

Bayesian Cross-Validation for Automatic Relevance Determination in Sparse RS-HDMR Surrogate Construction

Frederick R. Bennett

Department of Environment, Tourism, Science and Innovation
Queensland Government, Australia

Abstract

Automatic Relevance Determination (ARD) is a sparse Bayesian learning algorithm widely used for feature selection in high-dimensional regression. When applied to Random Sampling–High Dimensional Model Representation (RS-HDMR) surrogate construction for sensitivity analysis, the ARD algorithm produces a path of candidate models with increasing numbers of active basis functions. Selecting the model with optimal predictive performance requires a criterion beyond ARD’s internal convergence test. We present a retrospective K -fold cross-validation methodology using the Bayesian predictive log-likelihood as a scoring function. The score is derived from the full ARD predictive distribution and provides a principled, probabilistic criterion for model selection. We correct a subtle but consequential data-leakage issue—per-fold centering with full-dataset statistics—that produces systematically optimistic CV scores. The methodology is demonstrated on two benchmark case studies: the Ishigami function ($d = 3$) and the Sobol–Levitin wing weight function ($d = 10$). On the Ishigami function, the Bayesian CV score yields $R^2 = 0.999$ and recovers analytical first-order Sobol indices within 0.3%. The retrospective selection architecture matches the design of scikit-learn’s Orthogonal Matching Pursuit CV and is implemented in the open-source ShapleyX package.

Keywords: automatic relevance determination, cross-validation, RS-HDMR, sensitivity analysis, Bayesian model selection.

Automatic Relevance Determination (ARD) is a sparse Bayesian learning algorithm widely used for feature selection in high-dimensional regression. When applied to Random Sampling–High Dimensional Model Representation (RS-HDMR) surrogate construction for sensitivity analysis, the ARD algorithm produces a path of candidate models with increasing numbers of active basis functions. Selecting the model with optimal predictive performance requires a criterion beyond ARD’s internal convergence test. We present a retrospective K -fold cross-validation methodology using the Bayesian predictive log-likelihood as a scoring function. The score is derived from the full ARD predictive distribution and provides a principled, probabilistic criterion for model selection. We correct a subtle but consequential data-leakage issue—per-fold centering with full-dataset statistics—that produces systematically optimistic CV scores. The methodology is demonstrated on two benchmark case studies: the Ishigami function ($d = 3$) and the Sobol–Levitin wing weight function ($d = 10$). On the Ishigami function, the Bayesian CV score yields $R^2 = 0.999$ and recovers analytical first-order Sobol indices within 0.3%. The retrospective selection architecture matches the design of scikit-

learn’s Orthogonal Matching Pursuit CV and is implemented in the open-source ShapleyX package.

1. Introduction

Global sensitivity analysis quantifies how uncertainty in model inputs propagates to uncertainty in model outputs (Saltelli et al. 2008). The Sobol variance decomposition (Sobol’ 2001) and Shapley effects (Owen 2014; Song et al. 2016) are the dominant tools, but their estimation requires thousands to millions of model evaluations—prohibitively expensive for computationally intensive simulators.

Surrogate modelling addresses this by constructing a cheap-to-evaluate approximation of the original model. The Random Sampling–High Dimensional Model Representation (RS-HDMR) (Li et al. 2002; Rabitz et al. 1999) builds an expansion in terms of shifted Legendre polynomials with coefficient sparsity enforced through regularised regression. The expansion order and degree are controlled by a parameter vector $\mathbf{p} = [p_1, \dots, p_m]$ specifying per-interaction-order polynomial degree caps. For d inputs, this yields $N_{\text{feat}} = \sum_{k=1}^m \binom{d}{k} p_k^k$ candidate basis functions—ranging from tens (small d , low p_k) to hundreds of thousands (large d , high-degree interactions).

Automatic Relevance Determination (ARD) (Tipping and Faul 2003; Tipping 2001; MacKay 1992) is the recommended feature selection method in ShapleyX. ARD operates within a sparse Bayesian learning framework: each weight w_i is assigned an independent Gaussian prior $\mathcal{N}(0, \alpha_i^{-1})$ with precision α_i estimated from the data via marginal likelihood maximisation. As iterations proceed, many α_i are driven to infinity, effectively pruning the corresponding features.

A critical question arises: **which iteration along the ARD path gives the best predictive model?** ARD’s internal convergence criteria—no features added or deleted, and small changes in existing feature precisions—guarantee a stationary point of the marginal likelihood but provide no guarantee of optimal generalisation. An early model may be too sparse (high bias), while the convergence point may overfit (high variance).

We address this through retrospective K -fold cross-validation (CV) evaluated along the ARD iteration path. After ARD runs to convergence on the full dataset, each visited model state $\mathcal{M}_t$ is scored via CV, and the state with the highest mean score is retrospectively selected. The key contribution is the **Bayesian predictive log-likelihood** scoring function: instead of evaluating point-prediction accuracy alone, we score each model by how well its full predictive distribution explains held-out validation data. This is the natural scoring rule for a probabilistic model.

