a main MLP on $\mathbf{h}_j$ and a pair MLP on $\mathbf{g}_{jk}$ (Equation 4). The regression head maps $\mathbf{r}_{iy}$ of each query row to a bar distribution.

dataset follows the count prior with probability 0.5 and the rate prior otherwise, so $\pi(\gamma \mid p)$ is the average of the two priors (Appendix B.2). The share of datasets with no active main effect is 0.103.

Effect-size prior. Drawing c per dataset rather than fixing it follows the mixtures of g-priors (Liang et al., 2008). The hyperprior of c is calibrated so that the standardized effect of an active main effect, $|\beta_j|\,\mathrm{sd}(\mathbf{x}_j)/\mathrm{sd}(\mathbf{y})$, matches meta-analytic norms for published correlations in psychology: prior quartiles of 0.078, 0.169 and 0.294 against norms of 0.11, 0.19 and 0.29 (Gignac & Szodorai, 2016). The ratio ρ is fixed at 0.5, so that interaction effects are smaller than main effects.

3.2 Network and Training

Architecture. LinearPFN is based on nanoTabPFN (Pfefferle et al., 2025). Each cell of the table is embedded into a vector of dimension $D = 512$, $\mathbf{u}_{ij} \in \mathbb{R}^D$ for predictor j in row i and $\mathbf{u}_{iy} \in \mathbb{R}^D$ for the outcome, which enters as one more column and is masked for the query rows. The backbone stacks 8 transformer blocks, and each block updates every cell vector in three steps. First, in attention across features, each cell attends to the cells of its row. Second, in attention across rows, each cell attends to the cells of its column in the context rows. Third, an MLP transforms each cell vector separately. The output of the last block is the final representation (see Figure 1, bottom).

Three heads and their inputs. Let $\mathbf{r}_{ij}, \mathbf{r}_{iy} \in \mathbb{R}^D$ denote the cell vectors after the last block, the final representation. The selection and coefficient heads need one input vector per candidate effect, computed from the context rows $i = 1, \ldots, n_C$. We pool in two orders, averaging a column directly or averaging a row-wise product of columns:

$$\mathbf{h}_j = \frac{1}{n_C}\sum_{i=1}^{n_C} \mathbf{r}_{ij}, \qquad \mathbf{h}_{jk} = \frac{1}{n_C}\sum_{i=1}^{n_C} \mathbf{r}_{ij} \odot \mathbf{r}_{ik}, \qquad \mathbf{h}_{jky} = \frac{1}{n_C}\sum_{i=1}^{n_C} \mathbf{r}_{ij} \odot \mathbf{r}_{ik} \odot \mathbf{r}_{iy}, \tag{3}$$

where $\odot$ is the elementwise product. A main effect has its own column, and its input is $\mathbf{h}_j$. An interaction has no column of its own. The table holds x_j and x_k but not their product, so no cell belongs to the interaction $x_j x_k$. Its input is assembled from the columns j and k in four parts,

$$\mathbf{g}_{jk} = \big[\, \mathbf{h}_j + \mathbf{h}_k,\ \ \mathbf{h}_j \odot \mathbf{h}_k,\ \ \mathbf{h}_{jk},\ \ \mathbf{h}_{jky} \,\big], \tag{4}$$

where $[\cdot]$ concatenates, so that $\mathbf{h}_j, \mathbf{h}_{jk}, \mathbf{h}_{jky} \in \mathbb{R}^D$ and $\mathbf{g}_{jk} \in \mathbb{R}^{4D}$. Every part is symmetric in j and k, so a pair receives the same input in either order. The first two parts are *mean-then-combine* summaries ($\mathbf{h}_j + \mathbf{h}_k$, $\mathbf{h}_j \odot \mathbf{h}_k$): each column is averaged over the rows before the two means are

combined. They carry what the backbone has stored about each parent predictor on its own, which the interaction needs because under strong heredity it can be active only if both parents are. They cannot carry the evidence for the interaction itself, because in general $\mathbf{h}_j \odot \mathbf{h}_k \neq \mathbf{h}_{jk}$: the product of two means differs from the mean of the products. The last two parts are *combine-then-mean* summaries ($\mathbf{h}_{jk}$, $\mathbf{h}_{jky}$), which multiply within each row and average afterwards.

Three heads read the final representation of the shared backbone in the same forward pass (see Figure 3, Appendix C). Each targets a different posterior quantity:

1. *Regression head.* It maps the outcome cell $\mathbf{r}_{iy}$ of each query row i to a bar distribution, as in TABPFN (Pfefferle et al., 2025; Hollmann et al., 2022): the range of possible outcomes is divided into fixed bins and the head predicts the probability of each bin, which gives a piecewise-constant density over y. Its loss is minimized in expectation by the posterior predictive distribution under the prior (Müller et al., 2021). The other two heads cannot replace it, because they supply neither the noise variance nor the posterior covariance of the coefficients (Appendix C.2). Removing its loss also lowers selection and coefficient accuracy (Appendix G.5).
2. *Selection head.* This head returns a PIP for every candidate effect. One MLP scores each main effect from $\mathbf{h}_j$, and a second MLP scores each interaction from $\mathbf{g}_{jk}$. A PIP is the posterior probability that an effect is active.
3. *Coefficient head.* Like the selection head, it takes $\mathbf{h}_j$ and $\mathbf{g}_{jk}$ as its inputs for the two MLPs, and returns one number per effect, the posterior mean $\mathbb{E}[\beta_e \mid T]$. This is the marginal mean, averaged over all models. The conditional mean $\mathbb{E}[\beta_e \mid T, \gamma_e = 1]$ is instead the size of the effect given that it is active. The two are linked by the PIP, $\mathbb{E}[\beta_e \mid T] = \Pr(\gamma_e = 1 \mid T)\,\mathbb{E}[\beta_e \mid T, \gamma_e = 1]$, so an effect with a PIP of one half receives half its conditional mean. The output is dense, so every candidate effect receives a value and none is set to zero by a selection step.

Loss. Training minimizes, over datasets drawn from the prior, the sum of three terms:

$$\mathcal{L} = \underbrace{-\tfrac{1}{|Q|}\textstyle\sum_{i\in Q} \log q_\theta(y_i \mid \mathbf{x}_i, T_C)}_{\text{regression head: NLL}} + \underbrace{\tfrac{1}{d-1}\textstyle\sum_e \mathrm{BCE}(\mathrm{sigmoid}(s_e), \gamma_e)}_{\text{selection head: BCE}} + \underbrace{\tfrac{1}{d-1}\textstyle\sum_e (b_e - \beta_e)^2}_{\text{coefficient head: MSE}}. \quad (5)$$

Here T_C is the context part of a dataset, Q its query rows, $(\mathbf{x}_i, y_i)$ the predictors and outcome of query row i, and q_θ the bar distribution of the regression head with network weights θ. The sums over e run over the $d-1$ effects other than the intercept. For effect e, s_e is the selection logit, b_e the coefficient output, and γ_e and β_e the active indicator and coefficient of Equation 1, known because the dataset was simulated. The first term is the loss of a PFN, whose expected minimizer is the posterior predictive distribution under the prior (Müller et al., 2021). The second is a proper scoring rule whose expected minimizer is the posterior probability that the effect is active, so the head is trained toward the exact PIPs, and that of the third is the posterior mean of the coefficient.

Training and prediction. LINEARPFN has 31.5M parameters and was trained for 37,500 optimizer steps at an effective batch of 64 datasets, 2.4M datasets in total, on two H200 GPUs (model card, optimizer and schedule in Appendix C, Table 4). Fitting a dataset is one forward pass that returns the PIPs and the coefficients. Following the median probability model (MPM) (Barbieri & Berger, 2004), LINEARPFN selects every effect whose PIP exceeds one half.

3.3 EVALUATION

Exact posterior. Given the active set γ and the slab scale c, the prior is conjugate, so the marginal likelihood of every model has a closed form (Appendix D). For $p \leq 5$ there are at most 1,450 models, few enough to sum over all of them and to integrate out c numerically. This yields the exact posterior, the probability of every model, and from it the exact PIPs and posterior-mean coefficients. It uses the same implementation of the prior as the synthetic training data. We compare LINEARPFN with it on held-out prior draws, on six criteria. They cover the predictive distribution, the coefficients and the PIPs (Table 1, Appendix E).

Benchmarks. Practical performance is measured on three benchmarks using real predictor matrices:

Table 1: Validation of LINEARPFN against the exact posterior on held-out prior draws with $p \leq 5$ (coverage on every p). The thresholds were fixed before the paper's model was trained.

predictive distribution and coefficients			*inclusion probabilities*		
criterion	threshold	value	criterion	threshold	value
headroom	≥ 0.95	0.9605	AUC gap	≥ -0.01	-0.0026
coverage error	≤ 0.03	0.0005	ECE gap	≤ 0.01	0.0034
coefficient correlation	≥ 0.99	0.9931	resolution ratio	≥ 0.90 / 0.85	0.9640

Real predictor matrices, outcomes from the prior is the main benchmark. The predictor matrices are the 383 tables of the social-science packages in the RDATASETS collection (Arel-Bundock, 2026; R Core Team, 2025) that pass fixed inclusion and cleaning rules. A table with more than 1,999 rows or 30 predictors is reduced to a random subset of that size, drawn once with a fixed seed (Appendix F.1). Each dataset draws a predictor matrix uniformly from the 383 predictor matrices and an outcome from the prior of Section 3.1 on the standardized columns of that predictor matrix, $\mathbf{y} = \mathbf{Z}\beta + \epsilon$, so the active set and the coefficients are known and every score is computed against them. Each draw has a density, the share k_{main}/p of predictors with an active main effect, and an $R^2 = \text{var}(\mathbf{Z}\beta)/\text{var}(\mathbf{y})$, the share of outcome variance explained by $\mathbf{Z}\beta$ in the drawn dataset. High density and high R^2 are rare under the prior, so unstratified draws would not represent those regimes with enough datasets to compare methods within them. Each draw is therefore assigned to its cell in a 5×5 grid of density by R^2 bins, and draws continue until every cell holds 400 datasets, 10,000 in all, so that sparse and dense active sets and weak and strong signal are represented equally (Appendix F.2). Every method receives the same standardized predictors and the same candidate set of $p + \binom{p}{2}$ effects. The baselines are the lasso (Tibshirani, 1996), stability selection (Meinshausen & Bühlmann, 2010; Shah & Samworth, 2013), SUSIE (Wang et al., 2020; Zou et al., 2022), GLINTERNET (Lim & Hastie, 2026) and HIERNET (Bien & Tibshirani, 2020), each with its own selection rule and ranking score. LINEARPFN selects by the MPM rule and ranks by PIP (Appendix F.3). Selection is scored per dataset by the F_1 of each method's own hard selection against the true active set and by the area under the ROC curve (AUC) with which its ranking separates active from inactive candidate effects. Coefficients are scored by the root mean squared error (RMSE) against the true coefficient vector over every candidate effect. Every metric is averaged over datasets (Appendix F.4).