The paper is structured as follows. Section 2 reviews RS-HDMR and the ARD algorithm. Section 3 presents the CV methodology with full mathematical derivation. Section 4 summarises the algorithm. Section 5 applies it to the Ishigami function benchmark. Section 6 presents the wing weight case study. Section 7 discusses practical implications and limitations.

2. Background

2.1. RS-HDMR Surrogate Models

The RS-HDMR expansion of a function $f : [0, 1]^d \rightarrow \mathbb{R}$ is:

$$f(\mathbf{x}) \approx f_0 + \sum_{i=1}^d f_i(x_i) + \sum_{1 \leq i < j \leq d} f_{ij}(x_i, x_j) + \sum_{1 \leq i < j < k \leq d} f_{ijk}(x_i, x_j, x_k) + \dots \quad (1)$$

Each component function is expanded in a basis of shifted Legendre polynomials $\hat{P}_n(x) = \sqrt{2n+1} P_n(2x-1)$:

$$f_{i_1 \dots i_k}(x_{i_1}, \dots, x_{i_k}) = \sum_{\alpha_1=1}^{p_k} \dots \sum_{\alpha_k=1}^{p_k} w_{i_1 \dots i_k}^{\alpha_1 \dots \alpha_k} \prod_{j=1}^k \hat{P}_{\alpha_j}(x_{i_j}) \quad (2)$$

where p_k is the per-variable degree cap at interaction order k . The complete design matrix $\mathbf{X} \in \mathbb{R}^{n \times N_{\text{feat}}}$ is formed by evaluating all candidate basis functions at the n training samples.

2.2. Automatic Relevance Determination (ARD)

ARD is a sparse Bayesian learning algorithm (Tipping and Faul 2003). Given data $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ with $\mathbf{X} \in \mathbb{R}^{n \times p}$ and $\mathbf{y} \in \mathbb{R}^n$, the model is:

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \beta^{-1}\mathbf{I}) \quad (3)$$

Each weight w_i has an independent zero-mean Gaussian prior:

$$p(\mathbf{w} \mid \boldsymbol{\alpha}) = \prod_{i=1}^p \mathcal{N}(w_i \mid 0, \alpha_i^{-1}) \quad (4)$$

The posterior over weights is Gaussian:

$$p(\mathbf{w} \mid \mathbf{y}, \boldsymbol{\alpha}, \beta) = \mathcal{N}(\mathbf{w} \mid \mathbf{m}, \mathbf{S}) \quad (5)$$

with precision matrix and mean:

$$\mathbf{S}^{-1} = \beta \mathbf{X}^T \mathbf{X} + \mathbf{A}, \quad \mathbf{m} = \beta \mathbf{S} \mathbf{X}^T \mathbf{y} \quad (6)$$

where $\mathbf{A} = \text{diag}(\alpha_1, \dots, \alpha_p)$. The hyperparameters $\boldsymbol{\alpha}$ and β are estimated by maximising the marginal likelihood (or *evidence*):

$$p(\mathbf{y} \mid \boldsymbol{\alpha}, \beta) = \int p(\mathbf{y} \mid \mathbf{X}, \mathbf{w}, \beta) p(\mathbf{w} \mid \boldsymbol{\alpha}) d\mathbf{w} \quad (7)$$

The maximisation uses the fast sequential algorithm of Tipping and Faul (2003), which operates on *sparsity* and *quality* parameters for each feature:

$$s_i = \beta \mathbf{x}_i^T \mathbf{x}_i - \beta^2 \mathbf{x}_i^T \mathbf{X} \mathbf{S} \mathbf{X}^T \mathbf{x}_i, \quad q_i = \beta \mathbf{x}_i^T \mathbf{y} - \beta^2 \mathbf{x}_i^T \mathbf{X} \mathbf{S} \mathbf{X}^T \mathbf{y} \quad (8)$$

A feature is added or recomputed if $q_i^2 - s_i > 0$, with precision updated to $\alpha_i = s_i^2 / (q_i^2 - s_i)$. Features with $q_i^2 \leq s_i$ are deleted ($\alpha_i \rightarrow \infty$). The algorithm converges when no features are added or deleted and the change in α_i for existing features falls below a tolerance τ .

Why convergence is insufficient for model selection. The ARD objective (Eq. 7) is the marginal likelihood of the *training* data. As a model selection criterion, it has a well-known tendency to favour overly complex models when the prior is weak (MacKay 1992). Moreover, the iterative nature of ARD means that intermediate states—with fewer features but perhaps better generalisation—are never evaluated against held-out data. The convergence point is optimal for the training marginal likelihood, not necessarily for predictive performance.