Real predictor matrices, outcomes outside the prior is an ablation of the outcome model. It takes the same 10,000 datasets as the main benchmark and replaces one stage of the outcome model of Equation 1 at a time, holding every other quantity of the draw fixed. Given the draw $(\mathbf{X}, \gamma, c, \sigma^2, \beta)$ of a main-benchmark dataset, each variant keeps the predictor matrix $\mathbf{X}$, the active main effects, the slab scale c and the noise variance σ^2, and redraws the outcome as $\mathbf{y}' = \mathbf{Z}\beta' + \epsilon'$ with fresh noise from a random stream of its own. We test three variants.

1. **Fixed-magnitude coefficients.** In the main benchmark each active coefficient is drawn from a normal distribution, $\beta_e \sim \mathcal{N}(0, v_e\,\sigma^2)$. Here every active effect other than the intercept is set to $\beta'_e = s_e\sqrt{v_e\,\sigma^2}$ with a random sign $s_e \in \{-1, +1\}$.

2. **Many large interactions.** In the main benchmark interactions are rare. The median dataset has 1 active interaction (mean 3.50), 37.5% of the datasets have none, and each interaction has half the slab variance of a main effect ($\rho = 0.5$). This variant changes both properties. First, every pair whose two parent main effects are active becomes an interaction independently with probability one half, $\gamma'_{jk} \sim \text{Bernoulli}(1/2)$, so the count grows with the number of active main effects. Second, the slab variance of an interaction is raised from $c\rho$ to c, that is $\rho = 1$, so an interaction is on average as large as a main effect. The main-effect coefficients are kept, so strong heredity still holds.

3. **Heavy-tailed noise.** In the main benchmark the noise of row i is $\epsilon_i = \sigma z_i$ with $z_i \sim \mathcal{N}(0, 1)$. Here z_i is replaced by $t_i/\sqrt{3}$, where t_i is drawn from a Student-t distribution with three degrees of freedom. That distribution has variance 3, so $t_i/\sqrt{3}$ has unit variance like z_i, and the noise variance stays σ^2, but a few rows receive much larger errors, which act as outliers. The active set and the coefficients are unchanged.

Table 2: Benchmark on the real predictor matrices, 10,000 datasets with outcomes drawn from the prior. Over datasets, AUC and F_1 computed per dataset (/ds), with the first and third quartiles in brackets. Fit seconds are one fit on one CPU thread. On one NVIDIA L40 GPU, the same forward pass of LINEARPFN takes 0.03 seconds at the median [0.01, 0.10].

method	AUC/ds, mean	F_1/ds, mean	coef RMSE, mean	fit s, median
LinearPFN	**0.96** [0.95, 1.00]	**0.77** [0.67, 1.00]	0.041 [0.011, 0.051]	7.40 [1.07, 24.59]
hierNet	0.92 [0.88, 1.00]	0.53 [0.38, 0.67]	0.042 [0.011, 0.051]	25.45 [9.17, 79.28]
glinternet	0.79 [0.68, 0.97]	0.57 [0.43, 0.71]	0.044 [0.011, 0.052]	4.39 [1.18, 15.05]
SuSiE	0.85 [0.76, 1.00]	0.66 [0.46, 0.89]	0.051 [0.009, 0.066]	0.16 [0.08, 0.40]
lasso	0.82 [0.72, 0.96]	0.48 [0.33, 0.63]	0.049 [0.012, 0.059]	**0.14** [0.08, 0.71]
stability	0.82 [0.72, 0.95]	0.55 [0.40, 0.75]	0.068 [0.014, 0.074]	0.36 [0.33, 0.49]

Real outcomes scores LINEARPFN where no active set is known. We take 7 published tables that are widely used to demonstrate regression methods (Table 8). Here the reference is an MCMC sampler over the model space (Madigan et al., 1995) that targets the same posterior. Agreement is measured by the correlation and RMSE between the PIPs of LINEARPFN and of the sampler and between their posterior-mean coefficients over every candidate effect (Appendix G.4).

4 RESULTS

Fidelity to the exact posterior. LINEARPFN meets all six validation criteria (Table 1). Its gap to the exact posterior is -0.0026 in the AUC of the PIPs, and its posterior-mean coefficients correlate with the exact ones at 0.9931. On the 2,299 benchmark datasets within enumeration reach, its F_1 differs from that of the exact posterior by -0.030 and its coefficient RMSE by +0.015 (Appendix G.2). Beyond that reach, on 300 main-benchmark datasets, its F_1 differs from that of the MCMC sampler by -0.050 [-0.064, -0.036] and its AUC by -0.008.

Real predictor matrices, outcomes from the prior. Table 2 reports every method on the 10,000 datasets over the real predictor matrices (for paired differences see Appendix G.1). Figure 4 reports F_1, AUC and RMSE as a function of the number of predictors, number of rows, density and R^2.

Selection is ahead of every baseline in every slice. The F_1 margin grows with p and is widest at 200–499 rows, and the AUC margin shrinks as n grows (Figure 4). The lasso, GLINTERNET and HIERNET select many inactive effects, and SUSIE and stability selection miss more active ones. At high density HIERNET and GLINTERNET catch up. Against SUSIE, the closest baseline, the paired difference in F_1 per dataset is +0.118 [+0.114, +0.122] (bootstrap 95 % interval).

Coefficient error matches the best baseline. Except for SUSIE, the baselines return coefficients that are exactly zero outside their selected set, while LINEARPFN returns the dense posterior mean. We also score LINEARPFN hard-zeroed, setting to zero every coefficient whose PIP does not exceed 0.5. Against HIERNET, the best baseline, the paired difference in RMSE is -0.001 [-0.002, 0.000] for the dense posterior mean and 0.000 [-0.001, +0.001] hard-zeroed. For $p \leq 5$, the exact posterior has a coefficient RMSE of 0.050, and LINEARPFN exceeds it by +0.015 (Appendix G.2).

LINEARPFN *enables GPU acceleration.* On one CPU thread LINEARPFN takes 7.40 seconds at the median and is slower than every baseline except HIERNET (Table 2). Its forward pass also runs on a GPU, which the reference implementations of the baselines do not support, and on one NVIDIA L40 it takes 0.03 seconds. The baselines ran on one CPU thread, so this comparison includes the hardware. MCMC takes 57.43 seconds on one CPU thread, 9.1 times as long as LINEARPFN on the same thread and 1,924 times its GPU forward pass, on the same datasets.

Real predictor matrices, outcomes outside the prior. We test three variants of the outcome model (Equation 1): fixed-magnitude coefficients, many large interactions and heavy-tailed noise (Appendix G.3). Table 7 reports every method on each variant compared with the control.

LINEARPFN *keeps the highest AUC and F_1 in every variant.* The smallest F_1 margin is over SUSIE under many large interactions, +0.115 [+0.112, +0.119]. The one metric on which LINEARPFN is not first is coefficient RMSE under many large interactions. There its error of 0.064 is above that of three baselines, and the largest gap is to GLINTERNET, +0.005 [+0.003, +0.006].

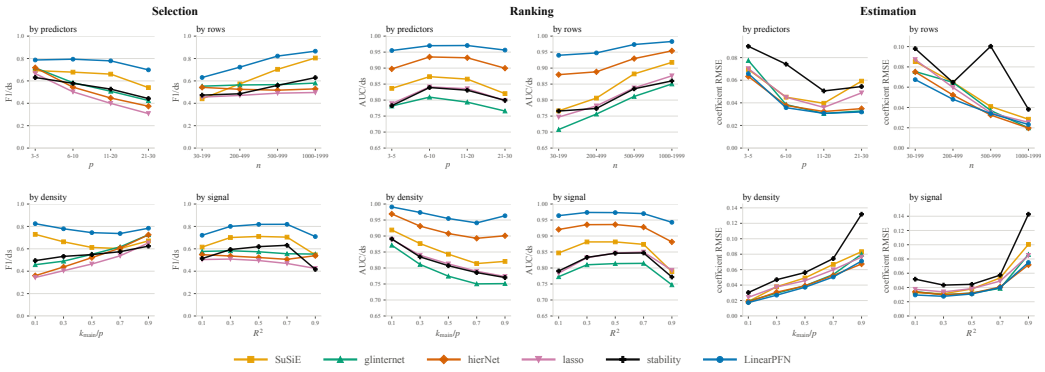

Figure 4: Benchmark on the 383 real predictor matrices, 10,000 datasets with outcomes drawn from the prior. Left: selection, F_1 per dataset at each method's own selection rule. Middle: ranking, AUC per dataset. Right: estimation, coefficient RMSE over every candidate effect. Each panel slices the same datasets by the number of predictors, the number of rows, the density k_{main}/p and the R^2; every point is the mean over the datasets of that slice. Density and R^2 are binned in the 5 bins of the benchmark grid, each 0.1 wide, and the axis marks the lower edge of each bin (Appendix F.2).

Many large interactions cost coefficient accuracy. They raise the coefficient error of every method, that of LINEARPFN by +0.023 [+0.022, +0.024], and lower the AUC of every method. Heavy-tailed noise changes the coefficient error of LINEARPFN by -0.001 [-0.001, -0.001] and raises F_1 slightly for every method. Only stability selection loses coefficient accuracy under it, through a few diverging refits that raise its mean error by +0.123 (Table 7).

Real outcomes. On 7 published tables with their actual outcome column, we compare LINEARPFN with the MCMC sampler that targets the same posterior (Table 8, Appendix G.4). The PIPs of the two correlate at 0.972 pooled over the 278 candidate effects, and the two MPMs are identical on 5 of the 7 tables. These tables have no known active set, so they do not measure selection accuracy.