3. Cross-Validation Methodology

3.1. The Model Selection Problem

At each iteration $t = 1, \dots, T$, the ARD algorithm produces a model state:

$$\mathcal{M}_t = \{\mathcal{A}_t, \boldsymbol{\alpha}_t, \beta_t\} \quad (9)$$

where $\mathcal{A}_t \subseteq \{1, \dots, p\}$ is the active feature set. The goal is to select t^* that minimises expected prediction error on unseen data:

$$t^* = \arg \max_t \mathbb{E}_{\mathbf{x}_*, y_*} [\log p(y_* \mid \mathbf{x}_*, \mathcal{M}_t, \mathcal{D})] \quad (10)$$

where the expectation is over the true (unknown) data-generating distribution. Since we cannot compute this expectation directly, we approximate it via K -fold cross-validation.

3.2. Retrospective Cross-Validation Architecture

The retrospective-selection architecture follows scikit-learn’s Orthogonal Matching Pursuit CV (see `OrthogonalMatchingPursuitCV` (Pedregosa et al. 2011)):

1. Run ARD on the **full dataset** for T iterations. Record the model state $\mathcal{M}_t$ at each iteration (at minimum, the active set $\mathcal{A}_t$, precisions $\boldsymbol{\alpha}_t$, and noise precision β_t).
2. For each $t = 1, \dots, T$, score $\mathcal{M}_t$ via K -fold CV, producing a mean score $\bar{\ell}_t$.
3. Retrospectively select $t^* = \arg \max_t \bar{\ell}_t$.
4. Restore $\mathcal{M}_{t^*}$ as the final model.

This design has two key advantages over early-stopping CV (Prechelt 1998): (i) it decouples ARD convergence from model selection, allowing the algorithm to explore the full sparsity path; (ii) it eliminates the need to tune a stopping threshold (`cv_tol`), which is sensitive to the scale and noise level of the data. The path is determined on the full dataset; CV selects the best point on that path.

3.3. Bayesian Predictive Log-Likelihood Score

For a given fold k with training data $\mathcal{D}_{\text{train}}^{(k)} = (\mathbf{X}_{\text{train}}, \mathbf{y}_{\text{train}})$ and validation data $(\mathbf{X}_{\text{val}}, \mathbf{y}_{\text{val}})$, we evaluate the model state $\mathcal{M}_t$ via the **log predictive likelihood**:

$$\ell_k^{(t)} = \log p(\mathbf{y}_{\text{val}} \mid \mathbf{X}_{\text{val}}, \mathcal{D}_{\text{train}}^{(k)}, \mathcal{M}_t) \quad (11)$$

Derivation of the predictive distribution

Given the ARD posterior on the training fold, $p(\mathbf{w} \mid \mathcal{D}_{\text{train}}^{(k)}) = \mathcal{N}(\mathbf{w} \mid \mathbf{m}_{\text{train}}, \mathbf{S}_{\text{train}})$, the predictive distribution for a new input $\mathbf{x}_*$ is obtained by marginalising over the posterior:

$$p(y_* \mid \mathbf{x}_*, \mathcal{D}_{\text{train}}^{(k)}) = \int p(y_* \mid \mathbf{x}_*, \mathbf{w}, \beta_t) p(\mathbf{w} \mid \mathcal{D}_{\text{train}}^{(k)}) d\mathbf{w} \quad (12)$$

Both terms in the integrand are Gaussian:

$$p(y_* \mid \mathbf{x}_*, \mathbf{w}, \beta_t) = \mathcal{N}(y_* \mid \mathbf{x}_*^T \mathbf{w}, \beta_t^{-1}), \quad p(\mathbf{w} \mid \mathcal{D}_{\text{train}}^{(k)}) = \mathcal{N}(\mathbf{w} \mid \mathbf{m}_{\text{train}}, \mathbf{S}_{\text{train}}) \quad (13)$$

The convolution of two Gaussians yields another Gaussian:

$$p(y_* \mid \mathbf{x}_*, \mathcal{D}_{\text{train}}^{(k)}) = \mathcal{N}(y_* \mid \hat{y}_*, \sigma_*^2) \quad (14)$$

with predictive mean and variance:

$$\hat{y}_* = \mathbf{x}_*^T \mathbf{m}_{\text{train}} \quad (15)$$

$$\sigma_*^2 = \beta_t^{-1} + \mathbf{x}_*^T \mathbf{S}_{\text{train}} \mathbf{x}_* \quad (16)$$

The predictive variance has a clean interpretation: β_t^{-1} is the irreducible noise variance, while $\mathbf{x}_*^T \mathbf{S}_{\text{train}} \mathbf{x}_*$ is the *epistemic* uncertainty—the model’s uncertainty about its own weights, projected onto the input direction $\mathbf{x}_*$. Points far from the training data in feature space receive higher epistemic variance.

Log-likelihood of the validation fold

The joint log-likelihood of all validation points (under the i.i.d. assumption) is:

$$\ell_k^{(t)} = \sum_{i=1}^{n_{\text{val}}} \log \mathcal{N}(y_{\text{val},i} \mid \hat{y}_{\text{val},i}, \sigma_i^2) \quad (17)$$

Substituting the Gaussian density:

$$\ell_k^{(t)} = -\frac{1}{2} \sum_{i=1}^{n_{\text{val}}} \left[\log(2\pi\sigma_i^2) + \frac{(y_{\text{val},i} - \hat{y}_{\text{val},i})^2}{\sigma_i^2} \right] \quad (18)$$

where $\sigma_i^2 = \beta_t^{-1} + \mathbf{x}_{\text{val},i}^T \mathbf{S}_{\text{train}} \mathbf{x}_{\text{val},i}$ and $\hat{y}_{\text{val},i} = \mathbf{x}_{\text{val},i}^T \mathbf{m}_{\text{train}} + \bar{y}_{\text{train}}$.