5 DISCUSSION

We identify four limitations of our amortized variable-selection method. First, when two predictors are nearly collinear, the posterior mass splits across them, both PIPs can fall below one half, and the MPM rule then selects neither (Zou et al., 2022). Unlike SUSIE (Wang et al., 2020), LINEARPFN reports no credible sets that group such predictors. Second, without a GPU, LINEARPFN is slow. On one CPU thread it is the second slowest of the six methods, ahead of HIERNET only. However, neither the cost on a CPU nor the fixed prior is intrinsic to the approach. Pruning, quantization and knowledge distillation are established routes to a cheaper forward pass, which we did not pursue. Third, in coefficient error LINEARPFN only matches the best baseline, although it leads in selection. Fourth, the network is trained on a single prior, which assumes strong heredity, Gaussian noise, numeric predictors and independent rows. Its lead in selection held when the coefficients, the interactions or the noise departed from this prior. Follow-up work could condition the network on the prior's hyperparameters, so that users can state their own prior without retraining.

LINEARPFN is a transformer pretrained on an explicit conjugate spike-and-slab prior over main effects and pairwise interactions. Its outputs are validated against the exact posterior wherever that posterior can be enumerated, and they are compared with an MCMC sampler on real outcomes. On real predictor matrices with outcomes drawn from its prior it selects better than every baseline. Amortization reduces variable selection to one forward pass, which returns the selected variables and their coefficients in 0.03 seconds at the median on one GPU. The MCMC sampler over the same posterior needs 57.43 seconds at the median on one CPU thread and meets its convergence criteria on 93.0% of runs. Bayesian model averaging with interactions thereby becomes feasible where it has to be repeated many times.

AI USE STATEMENT

The authors developed the theoretical models, conceptual frameworks, and research methodology. We used generative AI tools to discuss and give feedback on these models and on the design of the experiments, to implement methods, and to clean and reformat datasets. We used AI tools to find related work. The authors read the cited papers and checked each citation. We used AI tools to correct grammar, spelling, and wording. For interpretation, we used AI tools to suggest possible explanations for observed patterns. The authors checked these suggestions against the data and decided which conclusions to report. We did not use generative AI tools to provide critical steps in proofs of mathematical claims, to formulate mathematical claims, to generate synthetic datasets, to translate text, to support qualitative or thematic data analysis, or to propose or refine hypotheses. The authors reviewed all AI-assisted work, including all AI-generated code. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI. The synthetic datasets are drawn by code from the prior stated in Section 3.1 and were not produced by a generative model.

ETHICS STATEMENT

This work uses publicly available data only: 383 tables of the RDATASETS collection as predictor matrices, and 7 published regression datasets with their actual outcome (Appendix G.4). No data were collected from human subjects for this work, and no personal data are processed beyond what the sources already publish. The method is a statistical tool for variable selection. Its outputs are probabilities under a stated prior and do not by themselves support causal claims.

REPRODUCIBILITY STATEMENT

The prior is specified in Section 3.1 and Appendix B, with every fixed value in Table 3. The exact posterior and the MCMC sampler are described in Appendix D, the network, its training and the model card in Appendix C and Table 4, and the validation criteria in Appendix E. The construction of every benchmark, the methods and the metrics are given in Appendices F and G. The predictor matrices come from a public collection and are selected and cleaned by fixed rules (Appendix F.1), and the real-outcome tables are listed with their preparation in Appendix G.4. The network is implemented in PyTorch (Paszke et al., 2019), and the exact posterior and the MCMC sampler in NumPy (Harris et al., 2020) and SciPy (Virtanen et al., 2020). The lasso and stability selection use scikit-learn (Pedregosa et al., 2011), and the other baselines run in R (R Core Team, 2025). Code, benchmark data and the trained checkpoints will be made available upon acceptance.

REFERENCES

Leona S Aiken, Stephen G West, and Raymond R Reno. *Multiple regression: Testing and interpreting interactions*. Sage, 1991.

By Riyadh Alrawkan, Edward Boone, Ryad Ghanam, and Anton Westveld. Linear models, variable selection, artificial intelligence. *arXiv preprint arXiv:2604.27191*, 2026.

Vincent Arel-Bundock. *Rdatasets: A collection of datasets originally distributed in various R packages*, 2026. URL `https://vincentarelbundock.github.io/Rdatasets`. R package version 1.0.0.

Maria Maddalena Barbieri and James O. Berger. Optimal predictive model selection. *The Annals of Statistics*, 32(3), 2004. doi: 10.1214/009053604000000238. URL `http://dx.doi.org/10.1214/009053604000000238`.

Jacob Bien and Rob Tibshirani. *hierNet: A Lasso for Hierarchical Interactions*, 2020. URL `https://CRAN.R-project.org/package=hierNet`. R package version 1.9.

Denny Borsboom, Gideon J Mellenbergh, and Jaap Van Heerden. The theoretical status of latent variables. *Psychological Review*, 110(2):203, 2003.

Meghan K Cain, Zhiyong Zhang, and Ke-Hai Yuan. Univariate and multivariate skewness and kurtosis for measuring nonnormality: Prevalence, influence and estimation. *Behavior research methods*, 49(5):1716–1735, 2017.

Marne C Cario and Barry L Nelson. Modeling and generating random vectors with arbitrary marginal distributions and correlation matrix. Technical report, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois, 1997.

Merlise A Clyde, Joyee Ghosh, and Michael L Littman. Bayesian adaptive sampling for variable selection and model averaging. *Journal of Computational and Graphical Statistics*, 20(1):80–101, 2011.

Jacob Cohen, Patricia Cohen, Stephen G West, and Leona S Aiken. *Applied multiple regression/correlation analysis for the behavioral sciences*. Routledge, 2013.

Stefany Coxe, Stephen G West, and Leona S Aiken. The analysis of count data: A gentle introduction to poisson regression and its alternatives. *Journal of Personality Assessment*, 91(2): 121–136, 2009. doi: 10.1080/00223890802634175. URL https://doi.org/10.1080/00223890802634175. PMID: 19205933.

Philip I Davies and Nicholas J Higham. Numerically stable generation of correlation matrices and their factors. *BIT Numerical Mathematics*, 40(4):640–651, 2000.

Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2), 2004. ISSN 0090-5364. doi: 10.1214/009053604000000067. URL http://dx.doi.org/10.1214/009053604000000067.

Edward I George and Robert E McCulloch. Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.

Gilles E Gignac and Eva T Szodorai. Effect size guidelines for individual differences researchers. *Personality and Individual Differences*, 102:74–78, 2016.

Manuel Gloeckler, JP Manzano-Patrón, Stamatios N Sotiropoulos, Cornelius Schröder, and Jakob H Macke. Scalable simulation-based model inference with test-time complexity control. *arXiv preprint arXiv:2603.15292*, 2026.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

Trevor Hastie, Robert Tibshirani, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Roger W Johnson. Fitting percentage of body fat to simple body measurements. *Journal of Statistics Education*, 4(1), 1996.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Alex Kipnis, Marcel Binz, and Eric Schulz. metabeta–a fast neural model for Bayesian mixed-effects regression. *arXiv preprint arXiv:2510.07473*, 2025.

Eduardo Ley and Mark FJ Steel. On the effect of prior assumptions in Bayesian model averaging with applications to growth regression. *Journal of Applied Econometrics*, 24(4):651–674, 2009.

Feng Liang, Rui Paulo, German Molina, Merlise A Clyde, and Jim O Berger. Mixtures of g priors for Bayesian variable selection. *Journal of the American Statistical Association*, 103(481):410–423, 2008.

Rensis Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 1932.

Michael Lim and Trevor Hastie. *glinternet: Learning Interactions via Hierarchical Group-Lasso Regularization*, 2026. URL https://CRAN.R-project.org/package=glinternet. R package version 1.0.13.

Lars Lorch, Scott Sussex, Jonas Rothfuss, Andreas Krause, and Bernhard Schölkopf. Amortized inference for causal structure learning. *Advances in Neural Information Processing Systems*, 35: 13104–13118, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

David Madigan, Jeremy York, and Denis Allard. Bayesian graphical models for discrete data. *International Statistical Review/Revue Internationale de Statistique*, pp. 215–232, 1995.

Nicolai Meinshausen and Peter Bühlmann. Stability Selection. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 72(4):417–473, 2010. ISSN 1369-7412, 1467-9868. doi: 10.1111/j.1467-9868.2010.00740.x. URL https://academic.oup.com/jrsssb/article/72/4/417/7076513.

Theodore Micceri. The unicorn, the normal curve, and other improbable creatures. *Psychological Bulletin*, 105(1):156, 1989.

Toby J Mitchell and John J Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032, 1988.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do Bayesian inference. *arXiv preprint arXiv:2112.10510*, 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library, 2019. URL https://arxiv.org/abs/1912.01703.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Alexander Pfefferle, Johannes Hog, Lennart Purucker, and Frank Hutter. nanoTabPFN: A Lightweight and Educational Reimplementation of TabPFN. *arXiv preprint arXiv:2511.03634*, 2025. URL https://arxiv.org/abs/2511.03634.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2025. URL https://www.R-project.org/.

Stefan T. Radev, Ulf K. Mertens, Andreas Voss, Lynton Ardizzone, and Ullrich Köthe. BayesFlow: Learning Complex Stochastic Models With Invertible Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(4):1452–1466, 2022. doi: 10.1109/TNNLS.2020.3042395.

Cornelius Schröder and Jakob H. Macke. Simultaneous identification of models and parameters of scientific simulators. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 43895–43927. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/schroder24b.html.

Rajen D Shah and Richard J Samworth. Variable selection with error control: another look at stability selection. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 75 (1):55–80, 2013.

Nurit Tal-Or, Jonathan Cohen, Yariv Tsfati, and Albert C Gunther. Testing causal direction in the influence of presumed media influence. *Communication Research*, 37(6):801–824, 2010.

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

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Gao Wang, Abhishek Sarkar, Peter Carbonetto, and Matthew Stephens. A simple new approach to variable selection in regression, with application to genetic fine mapping. *Journal of the Royal Statistical Society, Series B*, 82:1273–1300, 2020. doi: 10.1101/501114.

Yuxin Zou, Peter Carbonetto, Gao Wang, and Matthew Stephens. Fine-mapping from summary data with the "Sum of Single Effects" model. *PLoS Genetics*, 18:e1010299, 2022. doi: 10.1371/journal.pgen.1010299.

A GLOSSARY

A.1 ABBREVIATIONS

AUC area under the ROC curve

BCE binary cross-entropy

MCMC Markov chain Monte Carlo

MPM median probability model

NPE neural posterior estimation

OLS ordinary least squares

PFN prior-data fitted network

PIP posterior inclusion probability

RMSE root mean squared error

A.2 NOTATION

Datasets and predictors.

- T: a dataset, a table of n rows. T_C: its context rows. Q: the index set of its query rows.
- n, p: the number of rows and of predictors of a dataset. n_C: the number of context rows.
- i: the row index, $i = 1, \ldots, n$. j, k: predictor indices.
- $\mathbf{X} \in \mathbb{R}^{n \times p}$: the predictor matrix, with z-standardized columns $\mathbf{x}_j$. $\mathbf{x}_i$ in Equation 5: the predictors of query row i.
- ℓ: the index of a real predictor matrix $\mathbf{X}_\ell \in \mathbb{R}^{n_\ell \times p_\ell}$, with expanded predictor matrix $\mathbf{Z}_\ell$.
- $\mathbf{y}$: the outcome vector. The prior does not standardize it.
- $d = 1 + p + \binom{p}{2}$: the number of candidate effects, intercept included. e: the index of a candidate effect.
- $\mathbf{Z}$: the expanded predictor matrix, a column of ones, the columns of $\mathbf{X}$ and their pairwise products $\mathbf{x}_j \odot \mathbf{x}_k$. $\odot$: the elementwise product.

Prior.

- $\gamma \in \{0, 1\}^d$: the active set, with the intercept always active. One value of γ is a model. $\mathbf{Z}_\gamma$: the columns of $\mathbf{Z}$ of the active effects.
- $\beta \in \mathbb{R}^d$: the coefficient vector, zero outside the active set. β_γ: its active entries.
- ϵ: the noise vector. σ^2: the noise variance, with prior $\mathrm{InvGamma}(a_0, b_0)$.
- δ_0: the point mass at zero, the spike.
- c: the slab scale, the variance of an active main effect relative to σ^2, with prior $\mathrm{LogNormal}(\mu_c, s_c^2)$ truncated to $[c_{\min}, c_{\max}]$.
- ρ: the fixed variance ratio of an active interaction to an active main effect.
- τ_0^2: the fixed slab variance of the intercept relative to σ^2.
- v_e: the slab variance of effect e relative to σ^2: c for a main effect, $c\,\rho$ for an interaction and τ_0^2 for the intercept.
- $\pi(\mathbf{X} \mid n, p)$: the predictor prior.
- $\pi(\gamma \mid p)$: the active-effects prior. π_{count}, π_{rate}: the count prior and the rate prior. α: the probability of the count prior.
- k_{main}, k_{int}: the numbers of active main effects and of active interactions. M_{int}: the number of eligible pairs.
- ζ_{main}, $\bar{k}_{\mathrm{main}}$, ζ_{int}, $\bar{k}_{\mathrm{int}}$: the zero masses and geometric means of the count prior. ω_{main}, ω_{int}: the inclusion rates of the rate prior.

- k_{main}/p: the density of a dataset.
- f: the share of columns in the factor block. R: the correlation matrix of the generic block. m: the number of latent factors. h_j^2: the communality of column j.
- R^2: $\text{var}(\mathbf{Z}\beta)/\text{var}(\mathbf{y})$ over the rows of a dataset.

Network and training.

- $\mathbf{u}_{ij}$, $\mathbf{u}_{iy}$: the embedded cell of predictor j and the embedded outcome cell in row i.
- $\mathbf{r}_{ij}$, $\mathbf{r}_{iy}$: the same cells after the last block, the final representation. D: the dimension of every cell vector.
- $\mathbf{h}_j$, $\mathbf{h}_{jk}$, $\mathbf{h}_{jky}$: the context-row means of $\mathbf{r}_{ij}$, of $\mathbf{r}_{ij} \odot \mathbf{r}_{ik}$ and of $\mathbf{r}_{ij} \odot \mathbf{r}_{ik} \odot \mathbf{r}_{iy}$ (Equation 3).
- $\mathbf{g}_{jk}$: the input of the pair MLPs for the interaction of predictors j and k (Equation 4).
- θ: the network weights. q_θ: the predictive distribution of the regression head.
- s_e, b_e: the selection logit and the coefficient output for effect e. $\text{sigmoid}(\cdot)$: the logistic function.

Evaluation and exact posterior.

- F_1: the harmonic mean of precision and recall of a hard selection against the true active set.
- β', γ', $\mathbf{y}'$, ϵ': the redrawn quantities of a variant with outcomes outside the prior. z_i, t_i: the standard normal and the Student-t draw of the noise of row i.
- $\mathbf{z}^*$, y^*: the expanded predictors and the outcome of a new row.
- w_γ: the posterior probability of model γ under the exact posterior.
- V_0, V_n, m_n, a_n, b_n: the prior slab matrix and the posterior quantities of the normal-inverse-gamma regression given a model.

B THE PRIOR

A dataset is drawn in six steps:

$$n \sim \text{U}\{n_{\min}, \dots, n_{\max}\}, \qquad p \sim \text{U}\{p_{\min}, \dots, p_{\max}\}, \tag{6a}$$

$$\mathbf{X} \sim \pi(\mathbf{X} \mid n, p), \qquad \mathbf{Z} = [\mathbf{1},\ \mathbf{X},\ (\mathbf{x}_j \odot \mathbf{x}_k)_{j<k}], \tag{6b}$$

$$\gamma \sim \pi(\gamma \mid p), \tag{6c}$$

$$c \sim \text{LogNormal}(\mu_c, s_c^2), \qquad \sigma^2 \sim \text{InvGamma}(a_0, b_0), \tag{6d}$$

$$\beta_e \mid \gamma_e, c, \sigma^2 \sim (1-\gamma_e)\,\delta_0 + \gamma_e\,\mathcal{N}(0,\ v_e\,\sigma^2), \qquad v_e = \begin{cases} c & \text{main effect,} \\ c\,\rho & \text{interaction,} \\ \tau_0^2 & \text{intercept,} \end{cases} \tag{6e}$$

$$\mathbf{y} = \mathbf{Z}\beta + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_n). \tag{6f}$$

Equation 6a draws the number of rows and of predictors. Equation 6b draws the predictor matrix from the predictor prior (Appendix B.1) and expands it by a column of ones and all pairwise products. Equation 6c draws the active set from the active-effects prior (Appendix B.2). Equation 6d draws the slab scale, truncated to $[c_{\min}, c_{\max}]$, and the noise variance. Equation 6e sets every inactive coefficient to zero and draws every active one from a normal slab whose variance depends on the type of effect. Equation 6f adds Gaussian noise to $\mathbf{Z}\beta$ to form the outcome. Table 3 lists every fixed value of the paper's model. The size (n, p) is drawn once per training batch, so the datasets of one batch share it (Appendix C.3). The prior on σ^2 fixes the scale of the outcome, so the posterior changes when $\mathbf{y}$ is rescaled. On data from outside the prior, the outcome is therefore standardized before the forward pass, as for the real outcomes (Appendix G.4).

Table 3: The prior of the paper's model: its fixed values, and the standardized effects $|\beta_e|\,\mathrm{sd}(\mathbf{z}_e)/\mathrm{sd}(\mathbf{y})$ it implies over the active effects of 10,000 prior draws, where $\mathbf{z}_e$ is the column of effect e in $\mathbf{Z}$.

Predictor prior	
rows n, predictors p	U{20, …, 2,000}, U{2, …, 30}
factor share f	0 with probability 0.15, else Beta(1.5, 3)
factors m	U{1, …, $\lceil q/3 \rceil$}, q the size of the factor block
communality: mean, concentration	U(0.1, 0.92), U(4, 40), cap 0.98
high band: probability, mean	0.15, U(0.9, 0.995), cap 0.995
share of h_j^2 on other factors	0.15
sign flip of a loading	0.3
loading $\lvert w_j \rvert$ of a generic column	U(0, 0.35)
marginal types	Gaussian 0.6, Likert 0.2, binary 0.1, count 0.05, skewed 0.05
binary base rate, Likert levels	U(0.2, 0.5), 2 to 7
Active-effects prior	
heredity	strong
probability α of the count prior	0.5
count prior, main effects: ζ_{main}, $\bar{k}_{\mathrm{main}}$	0.09, 3.5
count prior, interactions: ζ_{int}, $\bar{k}_{\mathrm{int}}$	0.5, 1.5
rate prior: ω_{main}, ω_{int}	U(0.02, 0.6), U(0.05, 0.4)
Effect-size prior	
slab scale c: median e^{μ_c}, s_c, truncation	0.15, 0.6, [0.01, 1.5]
interaction ratio ρ, intercept variance τ_0^2	0.5, 0.25
noise variance σ^2: a_0, b_0	3, 1
Implied standardized effects, Q1 / median / Q3	
active main effects	0.078 / 0.169 / 0.294
active interactions	0.046 / 0.099 / 0.172
meta-analytic norms (Gignac & Szodorai, 2016)	0.11 / 0.19 / 0.29

B.1 Predictor prior

The predictor matrix is drawn in three steps (Figure 2). The latent columns are drawn in two blocks, their order is shuffled (Section 3.1), and a Gaussian copula maps each column to its marginal. Table 3 lists the fixed values of every step.

Of the p columns, $\mathrm{round}(f\,p)$ form the factor block and the rest form the generic block. The point mass of f at zero keeps predictor matrices without factor structure frequent at every p. The columns of the generic block are jointly Gaussian with correlation matrix R. The matrix R is a random correlation matrix with a given spectrum, built by Givens rotations (Davies & Higham, 2000), and its eigenvalues are a Dirichlet draw scaled to sum to the block size. Some generic blocks are therefore nearly orthogonal and others nearly singular.

The factor block follows the factor model $\mathbf{F}\Lambda^\top + \Psi^{1/2}\mathbf{E}$ with m latent factors. The number m is drawn uniformly between one and a fixed fraction of the block size. Each column j receives a primary factor and a communality h_j^2, the share of its variance that the factors explain. The communalities follow a Beta distribution whose mean and concentration are drawn once per dataset, so one dataset has weak and another strong factor structure. A fixed share of h_j^2 goes to the primary loading. The remainder is spread over the other factors in Dirichlet-distributed proportions, the cross-loadings. Each loading changes sign with a fixed probability, as reverse-coded items do. The uniqueness is $1 - h_j^2$, so every latent column has unit variance. With a fixed probability the communality mean is drawn from a high band instead. A factor that carries a few columns with communalities near one produces a group of near-duplicate columns. Real predictor matrices contain such groups, and the low band produces them rarely. Each generic column also receives a small loading on a random direction in factor space, because the correlation between the blocks of real data is never exactly zero.