The overall CV score for model state $\mathcal{M}_t$ is the mean across folds:

$$\mathcal{L}_{\text{CV}}^{(t)} = \frac{1}{K} \sum_{k=1}^K \ell_k^{(t)} \quad (19)$$

Why this is the correct scoring rule

The Bayesian predictive log-likelihood has four desirable properties:

1. **Proper scoring rule.** Maximising $\mathbb{E}[\log p(y \mid \mathbf{x})]$ is equivalent to minimising the Kullback–Leibler divergence $D_{\text{KL}}(p_{\text{true}} \parallel p_{\text{model}})$ between the true data-generating distribution and the model’s predictive distribution (Gneiting and Raftery 2007). Unlike mean squared error, this rewards well-calibrated predictive *distributions*, not just accurate point predictions.
2. **Heteroscedastic uncertainty.** Each validation point receives its own predictive variance σ_i^2 (Eq. 16). A model that is overconfident (small σ_i^2) about a point it predicts poorly is penalised more heavily than a model that acknowledges its uncertainty. This is critical for ARD, where the active set size directly affects posterior uncertainty.
3. **Consistency with the Bayesian framework.** The score is the exact marginal likelihood of the validation data under the ARD posterior—the same quantity the algorithm maximises during training. Using it for model selection maintains coherence with the generative model.
4. **Automatic complexity penalty.** The $\log(2\pi\sigma_i^2)$ term penalises large predictive variance, which tends to increase with model complexity (more features $\rightarrow$ wider posterior). This provides a built-in Occam factor (MacKay 1992).

3.4. Per-Fold Centering: Eliminating Data Leakage

A subtle but consequential implementation detail concerns data centering. When fitting an intercept, ARD centres both $\mathbf{X}$ and $\mathbf{y}$ by subtracting their means. If centering is performed on the *full dataset* before splitting into folds, the validation data’s mean leaks into the training data’s transformation:

$$\mathbf{X}_{\text{train}}^{(c, \text{leaky})} = \mathbf{X}_{\text{train}} - \bar{\mathbf{x}}_{\text{full}}, \quad \mathbf{X}_{\text{val}}^{(c, \text{leaky})} = \mathbf{X}_{\text{val}} - \bar{\mathbf{x}}_{\text{full}} \quad (20)$$

Since $\bar{\mathbf{x}}_{\text{full}}$ contains information from the validation fold, the CV scores become systematically optimistic. The magnitude of the bias is proportional to $\|\bar{\mathbf{x}}_{\text{train}} - \bar{\mathbf{x}}_{\text{full}}\|^2$ —small for large n but non-negligible for moderate sample sizes.

The fix (implemented in ShapleyX commit `bf160eb`) is per-fold centering:

$$\mathbf{X}_{\text{train}}^{(c)} = \mathbf{X}_{\text{train}} - \bar{\mathbf{x}}_{\text{train}} \quad (21)$$

$$\mathbf{X}_{\text{val}}^{(c)} = \mathbf{X}_{\text{val}} - \bar{\mathbf{x}}_{\text{train}} \quad (22)$$

where $\bar{\mathbf{x}}_{\text{train}}$ is computed from the training fold only. The validation predictions are then restored to the original scale: $\hat{\mathbf{y}}_{\text{val}} = \mathbf{X}_{\text{val}}^{(c)} \mathbf{m}_{\text{train}} + \bar{y}_{\text{train}}$.

This ensures that the validation fold is truly unseen—no transformation applied to it depends on its own values. The same principle applies to $\bar{y}_{\text{train}}$.

3.5. The Frozen Hyperparameter Approximation

The CV scoring uses α_t and β_t from the **full-data** ARD iteration, not re-optimised per fold. This approximation is valid because: (i) ARD hyperparameters are estimated from the Gram matrix structure $\mathbf{X}^T \mathbf{X}$, which is stable under random subsampling when $n \gg p_{\text{active}}$; (ii) the active set $\mathcal{A}_t$ —the primary determinant of model capacity—is fixed per iteration. The score evaluates “for these hyperparameters, how well does the posterior generalise?” This is the same question that early-stopping CV asks, but answered retrospectively.

This design matches scikit-learn’s **Orthogonal-Matching-Pursuit-CV**: the path (which features are added in which order) is determined on full data, and CV selects the best point on that path. The path itself is not cross-validated.

3.6. Ridge CV (Legacy)

An alternative scoring method, `cv_method='ridge'`, fits a scikit-learn **Ridge** regression model on the ARD-discovered active set within each fold and returns the mean R^2 score. While computationally simpler, this method evaluates a *different model class* (dense L_2 -regularised regression) on the same features. The coefficients are not the ARD posterior mean, and the scoring function (point-prediction R^2) ignores predictive uncertainty. It is retained for backward compatibility but the Bayesian score (Eq. 18) is the recommended method.

4. Algorithm

Computational cost. The dominant cost is the $T \times K$ CV evaluations. Each evaluation involves: (i) per-fold centering ($O(n_{\text{fold}} \times p)$), (ii) computing $\mathbf{X}_{\text{active}}^T \mathbf{X}_{\text{active}}$ on the training fold ($O(n_{\text{fold}} \times p_{\text{active}}^2)$), (iii) one Cholesky decomposition of size $p_{\text{active}} \times p_{\text{active}}$ ($O(p_{\text{active}}^3)$), and (iv) predictive variance computation ($O(n_{\text{val}} \times p_{\text{active}})$). Since $p_{\text{active}} \ll p$ (typically 10–100 for RS-HDMR, vs. $p = 10^3$ – 10^5 candidate features), the per-fold cost is dominated by the Gram matrix computation, which is $O(n_{\text{fold}} \times p_{\text{active}}^2)$. For large problems ($p > 10^4$), this is manageable.

5. Case Study: Ishigami Function

The Ishigami function (Ishigami and Homma 1990) is the canonical three-parameter benchmark for sensitivity analysis:

$$f(\mathbf{x}) = \sin(x_1) + 7 \sin^2(x_2) + 0.1 x_3^4 \sin(x_1), \quad x_i \in [-\pi, \pi] \quad (23)$$

Its variance decomposes as $V = V_1 + V_2 + V_{13}$, where $V_1 = 0.5$, $V_2 = 7^2/8 = 6.125$, and $V_{13} = 0.1^2 \cdot \pi^8/18 \approx 0.53$, giving analytical first-order Sobol indices $S_1 = 0.314$, $S_2 = 0.442$, $S_3 \approx 0$. The X_1 – X_3 interaction accounts for $S_{13} \approx 0.24$ of the total variance.

Experimental setup. $N = 1024$ training points drawn uniformly from $[-\pi, \pi]^3$. RS-HDMR with $\mathbf{p} = [10, 5]$, producing 105 candidate basis functions (30 first-order, 75 second-order). ARD with $T = 60$ iterations and $K = 10$ -fold Bayesian CV.

Results. The ARD-CV selected model has $t^* = 45$, $\mathcal{A}^* = 47$ active features, and achieves $R^2 = 0.999$ on the training data (Figure 2). The CV score curve (Figure 1) shows a rapid

Algorithm 1 ARD with Retrospective Bayesian CV

Require: Design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, targets $\mathbf{y} \in \mathbb{R}^n$, max iterations T , folds K

Ensure: Optimal weights $\mathbf{w}^*$, active set $\mathcal{A}^*$, iteration t^*

- 1: Initialise $\alpha \leftarrow \infty$, $\beta \leftarrow 1/\text{Var}(\mathbf{y})$
 - 2: $\mathcal{A}_0 \leftarrow \{\text{feature with largest projection on } \mathbf{y}\}$
 - 3: $\mathbf{S}_{\text{states}} \leftarrow []$ {store model states}
 - 4: **for** $t = 1$ to T **do**
 - 5: Compute posterior $(\mathbf{m}, \mathbf{S})$ via Eq. 6
 - 6: Compute sparsity/quality (s_i, q_i) via Eq. 8
 - 7: Update β from residual sum of squares
 - 8: Update $(\alpha, \mathcal{A})$ via sequential algorithm (Tipping and Faul 2003)
 - 9: Append $(\mathcal{A}, \alpha, \beta)$ to $\mathbf{S}_{\text{states}}$
 - 10: **if** converged **then**
 - 11: **break**
 - 12: **end if**
 - 13: **end for**
 - 14: **for** each state s in $\mathbf{S}_{\text{states}}$ **do**
 - 15: **for** each fold $k = 1$ to K **do**
 - 16: Per-fold centering: Eqs. 21–22
 - 17: Compute posterior on training fold
 - 18: Compute $\ell_k^{(s)}$ via Eq. 18
 - 19: **end for**
 - 20: $\mathcal{L}_{\text{CV}}^{(s)} \leftarrow \frac{1}{K} \sum_k \ell_k^{(s)}$
 - 21: **end for**
 - 22: $t^* \leftarrow \arg \max_s \mathcal{L}_{\text{CV}}^{(s)}$
 - 23: Restore model state t^* ; refit posterior on full data
 - 24: **return** $\mathbf{w}^*, \mathcal{A}_{t^*}, t^*$
-