The copula (Cario & Nelson, 1997) pushes each latent column through the standard normal distribution function and then through the quantile function of a type drawn per column from the Gaussian, Likert, binary, count and skewed families. The parameters of the type (number of levels, base rate, mean count, skew) are drawn per column as well. A column that the transform makes constant draws its type again on the same latent column. Every column is then z-standardized. The interaction columns of $\mathbf{Z}$ are products of the standardized columns and are not standardized again. Discretization attenuates the latent correlations, and skewed marginals produce heavy-tailed interaction columns. A predictor prior with Gaussian columns only would produce neither.

B.2 The active-effects prior in full

Figure 5 shows one draw of the active set and the outcome. Both components of the active-effects prior first choose the active main effects and then the active interactions among the $M_{\text{int}} = \binom{k_{\text{main}}}{2}$ eligible pairs.

Under the count prior, a count over M candidates is zero with probability ζ. Otherwise it follows a geometric distribution on the positive integers with mean $\bar{k}$, truncated to at most M and renormalized. The main effects use $M = p$, $\zeta_{\text{main}} = 0.09$ and $\bar{k}_{\text{main}} = 3.5$. The interactions use $M = M_{\text{int}}$, $\zeta_{\text{int}} = 0.5$ and $\bar{k}_{\text{int}} = 1.5$, and their count is zero when no pair is eligible. Given the counts, the active main effects and the active interactions are uniform subsets of their candidates, so

$$\pi_{\text{count}}(\gamma \mid p) = \frac{P_p(k_{\text{main}})}{\binom{p}{k_{\text{main}}}} \, \frac{P_{M_{\text{int}}}(k_{\text{int}})}{\binom{M_{\text{int}}}{k_{\text{int}}}},$$

where P_M is the count distribution over M candidates.

Under the rate prior, an inclusion rate ω_{main} is drawn uniformly between 0.02 and 0.6, and each main effect is active independently with probability ω_{main}. A second rate ω_{int} is drawn uniformly between 0.05 and 0.4, and each eligible pair is active independently with probability ω_{int}. Integrating out the two rates gives

$$\pi_{\text{rate}}(\gamma \mid p) = \mathbb{E}_{\omega_{\text{main}}}\big[\omega_{\text{main}}^{k_{\text{main}}}(1-\omega_{\text{main}})^{p-k_{\text{main}}}\big] \, \mathbb{E}_{\omega_{\text{int}}}\big[\omega_{\text{int}}^{k_{\text{int}}}(1-\omega_{\text{int}})^{M_{\text{int}}-k_{\text{int}}}\big],$$

and each expectation over a uniform rate is a difference of incomplete Beta functions.

Each dataset follows the count prior with probability $\alpha = 0.5$ and the rate prior otherwise, so $\pi(\gamma \mid p) = \alpha\, \pi_{\text{count}}(\gamma \mid p) + (1-\alpha)\, \pi_{\text{rate}}(\gamma \mid p)$. The exact posterior evaluates this closed form for every model it enumerates. A dataset has no active main effect, and under strong heredity no active effect at all, with probability 0.103 under the prior. This share combines the zero mass ζ_{main} of the count prior with the probability that the rate prior includes no main effect, which is largest at small p (0.273 at $p = 2$).

C Network and training details

C.1 Architecture

Figure 3 shows the three heads. The query cells of the outcome column are filled with the mean of the context outcomes before embedding, which hides the query outcomes from the network. Each transformer block has 16 attention heads. The bar distribution of the regression head has 512 bins of equal width on $[-8, 8]$ and a half-normal tail on each side (Müller et al., 2021; Hollmann et al., 2025). The main and pair MLPs of the selection and coefficient heads have one hidden layer of 1,024 units each, and they share the pooled vectors of Equation 3, which are computed once per forward pass. The selection head returns one logit for every main effect and every pair, with no heredity mask. The per-effect heads read the context rows only, and their outputs do not depend on the order of those rows. During pretraining the query rows exist because the predictive term of the loss is scored on held-out rows.

C.2 Why the predictive distribution needs its own head

In a linear model the posterior mean of the coefficients determines the mean of the prediction exactly. For a new row with expanded predictors $\mathbf{z}^*$,

$$\mathbb{E}[y^* \mid T] \;=\; (\mathbf{z}^*)^\top \, \mathbb{E}[\beta \mid T],$$

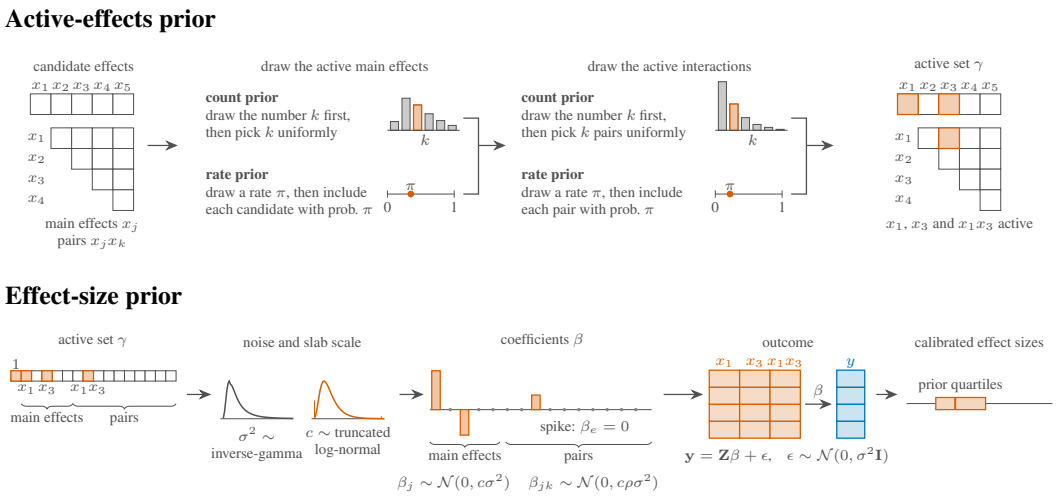

Figure 5: The active-effects and effect-size priors, one draw with $p = 5$. Top: the active main effects and then the active interactions are drawn from the candidate set, under the count prior (the number first, then a uniform subset) or the rate prior (a rate first, then one inclusion per candidate). Bottom: the noise variance and the slab scale are drawn, active effects receive slab coefficients and inactive ones a spike at zero, and the outcome is $\mathbf{y} = \mathbf{Z}\beta + \epsilon$.

so the coefficient head already implies the predictive mean. The predictive variance does not follow from the posterior means. It decomposes as

$$\mathrm{Var}(y^* \mid T) \;=\; \mathbb{E}[\sigma^2 \mid T] \;+\; (\mathbf{z}^*)^\top \mathrm{Cov}(\beta \mid T)\, \mathbf{z}^*.$$

The first term is the expected noise variance, the scatter of a new observation around the regression line even if the coefficients were known exactly. It does not shrink as the data grow, while the second term does, along every direction the data inform. The second term is the uncertainty about the regression line at that row. If coefficient vectors were drawn from the posterior, it would be the variance of the predictions $(\mathbf{z}^*)^\top \beta$ they give. Written out,

$$(\mathbf{z}^*)^\top \mathrm{Cov}(\beta \mid T)\, \mathbf{z}^* \;=\; \sum_e (z_e^*)^2 \,\mathrm{Var}(\beta_e \mid T) \;+\; \sum_{e \neq e'} z_e^* z_{e'}^* \,\mathrm{Cov}(\beta_e, \beta_{e'} \mid T),$$

so it depends on the covariances between the coefficients as well as on their variances.

C.3 Training

The datasets of one batch share their size (n, p). The share of query rows is drawn per batch, uniformly between 0.1 and 0.5, and the remaining rows form the context. Muon (Jordan et al., 2024) trains the weight matrices of the backbone and AdamW (Loshchilov & Hutter, 2019) every other parameter, at the learning rates of Table 4. Both follow 4,000 steps of linear warm-up and then a cosine decay. The gradients of all three loss terms reach the shared backbone. The run was data-parallel on one node. The accumulation steps were divided among the two processes, so the effective batch equals that of a single-GPU run. The scheduler requeued the job once. The run resumed from its last checkpoint, written every 30 minutes, at step 5,684, and the wall time in Table 4 counts every segment.

D The exact posterior and the MCMC sampler

D.1 Enumeration and quadrature

Given (γ, c), the prior of Equation 6 is a normal-inverse-gamma regression on the active columns $\mathbf{Z}_\gamma$. The coefficients follow $\beta_\gamma \mid \sigma^2 \sim \mathcal{N}(0, \sigma^2 V_0)$, where V_0 is the diagonal matrix of the slab

Table 4: Model card of LINEARPFN. Wall time sums the segments of the requeued job, and GPU-hours are wall time times the number of GPUs.

layers / heads / embedding	8 / 16 / 512	parameters	31.5M (31,541,766)
MLP hidden units	1,024	GPUs	$2 \times$ H200
optimizer steps	37,500	wall time	14.6 h
effective batch	64 (4×16)	GPU-hours	29.1
datasets seen	2,400,000	throughput	0.72 steps/s
learning rate	Muon 0.02, AdamW 0.001	peak GPU memory	57.3 GiB

variances v_e of the active effects, and $\sigma^2 \sim \mathrm{InvGamma}(a_0, b_0)$. The marginal likelihood is

$$\log p(\mathbf{y} \mid \gamma, c) = \frac{1}{2} \log \frac{|V_n|}{|V_0|} + \log \frac{\Gamma(a_n)}{\Gamma(a_0)} + a_0 \log b_0 - a_n \log b_n - \frac{n}{2} \log 2\pi, \tag{7}$$

with the posterior quantities

$$V_n = \left(V_0^{-1} + \mathbf{Z}_\gamma^\top \mathbf{Z}_\gamma\right)^{-1}, \qquad m_n = V_n \mathbf{Z}_\gamma^\top \mathbf{y},$$
$$a_n = a_0 + \tfrac{n}{2}, \qquad b_n = b_0 + \tfrac{1}{2}\left(\mathbf{y}^\top \mathbf{y} - m_n^\top \mathbf{Z}_\gamma^\top \mathbf{y}\right).$$

Because ρ is fixed, c is the only continuous hyperparameter, and the evidence of a model is

$$p(\mathbf{y} \mid \gamma) = \int p(\mathbf{y} \mid \gamma, c)\, p(c)\, dc.$$

The integral is computed by quadrature.