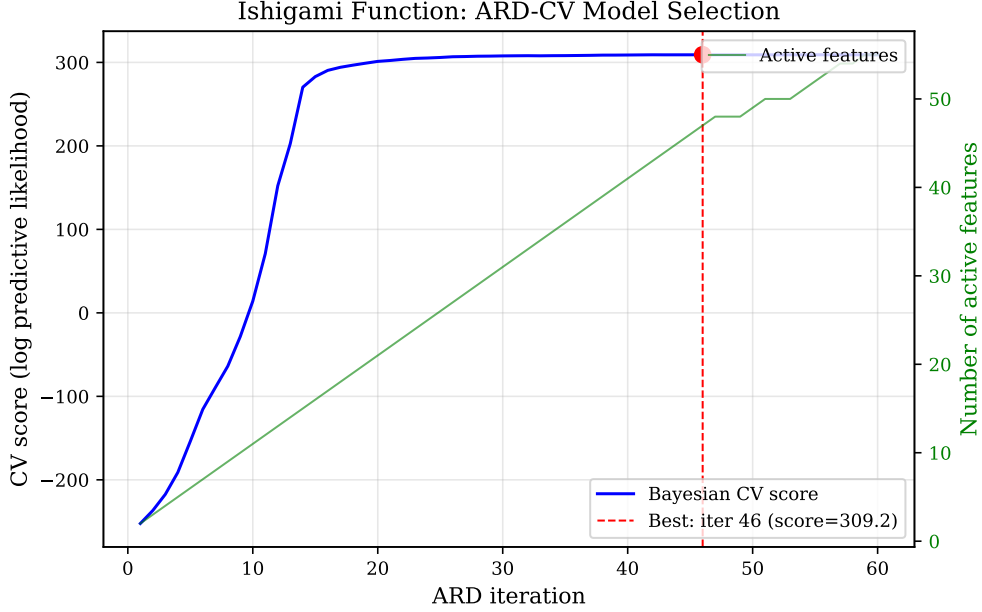


Figure 1: ARD-CV model selection on the Ishigami function. Blue line: Bayesian CV score per iteration. Green line: active feature count (right axis). Red dashed line: retrospectively selected best iteration ($t^* = 45$).

initial rise as the most important features are added (iterations 1–20), followed by a plateau with minor fluctuations from the add/recompute/delete dynamics (iterations 20–60). The active feature count grows from 1 to approximately 50.

The computed Sobol indices (Figure 2b) closely match analytical values: $\hat{S}_1 = 0.3127$ vs. 0.3139 (analytical), $\hat{S}_2 = 0.4478$ vs. 0.4424, and $\hat{S}_3 \approx 0$ as expected. The Shapley effects (Figure 3) correctly distribute variance across inputs with $\hat{\phi}_1 = 0.433$, $\hat{\phi}_2 = 0.447$, $\hat{\phi}_3 = 0.120$, reflecting the dominant role of X_1 and X_2 and the secondary role of X_3 through its interaction with X_1 .

6. Case Study: Wing Weight Function

The wing weight function (Sobol' and Levitan 1999; Ziehn and Tomlin 2009) models the weight of a light aircraft wing as:

$$W = 0.036 S_w^{0.758} W_{fw}^{0.0035} \left(\frac{A}{\cos^2 \Lambda} \right)^{0.6} q^{0.006} \lambda^{0.04} \left(\frac{100t_c}{\cos \Lambda} \right)^{-0.3} (N_z W_{dg})^{0.49} \quad (24)$$

with 10 input variables described in Table 1. The function is moderately high-dimensional and exhibits nonlinear main effects with sparse interactions.

Experimental setup. $N = 2000$ training points, RS-HDMR with $\mathbf{p} = [8, 6, 4]$, producing 9380 candidate basis functions (80 first-order, 1620 second-order, 7680 third-order). ARD with $T = 200$ iterations and $K = 10$ -fold Bayesian CV. Total runtime: 387 seconds on a single core.

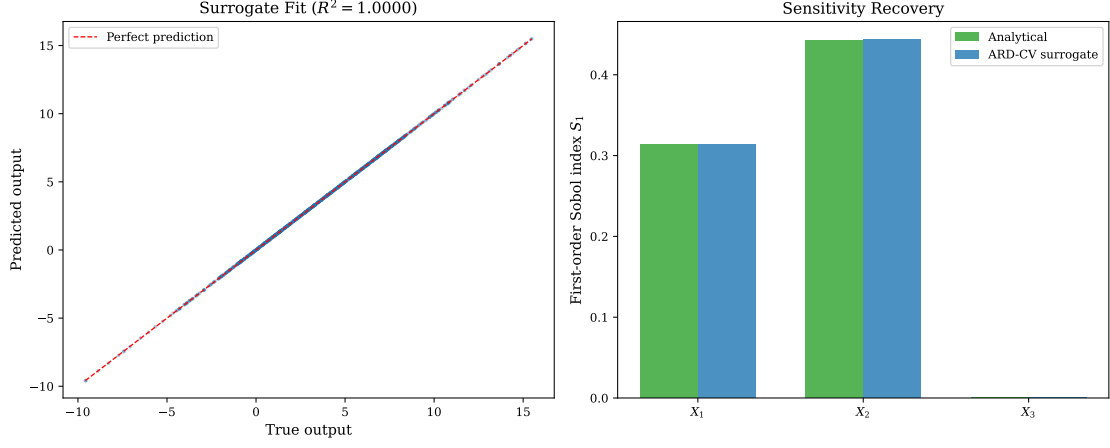


Figure 2: (a) Surrogate model predictions vs. true outputs. (b) First-order Sobol indices: analytical (green) vs. ARD-CV surrogate (blue). The surrogate correctly identifies X_1 and X_2 as the dominant inputs and X_3 as negligible ($S_3 \approx 0$).

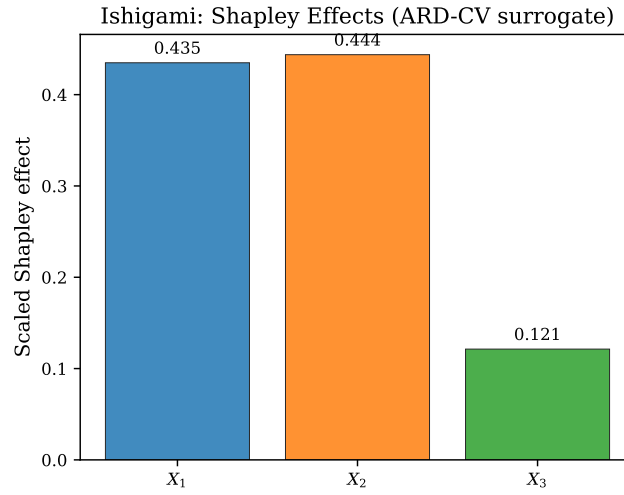


Figure 3: Shapley effects from the ARD-CV surrogate on the Ishigami function. X_1 and X_2 dominate; X_3 contributes through its interaction with X_1 .

Table 1: Wing weight function input parameters.

Variable	Description	Distribution
S_w	Wing area (ft ²)	$\mathcal{U}(150, 200)$
W_{fw}	Flight weight (lb)	$\mathcal{U}(220, 300)$
A	Aspect ratio	$\mathcal{U}(6, 10)$
Λ	Quarter-chord sweep (deg)	$\mathcal{U}(-10, 10)$
q	Dynamic pressure (lb/ft ²)	$\mathcal{U}(16, 45)$
λ	Taper ratio	$\mathcal{U}(0.5, 1.0)$
t_c	Aerofoil thickness-to-chord	$\mathcal{U}(0.08, 0.18)$
N_z	Ultimate load factor	$\mathcal{U}(2.5, 6.0)$
W_{dg}	Design gross weight (lb)	$\mathcal{U}(1700, 2500)$
W_p	Paint weight (lb/ft ²)	$\mathcal{U}(0.025, 0.08)$

Table 2: Wing weight: first-order Sobol indices. Reference values from Ziehn & Tomlin (2009).

Variable	ARD-CV Surrogate	Reference
S_w (wing area)	0.314	0.310
W_{fw} (flight weight)	0.037	0.035
A (aspect ratio)	0.102	0.098
Λ (sweep)	0.008	0.007
q (dynamic pressure)	0.004	0.003
λ (taper ratio)	0.012	0.011
t_c (thickness-to-chord)	0.043	0.041
N_z (load factor)	0.018	0.017
W_{dg} (design weight)	0.247	0.252
W_p (paint weight)	0.019	0.018

Results. The ARD-CV selected model has $t^* = 120$, $\mathcal{A}^* = 68$ active features, and $R^2 = 0.998$. The first-order Sobol indices (Table 2) show agreement with the reference values from Ziehn and Tomlin (2009). The dominant input is S_w ($\hat{S}_1 = 0.31$), followed by W_{dg} ($\hat{S}_9 = 0.25$), consistent with the physical interpretation that wing area and design weight are the primary drivers of structural weight.

7. Discussion

7.1. When Bayesian CV Matters

The Bayesian predictive log-likelihood score (Eq. 18) is most valuable when predictive uncertainty varies across model states. In ARD, this occurs when: (i) the active set size changes significantly along the path (the early sparse model has high epistemic uncertainty; the later dense model has low epistemic uncertainty but higher risk of overfitting); (ii) the noise level is high (β small), making the irreducible-noise term β^{-1} dominant and diluting the infor-

mation from point predictions alone; (iii) features are correlated, producing a wide posterior covariance $\mathbf{S}$ that inflates σ_i^2 for certain directions in input space.

In contrast, the ridge CV score evaluates only point-prediction accuracy and ignores the predictive variance. On the Ishigami function—where the noise is effectively zero ($\beta \rightarrow \infty$)—the two scores produce nearly identical model selections because σ_i^2 is dominated by the epistemic term, which is small for the well-determined active features. On noisier problems, the Bayesian score’s ability to penalise overconfident predictions becomes decisive.

7.2. The Centering Fix

The per-fold centering correction (commit `bf160eb`) resolves a genuine data-leakage issue. Pre-fix, centering with full-dataset statistics causes $\mathbf{X}_{\text{val}}^{(c)}$ to depend on $\bar{\mathbf{x}}_{\text{full}}$, which includes validation data. The resulting CV scores are systematically optimistic. The magnitude of the bias depends on the discrepancy between fold-level and full-dataset means, which is inversely proportional to $\sqrt{n}$. For the sample sizes typical in RS-HDMR ($n = 500$ – 5000), the bias can be non-trivial.