The reference enumerates every model with nonzero prior mass. These are the models with the intercept active in which every active interaction has both parents active. At $p = 5$, the largest p the reference enumerates, there are 1,450 of them. The posterior model probabilities are

$$w_\gamma \propto \pi(\gamma \mid p)\, p(\mathbf{y} \mid \gamma),$$

with the prior of Appendix B.2. They give the exact PIPs as $\sum_\gamma w_\gamma \gamma$ and the posterior-mean coefficients as the average of m_n over models and quadrature nodes, with zeros outside each model. For a new row with expanded predictors $\mathbf{z}^*$, each pair of a model and a node contributes a Student-t distribution with $2a_n$ degrees of freedom,

$$t_{2a_n}\left((\mathbf{z}_\gamma^*)^\top m_n,\ \frac{b_n}{a_n}\left(1 + (\mathbf{z}_\gamma^*)^\top V_n\, \mathbf{z}_\gamma^*\right)\right),$$

with its location and squared scale as arguments. The exact posterior predictive distribution is the mixture of these components, and its quantiles are found by bisection on the mixture distribution function.

The integral over c uses 20 Gauss–Legendre nodes on $\log c$ over $[\log c_{\min}, \log c_{\max}]$. The mass of a node is its quadrature weight times the Jacobian times the truncated lognormal density of c, and the masses are renormalized to sum to one. The posterior is then a discrete distribution over pairs of a model and a node. The computation runs in two passes. The first pass computes the evidence of every model at every node and sums over the nodes, so the model probabilities and the PIPs use every model. The second pass computes m_n, V_n and b_n only for the pairs that together carry all but 1.0×10^{-9} of the posterior mass, and the coefficient means and the predictive distribution use these pairs.

The node count was checked against a grid of 30 nodes on 12 prior draws in each cell that crosses 32 and 1,024 rows with 3 and 5 predictors. At 20 nodes the largest deviation occurs at $n = 1,024$ and $p = 5$. It is 1.4×10^{-7} in the PIPs and 9.8×10^{-6} in the log evidence of a model. Over the tested grids of 12, 15 and 20 nodes, the largest PIP deviation falls from 4.1×10^{-4} to 1.4×10^{-7}.

D.2 THE MCMC SAMPLER

Beyond the enumeration reach the reference is an MCMC sampler over the model space (Madigan et al., 1995) that targets the same posterior. It is a Metropolis–Hastings chain on the active set

with moves that add, drop or swap one effect. Its target is $\pi(\gamma \mid p)\, p(\mathbf{y} \mid \gamma)$ with the evidence of Equation 7 and the quadrature of Appendix D.1, so models that violate strong heredity have zero target mass. Several chains start from prior draws, and the burn-in states are discarded. PIPs are visit frequencies, and the posterior means average the posterior mean of each visited model. A run meets the convergence criteria when the largest split $\hat{R}$ is below 1.1 and the smallest effective sample size is above 200.

E VALIDATION AGAINST THE EXACT POSTERIOR

Networks trained on simulated data to return a posterior over model structure also exist for scientific simulators (Schröder & Macke, 2024; Gloeckler et al., 2026), but a simulator whose likelihood cannot be evaluated offers no exact posterior to compare with. The conjugate prior of LINEARPFN provides one for a small number of predictors.

The validation panel consists of prior draws. It crosses 32, 128, 512, 1,024 and 2,000 context rows with 3, 5, 8, 10, 15, 20 and 30 predictors, with 100 datasets per cell and 64 query rows per dataset. In the cells with $p \leq 5$, the exact cells, the reference of Appendix D is computed on the same datasets. The coverage criterion pools every cell of the panel, and the other five criteria use the exact cells.

1. *Predictive headroom.* In each exact cell, $(\mathrm{NLL}_{\mathrm{marginal}} - \mathrm{NLL}_{\mathrm{model}})/(\mathrm{NLL}_{\mathrm{marginal}} - \mathrm{NLL}_{\mathrm{exact}})$ is at least 0.95. Here the marginal is a Gaussian fitted to the context outcomes, and each NLL is the mean over the query rows. Cells in which the exact posterior improves on the marginal by less than 0.02 nats are reported and not scored, because the ratio of two small differences is unstable. The worst scored cell counts.
2. *Coefficient recovery.* The correlation between the coefficient head's posterior means and the exact posterior means over the candidate effects, pooled over the exact cells, is at least 0.99. The probe is reported beside it and not scored. It reaches 0.9958 on the same datasets.
3. *Coverage.* The central predictive intervals at the nominal levels 50, 80 and 90 % cover the query outcomes within 3 percentage points of nominal.
4. *Selection ranking.* The AUC of the PIPs against the simulated active set is at most 0.01 below the AUC of the exact posterior on the same datasets, in the worst of the pooled, main-effect and interaction subsets.
5. *Selection calibration.* The ECE of the PIPs exceeds the ECE of the exact posterior by at most 0.01, in the worst subset.
6. *Selection resolution.* The resolution term of the Murphy decomposition of the PIPs is at least 0.90 times that of the exact posterior on the pooled effects and at least 0.85 times on each subset.

Resolution alone can be raised by extreme PIPs at the cost of calibration, so it is scored together with the ECE.

The probe reads coefficients from the predictive distribution. It evaluates the mean of the predictive distribution at $3\,d$ probe points, which are context rows resampled with replacement plus Gaussian jitter with standard deviation 0.1 on each standardized predictor. It then regresses these means on the columns of $\mathbf{Z}$ by ridge regression with penalty 1.0×10^{-3}. Under the exact posterior the predictive mean is linear in $\mathbf{Z}$ (Appendix C.2), so the probe recovers the exact posterior means.

F BENCHMARK SETUP

F.1 THE REAL PREDICTOR MATRICES

The predictor matrices come from the RDATASETS collection (Arel-Bundock, 2026; R Core Team, 2025), an index of 3,648 datasets shipped with 104 R packages. A hand-made label file assigns each package a domain, and 20 packages are labeled as social-science or psychology data. Their datasets form the frame of 480 tables, after tables described as simulated, tables stored as time series or contingency tables, and tables with fewer than 30 rows or 3 usable columns in the index are set aside.

Table 5: RDATASETS entries that did not become a predictor matrix, by reason.

reason	entries	reason	entries
package outside the frame	2,908	described as simulated	17
index: fewer than 30 rows	93	fewer than 3 columns after cleaning	15
index: fewer than 3 usable columns	81	more than 50.0% of rows lost	13
time-series or contingency-table object	69	same title as another table	8
pure time series	32	download failed	3
same columns as a larger table	23	fewer than 30 rows after cleaning	3
total			3,265

One cleaning function with no per-dataset switches processes every table of the frame. It drops identifier columns. It admits numeric, logical and two-level columns, the last two coded 0/1, and drops categorical columns with more than two levels. It drops columns with more than 50.0% missing values and keeps the complete cases. A table that loses more than 50.0% of its rows in this step is excluded. The function then drops constant columns, every column whose absolute correlation with an earlier kept column is at least 0.95, and every column with fewer than 10 rows outside its most frequent value. A table is excluded if fewer than 30 rows or 3 columns remain. Pure time series and duplicate tables are removed. This leaves 383 eligible tables, and each of them is one predictor matrix of the main benchmark and of its variants with outcomes outside the prior. A table with more than 1,999 rows or 30 predictors is subsampled once with a seed derived from its name, in rows for 121 tables and in columns for 15. The rule on near-constant columns is applied again after the subsampling. Every table that does not become a predictor matrix carries a logged reason (Table 5).

The 383 predictor matrices come from 20 packages in 7 fields: economics (237 predictor matrices), political science (58), general social science (32), education (24), criminology (17), psychology (14) and demography (1). Their median numbers of rows and predictors are 758 and 9, and 87 of them have at most 5 predictors. The correlation rule acts before the subsampling, so a subsampled predictor matrix can exceed it. The largest absolute correlation between two columns of a predictor matrix has median 0.80 and maximum 0.965. In 61 of the predictor matrices the rows are repeated observations of the same units.

F.2 CONSTRUCTION OF THE MAIN BENCHMARK

Every dataset of the main benchmark is accepted by rejection sampling. A draw picks a predictor matrix uniformly from the 383 predictor matrices and draws $(\gamma, c, \sigma^2, \beta, \mathbf{y})$ from the prior of Appendix B on its standardized columns. Density and R^2 are defined as in Section 3.3. Each axis of the grid has 5 bins of width 0.1, with equally spaced lower edges from 0.1 to 0.9, and the top bin includes one. The bins do not touch. A draw whose density or R^2 lies between two bins or below the lowest bin is discarded, so the benchmark contains no dataset without an active main effect and none with an R^2 below the lowest bin. The 25 cells hold 19.1 % of the prior mass (Figure 6), and within them every cell is weighted equally. Each cell has its own random stream and draws until it is full. The share of accepted draws ranges from 3.6% in the most frequent cell to 0.013% in the rarest, because the prior rarely produces dense or high-R^2 datasets (Figure 6).

The density k_{main}/p_ℓ takes only multiples of $1/p_\ell$, so a predictor matrix with few predictors reaches only some density bins. None of the 2,299 datasets with $p_\ell \leq 5$ lies in the lowest density bin, because $1/p_\ell$ is at least the upper edge of that bin. Every predictor matrix appears in at least one cell, and a cell holds between 101 and 220 distinct predictor matrices. The cells therefore differ in the predictor matrices they hold, and a comparison across cells mixes the effect of density and R^2 with the effect of those predictor matrices.

F.3 METHODS

The lasso (Tibshirani, 1996) selects the effects that are nonzero at the penalty with the smallest cross-validated error, and it ranks effects by their point of entry on the regularization path. Stability selection with complementary pairs (Meinshausen & Bühlmann, 2010; Shah & Samworth, 2013)

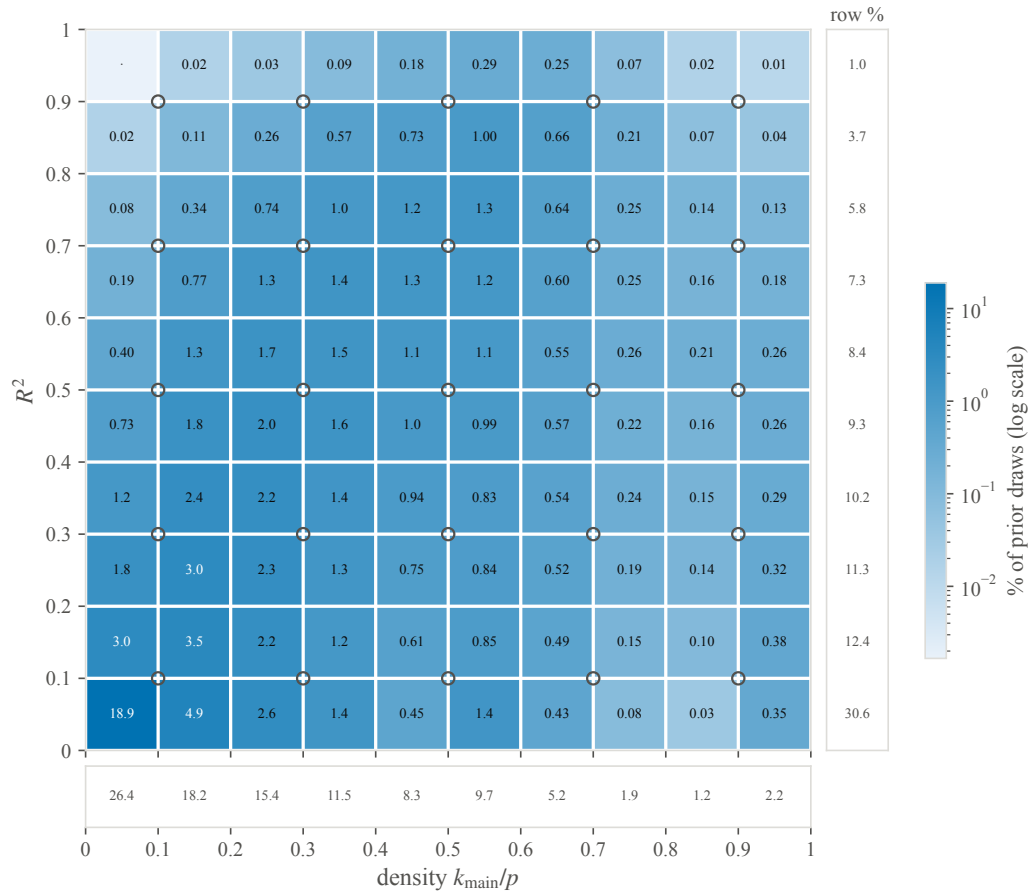

Figure 6: Share of 60,000 datasets drawn from the prior in each cell of the plane of density and R^2, on a log color scale, with the row and column totals in the margins. Rings mark the lower-left corners of the 25 cells of the main benchmark, which together hold 19.1 % of the draws. A dot marks a cell whose share rounds to zero.

fits the lasso path on 50 complementary pairs of half-samples. It selects the effects whose selection frequency is at least 0.75 and ranks by that frequency. It returns no coefficients of its own, so its coefficients are the least-squares fit on its selected effects, and its fit seconds include that fit. SuSiE (Wang et al., 2020) with at most 20 single effects selects the effects with a PIP above one half and ranks by PIP. glinternet (Lim & Hastie, 2026) and hierNet (Bien & Tibshirani, 2020) select the effects at the penalty with the smallest cross-validated error under their heredity constraints, and they rank by the point of entry on the path. The hard-zeroed read-out of LinearPFN (Section 4) keeps the intercept and involves no refit.

The coefficients of the lasso, glinternet and hierNet are penalized estimates. The exact posterior, like LinearPFN, returns a posterior mean over models, in which no coefficient is exactly zero.

F.4 Metrics

The metrics are those of Section 3.3, with the following details. The AUC of a dataset ranks the candidate effects by the method's own score, the PIP for LinearPFN and SuSiE, the selection frequency for stability selection and the point of entry on the path for the lasso, glinternet and hierNet. Tied scores receive their mean rank. The AUC is undefined on a dataset in which every candidate effect is active or none is, so the AUC average covers 9,974 of the 10,000 datasets.

Table 6: Main benchmark: paired differences, LINEARPFN minus each baseline on the same datasets, with bootstrap 95 % intervals. The last column pairs the hard-zeroed read-out of LINEARPFN with each baseline. Every Wilcoxon signed-rank p is below 0.001, except for the RMSE against HIERNET ($p = 0.155$) and the hard-zeroed RMSE against GLINTERNET ($p = 0.775$).

baseline	F_1/ds	AUC/ds	coefficient RMSE	RMSE, hard-zeroed
HIERNET	+0.244	+0.044	−0.001	0.000
	[+0.238, +0.249]	[+0.043, +0.046]	[−0.002, 0.000]	[−0.001, +0.001]
GLINTERNET	+0.205	+0.173	−0.004	−0.003
	[+0.200, +0.210]	[+0.170, +0.176]	[−0.005, −0.003]	[−0.004, −0.002]
SuSiE	+0.118	+0.110	−0.011	−0.010
	[+0.114, +0.122]	[+0.107, +0.113]	[−0.012, −0.010]	[−0.011, −0.009]
lasso	+0.293	+0.144	−0.008	−0.007
	[+0.288, +0.298]	[+0.141, +0.146]	[−0.009, −0.007]	[−0.008, −0.006]
stability selection	+0.219	+0.147	−0.027	−0.026
	[+0.214, +0.223]	[+0.144, +0.150]	[−0.043, −0.019]	[−0.042, −0.018]

The coefficient RMSE excludes the intercept. A paired difference is LINEARPFN minus the other method on the datasets that both score. Its interval is a seeded bootstrap 95 % interval of the mean difference over 2,000 resamples, and its p value is that of the Wilcoxon signed-rank test. Fit time is the wall time of one fit on one CPU thread of the cluster, measured inside each method's own process around the fit and its tuning.

G BENCHMARK RESULTS

G.1 MAIN BENCHMARK: FULL RESULTS

Table 6 gives the paired differences between LINEARPFN and each baseline. The means and quartiles of every metric are in Table 2.

The 61 panel predictor matrices contribute 1,645 datasets. The simulated outcomes are independent across rows on every predictor matrix, so the panel structure enters only through the predictors. LINEARPFN's F_1 is 0.78 on the panels and 0.77 on the cross-sections. Its margin over HIERNET is +0.241 and +0.244, and over SUSIE +0.110 and +0.120.

On one NVIDIA L40 GPU the forward pass takes 0.03 seconds at the median over the same 10,000 datasets, 0.20 at the 90th percentile and 0.42 at most. This is 222 times faster than the single-thread median, with a one-off 0.92 seconds to load the checkpoint. The exact posterior takes 0.04 seconds at the median on the 2,299 datasets it reaches. The MCMC sampler takes 57.43 seconds at the median and 866.31 at the 90th percentile on its 300 datasets, 9.1 times the single-thread median of LINEARPFN on the same datasets.

G.2 DISTANCE TO THE EXACT POSTERIOR AND TO MCMC

On the 2,299 datasets of the main benchmark with $p_\ell \leq 5$, the exact posterior is computed on the same data. LINEARPFN minus the exact posterior is -0.009 [-0.012, -0.007] in AUC per dataset and -0.030 [-0.035, -0.024] in F_1 per dataset. Its coefficient RMSE exceeds that of the exact posterior by +0.015 [+0.014, +0.017]. This excess is the amortization error in the coefficients. The RMSE of the exact posterior itself, 0.050, is the error that remains with exact inference under the prior.

Beyond the enumeration reach the reference is the MCMC sampler of Appendix D.2. It ran on 300 datasets of the main benchmark drawn at random over all p_ℓ, for 200,000 steps per chain at the median. 93.0% of the runs meet its convergence criteria, with a median largest split $\hat{R}$ of 1.001 and a median smallest effective sample size of 4695. LINEARPFN minus the MCMC sampler is -0.008 [-0.012, -0.005] in AUC, -0.050 [-0.064, -0.036] in F_1 and +0.009 [+0.007, +0.012] in RMSE, over all 300 datasets.

Table 7: Outcomes outside the prior: every method on the control (the main benchmark) and on the three variants, means over datasets, with the paired change from the control in parentheses. AUC and F_1 per dataset, coefficient RMSE over every candidate effect.

	control			fixed-magnitude coefficients		
method	AUC/ds	F_1/ds	RMSE	AUC/ds	F_1/ds	RMSE
LINEARPFN	0.96	0.77	0.041	0.98 (+0.011)	0.85 (+0.072)	0.044 (+0.003)
HIERNET	0.92	0.53	0.042	0.94 (+0.018)	0.52 (−0.006)	0.045 (+0.003)
GLINTERNET	0.79	0.57	0.044	0.85 (+0.062)	0.56 (−0.011)	0.048 (+0.004)
SuSiE	0.85	0.66	0.051	0.89 (+0.034)	0.70 (+0.044)	0.063 (+0.012)
lasso	0.82	0.48	0.049	0.86 (+0.040)	0.48 (+0.004)	0.056 (+0.007)
stability selection	0.82	0.55	0.068	0.86 (+0.042)	0.60 (+0.046)	0.098 (+0.030)

	many large interactions			heavy-tailed noise		
method	AUC/ds	F_1/ds	RMSE	AUC/ds	F_1/ds	RMSE
LINEARPFN	0.94 (−0.026)	0.76 (−0.017)	0.064 (+0.023)	0.97 (+0.002)	0.79 (+0.014)	0.040 (−0.001)
HIERNET	0.88 (−0.039)	0.56 (+0.034)	0.060 (+0.018)	0.92 (+0.002)	0.53 (+0.005)	0.040 (−0.002)
GLINTERNET	0.75 (−0.045)	0.60 (+0.029)	0.059 (+0.015)	0.80 (+0.005)	0.57 (+0.006)	0.043 (−0.001)
SuSiE	0.83 (−0.021)	0.64 (−0.014)	0.072 (+0.021)	0.86 (+0.002)	0.66 (+0.007)	0.051 (−0.001)
lasso	0.81 (−0.009)	0.52 (+0.044)	0.061 (+0.012)	0.82 (+0.003)	0.49 (+0.005)	0.047 (−0.002)
stability selection	0.80 (−0.015)	0.55 (0.000)	0.118 (+0.050)	0.82 (+0.003)	0.56 (+0.005)	0.191 (+0.123)

G.3 OUTCOMES OUTSIDE THE PRIOR

The variants are built on the datasets of the main benchmark (Appendix F.2). For each dataset the draw of the main benchmark is replayed from its random stream and checked against the stored dataset, which recovers c and σ^2, and each variant then applies its change from a random stream of its own. Under fixed-magnitude coefficients every active coefficient keeps the second moment of its slab, and no active effect lies near zero. Under many large interactions the median dataset has 3 active interactions (mean 9.09, maximum 135), and 15.7% of the datasets have none. The density k_{main}/p of every dataset is the same under all three variants. LINEARPFN and the exact posterior assume the prior of Section 3.1, so on the three variants the exact posterior is the posterior under a wrong prior, and its distance from the truth is the cost of the prior itself. The ECE is computed over every candidate effect of the panel and is reported for the methods that return PIPs. Every method returned a fit on every dataset, so every panel holds 10,000 datasets. Table 7 lists every method on every panel with its paired difference from the control. Figure 7 slices each variant by the number of predictors, the number of rows, the density and the R^2, as Figure 4 slices the main benchmark.