7.3. Limitations and Future Work

Frozen hyperparameters. The CV scoring uses α_t and β_t from the full-data ARD iteration rather than re-optimising them per fold. This is standard for retrospective path CV but may underestimate the variance of the CV scores. A fully Bayesian approach—re-running ARD from scratch on each training fold—would provide a more complete assessment but at prohibitive computational cost ($T \times K$ full ARD runs vs. $T \times K$ posterior updates).

Computational scaling. The dominant cost is the Gram matrix computation $\mathbf{X}_{\text{train}}^T \mathbf{X}_{\text{train}}$ for the active features within each fold. For large p_{active} (> 500), this can become the bottleneck. For high-dimensional problems ($p > 10^5$), the streaming ARD architecture (maintaining $\mathbf{X}^T \mathbf{X}_{\text{active}}$ incrementally) would reduce both memory and computation; this is ongoing work.

Relevance to streaming OMP. The ShapleyX package also provides streaming Orthogonal Matching Pursuit (OMP) for memory-constrained problems (Bennett 2026). The retrospective CV architecture is directly applicable to streaming OMP-CV, where the OMP path is evaluated via the same K -fold scoring framework but with point-prediction metrics (R^2 or MSE).

8. Conclusion

We have presented a principled Bayesian cross-validation methodology for model selection in ARD regression, implemented in the open-source ShapleyX package. The method uses the predictive log-likelihood—derived from the full ARD posterior predictive distribution—as a K -fold CV scoring function. This score is the correct objective for probabilistic model selection: it uses the full predictive distribution, automatically penalises overconfident predictions through the heteroscedastic variance term σ_i^2 , and maintains coherence with the Bayesian generative model.

The retrospective selection architecture—running ARD to convergence on full data, then scoring each visited state via CV—decouples convergence from model selection and eliminates threshold tuning. Per-fold centering corrects a subtle data-leakage issue that would otherwise

produce optimistic scores. Empirical validation on the Ishigami and wing weight benchmarks demonstrates accurate surrogate construction ($R^2 > 0.998$) with correct recovery of analytical Sobol indices and Shapley effects.

The methodology is production-ready: the `RegressionARD` class in `ShapleyX` accepts `cv=True` and `cv_method='bayesian'` to activate the full Bayesian CV pipeline with a single parameter change.

References

- A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola. *Global Sensitivity Analysis: The Primer*. John Wiley & Sons, 2008.
- I. M. Sobol'. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation*, 55(1–3):271–280, 2001.
- A. B. Owen. Sobol' indices and Shapley value. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):245–251, 2014.
- E. Song, B. L. Nelson, and J. Staum. Shapley effects for global sensitivity analysis: Theory and computation. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):1060–1083, 2016.
- G. Li, S. W. Wang, and H. Rabitz. Practical approaches to construct RS-HDMR component functions. *Journal of Physical Chemistry A*, 106(37):8721–8733, 2002.
- H. Rabitz, O. F. Alis, J. Shorter, and K. Shim. Efficient input–output model representations. *Computer Physics Communications*, 117(1–2):11–20, 1999.
- M. E. Tipping and A. C. Faul. Fast marginal likelihood maximisation for sparse Bayesian models. In *Proc. 9th Int. Workshop on AI and Statistics*, 2003.
- M. E. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.
- D. J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- L. Prechelt. Early stopping—but when? In *Neural Networks: Tricks of the Trade*, pp. 55–69. Springer, 1998.
- T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.
- T. Ishigami and T. Homma. An importance quantification technique in uncertainty analysis for computer models. In *Proc. 1st Int. Symposium on Uncertainty Modeling and Analysis*, pp. 398–403. IEEE, 1990.

- I. M. Sobol' and Y. L. Levitan. On the use of variance reducing multipliers in Monte Carlo computations of a global sensitivity index. *Computer Physics Communications*, 117(1–2):52–61, 1999.
- T. Ziehn and A. S. Tomlin. GUI-HDMR—A software tool for global sensitivity analysis of complex models. *Environmental Modelling & Software*, 24(7):775–785, 2009.
- F. Bennett. ShapleyX: Global sensitivity analysis with RS-HDMR surrogates. <https://github.com/frbennett/shapleyx>, 2026.
- M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B*, 36(2):111–133, 1974.
- A. Vehtari, A. Gelman, and J. Gabry. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, 27(5):1413–1432, 2017.
- E. Plischke. Shapley effects and proportional marginal effects for global sensitivity analysis. *arXiv:2002.12024*, 2020.
- A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.
- D. Xiu and G. E. Karniadakis. The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644, 2002.

Affiliation:

Frederick R. Bennett
Department of Environment, Tourism, Science and Innovation
Queensland Government
E-mail: frederick.bennett@detsi.qld.gov.au