The grid truncates and reweights the prior, and its cells hold 19.1 % of the prior mass (Figure 6). Even the exact posterior is therefore not calibrated on the benchmark, and the ECE here compares methods without testing calibration. The ECE of LINEARPFN is 0.009 on the control, against 0.041 for SUSIE, and 0.008, 0.026 and 0.007 under fixed-magnitude coefficients, many large interactions and heavy-tailed noise. On the datasets within the enumeration reach, the ECE of the exact posterior is 0.063 on the control, 0.028 under fixed-magnitude coefficients and 0.092 under many large interactions, against 0.081, 0.036 and 0.124 for LINEARPFN on the same datasets. The F_1 of LINEARPFN on these datasets is 0.79 against 0.82 for the exact posterior on the control and 0.78 against 0.82 under many large interactions.

G.4 REAL OUTCOMES

The 7 tables are the attitude survey and the Swiss fertility data shipped with R (R Core Team, 2025), the prostate cancer and Los Angeles ozone data (Hastie et al., 2009), the media-influence experiment of Tal-Or et al. (2010), the body-fat data (Johnson, 1996) and the diabetes data of Efron et al. (2004). They have n from 30 to 442 rows and p from 5 to 13 predictors. Each has a continuous, documented outcome. The 3 tables bodyfat, ozone and diabetes are included as collinearity stress tests. The tables did not pass through the inclusion and cleaning rules of Appendix F.1, so no correlation or near-constant rule was applied to them. Every column other than the outcome is a predictor, with two exceptions. The train/test indicator of the prostate data is dropped. From the body-fat data we

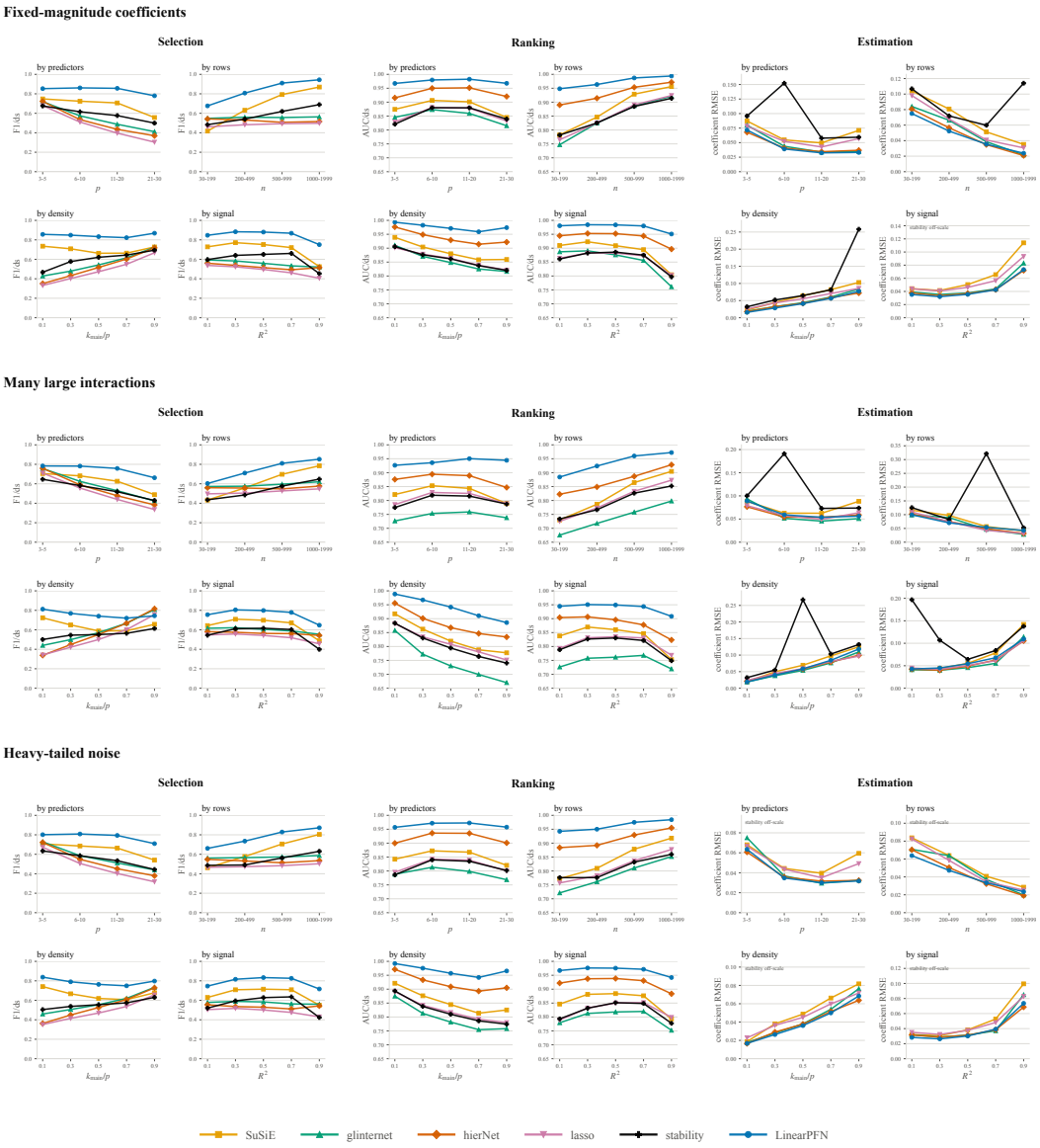

Figure 7: Outcomes outside the prior, one row per variant, each on 10,000 datasets. Left: selection, F_1 per dataset at each method's own selection rule. Middle: ranking, AUC per dataset. Right: estimation, coefficient RMSE over every candidate effect. Each panel slices the datasets by the number of predictors, the number of rows, the density k_{main}/p and the R^2; every point is the mean over the datasets of that slice. Density and R^2 are the bins of the main-benchmark dataset that the variant is built on, so the R^2 of a variant dataset can lie outside its bin. A method with an off-scale value in a panel is left out of that panel, and the panel names it.

Table 8: Real outcomes: LINEARPFN against the MCMC sampler on 7 published tables with their actual outcome. PIP agreement over the candidate effects (correlation on the main effects, RMSE and largest difference over all effects), RMSE between the posterior-mean coefficients, candidate effects on which the two MPMs agree, and held-out negative log predictive density.

			PIP			coefficient	MPM	held-out NLL		
table	n	p	r, mains	RMSE	max diff.	RMSE	agree	LINEARPFN	MCMC	OLS
attitude	30	6	0.998	0.016	0.043	0.006	21/21	1.095	1.122	1.177
swiss	47	5	0.854	0.074	0.157	0.019	15/15	1.277	1.312	1.765
prostate	97	8	0.990	0.028	0.107	0.009	36/36	0.927	0.894	0.876
tal_or	123	5	0.999	0.017	0.044	0.009	15/15	1.226	1.226	1.246
bodyfat	252	13	0.780	0.083	0.600	0.019	87/91	0.651	0.663	0.667
ozone	330	9	0.966	0.069	0.213	0.013	45/45	0.742	0.785	0.839
diabetes	442	10	0.961	0.061	0.248	0.011	53/55	1.037	1.027	1.039

drop the case number, the body density from which the outcome is computed, and three columns computed from other columns (the Brozek estimate of body fat, adiposity and fat-free weight). The media-influence table is also one of the 383 predictor matrices, where its outcome column is one of the predictors. The predictors are standardized as in the benchmarks, and the outcome is centered and scaled to unit variance, because the prior is calibrated for outcomes of unit scale.

The reference on every table is the MCMC sampler of Appendix D.2. It runs 4 chains of 100,000 to 500,000 steps, the step count rising with p. Every run met the convergence criteria, with a largest $\hat{R}$ of 1.006 and a smallest effective sample size of 899. Predictive quality is measured on a seeded split that holds out 19.9% of the rows. It is the mean negative log predictive density on the held-out rows for LINEARPFN, for the sampler and for OLS on the main effects, with the held-out rows standardized by the statistics of the fitting rows.

Pooled over the 278 candidate effects of the 7 tables, the PIPs of LINEARPFN correlate with those of the sampler at 0.972, at 0.937 on the main effects and at 0.917 on the interactions, with an RMSE of 0.065. The posterior-mean coefficients correlate at 0.991 with an RMSE of 0.014. The two MPMs agree on 272 of the 278 candidate effects. A Gaussian fitted to the fitting rows has a mean negative log predictive density of 1.385 on the held-out rows.

G.5 TRAINING WITHOUT THE PREDICTIVE LOSS

The regression head is the only source of the posterior predictive distribution, and the headroom and coverage criteria score it (Appendix E). A separate question is whether its loss also helps the other two heads. We retrained the paper's model with the predictive negative log-likelihood removed from Equation 5 and everything else unchanged, with the same prior, architecture, seed and number of steps (37,500). Each of the two models was trained once. The selection and coefficient heads of this two-head model learn from the simulated labels alone. On the 10,000 datasets of the main benchmark, the paper's model minus the two-head model is +0.005 in AUC per dataset ($p < 0.001$), +0.026 in F_1 per dataset ($p < 0.001$) and -0.005 in coefficient RMSE ($p < 0.001$), with Wilcoxon signed-rank p values. The paper's model has the smaller coefficient error on 67.0% of the datasets. The two-head model still has a higher F_1 than every baseline, 0.75 against 0.66 for SUSIE. On the validation panel of Appendix E, the two-head model meets the three selection criteria. Its AUC gap to the exact posterior is -0.0055, against -0.0026 for the paper's model. Its ECE gap is 0.0005, against 0.0034. Its smallest resolution ratio, on the interactions, is 0.9050, against 0.9640. It cannot meet the headroom and coverage criteria, because its regression head is never trained. The validation does not score the coefficient head of the two-head model. The remaining two loss terms were not reweighted, so part of the gap may be recoverable.